INTERNATIONAL DOCTORAL
SCHOOL OF THE USC

Yago
Fontenla Seco

PhD Thesis

Process-to-Text: A Framework for the Automatic Generation of Natural Language Descriptions of Processes

Santiago de Compostela, 2023

Doctoral Programme in Information Technology Research

ESCOLA DE DOUTORAMENTO
INTERNACIONAL DA USC

TESIS DOCTORAL

PROCESS-TO-TEXT: A FRAMEWORK FOR THE AUTOMATIC GENERATION OF NATURAL LANGUAGE DESCRIPTIONS OF PROCESSES

Autor

Yago Fontenla Seco

Directores:
Prof. Dr. Alberto J. Bugarín Diz
Prof. Dr. Manuel Lama Penín

Tutor:
Prof. Dr. Alberto J. Bugarín Diz

PROGRAMA DE DOCTORADO EN INVESTIGACIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

SANTIAGO DE COMPOSTELA

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

A mi abuela Ana

¿Te frío un hueviño?

USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Sometimes science is more art than science, Morty. Lot of people don't get that.

— Rick Sanchez, *Rick and Morty*

USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Agradecimientos

Dicen que *no puedes cambiar a las personas que te rodean, pero sí puedes cambiar a las personas de las que te rodeas*. Con total seguridad, no cambiaría a ninguna de las personas de las que me rodeé durante mi vida, especialmente, a ninguna de las que me acompañó durante estos últimos años y que me acompañará seguro durante muchos años más. Gracias, a todos.

Gracias a mis directores, Alberto Bugarín y Manuel Lama, por la confianza depositada en mí durante estos años. La relación entre un doctorando y sus directores no siempre es fácil, pero siempre que he acudido a vosotros me habéis recibido con los brazos abiertos, brindándome ayuda y apoyo. Me gustaría extender este agradecimiento al Departamento de Electrónica y Computación y al Centro de Investigación en Tecnoloxías Intelixentes (CiTIUS), de la Universidade de Santiago de Compostela, por el entorno de trabajo, el apoyo y los medios proporcionados que han hecho posible el desarrollo de esta tesis. Aprovecho también para agradecer al personal de apoyo del CiTIUS, que con su trabajo, facilitan enormemente que los investigadores del centro puedan dedicar su tiempo y esfuerzo a la ciencia.

Quiero aprovechar también para agradecer a Carlos Peña y Violeta González, cardiólogos en el Hospital Universitario de Santiago de Compostela por su ayuda e interés en la investigación desarrollada en esta tesis. Su colaboración ha permitido aplicar los métodos y técnicas desarrollados en un caso de uso real, de interés e impacto para la sociedad.

Por otro lado, me gustaría agradecer al profesor Marco Montali, de la *Libera Universitá di Bolzano* y a todos los compañeros — ahora amigos, en especial a Alessandro, Sarah y Marco P., del *KRDB Research Centre for Knowledge and Data*, que me acogieron durante mi estancia en Bolzano (Italia) y que hicieron de los 3 meses que pasé allí una experiencia inolvidable. *Ci vediamo presto!*

Me gustaría destacar que este trabajo no hubiese sido posible sin el apoyo financiero de múltiples entidades y por ello quisiera agradecer al Ministerio de Ciencia, Innovación

USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

y Universidades por el programa de ayudas para contratos predoctorales para la formación de doctores (PRE2018-083719) y por financiar los proyectos directamente relacionados con la investigación de esta tesis (TIN2017-84796-C2-1-R, PID2020-112623GB-I00, PDC2021-121072-C21), a la Consellería de Cultura, Educación e Universidade (acreditación 2019-2022 ED431G-2019/04) y al Fondo Europeo de Desarrollo Regional (FEDER), que reconoce al CiTIUS - Centro Singular de Investigación en Tecnoloxías Intelixentes da Universidade de Santiago de Compostela como Centro de Investigación del Sistema Universitario de Galicia. A este último, agradecer el apoyo financiero para la realización de mi estancia en la *Libera Universitá di Bolzano*.

En lo personal me gustaría agradecer a todos mis *amigos* (que no solo compañeros) del CiTIUS. Desde los que me encontré cuando entré por primera vez en el Emprendelab, a los que formaron parte del *difunto* LS02 y, en especial, al pack de la P1: Dani, Marcos, Lorenzo, Mariña, Pedro, Juan; sois lo mejor que me llevo del CiTIUS y de esta tesis. Gracias de corazón. A mis compañeros de fatigas, gracias por estar ahí todos los días. Dentro y fuera de los entrenamientos sois quienes me apoyáis y me empujáis a ser mejor en todos los ámbitos: Alex, Chris, David, Ferro; gracias.

Por último, y todavía más importante, quiero agradecer profundamente a mi familia. A mis padres y a mi hermana, por apoyarme siempre, fomentar mi curiosidad y hacerme creer capaz de conseguir todo lo que me propongo. Vuestro apoyo incondicional lo es todo. Y a la familia que se escoge, tanto a los que me han visto crecer desde niño como a los que me están viendo crecer como adulto, especialmente a Fátima, gracias por estar ahí, siempre. Chip, sé que tú no puedes entender esto, así que, *guau guau*.

Pumos secretos.

Diciembre de 2023

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Resumo

Rapaces, boas novas! Ensineille á torradeira a sentir amor!

— Profesor Farnsworth *(Futurama)*

Os procesos permiten representar e estruturar as actividades que teñen lugar dentro das organizacións en calquera dominio, así como a forma en que se xestionan os datos e os recursos. Desde a xestión de servizos de TI [1], ata o rendemento deportivo [2], a saúde [3, 4, 5, 6, 7] ou a enxeñaría [8, 9], os procesos representan unha peza fundamental do funcionamento organizacional. Propuxéronse unha variedade de métodos, ferramentas e técnicas para axudar na definición, análise e mellora dos procesos. Todos eles forman parte da Xestión de Procesos de Negocio (BPM), un enfoque global que se centra en descubrir, analizar, redeseñar, executar e monitorizar os procesos empresariais co obxectivo de mellorar a súa eficiencia. O marco da BPM pódese ver como un ciclo continuo ao longo do ciclo de vida dun proceso. O primeiro paso implica a identificación do proceso e a súa arquitectura, seguido da fase de descubrimento e documentación do proceso. Unha vez que se define un deseño de proceso final, o proceso ponse en execución e as súas operacións son monitorizadas para recoller datos sobre o seu progreso, facilitando a mellora cíclica.

A medida que as organizacións se expanden e os seus procesos se desenvolven, vólvese cada vez máis importante monitorizar e avaliar persistentemente o seu rendemento. Aínda que algúns cambios poden ser sinxelos de xestionar durante a fase de monitorización, a identificación de debilidades e problemas que poderían afectar a eficacia xeral do proceso require unha análise máis profunda. Os distintos retos e obxectivos de cada dominio fan que os procesos

USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

sexan inherentemente dinámicos, necesitando actualizacións e axustes frecuentes para manter a súa relevancia e eficacia en liña cos cambios ambientais. Este cambio constante, xunto coa extensa cantidade de datos producidos durante a execución do proceso, destaca a necesidade crítica de ferramentas e métodos que proporcionen información precisa, veraz e fiable. Tales ferramentas e métodos poden axudar significativamente na análise de procesos, apoiando a toma de decisións organizacionais e fomentando a evolución e mellora dos procesos.

A Minaría de Procesos (PM) é unha disciplina centrada nos procesos e orientada aos datos que ten como obxectivo explotar os datos de eventos rexistrados durante a execución dun proceso. Os seus principais obxectivos son a descuberta, o monitoraxe e, en última instancia, a comprensión e mellora dos procesos. A Minaría de Procesos serve como unha ponte de conexión, situada entre a Xestión de Procesos Empresariais (BPM) e a Intelixencia Empresarial dun lado, e a Minaría de Datos e Aprendizaxe Automática do outro, ligando efectivamente estes dous reinos. O marco da minaría de procesos baséase na extracción de información valiosa relacionada co proceso a partir de rexistros de eventos que capturan a progresión das actividades nun proceso. A través da análise de datos de eventos, a PM pode utilizarse para determinar o rendemento dun proceso e proporcionar percepcións precisas que iluminen o que realmente está a acontecer no proceso, en lugar do que se cre que está a acontecer.

As técnicas de PM pódense enmarcar en tres áreas principais. En primeiro lugar, o *descubrimento de procesos* céntrase en extraer as interaccións e restricións entre actividades nun proceso, baseándose no comportamento rexistrado nos rexistros de eventos. O obxectivo é crear un modelo que encapsule a estrutura do proceso. Os modelos de procesos son fundamentais para o método integrado de minaría de procesos, ofrecendo unha representación visual dun proceso que se converte nunha ferramenta clave para analizar a súa dinámica. Estes modelos adoitan ilustrarse utilizando formalismos como o Modelo de Proceso Empresarial e Notación (BPMN), redes de Petri, redes de fluxo de traballo ou Gráficos de Seguimento Directo (DFGs). Os modelos de procesos poden clasificarse en modelos de procesos *procedurais* ou *imperativos*, e modelos de procesos *declarativos*, con diferentes formalismos que ofrecen unha forma única de representar o comportamento e o fluxo de actividades dentro dun proceso. A segunda categoría nas técnicas de minaría de procesos son os *controis de conformidade*. Este método céntrase en verificar se un modelo de proceso coincide co comportamento rexistrado nun rexistro de eventos. O obxectivo é detectar e cuantificar calquera discrepancia entre o comportamento esperado definido polo modelo e o comportamento real observado no rexistro, identificando desviacións ou non conformidades para a súa corrección. Técnicas como a

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

análise de alineación e a reprodución baseada en fichas axudan nesta avaliación. Finalmente, as técnicas de *mellora* teñen como obxectivo analizar, refinar e optimizar un proceso utilizando percepcións da descuberta de modelos, a comprobación da conformidade e as execucións pasadas. A *reparación* é un método de mellora centrado en actualizar un modelo de proceso para mellor alinealo co comportamento real observado no rexistro. Outro método de mellora é a *extensión*, onde o modelo do proceso enriquecese correlacionándoo co rexistro de eventos, permitindo unha análise e inspección máis profundas do proceso, facilitando predicións dos tempos de execución, detección de estrangulacións e identificación de anomalías, entre outros aspectos. Podemos ver a *extensión* do proceso como unha ampliación do coñecemento do proceso baseándose nun modelo de proceso e un rexistro de eventos.

Estas técnicas fan da minaría de procesos unha ferramenta versátil para comprender, validar e optimizar os procesos empresariais, cruciais para acadar a excelencia operativa. Ademais, os procesos poden observarse desde diferentes perspectivas. Estas inclúen a perspectiva do *control de fluxo*, centrada na estrutura das secuencias de actividades; a perspectiva *organizativa*, que examina os recursos implicados nun proceso; a perspectiva do *tempo*, que analiza as duracións e identifica os estrangulacións; e a perspectiva do *caso*, centrada nos atributos únicos de casos individuais. Este enfoque multifacético permite unha análise completa dos procesos, con perspectivas complementarias que a miúdo se solapan. As técnicas de minaría de procesos adoitan adoptar unha estratexia multiperspectiva, abordando varias dimensións ao mesmo tempo, como a análise de aspectos de caso e tempo para a optimización do rendemento do proceso. Aínda que esta tese se centra principalmente nas perspectivas de caso e tempo, as técnicas desenvolvidas poden adaptarse con mínimo esforzo para incluír outras perspectivas tamén.

Ao longo das últimas dúas décadas, a minaría de procesos evolucionou significativamente, levando a métodos e ferramentas avanzados para a análise de rexistros de eventos. Inicialmente centrada en aspectos técnicos, a investigación recente comezou a explorar as dimensións xestionais e organizativas da minaría de procesos. Desenvolvéronse varias metodoloxías, como o Método de Diagnóstico de Procesos, o ciclo de vida L^* e a metodoloxía PM^2, para apoiar o uso da PM nas organizacións. Con todo, aínda non hai un consenso unificado ou un marco común para a realización de análises de minaría de procesos na práctica. Comprender cales son os aspectos relevantes dun proceso é crucial para aproveitar ao máximo as oportunidades de mellora proporcionadas pola PM. Na análise práctica da minaría de procesos, os profesionais utilizan unha variedade de ferramentas desenvolvidas por investigadores e expertos. Un

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

elemento clave común a todos os proxectos de minaría de procesos é o uso de ferramentas especializadas como *ProM* e *Celonis* que, entre moitas outras, permiten a visualización do comportamento do proceso e ofrecen varias perspectivas analíticas e métricas de rendemento. Centrais para estas ferramentas son as representacións visuais, que son clave para transmitir información sobre os procesos analizados. Os modelos de procesos, en particular, son o principal método para analizar procesos. En particular, máis de 30 ferramentas de software de minaría de procesos comerciais utilizan DFGs para a descuberta e análise de procesos. Estes modelos adoitan incluír datos adicionais sobre o rendemento ou aspectos organizativos, proporcionando percepcións máis profundas sobre varios aspectos dun proceso. Os *gráficos de puntos*, son outra técnica de visualización común, mostran eventos de procesos como puntos nun plano bidimensional, co eixe horizontal representando o tempo e o eixe vertical denotando a clase do evento. Estes gráficos capturan visualmente a distribución e os patróns dos eventos, axudando na identificación de tendencias, outliers e variacións nos datos do proceso. Xunto cos modelos de procesos e gráficos de puntos, utilízanse varias outras técnicas de visualización para explorar e analizar procesos. Estas representacións visuais permiten aos analistas comprender o fluxo de actividades, identificar desviacións dos modelos estándar e medir indicadores clave de rendemento como os tempos de ciclo e a utilización de recursos e serven como ferramentas instrumentais para detectar estrangulacións nos procesos e oportunidades de mellora. Normalmente, estas técnicas combínanse en interfaces de panel, ofrecendo unha visión multifacética dun proceso. Este enfoque holístico é esencial para unha análise completa en escenarios da vida real, integrando diferentes perspectivas para unha comprensión completa dos procesos.

Aínda que a minaría de procesos evolucionou ata converterse nunha ferramenta poderosa para a análise e mellora de procesos empresariais, aínda enfronta desafíos significativos, particularmente para usuarios non expertos. Os problemas principais inclúen a necesidade de experiencia técnica na análise de procesos e un entendemento de ambientes de datos complexos, como na atención sanitaria. Ademais, existe unha fenda na investigación sobre as habilidades e competencias necesarias para unha implementación e escalado efectivos da minaría de procesos. Unha dificultade clave reside na complexidade das visualizacións da minaría de procesos, que poden ser densas e intricadas, a miúdo abafadoras para aqueles sen coñecemento especializado. Modelos de procesos como os DFGs, a pesar de ser amplamente utilizados, son propensos a malinterpretacións. Por exemplo, os DFGs totalmente conectados poden non representar claramente a estrutura subxacente do proceso, levando a percepcións incorrectas. Ademais, a

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

medida que os procesos se volven máis dinámicos e multidisciplinares, os modelos resultantes, poden ser extremadamente complexos ou desafiantes de interpretar e analizar, motivo polo que, nunha metafora visualmente moi ilustrativa, son denominados *"procesos espagueti"*. Hai unha necesidade urxente de mellorar as técnicas de visualización e novos métodos para simplificar a interpretación dos resultados da minaría de procesos, facéndoos máis accesibles a un rango máis amplo de usuarios.

Ademais do dificultoso que pode resultar interpretar as visualizacións, a complexidade inherente das ferramentas e das técnicas de minaría de procesos, xunto coa experiencia necesaria para o seu uso, supoñen un importante desafío para usuarios non expertos, como adoitan ser os xestores dos procesos ou otras persoas implicadas na súa realización. Isto crea unha barreira para a adopción máis ampla das tecnoloxías de minaría de procesos. Para unha análise efectiva da minaría de procesos, os interesados a miúdo necesitan confiar en analistas de procesos para extraer percepcións significativas, o que pode levar a dificultades, especialmente cando os analistas carecen de coñecemento específico do dominio. Esta necesidade de implicación experta pode causar demoras, esforzo adicional e posibles inexactitudes debido a malentendidos. Xa que logo, hai unha necesidade crecente de facer as ferramentas de minaría de procesos máis accesibles, claras e amigables para o usuario, adaptadas ás necesidades dos usuarios tanto en contido como en presentación. O estado actual das ferramentas de minaría de procesos, a pesar da súa sofisticación, destaca a necesidade de métodos e representacións máis intuitivos para facilitar unha interpretación e interacción máis sinxelas para partes interesadas non expertas, particularmente á medida que a minaría de procesos se expande a varios sectores.

O campo da Xeración de Linguaxe Natural (NLG) ten cuberto esta fenda noutros ambitos, xa que convirte efectivamente información intrincada a un formato máis accesible, amigable e comprensible para o usuario. Estes sistemas de NLG son particularmente valiosos en áreas como visualizacións gráficas, onde interpretar correctamente datos complexos pode ser desafiante. A linguaxe natural, en contraste coas técnicas de visualización, é unha ferramenta fundamental de comunicación humana que non require dos usuarios entender o significado subxacente detrás de algúns datos ou gráfico. Este enfoque mellora significativamente a accesibilidade e a comprensión da información intrincada en varios campos para usuarios non expertos. Ademais, diversos estudos indican que os expertos no dominio son máis propensos a tomar decisións mellor informadas cando se lles proporcionan descricións en linguaxe natural, ou unha combinación de linguaxe natural e axudas visuais, en lugar de confiar só en representacións gráficas. Neste contexto, a NLG non só facilita o manexo de cuestións

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

complexas e ambiguas na comunicación en liña, senón que tamén permite que os sistemas informáticos conversen en linguaxe humana. Isto aumenta a claridade da información e axuda tanto a usuarios expertos como non expertos a tomar decisións máis informadas. En última instancia, os sistemas NLG son vitais para facer a información complexa accesible e comprensible para aqueles sen coñecemento experto. Baseándonos nestas ideas, podemos formular a hipótese sobre a que se constrúe esta tese: **as descricións en linguaxe natural dos procesos representan unha mellor maneira de transmitir información relacionada co proceso, permitindo que usuarios non expertos interactúen coas técnicas de minaría de procesos e deriven percepcións dun proceso dunha maneira comprensible**.

Baseándose nesta hipótese, esta disertación céntrase en abordar as limitacións dos métodos actuais empregados para analizar e transmitir información relacionada cos procesos, especialmente para usuarios non expertos, a través de tres obxectivos concretos. En particular, o primeiro obxectivo céntrase na proposta de descricións en linguaxe natural que aborden aspectos relevantes dun proceso. Este obxectivo abórdase a través da definición dunha serie de descricións en linguaxe natural que involucran información cualitativa e cuantitativa sobre un proceso, información temporal e de frecuencia e interrelacións entre actividades, así como información específica do dominio do proceso. Estas descricións en linguaxe natural de procesos, baseadas en técnicas de Descricións Lingüísticas de Datos (LDD) e enraizdas na Lóxica Difusa, defínense como protoformas. As protoformas representan a semántica e estrutura fundamental dunha frase e, permiten a extracción e verbalización de percepcións sobre datos na forma de frases curtas coñecidas como resumos lingüísticos. Baseándonos nestas ideas, presentamos nesta tese un modelo baseado en protoformas para a xeración automática de descricións en linguaxe natural de procesos, no que extendemos as protoformas cuantificadas clásicas difusas con información do dominio do proceso, incluíndo detalles sobre o tempo, frecuencia, a interacción entre actividades e información específica pertinente ao dominio. Ademais, propoñemos novas estruturas semánticas e un marco de composición de protoformas, resultando en resumos lingüísticos con estruturas de descrición máis complexas e completas. A eficacia destas descricións foi validada a través de avaliacións por expertos humanos, confirmando que cubren eficazmente aspectos relevantes do proceso e proporcionan información valiosa e comprensible. O modelo proposto combina datos cualitativos e cuantitativos da execución do proceso con percepcións estruturais obtidas a través da descuberta de procesos, resultando nunha representación máis rica sintáctica e semanticamente que as descricións en linguaxe natural de procesos existentes na literatura actual.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

O segundo obxectivo da tese consiste no deseño dun marco para a xeración de descricións en linguaxe natural de procesos baseado na aproximación Datos-a-Texto (D2T). A pesar das melloras propostas na expresividade — sintacticamente e semanticamente a través da proposta de novas protoformas — o uso de resumos lingüísticos como unha saída directa dun sistema orientado ao usuario aínda é menos que ideal. Por esta razón, presentamos o marco Proceso-a-Texto (P2T), baseado na arquitectura D2T máis estendida [10] e respaldado por un formalismo de representación de coñecemento baseado en lóxica difusa. Este formalismo, baseado nas protoformas previamente propostas, facilita a representación lingüística da información do proceso, ancorando a semántica dos termos lingüísticos na lóxica difusa. O marco P2T componse de diferentes elementos. Unha ontoloxía completa delinea os conceptos centrais dun proceso, variables lingüísticas e protoformas, cruciais para representar procesos lingüisticamente e xerar descricións en linguaxe natural usando protoformas difusas. Este marco estruturado facilita o razoamento modelando e almacenando coñecemento, complementado por un razoador externo baseado en cálculos de lóxica difusa. A través dun proceso de *identificación do sistema*, defínense variables lingüísticas a partir de datos de entrada. Este paso mestura a indución orientada aos datos con percepcións expertas para desenvolver conceptos precisos e relevantes que representen a información dun proceso de maneira lingüística. Ademais, un sistema de lóxica difusa interpreta rexistros de entrada a través da ontoloxía difusa. Este proceso de *difusificación* enche a ontoloxía con casos de procesos e termos lingüísticos, traducindo datos precisos en termos lingüísticos difusos. A través deste formalismo, as protoformas serven como representacións intermedias abstractas nas descricións finais en linguaxe natural que o sistema P2T xerará eventualmente. Isto serve como a etapa de determinación de contido do sistema P2T. Un sistema de inferencia difusa baseado en técnicas LDD, xera protoformas candidatas e poboan a ontoloxía de protoformas, extraendo das varias capas da ontoloxía. As protoformas xeradas combínanse en descricións máis complexas e ricas, que, a través dun proceso de realización lingüística se convirten nas descricións finais en linguaxe natural. Propónse unha extensión baseada no aprendizaxe automático desta tarefa. Aproveitando técnicas de vangarda de xeración de linguaxe natural neural combinadas cun proceso de razoamento, pódense xerar descricións precisas por deseño, contextualmente conscientes e lingüisticamente refinadas.

Na tese presentamos un caso de uso real do marco P2T no dominio sanitario. Ademais, este caso de uso emprégase para realizar unha avaliación da proposta. Os resultados da validación son moi positivos, xa que desde un punto de vista cuantitativo obtívose unha media global de 4,07/5,00 para as preguntas xerais relacionadas coa comprensibilidade e utilidade das

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

descricións en linguaxe natural, así como a capacidade dos expertos médicos (aqueles que validaron a aproximación) para completar tarefas máis facilmente e aumentar a calidade do seu traballo. En particular, os resultados mostran *i)* que a linguaxe natural é a modalidade máis eficiente para transmitir información relacionada cos procesos; *ii)* unha marcada preferencia polas descricións textuais sobre as representacións gráficas convencionais para comunicar información do proceso aos expertos; e *iii)* as descricións en linguaxe natural proporcionaron percepcións pertinentes e valiosas sobre o proceso, ofrecendo vías para a súa mellora.

O terceiro e último obxectivo consiste no deseño e implementación dunha ferramenta interactiva para a xeración de descricións en linguaxe natural de procesos. Este obxectivo abordouse a través da implementación do sistema de Xeración de Linguaxe Natural para Minaría de Procesos (NLG4PM). Un sistema interactivo, baseado no marco proposto P2T, capaz de xerar descricións en linguaxe natural dun proceso que aborden os intereses particulares dun usuario. Ademais, introducimos C-4PM como o primeiro axente conversacional para minaría de procesos declarativa. Este sistema innovador non só fai accesible a minaría de procesos declarativa a un rango máis amplo de usuarios, senón que tamén mellora a comprensión dos modelos de procesos, mitigando malinterpretacións e promovendo percepcións precisas, especialmente en ambientes complexos como a atención sanitaria. A través da definición dun módulo de Comprensión de Linguaxe Natural (NLU), C-4PM ten a capacidade de interactuar con usuarios a través dunha interface de texto, entender consultas en linguaxe natural e, conducir razoamento baseado en especificacións de procesos declarativos e, a través dun módulo NLG, proporcionar respostas en linguaxe natural. Isto fai que C-4PM sexa un avance significativo no campo, mostrando o potencial de integrar axentes conversacionais e minaría de procesos para unha análise de procesos máis efectiva e centrada no usuario. Esta aproximación innovadora ten o potencial de mellorar significativamente a interacción do usuario con especificacións de procesos declarativos e a minaría de procesos en xeral. C-4PM non só simplifica a interacción con datos de procesos complexos, senón que tamén amplía a accesibilidade das percepcións dos procesos a un público máis diverso. Esta interacción centrada no usuario fomenta unha experiencia de usuario máis natural e atractiva, rompendo barreiras técnicas e permitindo que usuarios de diversos niveis de experiencia consulten e comprendan datos de procesos intrincados de maneira efectiva.

Coas solucións propostas, consideramos plenamente cumpridos os obxectivos desta tese de doutoramento, xa que temos proposto unha serie de descricións en linguaxe natural relevantes de procesos, xunto cun marco para a súa xeración automática e un sistema que explota este

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

marco e permite aos usuarios adaptar o proceso de xeración as súas necesidades de información específicas.. Ademais, continuando coa liña de investigación proposta, presentamos o primeiro axente conversacional para minaría de procesos declarativa. Este axente é capaz non só de xerar descricións en linguaxe natural relevantes e contextualmente conscientes dun proceso, senón que faino nun entorno conversacional, mellorando enormemente a experiencia de interacción dos usuarios.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Summary

Good news, everyone! I've taught the toaster to feel love!

— Professor Farnsworth *(Futurama)*

Processes allow to represent and structure the activities that take place within organizations in any domain, as well as how data and resources are managed. From IT service management [1], to sports performance [2], healthcare [3, 4, 5, 6, 7] or engineering [8, 9], processes represent a foundational piece of organizational functioning. A variety of methods, tools, and techniques have been proposed to assist in the definition, analysis, and improvement of processes. These are all part of Business Process Management (BPM), an all-encompassing approach that focuses on discovering, analyzing, redesigning, executing, and monitoring business processes with the goal of enhancing their efficiency. The BPM framework can be seen as a continuous cycle throughout a process's lifecycle. The first step involves the identification of the process and its architecture, followed by the process discovery and documentation stage. Once a final process design is defined, the process is put into execution, and its operations are monitored to collect data on its progress, facilitating cyclic improvement.

As organizations expand and their processes develop, it becomes increasingly important to persistently monitor and assess how these processes perform. While some changes might be straightforward to manage during the monitoring phase, a deeper analysis aimed at spotting weaknesses and problems that could impact the process overall effectiveness is required. The distinct challenges and objectives of each domain make processes inherently dynamic, necessitating from frequent updates and adjustments to maintain their relevance and efficacy

USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

in line with environmental changes. This constant change, along with the extensive data produced during process execution, highlights the critical need for tools and methods that deliver precise, truthful, and reliable information. Such tools and methods can significantly aid in analyzing processes, supporting organizational decision-making, and fostering the evolution and improvement of processes.

Process Mining (PM) is a process-centric, data-oriented discipline aimed at exploiting event data recorded during the execution of a process. Its main objectives are the discovery, monitoring, and ultimately, the understanding and enhancement of processes. Process Mining serves as a connecting bridge, positioned between Business Process Management (BPM) and Business Intelligence on one side, and Data Mining and Machine Learning on the other, effectively linking these two realms. The process mining framework is based on the extraction of valuable process-related information from event logs which capture the progression of activities in a process. Through the analysis of event data, PM can be used to determine the performance of a process and provide accurate insights which shed light into what is actually happening in a process, rather than what is believed to be happening.

PM techniques can be framed into three main areas. Firstly, *process discovery* focus on extracting the interactions and constraints between activities in a process, based on the behavior recorded in event logs. The aim is to create a model that encapsulates the process structure. Process models are fundamental to the integrated method of process mining, offering a visual depiction of a process that becomes a key tool for analyzing its dynamics. These models are often illustrated using formalisms like Business Process Model and Notation (BPMN), Petri nets, Workflow nets, or Directly-Follows Graphs (DFGs). Process models can be classified into *procedural* or *imperative*, and *declarative* process models, with different formalisms offering a unique way to depict the behavior and flow of activities within a process. The second category in process mining techniques is *conformance checking*. This method focuses on verifying whether a process model matches the behavior recorded in an event log. The goal is to detect and quantify any discrepancies between the expected behavior defined by the model and the actual behavior observed in the log, identifying deviations or non-conformities for correction. Techniques such as alignment analysis and token-based replay aid in this assessment. Finally, *enhancement* techniques aim to analyze, refine, and optimize a process using insights from model discovery, conformance checking, and past executions. *Repair* is one enhancement method, focused in updating a process model to better align with the actual behavior observed in the log. Another enhancement method is *extension*, where the process model is enriched

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

by correlating it with the event log, enabling deeper analysis and inspection of the process, facilitating predictions of execution times, bottleneck detection, and anomaly identification, among other aspects. We can see process *extension* as extending process knowledge on the basis of a process model and an event log.

These techniques make PM a versatile tool for understanding, validating, and optimizing business processes, crucial for achieving operational excellence. Furthermore, processes can be observed through different perspectives. These include the *control-flow* perspective, focusing on the structure of sequences of activities; the *organizational* perspective, examining the resources involved in a process; the *time* perspective, analyzing durations and identifying bottlenecks; and the *case* perspective, focusing on the unique attributes of individual cases. This multifaceted approach allows for a comprehensive analysis of processes, where different perspectives complement each other and often overlap. Process mining techniques usually adopt a multi-perspective strategy, addressing various dimensions simultaneously, such as analyzing both case and time aspects for process performance optimization. While this dissertation primarily centers on case and time perspectives, the techniques developed can be adapted with minimal effort to include other perspectives as well.

Over the last two decades, process mining has significantly evolved, leading to advanced methods and tools for event log analysis. Initially focused on technical aspects, recent research has begun exploring the managerial and organizational dimensions of process mining. Various methodologies, such as the Process Diagnostics Method, the L* life-cycle, and the PM^2 methodology, have been developed to support the use of PM in organizations. However, there is still no unified consensus or common framework for conducting process mining analysis in practice. Understanding which are the relevant aspects of a process is crucial to fully leverage the improvement opportunities provided by PM. In practical process mining analysis, professionals use a range of tools developed by researchers and experts. A key element common to all process mining projects is the use of specialized tools like *ProM* and *Celonis* which, among many others, enable the visualization of process behavior and offer various analytical perspectives and performance metrics. Central to these tools are visual representations, which are key to conveying information about the processes being analyzed. Process models, in particular, are the main method for analyzing processes. In particular, over 30 commercial process mining software tools utilizing DFGs for process discovery and analysis. These models often include additional data on performance or organizational aspects, providing deeper insights into various facets of a process. *Dot charts*, are another common visualization

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

technique, display process events as points on a two-dimensional plane, with the horizontal axis representing time and the vertical axis denoting the event's class. These charts visually capture event distribution and patterns, aiding in the identification of trends, outliers, and variations in process data. Alongside process models and dot charts, various other visualization techniques are used to explore and analyze processes. These visual representations enable analysts to understand activity flow, identify deviations from standard models, and measure key performance indicators like cycle times and resource utilization and serve as instrumental tools in spotting process bottlenecks and improvement opportunities. Typically, these techniques are combined into dashboard interfaces, offering a multifaceted view of a process. This holistic approach is essential for a comprehensive analysis in real-life scenarios, integrating different perspectives for a complete understanding of the processes.

While process mining has evolved into a powerful tool for business process analysis and improvement, it still faces significant challenges, particularly for non-expert users. The primary issues include the need for technical expertise in process analysis and an understanding of complex data environments, such as in healthcare. Additionally, it exists a gap in research regarding the skills and competencies necessary for effective implementation and scaling of process mining. A key difficulty lies in the complexity of process mining visualizations, which can be dense and intricate, often overwhelming for those without specialized knowledge. Process models like DFGs, despite being widely used, are prone to misinterpretation. For example, fully connected DFGs might not clearly depict the underlying process structure, leading to incorrect insights. Furthermore, as processes become more dynamic and multidisciplinary, the resulting models, often referred to as *"spaghetti processes"*, can be challenging to interpret, complicating the analysis. There is a pressing need for improved visualization techniques and new methods to simplify the interpretation of process mining results, making them more accessible to a broader range of users.

In addition to the complexities of interpreting visualizations, the inherent complexity of process mining tools and techniques, along with their demanding expertise requirements, poses a significant challenge for non-expert users like managers or resources involved in the process. This creates a barrier to the broader adoption of process mining technologies. For effective process mining analysis, stakeholders often need to rely on process analysts to extract meaningful insights, which can lead to difficulties, especially when analysts lack domain-specific knowledge. This requirement for expert involvement can cause delays, additional effort, and potential inaccuracies due to miscommunication. Consequently, there is a growing need to

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

make process mining tools more accessible, clear, and user-friendly, tailored to the needs of users in both content and presentation. The current state of process mining tools, despite their sophistication, highlights the need for more intuitive methods and representations to facilitate easier interpretation and interaction for non-expert stakeholders, particularly as process mining expands into various sectors.

The Natural Language Generation (NLG) field was developed to bridge this gap, effectively translating intricate information into a format that is more accessible and user-friendly. These systems are particularly valuable in areas like graphical visualizations, where correctly interpreting complex data can be challenging. Natural language, in contrast to visualization techniques, is a fundamental human communication tool that does not require users the underlying meaning behind some data or graph. This approach significantly improves the accessibility and understanding of intricate information across various fields for non-expert users, enhancing their engagement with the data. Moreover, studies have indicated that domain experts are more likely to make better-informed decisions when provided with descriptions in natural language, or a combination of natural language and visual aids, rather than relying solely on graphical representations. In this context NLG not only eases the handling of complex, ambiguous issues in online communication but also enables computer systems to converse in human language. This increases information clarity and aids both expert and non-expert users in making more informed decisions. Ultimately, NLG systems are vital in making complex information accessible and comprehensible to those without expert knowledge. Based on these ideas, we can formulate the hypothesis upon which this thesis is built: **natural language descriptions of processes represent a better way of conveying process related information, allowing non-expert users to interact with process mining techniques and derive insights from a process in an understandable manner**.

On the basis of this hypothesis, this dissertation focuses on addressing the limitations of current methods used for analyzing and conveying process-related information, especially for non-expert users, through three concrete objectives. In particular, the first objective focuses on the proposal of natural language descriptions that address relevant aspects of a process. This first objective is addressed trough the definition of a series of natural language descriptions involving qualitative and quantitative information about a process, temporal and frequency information and interrelations between activities, as well as domain-specific process information. Based on Linguistic Descriptions of Data (LDD) techniques, these natural language descriptions of processes are defined as protoforms. Protoforms represent

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

the fundamental semantics and structure of a sentence and, on the basis of LDD techniques, grounded in Fuzzy Logic, allow for the extraction and verbalization of insights on data in the form of short sentences known as linguistic summaries. Based on this ideas, we present a process-focused, protoform-based model for the automatic generation of natural language descriptions of processes, extending classical fuzzy quantified protoforms with process-domain information, including details about the timing, frequency, the interplay between activities, and specific information pertinent to the domain. Furthermore, new semantic structures and a protoform composition framework are proposed, resulting in linguistic summaries with more complex and complete description structures. The effectiveness of these descriptions has been validated through evaluations by human experts, confirming that they effectively cover relevant process aspects and provide valuable, understandable information. The model proposed combines qualitative and quantitative data from process execution with structural insights obtained through process discovery, resulting in a more syntactically and semantically rich representation than existing natural language process descriptions in current literature.

The second objective consists on the design of a framework for the generation of natural language descriptions of processes based on the Data-to-Text (D2T) approach. Despite the proposed enhancements in expressiveness — syntactically and semantically through the proposal of new protoforms — the use of linguistic summaries as a direct output of a user-targeted system is still less than ideal. For this reason, the Process-to-Text (P2T) framework is presented. Building on the most extended D2T architecture [10], this framework is backed by a fuzzy logic-based knowledge representation formalism. This formalism, based on the previously proposed protoforms, facilitates the linguistic representation of process information, anchoring the semantics of linguistic terms in fuzzy logic. It is composed of different elements. A comprehensive ontology outlines the core concepts of a process, linguistic variables, and protoforms, crucial for linguistically representing processes and generating natural language descriptions using fuzzy protoforms. This structured framework facilitates reasoning by modeling and storing knowledge, complemented by an external reasoner based in fuzzy logic computations. Through a *system identification* process, linguistic variables are defined from input data. This step merges data-driven induction with expert insights to develop accurate, relevant concepts that represent the information of a process in a linguistic manner. Furthermore, a fuzzy logic computational system interprets input logs via the fuzzy ontology. This *fuzzification* process fills the ontology with process cases and linguistic terms, translating crisp data into fuzzy linguistic terms. Through this formalism, protoforms serve as abstract

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

intermediate representations on the final natural language descriptions the P2T system will eventually generate. This serves as the content determination stage of the P2T system. A fuzzy inference system based in LDD techniques, generates candidate protoforms and populates the protoform ontology, drawing from the ontology's various layers. The generated protoforms then undergo the micro-planning stage, where, on the basis of their semantics, protoforms are combined into more complex and rich descriptions. Finally, through a linguistic realization process the final natural language descriptions are generated. A machine-learning based extension of this task is proposed. By harnessing state-of-art techniques of neural Natural Language Generation combined with a reasoning process, accurate by-design, contextually aware, linguistically refined descriptions can be generated.

A real use case of the P2T framework in the healthcare domain is presented. Furthermore, this use case is used to conduct an evaluation of the proposal. Validation results are very positive, since from a quantitative point of view a global average of 4.07/5.00 was obtained for the general questions related to the understandability and usefulness of the natural language descriptions as well as the capability of medical experts (those who validated the approach) to complete tasks easier and increase the quality of their work. In particular, results show *i)* that natural language emerged as the most efficient modality for conveying process-related information; *ii)* a marked preference for textual descriptions over conventional graphical representations for communicating process information to experts; and *iii)* natural language descriptions yielded pertinent and valuable insights about the process, offering avenues for its enhancement.

The third and final objective consists on the design and implementation of an interactive tool for the generation of natural language descriptions of processes. This objective is addressed through the implementation of the Natural Language Generation for Process Mining (NLG4PM) system. An interactive system, based on the proposed P2T framework, capable of generating natural language descriptions of a process that address the particular interests of a user. Furthermore, C-4PM is introduced as the first conversational agent for declarative process mining. This novel system not only makes declarative process mining accessible to a broader range of users but also enhances the understanding of process models, mitigating misinterpretations and promoting accurate insights, especially in complex environments like healthcare. Through the definition of a Natural Language Understanding (NLU) module, C-4PM has the ability to interact with users via a text interface, understand natural language queries and, conduct reasoning on the basis of declarative process specifications and, through a NLG module, provide responses in natural language. This makes C-4PM a significant leap

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

forward in the field, showcasing the potential of integrating conversational agents and process mining for more effective and user-centric process analysis. This innovative approach has the potential to significantly enhance user interaction with declarative process specifications and process mining in general. C-4PM not only simplifies the interaction with complex process data but also broadens the accessibility of process insights to a more diverse audience. This user-centric interaction encourages a more natural and engaging user experience, breaking down technical barriers and enabling users of varying expertise levels to effectively query and comprehend intricate process data.

With the proposed solutions, the objectives of this Ph.D dissertation can be considered fulfilled. As a series of relevant natural language of processes have been proposed, together with a framework for its automatic generation and a system which exploits this framework and allows users to tailor the generation process for their particular interests. Furthermore, continuing with the proposed research line, the first conversational agent for declarative process mining is presented. This agent is capable not just of generating relevant, contextually-aware natural language descriptions of a process, but does this in a conversational environment, greatly enhancing the interaction experience of users.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Contents

Resumo **xi**

Summary **xxi**

1 Introduction **1**
1.1 Motivation 1
1.2 Hypothesis and objectives 16
1.3 Contributions 17
1.4 Methodology 21
1.5 Dissertation structure 23

2 Related work **25**
2.1 Natural Language in Process Mining 28
2.2 Concluding remarks 44

3 A protoform-based model for the description of processes in natural language **49**
3.1 Linguistic Descriptions of Data 51
3.2 Protoforms for the description of processes in natural language 69
3.3 Evaluation of the proposed protoforms 78
3.4 Discussion and final remarks 80

4 A framework for the automatic description of processes in natural language **83**
4.1 The Data-to-Text (D2T) architecture 85
4.2 Process-to-Text (P2T): A framework for the automatic generation of natural language descriptions of processes 90

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

4.3 A use case in the healthcare domain 127
4.4 Evaluation of the P2T framework 136
4.5 The Natural Language Generation for Process Mining (NLG4PM) system . . 144
4.6 Discussion and final remarks 150

5 C-4PM: a Conversational Agent for Declarative Process Mining 153
5.1 A conversational agent for declarative process mining 154
5.2 Understanding natural language inputs in C-4PM 157
5.3 Reasoning with declarative process specifications 159
5.4 C-4PM architecture 164
5.5 Discussion and final remarks 174

6 Conclusions 177

Appendix A List of publications included in this Ph.D. dissertation 185

Bibliography 189

List of Figures 223

List of Tables 227

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

CHAPTER 1

INTRODUCTION

To begin is half the work, let half still remain;
again begin this, and thou wilt have finished.
— Marcus Aurelius

1.1 Motivation

Processes effectively represent and organize the activities and resources within an organization. From IT service management [1], to performance in high-level athletes [2], healthcare [3, 4, 5, 6, 7] or software engineering [8, 9], processes are a cornerstone of organizational functioning. Given the unique challenges and goals particular to each domain, processes are inherently dynamic, requiring regular updates and modifications to ensure they stay relevant and effective, mirroring the shifts in their environments. This dynamism, coupled with the vast amounts of data generated by their execution, underscores the essential need for tools and methodologies that provide accurate, truthful, and reliable information. Potentially facilitating process analysis, aiding decision-making within organizations, and paving the way for process adaptation and enhancement.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

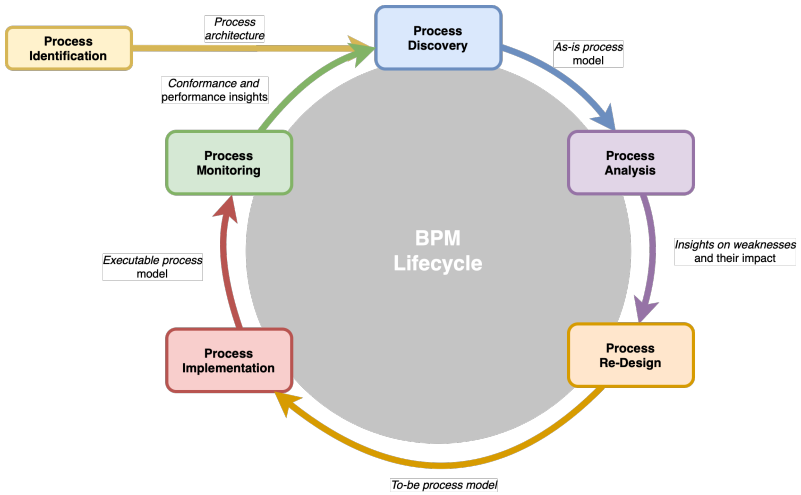


Figure 1.1: Business Process Management framework lifecycle (adapted from (11)).

Definition 1: ***Process***

A collection of inter-related events, activities, and decision points that involve a number of actors and objects, which collectively lead to an outcome that is of value to at least one customer [11].

Several methodologies, tools, and techniques have been developed to facilitate process modeling, analysis, and enhancement. All these techniques collectively fall under the realm of Business Process Management (BPM), a comprehensive framework aimed at identifying, discovering, analyzing, redesigning, executing, and monitoring business processes to ultimately optimize their performance [11]. This framework, illustrated in Figure 1.1, can be seen as a continuous cycle throughout a process's lifecycle.

As processes evolve and organizations grow, it becomes crucial to continuously observe and evaluate their performance. During the monitoring phase, processes are executed while being monitored for potential necessary changes. Some of these changes can be easily handled, however, the process is not redesigned and only predefined controls are used to adapt or reconfigure the process in minor ways [12].

For organizations aiming to delve beneath the surface and gain a profound understanding of their processes, process monitoring alone isn't enough. This deeper exploration is facilitated by

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

a new discipline and family of techniques called Process Mining (PM): a data-driven, process-centric approach, whose aim is to exploit the event data recorded during the execution of a process. Process Mining serves as a connecting bridge, positioned between Business Process Management (BPM) and Business Intelligence on one side, and Data Mining and Machine Learning on the other, effectively linking these two realms. The process mining framework, shown in Figure 1.2, is based on the extraction of valuable process-related information from event logs which capture the progression of activities in a process as events and traces. Through the analysis of these logs, process mining techniques can be used to provide insights, determine performance, and detect and identify bottlenecks, allowing to interpret what is actually happening in a process, rather than what is believed to be happening; ultimately helping to anticipate problems, understand, and improve processes [12].

For this reason, the process mining framework is usually positioned in the BPM life-cycle as a support for the implementation, monitoring and diagnosis or analysis phases. In this Ph.D. dissertation, we have specifically focused on the analysis phase, providing an in-depth look into a process in an off-line scenario using "post-mortem" event data i.e.: we analyze event data that corresponds to process instances that have already completed. However, the approach here presented could be adapted to an operational support scenario i.e.: an online in-depth monitoring scenario achievable by applying process mining techniques to "pre-mortem" events corresponding to process instances that are still actively worked on [13]; with a reasonably low effort.

Definition 2: *Process Mining*

The idea of process mining is to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs readily available in today's systems [12].

During the execution of a process, event logs record the execution data in a way in which can be later used by process mining techniques. Each entry in a log represents an individual *event* i.e.: the execution of an *activity* –being an activity each well-defined step in a process– in a particular instant. Each event is defined by three mandatory *attributes*: the name of the executed activity, a *timestamp* denoting the precise instant the activity was executed which allows for the chronological sequencing of events, and an identifier relating each event to a particular process instance. Multiple events grouped together form a *case* representing each particular instance

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

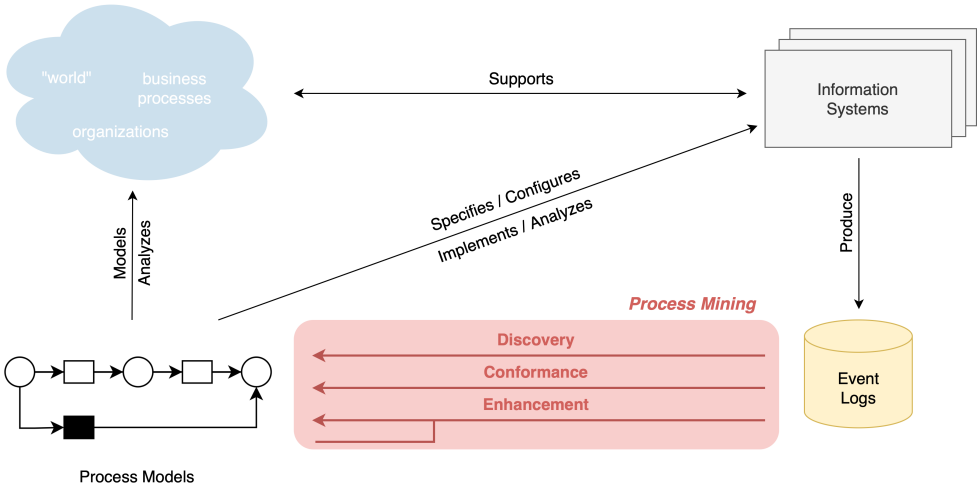


Figure 1.2: Process Mining framework (adapted from (12)).

of the process. Cases can represent multiple things depending on the particular process they belong to, like a specific customer order or a unique patient visit. The ordered sequence of events related to a single case is referred to as a *trace*. Each case is compulsorily defined by its corresponding trace and an identifier (the same used by the events that belong to the case). Besides the core elements mentioned above, events and cases can have other attributes, such as the resource responsible for the event (e.g., a specific employee or machine), the cost of the event, the outcome of the activity, and other relevant data points. Definition 3 formalizes the previous definition of event log and Table 1.1 shows an example log of a fictional company that can be used as an illustrative example. In this event log, we can identify 3 different traces, 15 events, 5 distinct activities and 6 resources.

Definition 3: *Event Log*

An event log is a structured record of events capturing the execution of a process. Consists of *cases* and cases consist of *events*. The events for a case are represented in the form of a *trace*, i.e., a sequence of unique events. Formally, an event log L can be defined as follows:

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Each *event e* in an event log is an instance of the execution of an *activity* α, where an activity α represents a well-defined step in a process.
- An *event e* is characterized by a set of *attributes*, of which three are mandatory:
  1. The name of the executed activity α.
  2. A *timestamp* indicating the precise instant at which the activity was executed.
  3. A *case identifier* that relates the event to a specific instance of the process.
- A *case c* is a specific instance or execution of the process. As events, cases have attributes, of which two are mandatory:
  1. A *case identifier* that uniquely identifies each individual instance of a process.
  2. A *trace*, defined as the ordered sequence of events $\{e_1, e_2, \ldots, e_n\}$ that constitutes a case. It represents the flow of events in one case from start to finish. All events in the case share the same *case identifier*.
- An event log L is a collection of cases, formally represented as a set of traces $L = \{c_1, c_2, \ldots, c_m\}$.
- Additional attributes for each event and case may include information such as the resource responsible for the event, cost, outcome, and other relevant data points.

As illustrated in Figure 1.2, process mining is a multifaceted field with distinct approaches typically framed into three main areas [12]. On the first point, *process discovery* techniques extract the relationships and constraints among the activities of a process based on the behavior recorded in an event log, in order to construct a model encapsulating the structure of a process. Examples of state-of-the-art algorithms that have been proposed for this task are the α-algorithm [14], Inductive Miner [15], Heuristics Miner [16], Split Miner [17], and ProDiGen [18], to name a few. The resulting process models serve as the cornerstone of process mining's integrated approach, providing a diagrammatic representation of a process that can serve as the primary means to analyze what is happening in a process. These models are typically represented using formalisms such as Business Process Model and Notation (BPMN) [19], Petri nets [20], Workflow nets [21] or Directly-Follows Graphs (DFGs) [12], among others; each formalism offers a unique way to depict the behavior and flow of activities within a process. As an example, Figure 1.3 shows the process model corresponding to the behavior recorded in the log

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 1.1: Example of a log with 15 events belonging to 3 different traces. It records the execution of 5 different activities by 6 resources in a fictional company.

Case	Timestamp	Activity	Resource	Branch
#aaa	01/10/2021 08:01	Sales Call	Jim	Scranton
#aaa	01/10/2021 08:53	Stock Check	Oscar	Scranton
#aac	01/10/2021 09:12	Sales Call	Andy	Scranton
#aac	01/10/2021 09:33	Stock Check	Oscar	Scranton
#aaa	01/10/2021 08:57	Delivery Preparation	Darryl	Scranton
#aab	01/10/2021 11:40	Sales Call	Michael	Scranton
#aac	01/10/2021 11:48	Delivery Preparation	Darryl	Scranton
#aab	01/10/2021 11:49	Stock Check	Oscar	Scranton
#aaa	01/10/2021 13:37	Delivery	Darryl	Scranton
#aac	01/10/2021 12:16	Delivery	Darryl	Scranton
#aaa	01/10/2021 13:45	Confirm Delivery	Pam	Scranton
#aab	01/10/2021 16:18	Delivery Preparation	Darryl	Scranton
#aac	01/10/2021 16:22	Confirm Delivery	Michael	Scranton
#aab	01/10/2021 17:23	Delivery	Darryl	Scranton
#aab	01/10/2021 17:35	Confirm Delivery	Andy	Scranton

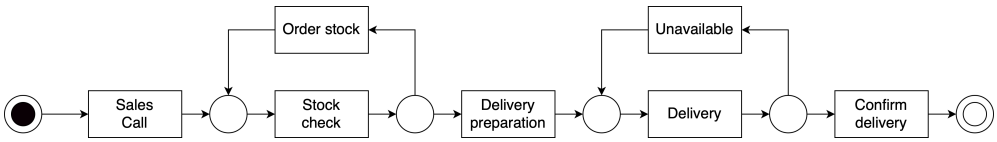


Figure 1.3: Example of a process model using the Petri Net formalism corresponding to the event log in 1.1.

presented in Table 1.1 in the form of a Petri Net. A process model can be formally defined as follows:

Definition 4: ***Process model***

Let U_α be the universe of activity names, U_{case} be the universe of case identifiers, and U_M be the universe of process models. A process model $M \in U_M$ defines a set of traces $lang(M) \subseteq U_\alpha^*$.

Where U_α represents the universe of all possible activity names in a process, U_{case} denotes the comprehensive set of unique identifiers for each case within a process, and U_M is the universe of process models, encompassing all possible process models that can be defined. In this context, a process model M, which belongs to U_M, defines a set of traces. These

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

traces are the sequences of activities from U_α that are valid within the context of the process. The notation $lang(M) \subseteq U_\alpha^*$ means that the language of the process model M — denoted as $lang(M)$ — is a subset of all possible sequences of activity names from U_α. The use of the *Kleene star* operator in U_α^* represents the set of all possible sequences (of any length) over symbols in U_α. This definition establishes a formal relationship between a process model and the valid sequences of activities it defines, reflecting the potentially infinite permutations of activity sequences that can occur within the framework of a given process model.

The second group of techniques are grouped under the *conformance checking* task. Rather than merely discovering process models, this approach emphasizes validating if the model of a process aligns with the real behavior recorded in its corresponding event log by comparing an existing process model —either discovered or manually designed— with an event log of the same process. The objective is to identify and quantify discrepancies between the expected behavior —represented in the process model— and actual behavior —as captured in the event log— to pinpoint deviations or non-conformities in order to be addressed. Multiple metrics allow to quantify this deviance, namely, the fitness metric measures the fraction of behavior observed in the event log that is supported by the model [22] i.e.: how much of the observed behavior deviates from the expected. Or the precision metric, the measures the fraction of allowed behavior (by the model) that is observed in the log [22] i.e.: how much unobserved behavior is supported by the model. Techniques like alignment analysis and token-based replay are used to assess conformance [12].

Lastly, *enhancement* techniques aim to analyze, refine and optimize a process by leveraging the insights gained from model discovery, conformance checking and previous process executions. One type of enhancement is *repair*, where the constraints of the model are updated to better support the actual behavior as observed in the event log [23]. *Extension* is another type of process enhancement technique, where the model of a process is augmented by cross-correlating it with the event log. This facilitates a more in-depth analysis and inspection of the process, allowing to predict execution times [24], detect bottlenecks [25], or detect anomalous behavior [26], among others.

Each of these areas offers a unique lens to understand, validate, and optimize business processes, making process mining an invaluable tool for organizations aiming for operational excellence. It is important to highlight that these areas are not exclusive but interplay usually takes place. When extending process models, additional perspectives are added, enabling the application of discovery and conformance techniques to more than just the control-flow

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

perspective. Hence, in addition to the three primary types of mining, processes can also be examined through different but complementary perspectives [12]:

- **Control-flow Perspective:** Focuses on the sequencing of the activities in the process. It maps the order and establishes relationships between activities, capturing the essential paths of the process. For instance, constraints like "The first activity must always be to register the order" or "A purchase from the supplier must always be preceded by a stock checking" are determined under this perspective. It forms the bedrock of a process model and typically heralds the onset of process mining analysis.
- **Organizational Perspective:** Sometimes classified as the resources perspective, this delves into the details about resources, be it human or automated systems, responsible for executing activities. The key objectives here include analyzing organizational structure i.e. what resources are involved and how they are related, via social-network graphs and identifying resource constraints. An illustrative example might be, "The person who performs the purchase from the supplier must be the same as the person who checked the stock".
- **Time Perspective:** As the name suggests, this perspective sheds light on the temporal facets of a process. It evaluates the time duration of each activity, as well as the interim periods between consecutive tasks. The ultimate objective is to unveil bottlenecks, make predictions regarding ongoing activities, and recognize potential inactivity intervals. Examples of insights drawn could be, "The preparing and packing of an order must not take more than 20 minutes" or "On average, the orders of May have been processed and prepared in 7 hours and shipped in 5 hours".
- **Case Perspective:** This perspective emphasizes the attributes and properties unique to individual cases. While the path a case takes through a process or the individuals working on it can define a case, it's often the associated data elements that provide deeper insights. For instance, if multiple cases of a process don't align with the expected behavior, examining the context of these instances can reveal commonalities that may allow to predict and anticipate future deviations. An example of this could be, "All orders from Supplier X have a delay of more than 5 days in delivery." This example highlights a specific attribute (the supplier) and a particular outcome (delay in delivery) that characterizes a subset of process instances. Understanding details like these are crucial to many processes.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

This taxonomy provides a good non-exhaustive characterization of the aspects that process mining aims to analyze. Note that the different perspectives are non-exclusive and partially overlapping and, moreover, process mining techniques do not have to focus exclusively on a single perspective. In practice a multi-perspective approach is usual, where aspects on the different dimensions of a process are addressed at the same time —e.g., the analysis of the performance and the optimization of a process can be done by simultaneously analyzing the case and time perspectives. In this Ph.D. dissertation, we have mostly focused on the case and time perspectives, the later one heavily relying on top of the control-flow perspective. However, the designed techniques and algorithms could be adapted to take into account also other perspectives with a reasonably low effort.

1.1.1 Process mining in practice

Having established the foundational aspects of process mining, including its various types, perspectives and the concepts of event logs, process models and its main components, it is imperative to delve into how practitioners harness process mining techniques. Over the past twenty years, the field of process mining has grown and matured, leading to the creation and refinement of methods and tools for analyzing event logs [27, 28]. While the predominant focus in the process mining domain has been mostly on technical dimensions, a number of recent studies have advocated for the exploration of the managerial and organizational facets of process mining [29]. In the past few years, many different process mining methodologies have been proposed tailored towards supporting organizations and their process mining projects such as the *Process Diagnostics Method* [30], the L^* *life-cycle* [12] or the PM^2 methodology [31]. Still, to this present moment, we are only beginning to understand how process mining is used in practice [32], as there is still no consensus or a common framework on how process mining analysis is or should be performed. Assessing how practitioners lead and carry out process mining projects in practice [28, 32, 33, 34] and grasping the organizational perspective entailed in process analysis is imperative to fully harness the capabilities of process mining [35, 36].

When performing process mining analysis, practitioners can draw from the plethora of techniques developed over the years by researchers and professionals. A common aspect to all process mining projects and methodologies is the use of specialized process mining tools such as ProM[1] [37] or Celonis[2] that incorporate all these techniques for performing a complete

[1] https://promtools.org
[2] https://celonis.com

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

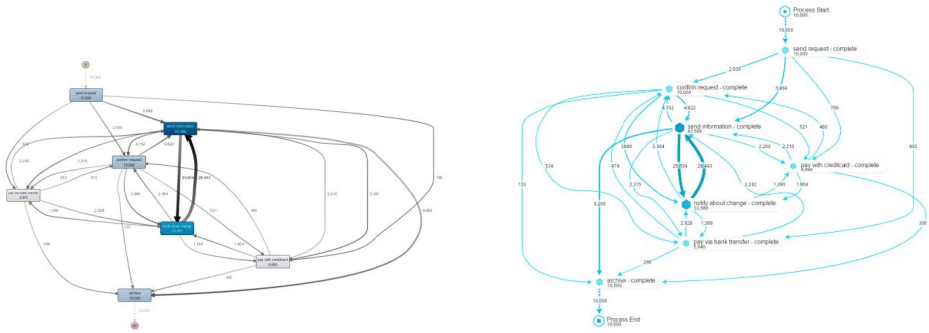


Figure 1.4: DFGs generated with Disco[3]and Celonis (extracted from (12)).

process analysis. ProM is the leading open-source process mining tool, an extensible framework for process mining, that provides a wide range of plugins for the discovery, conformance checking, and enhancement of process models, seamlessly integrating process mining essential components into its framework [12, 37]. Similarly, Celonis offers an online platform with advanced visualizations and dashboards, aiding practitioners in dissecting processes, identifying inefficiencies, and recommending optimizations based on real-time data. These tools, among many others (e.g., ProM, Celonis, Disco, ProcessGold, InVerbis, myInvenio, PAFnow, Minit, QPR, Mehrwerk, Puzzledata, LanaLabs, StereoLogic, Everflow, TimelinePI, Signavio, and Logpickr; cf. [38]), facilitate the visualization of process behavior alongside various other analytical perspectives and performance metrics.

Central to these tools are visual representations, serving as the primary medium for conveying information related to the process under analysis [39]. In particular, process models constitute the primary way of analyzing a process, being Directly-Follows Graphs the most common notation for representing them, with more than 30 commercial process mining software tools using DFGs as a base for process discovery and analysis [38]. These models are typically enriched with performance or organizational information, giving insights into different perspectives of a process. Figure 1.4 shows two process models in the form of DFGs generated with two of the most prominent process mining analysis platforms.

Other typical visualization technique used are the so-called Dot charts. Dot charts provide an overview of an entire process where each event is visualized as a point on a two-dimensional plane. The horizontal axis corresponds to the event's timestamp, while the vertical axis

[3] `https://fluxicon.com/disco/`

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

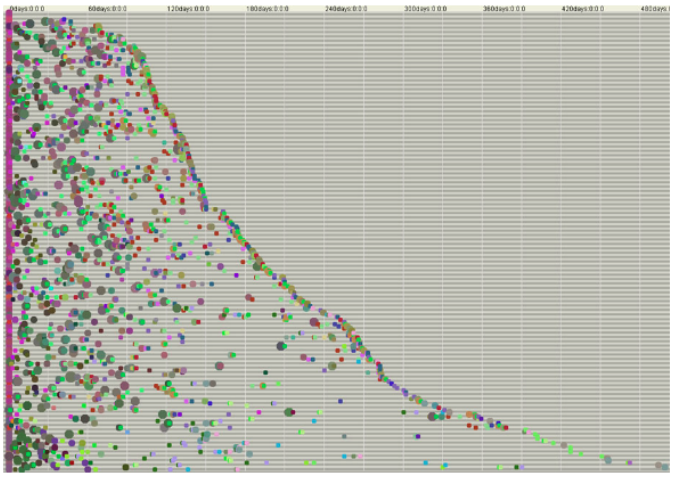


Figure 1.5: Example of a dot chart in which all cases start at time zero. The chart is used to analyze variation in flow times i.e. why some cases are handled in a few days whereas others take almost a year (extracted from (12).

represents the event's class (this could be the corresponding activity, case or any other attribute of an event) [12]. Dot charts provide a graphical representation of the distribution and pattern of events, allowing analysts to identify trends, outliers, and clusters in the process data [40]. These charts used for comparing the performance of different process instances and analyzing variations over time. An example of a dot chart can be seen in Figure 1.5.

Together with process models and dot charts, heatmaps, charts, network diagrams and an array of other visualization techniques, are used by practitioners to explore and analyze a process. Through these visual representations, analysts can gain insights into the flow of activities (process behavior), pinpoint deviations from expected process models, measure cycle times, resource utilization, and many other key performance indicators (KPIs), aiding in the identification of process deviations, bottlenecks, and opportunities for improvement [31, 39]. Usually, these different visualization techniques are aggregated in the form of dashboard-like interfaces, which provide multiple views of a process at a time. So when analyzing processes in real-life scenarios, a holistic approach is adopted, integrating and observing the various process perspectives at the same time to gain comprehensive insights into the underlying processes. Figure 1.6 shows a screenshot of ProM's interface, where a conjunction of different visual representations is used for conveying information derived from an event log: a process model

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

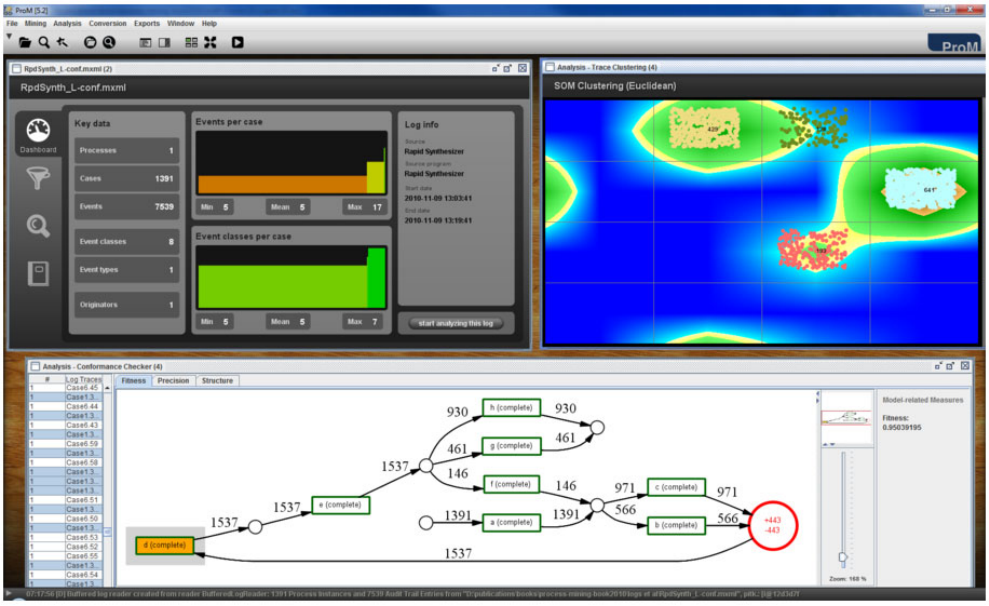


Figure 1.6: Screenshot of ProM 5.2. interface (extracted from (12)). It can clearly be seen how visualization techniques conform the core aspect for process analysis with this tool.

represented as a WorkFlow Net used for conformance checking is shown in the bottom window, the top right window shows a trace clustering visualization using Self Organizing Maps (SOM), and the top left window shows a histogram depicting the number of events per case (process instance) in the event log that is being analyzed [12].

1.1.2 The need for new ways of interacting with process related information

Process mining techniques have gained significant attention for their ability to analyze, monitor, and improve business processes. However, their use still poses a series of challenges, specially for non-expert users: *i)* highly technical knowledge (process analytical expertise and in-depth knowledge about the data) is needed to perform process analysis, particularly in complex environments (e.g., healthcare) [5, 6, 7] and *ii)* there exists a research gap around the capabilities and competencies that are required to successfully implement and scale process mining [28, 33]. Making process mining techniques ubiquitous and accesible for non-experts is widely

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

recognized as an open challenge by the process mining community [5, 41, 42, 43, 44, 45],

One of the challenges when utilizing process mining techniques is their complexity, especially when it comes to visualization [5, 6, 7]. Process mining visualizations often involve intricate and information-dense graphs: process models have many nodes and links that encapsulate multiple dimensions of data, such as time, frequency, and resource allocation, dot charts condense information about potentially thousands of events in a single image, heatmaps utilize different color scales to represent frequencies and times, etc. The correlation of this graphical information to the different elements of a process may not be intuitive for someone unfamiliar with a particular visualization method or the structure of the underlying data. Understanding these complex multidimensional visualizations requires specialized knowledge that non-expert users often do not possess [46]. Furthermore, studies have shown that process models —more specifically, DFGs— are prone to misinterpretation and can lead to incorrect insights [28, 38]. As an example, the DFGs shown in Figure 1.4 are almost fully connected, not showing any clear underlying process structure [38]. This highlights the lack of good visualization techniques for process mining analysis in the state-of-the-art and emphasizes the need for improved visualization techniques, visual analytics and the proposal of new methods to facilitate the interpretation of process mining analysis results [5].

Added to the inherent difficulty of understanding visual analytics regardless of the domain or data they convey [46], as processes become more dynamic, rich (in terms of relations among activities, amount of data captured, etc.), ad-hoc and multidisciplinary [47], their corresponding process models need to capture a multitude of paths, decisions, and activities. To obtain a good level of fitness replay, simplicity and precision need to be sacrificed, often deriving in the so-called *spaghetti processes* [48]. As an example of this kind of models, Figure 1.7 depicts the model of the management and treatment process of patients diagnosed with Aortic Stenosis in a Spanish hospital [49]. These processes and their derived analysis results are referred to be quite difficult to understand, leading to much harder process analysis [5, 6, 7, 22].

In addition to the complexities of interpreting visualizations, the use of process mining tools and techniques themselves demands a level of expertise that is often beyond the reach of non-expert users [31, 33, 41, 45]. This effectively creates a barrier for business analysts, managers, or stakeholders who could otherwise benefit from insights derived from process mining, thereby limiting the widespread adoption of process mining technologies. To be able to carry out process mining analysis projects, process stakeholders are then often required to engage process analysts and domain experts to acquire the desired insights from a process. As

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

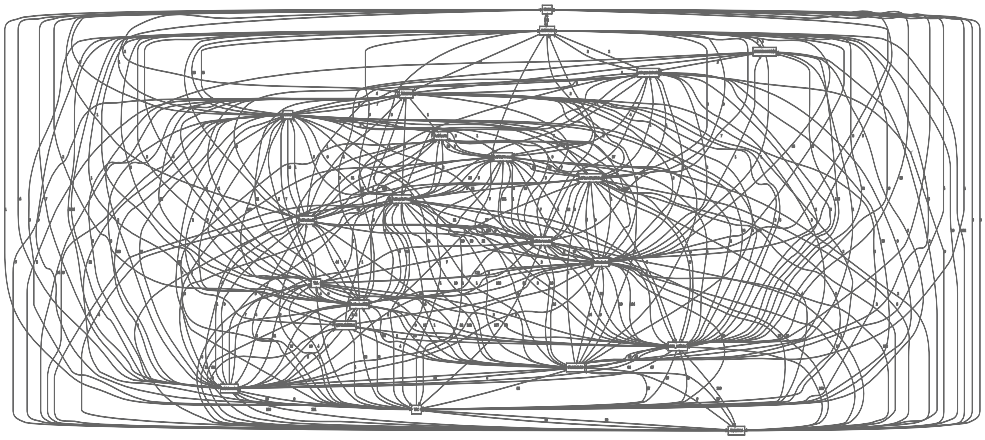


Figure 1.7: Complex process model (spaghetti model) of an Aortic Stenosis Integrated Care Process in a Spanish hospital (49). Generated with ProM (37).

often, process analysts are not domain experts for the process they are analyzing difficulties determining the causes of unexpected analysis results usually arise [30, 50]. Therefore, it is essential that process experts are involved during the process analysis stage, guiding the analysis to make sure the results are correctly interpreted and are useful for their organization. This causes significant setbacks due to extra effort and delays, and potentially inaccurate process insights resulting from miscommunication [44].

All this problematic leads to an increasing necessity of endowing, integrating and enriching process mining tools and techniques with mechanisms able to convey the most salient aspects of a process in a clear, direct and understandable manner, adapted to the real need of users both in content and form, arises. We can then conclude the following:

> Current tools and techniques in process mining, while advanced, can often seem intimidating or inaccessible to those who stand to benefit most from them. While there is no denying the power and potential of process mining, it underscores the urgent need for more user-friendly, intuitive tools and representations, especially as process mining continues its penetration into diverse sectors. This makes it imperative to provide non-expert process stakeholders with techniques for the correct interpretation and interaction with process mining analysis.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

The presented challenges associated with human-computer interaction are not exclusive to the field of process mining. Understanding complex information and graphical visualizations can be challenging, especially for non-expert users across various domains [46, 51, 52]. Supporting this hypothesis, research indicates that expertise is often required to comprehend graphical information [53, 54].

One promising approach is the use of natural language, which can serve as an effective alternative to current visualization techniques. Natural Language Generation (NLG) systems were originally proposed to address this difficulty by conveying complex information in a more accessible manner. These systems are particularly useful in areas like graphical visualizations where acquiring correct knowledge is a challenge [55]. Unlike visualization techniques, natural language is a core communication tool for humans that does not rely on the ability of users to understand data or graphs. This approach facilitates the access and comprehension of complex information in various domains for non-expert users, enhancing their understanding and interaction with the data [56]. Additionally, research has shown that domain experts are more likely to make informed decisions when provided with natural language descriptions, or a blend of natural language and visual aids, as opposed to depending exclusively on graphical displays [52]. In this context, NLG not only simplifies the management of complex, ill-defined issues in online communication [57] but also empowers computer systems to interact in human language. This enhances information clarity and supports both expert and non-expert users in making well-informed decisions [58, 59]. In conclusion, NLG systems play a crucial role in making complex information accessible and understandable to non-expert users.

Building on the role of NLG systems, Data-to-Text (D2T) systems focus on generating natural language descriptions from non-linguistic data, similar to how an expert would interpret it [10, 56]. These systems were developed with the belief that natural language is often more effective than traditional numeric or visual (graphical) data presentations, which require relatively complex cognitive processing [60]. Over the past two decades, D2T systems have shown their worth across various domains [56, 61, 62], successfully converting complex data into understandable narratives. This success underscores the role of NLG and D2T systems in transforming intricate data into accessible information, aiding decision-making for both experts and non-experts.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

1.2 Hypothesis and objectives

In order to enhance process mining ubiquity, mitigate the current challenges on process mining analysis interpretability and enhance the accessibility of process mining for non-expert users, a growing interest in developing more intuitive and user-friendly methods for conveying process information and interacting with process mining techniques emerges.

A promising Artificial Intelligence (AI) technique for this purpose is that of Natural Language Generation (NLG), capable of aggregating and distilling multiple sources and modalities of information into useful and comprehensible texts [63]. Within the NLG area, Data-To-Text (D2T) systems focus on processing large amounts of numerical or symbolical data, producing readable texts that convey relevant information for users in an understandable manner [10]. This architecture looks specially promising for generating process mining insights in the form of natural language descriptions of processes, allowing users to get insights of a process in a more natural and accessible manner than state-of-art visualization techniques.

Considering this, the base hypothesis of this Ph.D. thesis can be formulated as follows:

> Natural language descriptions of processes represent a better way of conveying process related information. Allowing non-expert users to interact with process mining techniques and derive insights from a process in an understandable manner and without the need of process mining practitioners.

Supporting this hypothesis, research shows that in many domains knowledge and expertise are required to understand graphical information [53, 54] and proves that domain experts can take better decisions based on natural language descriptions or a combination of natural language and visual support, than solely relying on graphical displays [52]. Therefore, the overall objective of this thesis is to investigate the usage of Artificial Intelligence-based techniques for the generation of natural language descriptions of processes through a process mining-based Data-to-Text system. This main objective can be broken down into the following objectives:

- **O1. To propose a series of relevant natural language descriptions of processes**.

  Specifically, to propose a series of natural language descriptions involving qualitative and quantitative information about a process, temporal and frequency information and interrelations between activities, as well as domain-specific process information.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Based on the proposed descriptions, the objective is to create a formal model in the form of an ontology that defines the proposed process descriptions and their components. This model should consider both qualitative and quantitative information sourced from a process execution data, as well as structural information extracted from the process execution data through process discovery, rather than relying on synthetic (theoretical) process models. Furthermore, this model has to achieve greater syntactic and semantic richness than the existing natural language descriptions of processes in the literature.

- **O2. Design a framework for the generation of natural language descriptions of processes taking a genuine Data-to-Text approach.**

  Building upon the formal model previously defined and based on the Data-To-Text pipeline [10, 56, 63, 64] to define a complete process-mining-based framework for the generation of natural language descriptions of a process. An evaluation with a real process must be performed to assess the validity of the proposed formal model and generation framework.

- **O3. Design and implement an interactive tool for the generation of natural language descriptions of processes.**

  Based on the proposed framework, the objective is to design and implement a comprehensive Natural Language Generation (NLG) system capable of generating natural language descriptions of a process that address the particular interests of a user.

1.3 Contributions

The research developed in this thesis has led to the following contributions:

- **C1.** The proposition of a series of relevant natural language descriptions for the description of processes. By taking into account both quantitative and qualitative process information and structural data sourced from a process itself these protoforms are more expressive than general fuzzy quantified sentences.

- **C2.** The definition of a formal model for the natural language description of processes that integrates in a high-level ontology fuzzy logic and process mining elements to create user-friendly, context-sensitive descriptions of complex processes.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- **C3.** The design of a D2T framework based on computational intelligence and natural language generation for automatically explaining processes. This model allows for the construction of more complex linguistic descriptions than those usually proposed in related work, based on the unary and binary quantified sentences, which are uninformative in many environments, especially real use cases.

- **C4.** The application of the Data-To-Text framework and formal model in a real application problem for the generation of natural language descriptions of healthcare processes. In particular, focused on describing a process related to the treatment of patients of patients with valvular heart diseases in the cardiology service of the University Hospital Complex of Santiago de Compostela (CHUS) [49].

- **C5.** The empirical evaluation of the validity and usefulness of the proposed natural language descriptions for processes in a real process in the healthcare domain. A domain known for complex, ad-hoc and interdisciplinary processes [5, 6, 7].

- **C6.** The implementation of the proposed framework in the NLG4PM interactive tool. This tool allows for the generation of natural language descriptions of a process targeted to the particular interests of a user.

- **C7.** The revision of the D2T framework for harnessing the strengths of Large Language Models (LLMs) in a D2T architecture-based system. This revised framework modifies and enhances the linguistic *realization* stage of said systems with a LLM-based approach, as a way of generating more fluent and natural texts that are syntactically richer and more appealing for users.

- **C8.** The design and implementation of C-4PM, a conversational agent for interacting with declarative specifications of a process in natural language. Furthermore, this agent is grounded in reasoning and logical inference, which allows to affirm that the answers a user gets from the interaction with the system are correct by design. This contribution was developed during a research stay in the Free University of Bozen-Bolzano under the supervision of *prof.* Marco Montali.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

1.3.1 Publications

All the contributions detailed above are included in the following publications, which constitute the main scientific output of the research conducted during the development of this thesis:

Journals

- **J1.** Fontenla-Seco, Y., Lama, M., González-Salvado, V., Peña-Gil, C., & Bugarín-Diz, A. (2022). **A Framework For The Automatic Description Of Healthcare Processes In Natural Language: Application in an Aortic Stenosis Integrated Care Process**. *Journal of Biomedical Informatics*, 128, 104033. ISSN: 1532-0464. doi: 10.1016/j.jbi.2022.104033.

  > Journal Impact Factor (JCR 2022): *4.5*
  > Computer Science, Interdisciplinary applications (Q2, 39/110)
  > Medical Informatics (Q2, 12/31)
  >
  > Journal Impact Factor (SJR 2022): *1.08*
  > Computer Science Applications (Q1)
  > Health Informatics (Q1)

International conferences

- **IC1.** Fontenla-Seco, Y., Winkler, S., Gianola, A., Montali, M., Lama, M., & Bugarín-Diz, A. (2023). **The Droid You're Looking For: C-4PM, a Conversational Agent for Declarative Process Mining**. In *Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2023 co-located with 21st International Conference on Business Process Management (BPM)* (pp. 112-116). Utrecht, The Netherlands. Available at ceur-ws.org/Vol-3469/paper-20.pdf.

  > Conference ranking (GRIN-SCIE): A, Class 2.

- **IC2.** Fontenla-Seco, Y., Lama, M., González-Salvado, V., Peña-Gil, C., & Bugarín-Diz, A. (2022). **A Framework For The Automatic Description Of Healthcare Processes In Natural Language**. In *Journal First Track at the 20th International Conference on Business Process Management (BPM)*, Münster, Germany. (Poster).

  > Conference ranking (GRIN-SCIE): A, Class 2.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- **IC3.** Fontenla-Seco, Y., Bugarín-Diz, A., & Lama, M. (2021). **Fuzzy Temporal Protoforms For The Quantitative Description Of Processes In Natural Language**. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–6). Luxembourg, Luxembourg. ISBN: 978-1-6654-4407-1. doi: 10.1109/FUZZ45933.2021.9494444.

  > Conference ranking (GRIN-SCIE): B, Class 3.

- **IC4.** Fontenla-Seco, Y., Lama, M., & Bugarín-Diz, A. (2020). **Process-To-Text: A Framework for the Quantitative Description of Processes in Natural Language**. In *Trustworthy AI - Integrating Learning, Optimization and Reasoning (TAILOR 2020) at ECAI 2020*. Lecture Notes in Computer Science, vol. 12641 (pp. 212-219). Springer, Cham. ISBN: 978-3-030-73959-1 (online). doi: 10.1007/978-3-030-73959-1_19

  > Conference ranking (GRIN-SCIE): A, Class 2.

- **IC5.** Fontenla-Seco, Y. (2020). **A Framework for the Automatic Generation of Natural Language Descriptions of Processes**. In *Proceedings of the 1st Doctoral Consortium at the European Conference on Artificial Intelligence (DC-ECAI 2020)* (pp. 45-46). Santiago de Compostela, Spain. Available at hdl: 10347/23263

  > Conference ranking (GRIN-SCIE): A, Class 2.

National conferences

- **NC1.** Fontenla-Seco, Y., Lama, M., González-Salvado, V., Peña-Gil, C., & Bugarín-Diz, A. (2023). **A Framework For The Automatic Description Of Healthcare Processes In Natural Language** (Summary). *Jornadas de Ingeniería de Ciencia e Ingeniería de Servicios (JCIS)*, Ciudad Real, Spain. (Summary). Available at hdl: 11705/JCIS/2023/5992.

- **NC2.** Fontenla-Seco, Y., Lama, M., González-Salvado, V., Peña-Gil, C., & Bugarín-Diz, A. (2022). **Framework para la Descripción Automática de Procesos en Lenguaje Natural: Aplicación en un Proceso Asistencial Integrado de Estenosis Aórtica**. *XXI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF)* (pp. 25-26), Toledo, Spain. ISBN: 978-84-09-42613-3. doi: 10.13140/RG.2.2.22529.22886

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

1.3.2 Other contributions

In addition to the publications and projects detailed above, during the development of this doctoral thesis, some results and demos have been transferred to other institutions. Furthermore, in the spirit of bolstering collaborative research and furthering the state of the art in the field, we have made the algorithms developed during the course of this dissertation openly available for academic purposes. The source code provides researchers with a comprehensive toolkit to replicate, extend, or integrate these algorithms into their own work.

- *NLG4PM (Natural Language Generation For Process Mining)*[4] — contribution C6: Is an interactive tool for the generation of natural language descriptions of processes. Integrating process mining and soft computing techniques in a D2T architecture, user targeted descriptions of a process are generated. Through its interface, a user can explore the data present in an event log and select the relevant features for which descriptions should be generated. Additionally, temporal intervals can be selected so the analysis of the process is focused in particular time frames.

- *C-4PM (Conversational agent For declarative Process Mining)*[5] — contribution C8: With C-4PM we introduced the first conversational agent for declarative process specifications. C-4PM facilitates tasks such as consistency, conformance, and model checking through a conversation-driven modular pipeline. This work was carried out during a research stay in the Free University of Bozen-Bolzano under the supervision of prof. Marco Montali. C-4PM was presented at the 21st International Conference on Business Processes Management (BPM 2023) [65] and a demo is available online[6].

1.4 Methodology

The research delineated in this dissertation is conducted employing the scientific method [66]. Originating from the objectives outlined in Section 1.2, a thorough analysis of existing solutions within the state of the art is undertaken. Subsequently, a solution is formulated, validated, and based on validation results, iterative refinements are made to the proposed methods. If possible, the proposal is applied in real cases. It's imperative to underscore that this process is not a

[4] https://gitlab.citius.usc.es/yago.fontenla.seco/nlg4pm/
[5] https://github.com/Yagouus/c-4pm
[6] https://tec.citius.usc.es/c-4pm/

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

rigid but flexible approach, allowing for a non-linear progression of activities adapting to the conditions of research. Namely, certain stage may start prior to the completion of preceding ones, and retrospective revisits are permitted to augment or improve the results —for instance, a set of tests may be performed before completing the complete development of the research to have references to its progress. The aforementioned steps are specifically enacted as follows:

- Objective definition: The initial phase, detailed in Chapter 1, involves establishing the objectives and research goals to be achieved. Including a description of the problem, the criteria for evaluating the proposed solution and its limitations. A comprehensive analysis of these objectives is conducted to delineate them further and identify the sub-objectives encompassed within them.

- Review of related work: This stage is detailed in Chapter 2, where an analysis of the state-of-the-art is carried out to go over all the related works proposed so far. This helps in understanding the current state of this research field and evaluating the existing models and techniques, including their weaknesses and strengths. This will pinpoint the needs that the proposed research should address.

- Research development: Based on the insights obtained from the state-of-the-art analysis, a distinct solution is proposed to tackle the identified problem and objectives established initially. This stage includes the proposal of relevant natural language descriptions for processes, the definition of a formal model for these descriptions, developing a theoretical Data-to-Text framework for its generation, designing algorithms, and implementing the system. A step-by-step approach is used for software development, alternating between feature implementation and testing. Chapters 3, 4 and 5 include the details of the development of each of the solutions developed to achieve the objectives previously proposed.

- Validation of the solution: Once the approach has been developed, a series of tests are performed to validate the proposal. In the case the results of the validation do not meet the expectations, the process goes back to previous steps to perform the needed modifications. In this particular case, multiple validation stages are performed, as different elements need to be validated and integrated into a complete solution. Chapters 3, 4 and 5 address the different evaluations performed for each devised method.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Application in real cases: In this particular Ph.D. dissertation, this stage is parallel to all others, as for the approach here presented real data is of uttermost importance. For both the proposal and evaluation of relevant natural language descriptions of processes real data is needed in order to be able to assess the truthfulness, usefulness and relevancy of the proposed descriptions.

1.5 Dissertation structure

This Ph.D. dissertation is composed of chapters that analyze prior related work, describe in detail the contributions of this thesis, analyze and discuss the results obtained and present a series of concluding remarks. More specifically, the document is structured as follows:

- Chapter 2 presents an exhaustive analysis of the state of the art. Firstly, this chapter includes a primer on Natural Language Processing (NLP) with particular attention to Natural Language Generation (NLG) and Linguistic Descriptions of Data (LDD) fields is presented. Then, a general review of the literature on the use of natural language within Process Mining is included. A deeper review on the state-of-art proposals related to the objectives of this thesis is presented.

- Chapter 3 focuses on objective O1. A series of relevant natural language descriptions relating quantitative and qualitative information about a process are proposed — contribution C1. And introduces the basis for a formal model based on fuzzy protoforms to support these process descriptions — contribution C2.

- Chapter 4 focuses on objective O2, describing the proposed framework that builds on the previously proposed model based on fuzzy quantified descriptions for generating natural language descriptions of processes based on the Data-to-Text architecture — contribution C3. This framework is applied in real use case, in particular, in the management and treatment process of patients with valvular heart diseases in the cardiology service of the University Hospital Complex of Santiago de Compostela (CHUS) [49] — contribution C4. Details on the empirical validation of the proposed natural language descriptions and Data-to-Text system are also given in this chapter —contribution C5. This chapter also focuses on objective O3, presenting how the NLG4PM tool, based on the proposed D2T architecture, is built — contribution C6. Moreover, this chapter goes into details on

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

how the original D2T architecture is modified to support LLMs and enhance the quality of the generated texts — contribution C7.

- Chapter 5 also in objective O3, introducing the C-4PM system designed for the interaction with declarative process specifications in natural language. The C-4PM system complements the rest of the work done in this thesis and explores the capabilities of declarative process mining.
- Finally, Chapter 6 presents the concluding remarks of this dissertation. This chapter includes a summary and a discussion of the contributions of the research conducted and outlines some future research lines that could be explored to extend the approaches described in this dissertation.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

CHAPTER 2

RELATED WORK

You will accumulate more knowledge and more books as you grow older, and the growing number of unread books on the shelves will look at you menacingly. Indeed, the more you know, the larger the rows of unread books. Let us call this collection of unread books an antilibrary.

— Nassim Taleb on the *antilibrary* concept introduced by Umberto Ecco

Human-computer interaction challenges extend beyond the realm of process mining [46, 51, 52]. The use of natural language stands out as a promising strategy, offering an effective alternative to traditional techniques. Natural language serves as a fundamental communication tool that does not require users to have specialized skills in interpreting data or graphs. This has the potential to significantly enhance the accessibility and comprehension of complex information across various domains, particularly for non-expert users, by improving their understanding and interaction with data [56]. At the intersection of Artificial Intelligence, Computer Science, and Linguistics, Natural Language Processing (NLP) emerges as a field whose goal is to enable computers to understand, interpret, generate, and respond to human language in a valuable way. NLP holds great potential in facilitating human-computer interaction

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

through the use of natural language: as the primary mean of communication for humans, natural language is independent of users' abilities to extract relevant information from data or graphs; facilitating the accessibility of computational tools and resources for non-expert users [56]. This technology appears as a critical factor in many applications such as voice-activated assistants, chatbots, and many other AI-powered applications across multiple domains such as healthcare [67, 68], sports [69], economics [70] or meteorology [71, 72] among many others.

Within NLP, two main branches can be identified, namely, Natural Language Understanding (NLU) and Natural Language Generation (NLG). The former, deals with providing machines with the means and capabilities for understanding natural language. This is particularly useful in domains and applications such as information retrieval (IR) [73], question answering (QA) [74], sentiment analysis [75] or misinformation detection [76]. Conversely, Natural Language Generation (NLG) concentrates on producing coherent and interpretable textual outputs that synthesize information drawn from diverse sources and formats of information [63]. Additionally, an array of systems straddle the boundaries between NLU and NLG, possessing both interpretative and generative functionalities. Notably, machine translation systems [77] and dialogue or conversational systems [78] are key examples of such dual-capability platforms.

Natural language generation systems have been around for many years. For instance, ELIZA, the first natural processing system to carry out certain types of machine-human conversations, was proposed more than 50 years ago [79]. However, despite the foundational ideas being laid much earlier, it wasn't until the 1980s that marked the first appearance of data-to-text generation systems that could turn data into insightful texts [80]. This period marked the beginning of a significant evolution in NLG, with systems emerging across diverse sectors, including meteorological reporting [71, 72], project management and engineering documentation [81, 82], operating systems monitoring [83], clinical reporting in healthcare [67, 68], and the interpretation of conceptual models [84, 85].

NLG techniques are deployed across a diverse range of applications, with a strong focus on text-to-text generation systems. These are applications that process existing textual material to automatically generate new, coherent texts as outputs [86]. These methods effectively refine and repurpose information, encompassing summarization — which condenses extensive texts into their core messages [87], narrative generation — combining different sources of text into engaging stories [88], and spelling and grammar correction tools —designed to improve textual accuracy [89]. Another significant branch within the NLG domain focuses on systems that produce natural language outputs from non-linguistic data sources, commonly referred to as

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

data-to-text generation or Data-to-Text (D2T) systems. These systems translate numerical or structured data into coherent and understandable natural language texts, a pivotal process for making sense of complex datasets [10, 86].

In recent years, the field of NLG has been revolutionized by the advent of neural-based methods and Large Language Models (LLMs) like GPT-3 [90]. These models leverage complex neural network architectures to significantly improve the generation of human-like text. Recurrent Neural Network-based Encoder-Decoder models, for example, have enhanced dialogue systems by producing more relevant and fluent language [91]. However, these advancements predominantly focus on text-to-text generation, where the input and output are both textual. This can sometimes lead to challenges, such as the model "hallucinating" information — generating plausible but inaccurate or unrelated content [92, 93]. Additionally, these models may omit [94] crucial information or struggle with maintaining coherence over longer texts. Despite the substantial progress represented by LLMs, these limitations highlight ongoing challenges in the field.

One notable deficit within NLG is the absence of semantic modeling. This absence becomes critical in data-to-text systems, as these systems must accurately interpret the underlying meaning of their input data to generate correct output texts effectively. To navigate this challenge, different approaches have been proposed for handling the uncertainty intrinsic to natural language. Within the Fuzzy Set Theory, Linguistic Descriptions of Data (LDD), linguistic summaries of data or linguistic data summaries, have been developed to extract pre-defined patterns from some data and verbalize them in the form of short sentences [95, 96, 97, 98, 99]. These pre-defined patterns, known as protoforms — or prototypical forms — represent the fundamental semantics of a sentence [100]. Instanced protoforms, represent knowledge about some data [101]. Fuzzy quantified sentences are the most prominent protoform in the literature [60, 97, 102, 103]: type-I fuzzy quantified sentences represent the most basic form of fuzzy quantification and follow the pattern "$Q\ Xs$ are A". This pattern allows for the generation of summaries like *"Most days are rainy"* or in its extended version — the type-II fuzzy quantified sentence — *"Many rainy days are cold"*. This type of sentences can be used for summarizing knowledge about variables and their values in a given domain. As they provide quantitative information about the fulfillment of a series of properties of interest in a particular dataset.

Despite constituting informative pieces of language, the limited expressiveness and lack of linguistic refinement of protoforms and in particular, fuzzy quantified sentences, makes them less than ideal to be directly fed to users [60, 104]. However, if understood as an intermediate

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

language, protoforms represent a powerful formalization of knowledge in the context of larger NLG systems, particularly, for D2T systems.

2.1 Natural Language in Process Mining

The development of specialized hardware [105], an increase in data availability [106], together with seminal breakthroughs in machine learning [107], fuel the current surge in Artificial Intelligence applications, from autonomous systems [108] to computer vision tasks [109]. Among these advancements, the capabilities of NLP systems have significantly improved. State-of-the-art methodologies like transformer-based architectures, are pushing the field into new areas of discovery. Large Language Models (LLMs) like GPT-3 [90] and BERT [110] have demonstrated unprecedented capabilities in tasks like text summarization and translation, setting new benchmarks in the field [90]. These innovations, however, come accompanied by a series of emerging risks and challenges [92, 93].

Driven by these significant advances, akin to many other domains, the use of NLP techniques has taken a quantum leap in the Business Process Management and, in particular, in the Process Mining community. The advent of machine learning-based techniques and more specifically LLMs, makes more accessible the use of natural language for many different tasks, broadening the scope of NLP techniques. This is reflected by the many research initiatives that have emerged recently assessing the use of this kind of approaches to support (or even subsume) process mining techniques [111, 112, 113, 114, 115].

Despite its just recent spike in popularity, the use of natural language in Process Mining extends beyond the past decade [116, 117, 118]. Since those early days, a rich body of research has been undertaken and numerous NLP techniques have been applied in BPM, process mining, workflow management, event logs and related realms; the state of the art is extensive both in terms of number and diversity of proposals. Moreover, techniques originally devised for NLP have been successfully repurposed for process mining tasks [119].

Table 2.1 highlights specific instances of research where natural language is used to directly address process mining challenges. Examples include natural language interfaces aiming at interpreting and understanding natural language utterances and queries for computing Process Performance Indicators (PPIs) [41, 44], retrieving information from an event log [132, 134] or querying process mining platforms [45, 135]; detection, generation and refactoring of natural language features for different process model elements [117, 118, 120, 121, 124];

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 2.1: Classification of the different proposals for the use of NLU and NLG techniques in BPM, process mining, workflow management, event data and related realms.

Approach	Task	Input	Output
Friedrich et al. [116]	Generate process models from text	Text process description	BPMN process model
Leopold et al. [117, 120]	Classify grammatical style of activity labels	Activity labels	Label style and semantics
Leopold et al. [118]	Generate labels for process activities	Process model	Activity labels
Leopold et al. [121]	Refactoring non-standard activity labels	Activity labels	Corrected labels
Leopold et al. [122, 123]	Generate textual descriptions of process models	BPMN process model	Textual description
Leopold et al. [124]	Multi-language naming violations detection	Process model element	Naming convention
van der Aa et al. [125]	Inconsistency detection in text- vs model-based process descriptions	Text- and model-based process description	Likelihood of inconsistencies in model-text pair
Ackermann et al. [126]	Text descriptions of declarative process models	DPIL model	Textual description
Wilbik and Dijkman [127]	Linguistic summarization of event logs	Event log	Linguistic summaries set
Dijkman and Wilbik [128]	Linguistic summarization of event logs	Event log	Linguistic summaries set
van der Aa et al. [44]	Translate PPI descriptions to formal notation	Text describing the PPI	Formal PPI definition
Qian et al. [129]	Generate textual descriptions of process models	BPMN process model	Textual description
Sànchez-Ferreres et al. [130, 131]	Align text- and model-based process specifications	Text- and model-based process description	Optimal alignment
Han et al. [132]	Query event logs in natural language	Text query	DSL query and result
Alman et al. [133]	Declarative specification generation from text	Text process description	Declarative process model
Kobeissi et al. [134]	Query event logs in natural language	Text query	Cypher query and result
Barbieri et al. [45, 135]	Query process mining tools in natural language	Text query	API call to PM tool
Chouhan et al. [136]	Linguistic summarization of event log anomalies	Event log	Linguistic summaries set
Barón-Espitia et al. [137]	Configure simulation scenarios conversationally	Simulation parameters	Result of simulation
Caspary et al. [138]	Semantic-based anomaly detection	BPMN model	Detected anomalies
Resinas et al. [41]	Compute PPIs based on textual descriptions	Text describing the PPI	Value of the PPI
Revina and Aksu [111]	Process complexity estimation supported by text	Text data and event log	Complexity of the process

conformance and complexity analysis supported by natural language [111, 138], the generation of textual descriptions of process models [122, 123] and linguistic summarization of event logs [127, 128, 136, 139]. For each proposal we indicate the problem they try to solve, the source of information they use for the proposed task, and the output of the system. Enhancements or expansions of proposals, have been grouped to ease the reading of the table. Further details the most relevant proposals are given in the following sections.

It is imperative to note that while Table 2.1 showcases a wide spectrum of applying natural language to process mining challenges, it does not provide an exhaustive enumeration of all existing approaches within the domain. Other studies and publications, while significant, have not been included in the table. Mainly, works describing the aggregation of different

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

existing techniques into tools [140], theoretical discussions on the challenges and potentials of natural language in process mining [141, 142], with special attention to LLMs [112, 114, 115], state-of-the-art reviews [113] and books that provide a summary of various techniques in this domain [143]. While these works contribute to the overall understanding and development of the field, they do not introduce or apply new NLU or NLG methods for process mining and thus are excluded from the table, which primarily aims to catalog novel or distinct approaches in applying NLP or NLG to process mining.

Given the emphasis of this Ph.D. dissertation on the generation of natural language descriptions in process mining, special attention is paid to the evaluation and comparison of the capabilities in various NLG approaches. Additionally, while NLU methodologies are not the primary concern of this thesis, a preliminary discussion on relevant proposals grounded in NLU is provided due to their contributory insights that have influenced certain aspects of the research and solutions devised in this thesis.

2.1.1 NLU in process mining

Natural language understanding systems try to grasp the semantics of natural language utterances and translate them machine-readable representations. These systems confront problems such as the ambiguities inherent in linguistic expression and the variability of syntactical and semantic structures, making this a very challenging task. Advanced NLU techniques have been developed to tackle these issues, including syntactic parsing, semantic role labeling, and entity recognition, which together facilitate a the comprehension of the context and intent of a natural language text. Many of these techniques haven been used in the process mining domain targeted at multiple problems.

One of the pioneering works in the use of language in process mining is [116]. Authors propose an automatic technique to generate BPMN (Business Process Model and Notation) models from unstructured natural language text, leveraging a combination of NLP (Natural Language Processing) tools and an anaphora resolution mechanism specifically developed for this purpose. From the natural language input, a syntactic structure is created identifying words and their grammatical relationships. After this, semantic analysis is performed based on FrameNet[1] and WordNet[2] for identifying semantic roles and retrieving meaningful related concepts for each word, respectively. After this, the anaphora resolution mechanism resolves

[1] framenet.icsi.berkeley.edu
[2] wordnet.princeton.edu

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

references made by pronouns or demonstratives to the appropriate referenced terms. With this annotated information, a BPMN model is constructed. In this work, authors present an innovative approach to business process modeling, addressing the significant time and resource expenditure involved in capturing as-is models for process management projects. However, not without its limitations. The complexity of natural language presents challenges in accurately capturing the nuances and implicit knowledge within process descriptions. The system's reliance on the quality and specificity of the input text may limit its effectiveness, as ambiguities or variations in linguistic expression can lead to inaccuracies in the generated models.

Another series of works focuses on analyzing activity labels in process models for recognizing different labeling styles [117, 120], detecting naming-convention violations [124] and refactoring or generating new activity labels [118, 121]. In [117] authors focus on the systematic analysis of activity labels by proposing a rule-based algorithm that leverages part of speech tagging and grammatical structure analysis to identify various labeling styles found in business process models. This approach serves as a tool for quality assurance in process repository management by flagging labels that require human correction. One issue is that part of speech taggers typically require complete sentences to perform optimally, but activity labels often consist of only a few words, which can lead to inaccuracies. Moreover, while the algorithm can flag potentially poor labels, the task of correcting or refining these labels still falls to human modelers, which can be a resource-intensive process.

Building upon this previous foundation, in [121], authors focus on the refactoring of activity labels that do not meet quality standards, aiming to improve their quality by standardizing the labeling styles. The system operates by first recognizing the labeling style of each activity label. Once the labeling style is recognized, the technique then focuses on deriving an action and a business object from the activity label. This task as involves parsing the label to identify the main verb that represents the action and the noun that represents the business object involved. After identifying the action and business object, the final phase is the composition of a verb-object activity label. This newly composed label is intended to be clearer and more aligned with best practices in process model labeling, which favor the verb-object style for its superior comprehensibility.

In [124], introduce an automatic technique for detecting naming convention violations on different process model elements (activities, events and gateways) that is based on text corpora and independent of linguistic resources such as WordNet, making it adaptable to a broad set of languages. Authors employ text corpora and linguistic knowledge to create a

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

part of speech determination and a disambiguation component. The ambiguity resolution process takes the context of a label in incremental steps (the label itself, the process model, the process model collection and knowledge on word frequencies derived from the employed text corpus) to determine the correct part of speech of each term. With the computed part of speech information, they decide on the corresponding linguistic patterns and respective violations. Accepted conventions are derived from guidelines and recommendations from both research and practice, as well as from consultations with industry partners who provide their internal process model guidelines. This corpus-based approach represents a significant upgrade to rule-based methods, providing a more adaptable tool for organizations to maintain the integrity of their process model repositories.

In a similar manner, in [120], authors take an update on previous proposals by leveraging machine learning-based techniques — in particular Hidden Markov Models (HMMs) — to overcome the inherent limitations of rule-based systems like on the previous proposal. This new technique does not require the manual specification of rules, which makes it more flexible and capable of handling a wider variety of grammatical styles, including less common ones. However, the reliance on HMMs also introduces potential problems. The accuracy of the model is highly dependent on the quality and representativeness of the training data. If the training data is not comprehensive or lacks certain grammatical patterns, the model may not perform as expected.

Natural Language Interfaces

A natural language interface (NLI) is a type of user interface that allows users to interact with a system using natural language. The core idea behind an NLI is to enable users to communicate with a system or computer in natural language, rather than through a series of commands or structured queries that require specific syntax. NLIs are built upon the capabilities of NLU enabling the system to parse, understand, and interpret the user's natural language inputs and then perform actions, provide responses, or retrieve information based on that understanding. Recent work in PM includes the proposal of natural language interfaces for dealing with information retrieval from event data, interaction with process mining tools and the definition of PPIs, among others.

In [41, 44], authors propose a NLI for translating textual descriptions of PPIs into a formal notation whose components can be mapped to concepts used in specific PPI frameworks, such as PPINOT [144], that allow for their computation. In [44], authors define a series of PPI

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

templates to which the natural language inputs of the users are matched. This is done through the semantic annotation of the input text with HMMs. These models are trained to recognize patterns in the text that correspond to different components of a PPI, such as measurement types (e.g., time, count), aggregation functions (e.g., average, sum), and related process events (e.g., order received, order completed). Once the relevant concepts are extracted, the second step involves aligning these concepts with the corresponding elements of a process model. They refer to this second step as *domain value resolution* and is performed with semantic similarity measures together with a series of constraints established based on process-expert knowledge in oder to obtain correct results from a semantic perspective.

In [41] authors update and extend this previous approach. The previous *semantic annotation* and *domain value resolution* stages are now called *entity extraction* and *entity matching* respectively, but they still have the same targets as before. A fine-tuned LLM [3] is now used for entity extraction, which represents a more flexible and potentially more accurate alternative to HMM-based systems. The use of a model that has been trained on a large corpus of text allows to understand language in a more nuanced and flexible way, as the model can adapt to the context and variety of expressions found in natural language descriptions of PPIs. The matching of annotated entities to elements for the process is performed based in semantic similarity measures as before, but in a more nuance way, as authors introduce a series of different target domains for each possible semantic tag. For example, chunks with AGR tag correspond to aggregation functions, and the matching function targets the domain U_{agg} which includes functions like average, maximum, minimum and percentage. Additionally, authors introduce a series of heuristics for improving the matching process and completing PPI descriptions that may be missing some information. In summary, the inclusion of a fine-tuned language model represents an evolution from the HMM-based approach, offering more flexibility and greater accuracy in interpreting and matching textual PPI descriptions with event log data. However, both approaches must deal with the inherent complexity of natural language and the specific challenges of aligning it with structured process data. While both approaches may handle a wide variety of linguistic expressions, it may still struggle with very domain-specific jargon or extremely irregular patterns that were not covered in the training data.

In [45, 135] authors propose a conversational interface that enables users to interact with process mining tools through natural language queries. This interface is built upon a reference

[3] Refers to the process of taking a pre-trained model and further training it on a smaller, domain-specific dataset. This allows the model to adapt to the specifics of a particular task or dataset without learning from scratch.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

architecture that combines classic NLP techniques with an abstract logical representation for process mining queries, aiming to translate user questions into API calls for existing process mining tools. These techniques are used to interpret the user's intent and convert it into a logical query that can interact with process mining tools via API calls. The architecture is designed to be independent of any specific process mining tool, which adds versatility to their approach. A significant contribution of their work is the creation of a dataset of natural language questions related to process mining, which is intended to support the development of conversational interfaces in this domain. Additionally, they present a proof of concept by integrating their architecture with Everflow, a commercial process mining tool. While the approach is innovative, it faces typical NLP challenges such as handling ambiguity and variability in natural language, as well as ensuring context awareness in conversations. These issues could affect the system's ability to accurately interpret and respond to user queries, especially when dealing with complex or domain-specific language. Authors emphasize the need for further research to enhance the robustness and accuracy of the conversational interface, especially in handling a wider range of natural language expressions and maintaining context over a conversation.

Natural Language Interfaces to Databases (NLIDBs) were proposed for querying database information through natural language without the need to learn complex query languages such as SQL. They aim to remove the knowledge barrier users face when querying databases, as they need not just to know the structure of the data but also a particular querying language [51]. The core challenge when creating an NLIDB system involves grasping what the user means in their natural language query and translating the user intent into the correct database query to retrieve the correct data from the database.

Following this research line, in [132], authors introduce an interface for querying event data stored in Elasticsearch[4] databases using natural language. Their proposal is backed by the ontology-based NLIDB system ATHENA [51, 145]. which they extend with a bootstrapping method that derives a domain ontology from the stored event data. The ATHENA system uses the derived ontology for parsing and annotating the input query with the ontology elements that is capable of identifying. Based on these annotations, an Ontology Query Language (OQL) query is generated and, through a rule-based system, translated into a Domain-Specific Language (DSL) query that can be run agains an Elasticsearch database. This approach presents a series of inherent limitations. The reliance on domain-specific artifacts to enrich the ontology may not capture all nuances of natural language used by users, potentially leading to

[4] elastic.co

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

misunderstandings of user intent. Furthermore, the only possible insights that may be extracted from the data are those directly present or obtainable through the operations (groupings, aggregations, etc.) supported by Elasticsearch. Additionally, the system's performance is benchmarked on datasets that may not fully represent the complexity of real-world business queries, which could affect its effectiveness in practical applications.

In a similar manner, in [134] authors present a natural language interface (NLI) system designed to facilitate querying of event data within process mining. The system allows users to input queries in plain text, which are then automatically translated into structured Cypher[5] queries to be executed over a graph database. The authors focus on structural queries, which pertain to event data, and propose a hybrid system that combines machine learning for natural language understanding (intent detection and named entity recognition) with a rule-based approach for constructing Cypher queries. The system is evaluated using a real-life event log from the BPI challenge'17, demonstrating increased accuracy in the NLI system. The input to the system is a user's natural language question about some aspects of event data, and the output is the corresponding structured Cypher query that can be executed over the graph database to retrieve the desired information. Authors introduce an innovative approach to querying process execution data using natural language, which is a significant step forward in making data analytics more accessible. However, the reliance on a hybrid system that combines machine learning with a rule-based approach for constructing Cypher queries may present limitations. Machine learning models require extensive training data, which can be a bottleneck if diverse linguistic expressions are used in queries. The rule-based component, while providing structure, may not be flexible enough to handle the nuances and variability inherent in natural language, potentially leading to misinterpretations or a lack of support for more complex queries. Compared to previous works, the shift towards intent-based query interpretation is a notable advancement. However, the challenges of semantic understanding, context retention, and the translation of user intent into accurate database queries remain persistent issues. These systems often struggle with the ambiguity of natural language and the mapping of terms to their database counterparts, which can be further complicated by domain-specific jargon. The evaluation on a specific dataset like the BPI challenge'17 event log suggests that while the system performs well in controlled scenarios, its adaptability to other domains or datasets is not guaranteed, which is a common critique in NLIDB research.

These systems present innovative methods for leveraging natural language in the process

[5] neo4j.com/developer/cypher/

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

realm. However, they all exhibit a common challenge. These proposals are primarily focused on the input side of the interaction — understanding and translating user queries into formal representations for different types of data queries — while less attention is given to expressing complex results back to the user in an equally natural and comprehensible manner. The lack of natural language output means that while querying may be simplified, the interpretation of results still requires a deep understanding of process mining concepts and data, limiting the usability for non-expert users. The gap between the sophisticated outputs of process mining and the users' ability to fully comprehend these results without specialized knowledge remains a significant barrier to the democratization of process mining tools. Additionally, users need to be cognizant on the limits of the underlying databases or systems used for the computation of the responses to their questions or inputs.

2.1.2 NLG in process mining

The integration of natural language within BPM and Process Mining has predominantly leveraged natural language understanding systems for interpreting and analyzing process-related language data. On the contrary, the application of NLG techniques stands as a frontier to explore despite the potential of natural language to enhance the ubiquitousness of process mining and BPM, specially, increasing the accessibility of process mining to non-expert users. The intersection of NLG and process mining has the potential not only to expand the reach of process analytics to a broader audience but also facilitating a more interactive and user-friendly approach to understanding and improving processes.

More than a decade ago the difficulty in understanding graphical representations of process models was recognized [122, 123]: while business analysts are adept at creating and working with process models, domain experts find it challenging to interpret and validate these models due to their complexity and the specialized knowledge required to understand them. This disconnect can lead to inefficiencies and misunderstandings in the process design and improvement efforts. Trying to address this communication gap between business analysts and domain experts, authors propose a technique to convert the non-sequential structure of process models into sequential, understandable text i.e.: starting from a BPMN process model, a natural language text describing the process model is constructed. In their approach, authors follow the classical NLG pipeline [56, 63] proposed by Reiter and Dale that consists on three main tasks: text planning, sentence planning and linguistic realization. In an initial phase, linguistic information extraction is performed in a similar manner as in [117, 118, 121]. Following this,

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

The process begins when the Room-Service Manager takes down an order. Subsequently, the process contains a region which allows for different execution paths. One option from start to end is the following:

- *The Room-Service Manager checks the customer solvency. If it is not ok, the Room-Service Manager debits the guest's account.*

However, the region also allows for a number deviations:

- *In case of a premium customer the Room-Service Manager approves the order without debit.*
- *After checking the customer solvency, the Room-Service Manager may also approve the order without debit.*

Then, the process is split into 2 parallel branches:

- *If it is necessary, the Room-Service Manager gives the order to the sommelier. In case alcoholic beverages were ordered, one or both of the following activities are executed:*
  - *The Sommelier prepares the alcoholic beverages.*
  - *The Sommelier fetches the wine from the cellar.*
- *The Room-Service Manager submits the order ticket to the kitchen. Then, the Kitchen prepares the meal.*

Once both branches were executed, the Waiter delivers to the guest's room. Afterwards, the process is finished.

Figure 2.1: Example of a process description generated by the tool (extracted from (123))

a Refined Process Structure Tree (RPST) is generated, which helps in structuring the process model for text generation. The RPST serves as a backbone for the text structuring phase, ensuring that the narrative flow is logical and follows the process model's sequence of activities. The next step involves DSynT (Dependency-Syntactic Tree) message generation. DSynT is a representation that captures the grammatical relationships between words in a sentence, which is essential for generating sentences that are syntactically correct and semantically meaningful. This representation is then used to refine the messages, ensuring that the information is presented in a clear and concise manner. Finally, the RealPro realizer takes the XML-based DSynT as input and outputs grammatically correct sentences. An example of the resulting textual-based specification of a process that can be generated with this approach is shown in Section 2.1.2.

This approach presents the first significant advancement in the domain of process model

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

accessibility and comprehension. Providing non-expert users with the means to analyze processes through detailed textual descriptions, thereby democratizing the understanding of process models. However, this approach has a notable limitation: the technique is limited to the structural description of process models and does not extend to the actual execution of processes. This distinction is crucial because a process model is merely a blueprint or a design, while the process itself is the enactment of that design in a real-world setting. The absence of execution data, such as event logs, means that the generated texts provide a static and potentially idealized view of processes, particularly if the model used as input is a theoretical process model and no discovery techniques are applied. These process models do not capture the dynamic nature of process execution, where deviations, exceptions, and inefficiencies often occur; usually, many discrepancies are found between the designed process model and the actual running processes [3, 4]. Consequently, the approach may overlook the rich insights that can be gleaned from analyzing the actual performance of a process, insights that are invaluable for process enhancement and optimization. By not incorporating execution data, the approach misses the opportunity to offer a retrospective analysis of the process, which could identify bottlenecks, compliance issues, or variations from the intended model. Such analysis is essential for continuous process improvement and for bridging the gap between theoretical models and practical execution. Therefore, while the technique is adept at translating the static structure of BPMN models into natural language, its utility in supporting the analysis and enhancement of processes in operation is inherently limited. Furthermore, when dealing with more flexible o unstructured processes — that usually derive in spaghetti processes — the textual description of a process model would render a text so long and intricate that would be as difficult to understand as a model-based representation.

In [129], authors propose an enhanced algorithm named Goun to address the limitations of the algorithm previously introduced by Leopold et al. (that they abbreviate as Hen). The Hen algorithm was designed to translate BPMN models into textual descriptions to bridge the understanding gap between various stakeholders, but it struggled with complex elements, such as text annotations and complex gateways, and was limited to free-choice models, leading to potential information loss. Goun aims to overcome these issues by introducing an Extended Process Structure Tree (EPST) to express more model patterns and developing a new linearization technique to improve the coverage and expressiveness of the generated text. While Goun demonstrates an ability to support more elements and complex structures, thereby enhancing the model-to-text conversion, it is not without the same drawbacks as before. The

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

approach still primarily focuses on the structural description of process models, rather than the processes themselves, which means it does not take into account the execution data from event logs. This could limit the approach's utility for analyzing and enhancing processes based on actual performance data, as it provides a static view that may not reflect the dynamic nature of process execution.

In a similar manner, in [126] authors address the complexity of declarative process models and the challenges they pose for business analysts who often struggle with their interpretation. The paper proposes a technique to transform these declarative models into intuitive natural language texts, focusing on content determination and structuring the output texts. This transformation is particularly challenging due to the flexible nature of declarative models, which define execution constraints rather than a fixed sequence of steps. The authors build upon the Declarative Process Intermediate Language (DPIL) for its expressiveness and multi-perspective capabilities, which include control flow, data, and resources. The approach distinguishes between two contexts: report generation and execution assistance, tailoring the tense and complexity of the language accordingly. In the report generation context the generation process starts from an event log and the DpilMiner discovery algorithm [146] is used to generate a declarative process model from the execution data. As previous approaches, this paper follows Reiter and Dale's pipeline [56, 63] model for natural language generation, which involves document planning, micro-planning, and surface realization, to ensure the generated texts are grammatically and orthographically correct. Figure 2.2 shows an example report generated by this approach.

While this approach offers a novel way to aid in the understanding of declarative process models, it may still faces limitations due to the capabilities of DPIL. in capturing the dynamic nature of process execution and the nuances that come with it, similar to the previous articles discussed. The simplification of structure patterns into a linear narrative can obscure the inherent complexity and flexibility of declarative models, which are not strictly sequential but are defined by a set of constraints that allow for multiple valid execution paths. This linearization could potentially lead to oversimplification, where the rich interplay of constraints and choices available in the actual process is not fully conveyed. DPIL, despite its multi-perspective nature, may not fully capture the nuances of business processes. It is designed to represent control flow, data, and resources, but the static nature of the language may not reflect the dynamic and sometimes unpredictable ways in which processes unfold in real-world scenarios. While the use of event logs can provide historical data to inform the generation of natural language

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Overview
Two tasks have been performed where one role and one concrete person were involved. The involved role is professor. SJ is the only fixed identity, which means that he was involved very frequently.
One mandatory rule seems to restrict the process and in two cases there is a recommendation.
Performed Tasks
We only consider tasks that were performed with a certain frequency. These tasks are Approve Application and Book Flight. All restrictions are explained in the following section.
Constraints and Advice
1: The application was always approved by a professor and usually by SJ.
2: Furthermore, the flight was booked after approval of the application.

Figure 2.2: Exemplary natural language report generated with the DPIL approach (extracted from (126)).

descriptions, this retrospective view does not necessarily offer predictive insights or a deep understanding of the process dynamics. Event logs can tell us what has happened, but they do not always explain why it happened or predict what will happen next, especially in a declarative model where many different paths might be valid. Therefore, despite the use of event logs, the approach may still fall short in providing comprehensive insights into the processes. It excels in creating readable descriptions for those who are not familiar with process modeling notations, but for process enhancement and analysis, particularly for the purpose of understanding complex process behaviors and improving them, this method may not be sufficient. The descriptions generated are useful for communication and documentation purposes but may not replace the nuanced analysis required for process optimization and decision-making.

Linguistic summarization of process data

LDD techniques have been extensively applied in the literature in multiple problems and domains, from querying interfaces [147] and data mining [148] to the description of time series [149], including weather reports [71] or healthcare data [68]. Based on fundamental ideas of linguistic summarization proposed by authors in [97, 98, 99], a series of works [127, 128, 139, 141] elaborate on a technique for the linguistic summarization of event logs.

In [141] authors introduce a research agenda and reflect over the need for techniques like linguistic summarization to extract relevant insights from process execution data. The system

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

they envision would utilize event logs from business processes to generate linguistic summaries that reflect the main characteristics of these processes. This approach is seen as complementary to data visualization, providing a verbalization of data insights. The authors identify several key challenges, including the need for preprocessing event logs and the difficulty of summarizing sequences of activities within processes. They also recognize the need for a framework to determine which features of the event logs are most pertinent for analysis — and therefore, more relevant to be described.

In [127], authors start addressing the previously stated challenges by proposing a series of protoforms focused on process data. Namely, a case focused protoform for summarizing information about cases *"In Q cases, there was P"* and a sequence focused protoform for describing sequences of events *"In Q cases, sequence XYZ was P"* and their extended versions *"In Q cases, when condition R was fulfilled, there was P"* and *"When condition R was fulfilled, sequence XYZ was P in Q cases"*, respectively. These protoforms serve as a template for the generation of summaries like *"In most cases, there was 'accept' as the end result"* or *"In most cases sequence "Registration, Analysis" was in the beginning"*. These protoforms are designed to extract insights on frequently executed tasks or sequences and their attributes like throughput and service time. The paper presents a series of example linguistic summaries through a case study with real-world process data.

The main challenge when generating linguistic summaries is to automatically extract true summaries from the data. To quantify this validity or *truthfulness* of a summary, different quantification methods grounded on fuzzy set theory can be used [102, 150]. The elements involved in the protoforms, namely, quantifiers, summarizers, and qualifiers have their semantics grounded on fuzzy sets with an associated membership function that evaluates how well elements of the target dataset are represented by a particular linguistic term. The most naive strategy to generate a series of linguistic summaries of a particular dataset is to iterate over all possible summaries — based on the exhaustive combination of all the possible elements that conform the protoforms proposed for a particular task — and keep the ones that are true based on the chose quantification method. This approach results in a really large number of statements due to the all possible combinations of the elements that conform a protoform.

For this reason, in [139] authors propose a method that leverages clustering techniques to group similar sequences of events, thereby reducing the number of summaries while retaining significant information. This method is particularly focused on sequences described as "Q cases contain a sequence like ABC", where ABC represents an ordered set of actions. This is a

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

particular case of the case focused protoform proposed previously by authors in [127]. The proposed method aims to provide a quick overview of an event log without overwhelming the user with too much information. The approach involves generating all possible subsequences present in the data, calculating the degree to which each sequence is present in each trace, and then selecting potential summarizers based on frequency. A distance matrix is built for these sequences, and a feasible number of clusters is estimated. Levenshtein distance is used to measure the similarity between sequences. Then, for each cluster, a medoid is identified to serve as the summarizer of all sequences that belong to it. Finally, the best quantifier is chosen for each summarizer to form a summary.

In their previous approach [127], Levenshtein distance was already used to evaluate whether a sequence present to some degree in a particular case. However, they use a verbalization for this protoform that does not hint to this possible difference between the sequence described in a summary and the actual sequence that was present in a set of cases. An example of this is summary *"In most cases, there was a sequence 'Registration, Analysis, Decision"'*. In this second approach [139], authors introduce the word "like" in the case focused protoforms that use sequences as summarizers, to effectively indicate that the sequence present in the may not be exactly the one described in the summary, but a similar sequence of events.

In [128] authors build upon previous work by introducing conjunctive conditions, different similarity measures for clustering summary statements, and an efficient algorithm for managing the summary space. The clustering of sequences is improved and authors explore three different similarity metrics: edit distance similarity, Jaccard similarity and an adaptation of the Markov clustering authors refer to as Markov similarity. The clustering is based purely on the similarity of sequences, and while the paper discusses the potential for a more semantically rich grouping of sequences, it does not commit to a specific similarity metric for the final approach. by utilizing pruning and clustering of linguistic summaries. In this work, authors also propose a pruning technique which further reduces the number of summary statements compared to previous work. This work's novelty lies in its algorithm for exploring and pruning the large space of possible summary statements and its exploration of different measures of similarity for clustering. Authors evaluate the technique's effectiveness in reducing summary statements and its practical utility among practitioners.

In a similar fashion, in [136], authors propose an algorithm for the generation of linguistic summaries of anomalies (differences) among groups of similar cases. The proposed method assumes anomalies have already been detected and focuses on isolating these from normal cases

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

to study their behavior in comparison. The approach involves finding similar normal cases for each anomalous case, determining differences, and then generating linguistic summaries to explain these differences. However, no details on how the differences between normal and anomalous cases are discovered are provided. Furthermore, the generation of summaries is only based on protoforms; the identified differences are not explicitly used in the generation process.

These works (cf. [127, 128, 136, 139, 141]) introduce an innovative approach to the generation of natural language descriptions of event logs based on linguistic summarization. Aiming to make the vast amounts of process data more comprehensible to non-expert stakeholders. However, a critical examination reveals several potential shortcomings, primarily stemming from the absence of underlying process models in the summarization process. This omission is significant as it overlooks the complex nature of business processes, which often include parallel executions and decision points that cannot be captured through the linear sequences of activities captured in event logs alone. The proposed clustering technique, while adept at reducing information overload by grouping similar sequences, relies heavily on syntactic similarity measures. This reliance could lead to the combination of distinct process instances that, while syntactically similar, may have vastly different semantic implications and outcomes. Such an approach risks generating summaries that are not only semantically impoverished but could also misrepresent the true nature of the underlying processes. For instance, two process instances that differ only in the execution of parallel tasks might be clustered together, obscuring important differences in performance or outcomes. Potentially leading to biased and incorrect insights that may result in changes that worsen the performance of a process instead of enhancing it.

In their initial work, authors recognize the need for a framework to determine which features of the event logs are most pertinent for analysis — and therefore, more relevant to be described. However, the lack of integration with more sophisticated data analysis techniques further than quantification methods for addressing the most frequent aspects of an event log limits the utility of the proposed method. In this sense, research shows that not only frequent information is relevant, but many times, is the infrequent behavior the most salient and the one that needs to be addressed when improving a process [151].

In conclusion, while the articles contribute to the field of process data analysis by offering a method to distill complex event logs into more digestible summaries, the approach could benefit from incorporating process models to ensure that the generated summaries accurately reflect the multifaceted nature of business processes. While holding the potential to address

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 2.2: Approaches for the generation of natural language descriptions of processes from the state of the art and the types of descriptions or summaries they support.

	Activity frequency	Attribute quantification	Attribute correlation	Temporal context	Transition frequency	Transition duration	Attribute conformance	Group comparison	Control-flow
Wilbik and Dijkman [127, 128, 139]	+	+	−	±	−	±	−	−	−
Chouhan et al. [136]	+	+	−	−	−	−	−	±	−
Leopold et al. [122, 123]	−	−	−	−	−	−	−	−	+
Ackermann et al. [126]	±	±	−	−	−	−	−	−	+
Qian et al. [129]	−	−	−	−	−	−	−	−	+
Our approach	+	+	+	+	+	+	+	+	+

the lack of execution data in model-based description approaches, this approach lacks of the opposite. Without the integration of process models and event data, there is a risk that the linguistic summaries may lead to an oversimplified view of processes, potentially resulting in decisions that do not align with the best interests of process optimization.

2.2 Concluding remarks

In this chapter, an introduction to Natural Language Understanding (NLU), Natural Language Generation (NLG) and Linguistic Descriptions of Data (LDD) was provided, along with a review of the state of the art in the use of natural language in the Business Process Management (BPM) and Process Mining (PM) domains. Specifically, we described how natural language can be harnessed in process mining and how NLP techniques can be leveraged for allowing the use of natural language both as input and output of process mining techniques. Through the analysis of a series of works it was shown how natural language represents a powerful means for enhancing the accessibility and ubiquity of process mining. From allowing to query event data through natural language queries, to the generation of natural language descriptions of process models.

The current landscape of research on natural language processing within the domain of process mining reveals a dichotomy between two methodologies: one that generates textual descriptions of process models without considering execution data, and another that provides linguistic summaries of event logs without reference to the underlying process model. On one hand, the description of process models offers a static and potentially idealized view, neglecting the rich insights that could be gleaned from actual process execution data. On the other hand,

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

the linguistic summarization of event logs, while capturing the dynamics of process execution, often overlooks the structural and behavioral constraints imposed by the process models. This disconnection can lead to a superficial analysis, where the generated natural language outputs may not fully encapsulate the complexities and nuances of real-world process interactions. Table 2.2 contrasts these approaches, highlighting the gaps in their support for a series of comprehensive insights identified as significant in the literature [3, 152, 153, 154, 155, 156]. The table underscores the need for an integrated approach that leverages the power of process models with the dynamic, execution-based insights from event logs, providing a more holistic and actionable understanding of a process.

To address these shortcomings, research should aim to integrate both model and data perspectives, leveraging the strengths of both approaches to produce narratives that not only describe the *"what"* and *"how"* of process models but also the *"why"* and *"when"* of process execution. Such integration requires sophisticated methods that can interpret and correlate the static structure of process models with the dynamic behavior of event logs, potentially through advanced machine learning techniques or hybrid models that combine rule-based systems with statistical learning and, ultimately, harnessing the capabilities of process mining techniques. Only then can we approach the generation of comprehensive natural language descriptions that encapsulate all perspectives of a process, thereby enabling more informed decision-making and process refinement.

This approach should focus on providing comprehensible process information through natural language by leveraging the capabilities of NLG systems for converting different data sources into elaborate and understandable and the capabilities of linguistic summarization techniques for handling the imprecision of language by defining linguistic concepts through fuzzy sets. An NLG-based solution for generating insights on data are the so-called Data-to-Text (D2T) systems. These systems are designed to generate natural language texts from non-linguistic (typically numerical) input data, effectively providing linguistic descriptions of the data. In essence, a linguistic description conveys insights extracted from data in natural language. This involves two principal activities: the extraction of knowledge, akin to Knowledge Discovery in Databases (KDD), which seeks to uncover new, potentially useful, and comprehensible insights from data; and the formulation of this knowledge into natural language, enhancing its accessibility and practicality. A core element for the communication between both tasks is a knowledge representation formalism that describes the underlying semantics of the text that conforms the linguistic descriptions of the data. In this sense, it

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

is widely recognized the suitability and potential of Fuzzy Set Theory for representing the semantics of natural language expressions [101, 157]. Moreover, process mining techniques shall be harnessed for discovering the real model of a process according to the execution data, relating the execution data to the process model to obtain a holistic view of a process and exploit both conformance and enhancement process mining techniques to generate new insights on the execution data of a process that can be leveraged by statistical and machine learning techniques to determine the most salient aspects of a process.

In spite of the recent advances in NLG, identifying the most relevant information to communicate in a text is still challenging. In the particular case of PM, there is no common agreement on which are the relevant aspects of a process, as these are domain-, process- and potentially, user-dependent. For this reason, the incorporation of an interactive component is essential to fully harness the potential of natural language-based process mining analysis. Such interactivity would enable the tailoring of descriptions to specific user needs, ensuring that the insights provided are not only accurate but also contextually relevant and immediately applicable. This need for interactivity in process analysis tools is underscored in [128], who emphasizes the importance of user-driven attribute selection in linguistic summarization. The evolution towards an interactive platform thus represents a crucial step in the progression of the generation of natural language descriptions of processes, ensuring their applicability and effectiveness in a wider range of real-world scenarios.

This Ph.D. dissertation fully addresses these points in the context of generating natural language descriptions of processes:

- In Chapter 3 new protoform-based natural language descriptions description of processes are proposed involving quantitative and qualitative information about a process together with temporal, frequency and domain-specific information about a process.

- In Chapter 4, a fuzzy logic-based knowledge representation formalism which formalizes the previously proposed protoform-based model together with the process domain into an ontology is proposed. This formalism serves as the base for the Process-to-Text architecture. A process mining-based D2T architecture for the automatic generation of natural language descriptions of processes. This architecture is also presented in Chapter 4, together with its application in a use-case in the healthcare domain and a evaluation assessing the validity of the proposal.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Chapters 4 and 5 focus on introducing interactivity into the generation process. Chapter 4 introduces the details on how an interactive tool for the generation of natural language descriptions of processes and how the linguistic realization stage of the D2T architecture is with machine learning-based models to enhance the quality of the generated texts.
- Chapter 5 introduces C-4PM, the first conversational interface for declarative process mining. The C-4PM system allows for the interaction with declarative process specifications through natural language. By understanding natural language user queries, preforming the according reasoning tasks and generating appropriate responses, in natural language, C-4PM allows non-expert users to interact with declarative process specifications in an accessible and transparent manner.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

CHAPTER 3

A PROTOFORM-BASED MODEL FOR THE DESCRIPTION OF PROCESSES IN NATURAL LANGUAGE

In general, complexity and precision bear an inverse relation to one another in the sense that, as the complexity of a problem increases, the possibility of analyzing it in precise terms diminishes. Thus fuzzy thinking' may not be deplorable, after all, if it makes possible the solution of problems which are much too complex for precise analysis.

— Lofti A. Zadeh

In Chapter 1 we introduced the necessity for new means of conveying process-related information, aiming to make process mining ubiquitous. Natural language was presented as a potential alternative to current visualization techniques and process mining interfaces due to its potential in facilitating the understandability of process-related information and thus the accessibility of process mining analysis and techniques for non-expert users. Compared

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

to visual representations of data and complex human-computer interfaces — the state-of-art methodologies for interacting with processes — natural language texts represent a better way to capture and summarize the most characteristic aspects of some data in a way that can be easily consumed by users. As *i)* natural language is the inherent way of communicating for humans, therefore it does not rely on their capabilities to identify or understand patterns, trends, etc. from visual representations; and *ii)* it may include uncertain terms or expressions, which are very effective for communication. Supporting this hypothesis, research shows that in many domains knowledge and expertise are required to understand graphical information [53, 54] and proves that domain experts can take better decisions based on natural language descriptions or a combination of natural language and visual support, than solely relying on graphical displays [52].

In Chapter 2 techniques for the generation of insights from data in the form of natural language texts were introduced, namely Natural Language Generation (NLG), with a particular focus on Data-to-Text (D2T) systems as well as Linguistic Descriptions — or summaries — of Data (LDD). These techniques aim to provide users with natural language texts that capture and summarize the most salient aspects of some data. D2T systems can process large amounts of numerical and symbolical data, distilling it into comprehensible texts that contain relevant and understandable information. These systems include an initial data analysis stage where the most relevant information is identified and abstracted from the data, and a generation stage where the extracted information is aggregated and distilled into natural language texts that convey the most salient aspects of the input data. Nevertheless, a significant limitation of these systems is the lack of semantic modeling. This shortfall is particularly relevant, as the systems need to precisely discern and interpret the underlying semantics of input data to efficiently generate accurate and meaningful output texts.

In this chapter, we propose a solution based in Fuzzy Set Theory, in particular, a protoform-based model that can serve as a knowledge representation formalism in a D2T system designed for the generation of natural language descriptions of processes. A series of natural language descriptions that highlight salient aspects of a process are proposed together with a series of protoforms supporting the semantics of these sentences. In Section 3.1 a deeper background in LDD techniques, protoforms and fuzzy quantification methods is given. Section 3.2 introduces the proposed protoforms together with a framework for enhancing their expressiveness through a semantic-based approach for the combination of protoforms. This framework allows for protoforms that are more expressive both in terms of semantics and syntactic structures than

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

state-of-the-art proposals. Section 3.3 presents a preliminary evaluation of the proposal and some concluding remarks.

The contents of this chapter are extracted from the following publications:

© 2021 IEEE. Reprinted, with permission, from Fontenla-Seco, Y.[a], Bugarín-Diz, A.[a], & Lama, M.[a] (2021). **Fuzzy Temporal Protoforms For The Quantitative Description Of Processes In Natural Language**. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–6). Luxembourg, Luxembourg. ISBN: 978-1-6654-4407-1. doi: 10.1109/FUZZ45933.2021.9494444.

[a]Centro Singular de Investigación en Tecnoloxías Intelixentes, Universidade de Santiago de Compostela, Spain.

Fontenla-Seco, Y.[a], Lama, M.[a], González-Salvado, V.[b], Peña-Gil, C.[b], & Bugarín-Diz, A.[a] (2022). **A Framework For The Automatic Description Of Healthcare Processes In Natural Language: Application in an Aortic Stenosis Integrated Care Process**. *Journal of Biomedical Informatics*, 128, 104033. ISSN: 1532-0464. doi: 10.1016/j.jbi.2022.104033.

[a]Centro Singular de Investigación en Tecnoloxías Intelixentes, Universidade de Santiago de Compostela, Spain.

[b]Department of Cardiology, University Clinical Hospital of Santiago de Compostela, SERGAS IDIS CIBERCV, Santiago de Compostela, Spain.

3.1 Linguistic Descriptions of Data

Under the Soft Computing umbrella, within the Fuzzy Set Theory, Linguistic Descriptions of Data (LDD) — or linguistic (data) summaries — have been developed to extract pre-defined patterns in some data and verbalize them in the form of concise, natural language sentences [95, 96, 97, 98, 99]. In other words, linguistic summarization can be defined as a descriptive knowledge discovery technique which aims to find out patterns that summarize the existing relationships among attributes in a dataset through abstract conceptual representations [158]. Whilst traditional data summarization often relies on statistical measures such as median, mean, and standard deviation, which, while precise, may not always convey insights in an intuitively understandable manner. Linguistic data summarization seeks to bridge this gap by transforming data into natural language statements that are more aligned with everyday language, thus enhancing comprehension and relatability [97].

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

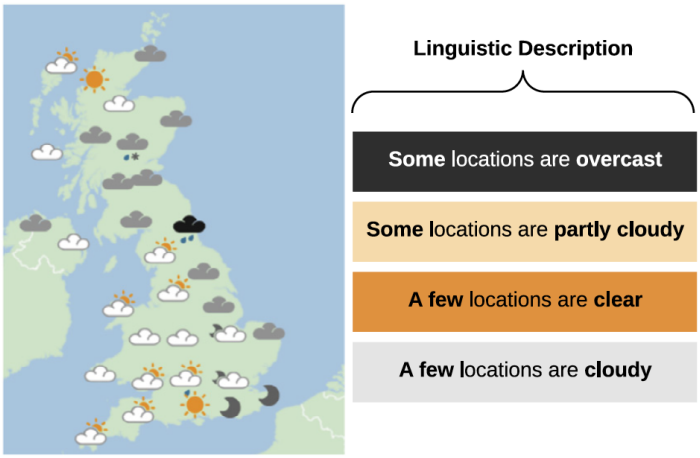


Figure 3.1: The straightforward nature of this example facilitates a clear comprehension of the compositional relationship between a linguistic description and its constituent sentences (or instantiated protoforms). This distinction is crucial, as the literature sometimes blurs these concepts by using them interchangeably, leading to confusion. (extracted from (103)).

The term LDD refers both to the discipline itself and the techniques it encompasses as well as the typical output of these techniques. In this thesis, we will use LDD (Linguistic Descriptions of Data) to refer to the discipline itself, and linguistic (data) summary[1] to make reference to the output of LDD algorithms. A linguistic summary is then understood as a set of sentences that summarize knowledge about some data [103]. Usually, single sentences composing a linguistic summary are also referred to as *linguistic summary*, as authors use indiscriminately the same term to refer to a set of sentences or a single sentence generated with LDD techniques (cf. [97]).

Sentences that are part of a linguistic summary follow pre-defined patterns or formal structures. These patterns, or conceptual representations are known as protoforms — short for *prototypical form*. Protoforms serve as a knowledge representation formalism which can be employed to capture the semantics of sentences that make up a linguistic summary through the semantics of protoforms themselves and their components [60, 100]. For example, the protoform *X is A* — known as a *possibilistic constraint* [95] — can be used to model the

[1] In the literature the term linguistic summary is often used as a synonym for the concept of *linguistic description*, specially in the NLG field [60] (cf. [159]).

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

semantics of the sentence *"temperature is high"*. Different protoforms have been proposed in the literature tailored towards specific tasks. These range from basic qualitative summaries such as *"the average salary of the employees is low"*, to more nuanced summaries that utilize linguistic quantifiers to convey the extent or frequency of a particular attribute within a dataset, exemplified by *"many rainy days are cold"*. The latter, is known as a fuzzy quantified sentence.

It is important to notice that there is not a one to one mapping between protoforms and their final verbalizations. While protoforms serve as abstract representations for the generation of linguistic summaries, they often do not align precisely with the final verbalization in these summaries. This misalignment is due to the inherent complexity and richness of natural language, which protoforms may not fully encapsulate. Despite their suitability for the extraction of relevant knowledge from a dataset and its linguistic representation, further linguistic refinement is necessary for the purpose of providing users with rich and comprehensible texts [160, 161].

Fuzzy quantified sentences can be categorized based on the number of properties they describe, known as the n-arity of the quantifier. Type-I fuzzy quantified sentences (unary) involve one single property and represent the most basic form of fuzzy quantification, following the pattern "$Q\ Ys$ are S". This pattern allows for the generation of summaries like *"Most days are rainy"*. Type-II fuzzy quantified sentences involve two properties and follow the pattern "$Q\ KYs$ are S", giving a more nuanced look inside data, an example is *"Many rainy days are cold"*. Fuzzy quantified sentences are recognized in the literature as the most prominent form of protoform [103, 158], in particular, unary and binary quantified propositions [60, 102, 103]. They have been extensively applied in multiple domains [162]; from data mining [147, 148] or text categorization [163], to time series summarization [149] of healthcare data [68] or the generation of weather reports [71].

Although type-I and type-II protoforms are the most used fuzzy quantified sentence structures in the literature, their lack of expressiveness and limited semantics [103, 161] restrict their efficacy in supporting and accurately conveying information from more complex domains. It becomes necessary to propose new and different types of more complex protoforms to give support to richer and more insightful linguistic summaries.

3.1.1 The process of linguistic summarization

Linguistic summaries are understood as sets of instanced protoforms that are computed according to a dataset and a knowledge base for a given application and domain with the

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

objective of summarizing knowledge about variables and their values [103]. For the formulation of linguistic summaries, a series of essential elements are needed:

- A **referential** $Y = \{y_1, \ldots, y_n\}$, a set of objects for which certain property or set of properties holds (e.g. the set of cases from an event log) to be described.
- **Linguistic values** used to indicate the value of some property of the referential of interest, in a linguistic manner. For example, a linguistic value could be *"long wait time"*. Linguistic values are used as *summarizers* and *qualifiers*.
  - *Summarizers* succinctly describe — summarize — a specific property of the referential. A summarizer like *"long wait time"* encapsulates a quantitative measure into a qualitative, easily graspable linguistic expression.
  - *Qualifiers* provide context or additional descriptive layers to the summarizers. They enhance the precision or scope of the summarizers, allowing for more detailed and accurate descriptions.
- **Aggregation operators** such as *"and"* or *"or"*, are used to combine multiple linguistic values into a single, more complex expression, such as *"long wait time and negative outcome"*. This allows for the creation of descriptions that capture multiple aspects or characteristics of the referential.
- **Fuzzy quantifiers** (in the case of quantified sentences) such as *"many"* or *"several"* which are used to express the quantity or proportion of elements in the referential which fulfill the property indicated by the summarizer.
- **Protoforms** used as templates for structuring linguistic summaries, where the referential, linguistic values, and quantifiers are filled in. Protoforms have associated semantics, guiding the formation of meaningful, coherent sentences that accurately reflect the data. They are crucial in establishing a standardized, yet flexible, framework for the linguistic summarization of data.
- **Evaluation criteria** used to determine the relevance of each generated sentence. The degree of truth T is the fundamental evaluation criteria, as it represents how well a sentence reflects the data.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

For instance, based on these elements, a fuzzy quantified sentence like *"In several cases there was a long wait time"*, can be generated using the type-I protoform

$$Q\ Y\text{s are } S\,, \tag{3.1}$$

where Q is a quantifier (*"several"* in the example), Y is a referential (*"cases"* in the example) and S is a summarizer of Y (*"long waiting time"* in the example). In type-II fuzzy quantified sentences, a qualifier is added to the description to better define the scope of the sentence, giving place to type-II protoforms. A qualifier K, as the summarizer S verbalizes the value of a property or aggregation of properties of the referential. Sentences like, *"In most cases where patients were males, there was a long wait time"* can be generated using the type-II protoform:

$$Q\ KY\text{s are } A\,. \tag{3.2}$$

Aggregation operators are essential tools for building richer and more complex expressions that effectively capture multiple characteristics of the referential under description in one single statement. These operators, like *"and"* or *"or"*, aggregate different linguistic values to form a more detailed description, such as *"In most cases patients were males and had a long wait time"*.

3.1.2 Linguistic variables

Both summarizer and qualifier are defined as linguistic values [164]. A linguistic variable, such as *"Age"* or *"Wait time"* is named with a term that describes a quality or characteristic of the elements in a dataset. The values of a linguistic variable — known as linguistic values — are words or phrases from a natural language rather than numerical values. For example, the linguistic variable *"wait time"* might have linguistic values like *"low"*, *"medium"* and *"high"*. Linguistic variables model the universe of discourse (domain) of a non-linguistic variable — categorical or numerical — into several linguistic values. Linguistic values are defined as expressions, typically words or phrases, each semantically represented by a fuzzy set [99]. They are associated with membership functions that quantify the degree to which values in the domain of the variable in question satisfy the properties implied by these linguistic terms. This definition can be formalized as follows:

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Definition 5: ***Linguistic Variable, Linguistic Value, Membership Function, Membership Degree***

Let us consider an observable quality V that objects from the referential Y manifest. The quality V could be any conceivable characteristic, like age, salary, hair color, years of education, etc. and take on both numeric and nonnumeric values in $X = \{x_1, \ldots, x_n\}$. This quality V can be modeled by a linguistic variable, which partitions the domain of V into linguistic values. Each linguistic value is associated with a membership function that quantifies the extent to which an element in Y exhibits such linguistic value of the property V.

Let S be a linguistic value corresponding to a property of V. The membership function of S, denoted as μ_S, is defined as:

$$\mu_S : Y \to [0, 1] \tag{3.3}$$

where for each element $y \in Y$, the value $\mu_S(y)$ represents the degree of membership of y in S. This function characterizes the extent to which y possesses the property represented by S, which is a linguistic interpretation of the observable quality V.

The function μ_S is characterized as follows:

- For a **crisp (non-fuzzy) set**, $\mu_S(y)$ is called indicator function and takes values in the set $\{0, 1\}$, where:
  - $\mu_S(y) = 1$ if y exhibits the property S fully,
  - $\mu_S(y) = 0$ if y does not exhibit the property S.
- For a **fuzzy set**, $\mu_S(y)$ takes any value in the interval $[0, 1]$, indicating the degree to which y belongs to S. A value of 1 represents full membership, 0 represents non-membership, and any value between 0 and 1 represents partial membership in the linguistic interpretation of V.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

We will denote the value of V for a specific object $y \in Y$ as $V(y)$, thus establishing a mapping $V : Y \to X$. So, for obtaining the membership degree of elements in the referential Y to S, the following composition is used:

$$\begin{aligned} Y &\xrightarrow{V} X \xrightarrow{\mu_S} [0,1] \\ y &\longmapsto V(y) \longmapsto \mu_S(V(y)) \end{aligned} \tag{3.4}$$

For simplification, we assume the mapping $V : Y \to X$ as implicit when evaluating elements in Y against linguistic value S. As a result, we will use $\mu_S(y)$ to refer to the membership degree of $y \in Y$ to S.

Consider, for example, a numerical variable containing the possible values of the wait time between two events, expressed in days and defined in $\{1, 2, ..., 600\}$. Within this context, we define the linguistic variable *"Wait Time"* in the universe of discourse $\{1, 2, ..., 600\}$, with linguistic values {*"minimum", "low", "medium", "high", "really high"*}, each represented as a fuzzy set, associated with a trapezoidal membership function, as illustrated in Figure 3.2. For instance, linguistic value *"low"* is defined using a non-increasing membership function, $\mu_{low}(x)$, over the universe of discourse. In a simplified piece-wise linear model[2], this membership function can be described as follows:

$$\mu_{\text{low}}(x) = \begin{cases} 0, & \text{if } x < 0, \\ 1, & \text{if } 0 \leqslant x \leqslant 200, \\ \frac{260-x}{260-200}, & \text{if } 200 < x \leqslant 260, \\ 0, & \text{if } x > 260. \end{cases} \tag{3.5}$$

It is important to note that the definition of what constitutes *low* in terms of wait time is inherently subjective. The definition of linguistic variables, values and their corresponding membership functions is a domain dependent problem, where knowledge about the variables and their values is needed in order to properly define the semantics of linguistic terms. Thus, the exact shape and parameters of membership functions could vary depending on the specific

[2]A piece-wise linear model refers to a method of modeling a function by dividing it into several linear segments, each defined over a specific interval of the independent variable(s). This approach simplifies the representation of complex functions by treating them as a series of linear equations, each applicable to a distinct sub-range of the data, effectively approximating non-linear relationships in a more manageable and interpretable form. In the context of fuzzy logic, such a model is used to define the membership functions of fuzzy sets in a more straightforward manner.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

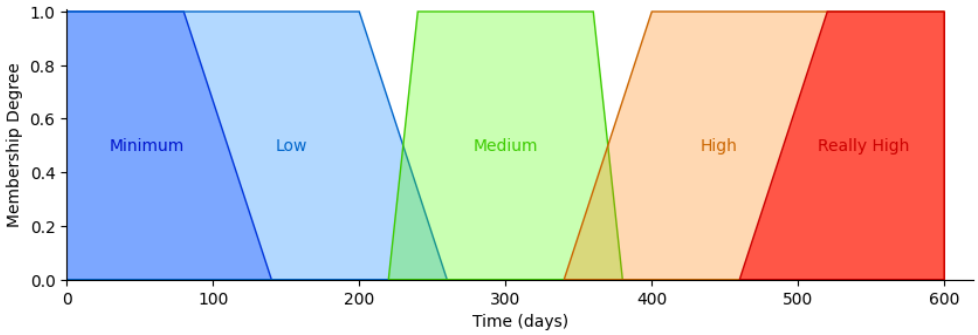


Figure 3.2: Graphic representation (as fuzzy sets) of the five linguistic values (*"minimum", "low", ..., "really high"*) of the linguistic variable *"Wait time"*.

problem, context or user interpretation. In terms of the type or shape of membership functions, any function that maps values from the input domain on $[0, 1]$ may be used, being triangular, trapezoidal, and Gaussian functions the most used in the literature. Membership functions based on classical (crisp) sets, known as indicator or characteristic functions, map values on the input domain to the set $\{0, 1\}$. These can be seen as particular cases of fuzzy sets [161]. For instance, consider a linguistic value defining boolean condition *"on"*, which is inherently crisp, may be defined with a crisp membership function as follows:

$$\mu_{on}(y) = \begin{cases} 1 & \text{if } y = on \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

In practical applications, the definition of membership functions can be either predefined based on standard criteria or tailored based on user input or expert knowledge. This applies not only to this particular example, but to the general process of defining linguistic variables and values. The general process of defining linguistic variables in a fuzzy system is known as *system identification* [165] (see Section 4.2.3).

The evaluation of an element from a referential (i.e., a dataset) against a linguistic value involves the use of its membership function, which quantifies the degree to which the element belongs to the fuzzy set represented by the linguistic value, i.e., the extent to which the element from the referential shows the property enunciated by the linguistic value. The output of this process is the *membership degree* of the element. This numerical value then enables the association of the linguistic expression associated with the linguistic value to the element, representing a qualitative interpretation of its value in the context of the referential.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

3.1.3 Quantifiers

A *quantifier* in fuzzy logic is a linguistic term used to describe the quantity of elements within a set that meet a specific criterion [96]. It extends the concept of linguistic variables and values by providing a way to express amounts or degrees in linguistic terms, rather than precise numerical values. A myriad of fuzzy quantifiers have been introduced in the literature, such as universal, existential, absolute, proportional, comparative, exception, and more [96, 166, 167, 168]. Among these, relative quantifiers, such as *"most"*, *"many"*, and *"several"*, and absolute quantifiers like *"about 5"* or *"around 10"*, are particularly prevalent. Each of these quantifiers can be interpreted as a fuzzy set over a universe of discourse, which in this case, represents proportions or quantities. Absolute fuzzy quantifiers are defined in $\mathbb{R}$, as the universe of discourse is the set of all values that the fuzzy quantifier can possibly take. On the other hand, relative quantifiers, as they defined proportions or relative frequencies, are defined in $[0, 1]$. For instance, a set of proportional or relative quantifiers — those quantifiers relating a proportion over the domain $[0, 1]$ — can be defined as a series of fuzzy sets as in Figure 3.3, where:

- The quantifier *"most"* is represented by a fuzzy set where values close to 1 indicate a high degree of agreement with the concept of *"most"*, while values closer to 0 indicate a lower degree of agreement.
- Quantifier *"many"* might have a membership function that peaks at a lower value than *"most"*, reflecting its interpretation as a smaller quantity.
- This same notion applies to the rest of the quantifiers defined in this example, each membership function will have a maximum value at the point or interval which better reflects the quantity conveyed by the linguistic term associated to the quantifier.

The natural way of representing fuzzy quantifiers is as fuzzy numbers. These are a special type of fuzzy set used to represent quantities that are uncertain or imprecise. Fuzzy numbers are often used to represent quantities like *"approximately 5"* or *"around 10"*, but they can also be used to represent any type of imprecise numerical value. For example, when talking about temperature, *"almost 30"* could be a fuzzy number where the value 30 has the highest degree of membership, and values close to but lesser than 30 have progressively lower degrees of membership and values higher that 30 have zero membership. In this role, they function similarly to absolute fuzzy quantifiers, providing a way to deal with numerical imprecision in a linguistic or qualitative manner. However, it should be noted that the idea of a fuzzy

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

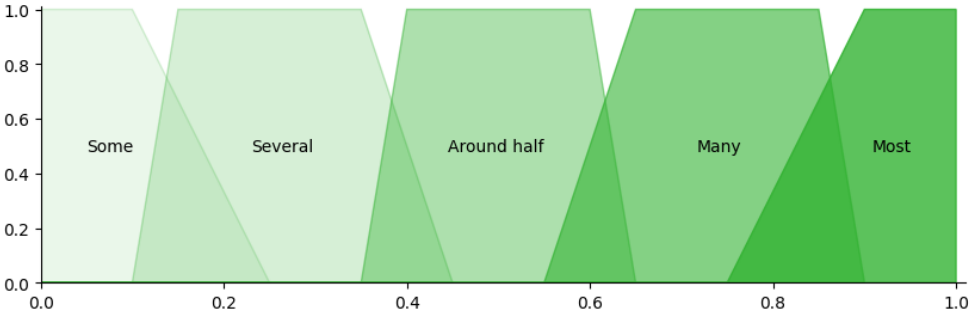


Figure 3.3: Graphic representation (as fuzzy sets) of relative quantifiers.

quantifier is not identical to that of a fuzzy number. In a broader sense, a fuzzy quantifier is then seen as a fuzzy number that offers a fuzzy description of the absolute or relative number of elements in one or more fuzzy or non-fuzzy sets. Take, for instance, the phrase *"Tom owns a few books"*. Here, *"a few"* serves as a fuzzy description of the quantity within the non-fuzzy set of *books owned by Tom*. Or the phrase *"many cases had a really high wait time"*, where *"many"* quantifies in a fuzzy manner the number of cases with a *"really high wait time"*.

Fuzzy numbers, in particular, possess a series of specific mathematical characteristics that differentiate them from general fuzzy sets. However, for the scope of this thesis, we will adopt a simplified approach. We will define both fuzzy numbers and fuzzy quantifiers as types of linguistic values, each associated with its own membership function. The set of all linguistic values defined for a particular problem, partitioning the domain, conform a quantity linguistic variable. As for any linguistic value, the membership function associated to a fuzzy number assigns a degree of membership to each possible value in its domain. For instance, for the quantifiers defined in Figure 3.3, a set of trapezoidal membership functions were defined. A trapezoid function $T[a, b, c, d]$ can be defined as follows:

$$\mu_{T[a,b,c,d]}(x) = \begin{cases} 0, & (x \leqslant a) \; or \; (x > d) \\ \frac{x-a}{b-a} & a < x \leqslant b \\ 1, & b < x \leqslant c \\ \frac{d-x}{d-c} & c < x \leqslant d \end{cases} \tag{3.7}$$

In a similar manner to what happens when defining linguistic variables and values, the interpretation of what *"many"* means is subjective and depends in aspects such as the data

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

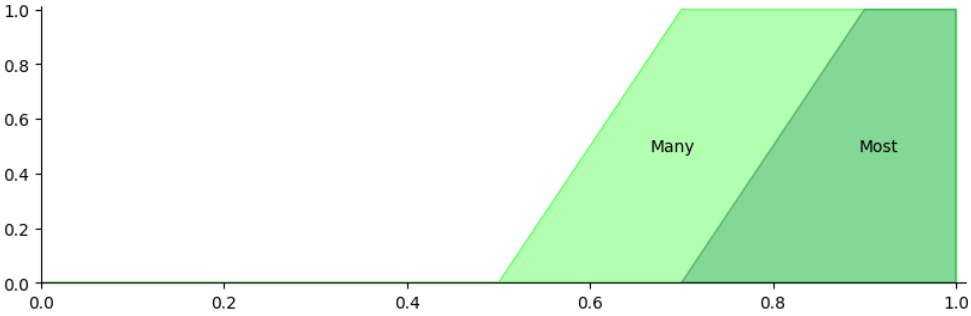


Figure 3.4: Monotonically, non-decreasing relative quantifiers.

under description or the context in which a particular sentence is generated or delivered to the user. The definition of the semantics of quantifiers follows then the same process as linguistic variables and its values. For instance, Figure 3.4 shows the definition of relative quantifiers representing values *"many"* and *"most"* as two monotonically non-decreasing fuzzy sets, in contrast with the definition of these same quantifiers provided in Figure 3.3.

Cardinality of a set

A concept which plays an essential role in the treatment of fuzzy quantifiers is that of the cardinality, a concept similar to the notion of cardinality of a classical (crisp) set. Informally, the cardinality of a referential Y $card_Y$ of a fuzzy set F is a real or fuzzy number which serves as a count of the number of elements *"in"* F. A simple way of extending the concept of cardinality concept of crisp sets to fuzzy sets is through a scalar definition of the fuzzy cardinality, based on the $\Sigma count$ (*sigma-count*) or power of a fuzzy set [96, 168], which is the arithmetic sum of the membership degrees of the elements of a referential — $e \in E$ — to a fuzzy set F. This is expressed as:

$$card_Y(F) := \Sigma\, count(F) = \sum_{y \in Y} \mu_F(y) \tag{3.8}$$

Fuzzy cardinality extends the classical notion of cardinality (which counts the number of elements in a set) to fuzzy sets, where elements have varying degrees of membership. Instead of just counting the number of elements, fuzzy cardinality considers the sum of their membership degrees in the fuzzy set. This concept is specially useful for linguistic summarization as it applies to both fuzzy and crisp sets, with the cardinality of a crisp set being a specific instance

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

of fuzzy cardinality. In our case, we will talk about the cardinality of a referential $card_Y$ — or the cardinality of the predicate when talking about a linguistic summary — to express the sum of the membership degrees of the elements of the referential to a linguistic value S, obtained through the evaluation of these against the membership function of the value.

So when generating linguistic summaries, first, the cardinality of the referential is computed and then, the *compatibility* of the cardinality with a quantifier Q is computed in order to obtain a single numerical value relating the validity of the proposed predicate. A quantifier allows to evaluate the aggregate membership degree of all the elements in the referential by evaluating the cardinality against its membership function.

3.1.4 Fuzzy temporal references

Apart from being used as quantifiers, in practical applications, fuzzy numbers are often used in scenarios where precise numerical data is either unavailable or unnecessary. For example, in estimating the time required to complete a task, one might use a fuzzy number to represent *"about 30 minutes"*. Knowing that the time aspect is of high relevance when dealing with processes (see Section 3.2) as it is precisely the control-flow and temporal aspects that difference the analysis of processes from other type of structured data. Based on a simplified notion of fuzzy number — as for quantifiers — we propose fuzzy time references to represent and handle the use of imprecise temporal expressions in natural language descriptions of processes.

We use as a base the fuzzy temporal model introduced in [61] for the definition or fuzzy temporal references. In particular, in the proposed model, time is conceptualized as a discrete and linear sequence, represented by an ordered set $\tau = t_0, t_1, ..., t_n$. Here, t_0 denotes the origin of time, and each subsequent t_k signifies a distinct and precise moment in time. A uniformly spaced temporal framework is defined by ensuring a constant difference, denoted as δ, between any two consecutive time points (t_k and t_{k-1}). Thus, $\delta = t_k - t_{k-1}$ is the same for all adjacent pairs in the sequence.

Given this structure, we define an event as an occurrence that takes place at a specific instant in time, without any extended duration. Under this definition, if the temporal separation between two events is less than the interval δ, we categorize these events as occurring simultaneously. This approach allows us to discretize time in a manner that aligns with the temporal granularity determined by δ, facilitating a consistent and systematic interpretation of event timings.

As any other linguistic value, a fuzzy time reference will have a linguistic term and a membership function associated. Despite the underlying temporal representation being discrete,

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

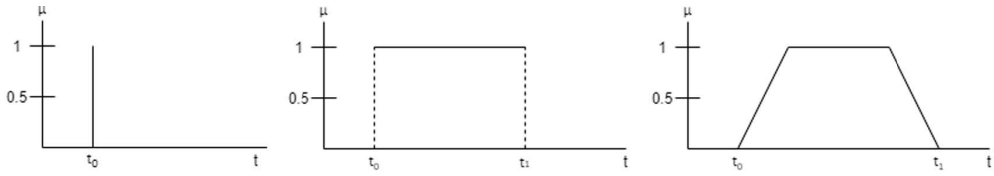


Figure 3.5: Fuzzy temporal references defined from left to right as an instant, an imprecise time interval and a fuzzy time interval (extracted from (61)).

the membership function of a fuzzy temporal interval is defined in $\mathbb{R}^+$ for practicality. The linguistic term associated with a fuzzy reference effectively conveys the underlying temporal notion of that interval. This flexibility in representation allows the same period to be expressed in different ways, accommodating various linguistic styles and contexts. For instance, a specific time span can be referred to as the *"first semester of the year"* or alternatively as *"between January and June"*, both conveying the same time frame but in distinct linguistic forms. This versatility is key in adapting to diverse communication needs and preferences, providing a range of descriptive options for the same temporal interval. Figure 3.5 shows three different fuzzy temporal intervals defined with different membership functions.

In order to effectively apply fuzzy sets and fuzzy logic to temporal data within our reasoning system, it is essential to convert date representations into numerical values. This conversion process enables the use of fuzzy sets and fuzzy logic in computations involving time references. By representing dates as numerical values, we can directly apply fuzzy membership functions to these dates, allowing us to evaluate the degree to which a specific date aligns with a given fuzzy time reference. An important aspect of this process is the selection of an appropriate level of temporal granularity. The choice of granularity – whether it be days, hours, minutes, or another unit – should be made in the reasoning system using this fuzzy temporal representation, as this choice determines the resolution at which time is quantified and influences how dates are converted to numerical values. The integration of this conversion mechanism within a reasoning system allows for the seamless application of fuzzy logic principles to temporal data. It not only facilitates the handling of dates in a computationally efficient manner but also ensures that the fuzzy temporal references are applied consistently and meaningfully, in line with the chosen level of temporal granularity.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

3.1.5 Evaluating a linguistic summary

For the generation of a linguistic summary — here understood as a single sentence, a protoform must be instantiated. Instancing a protoform involves assigning values to the elements on its structure — quantifier, qualifier and summarizer in the case of fuzzy quantified sentences, and then computing its associated truth degree i.e.: how well the proposition summarizes the referential under description. This way, an instanced protoform is characterized by the elements that are used to fill its structure (quantifier, summarizer, qualifier and the underlying referential) and its corresponding *truth degree*. Through this process, a sentence that summarizes knowledge about the referential is generated.

The truth degree — or truth value — of a sentence can be calculated with the use of a quantification model. Multiple models have been proposed in the literature [102, 150], being Zadeh's the most extended one [169, 170]. Fuzzy quantification methods enable the evaluation of the aggregation of membership degrees — i.e., the cardinality of the referential — with a fuzzy quantifier. Through this evaluation process, a single truth degree $T \in [0, 1]$ that characterizes the whole sentence is derived from the individual membership values of each element in the referential to the properties enunciated by the sentence. This truth value indicates the alignment of the sentence with the underlying data. The closer it is to one the more truthful the summary is. For instance, T *("Most places are wet")* $= 1$ means that *"most"* values of the referential fulfill the property *"wet"* in a high degree. The truth value is a measure of compatibility relies on four key factors: *i)* the dataset itself, *ii)* how the quantifiers are defined, *iii)* the linguistic terms derived from the properties within the referential, and *iv)* the chosen method for fuzzy quantification applied to assess the fuzzy quantified statement.

So when generating linguistic summaries, the computation of the truth degree is a two-fold process. First, the cardinality of the referential $card_Y$ is computed. This is done by computing the membership degree of each element in the referential to the property enunciated by the summarizer and then, through the chosen quantification method, aggregating all these membership degrees into a single value — the cardinality. Using Zadeh's quantification model, the $\Sigma count$ (see Section 3.1.3) is used as a cardinality measure. The second step involves evaluating the *compatibility* of the cardinality of the referential with a quantifier Q on the basis of its membership function. This compatibility measure results in a single numerical value relating the truth degree, or validity, of the sentence. Using Zadeh's quantification model, the process for instancing a protoform and computing its truth degree is as follows:

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

1. Consider a referential $Y = \{y_1, y_2, \ldots, y_n\}$ with a quality V that takes values in X.
2. Define a summarizer S as a linguistic value over X.
3. Let Q be a fuzzy quantifier, represented as a fuzzy subset of the interval $[0, 1]$ (relative quantifier) or $\mathbb{R}^+$ (absolute quantifier).
4. For each y in Y, calculate the membership degree $\mu_S(y)$, representing how well y satisfies S.
5. Compute the cardinality $card_Y$ of the referential i.e., the proportion of the dataset Y satisfying S — using the $\Sigma count$. With $|Y|$ being the crisp cardinality i.e., the number of elements, of Y.

   When using relative quantifiers such as *"several"*, this is done as the average of these membership degrees:

   $$card_Y = \frac{\sum_{y \in Y} \mu_S(y)}{|Y|}$$

   When using absolute quantifiers, such as *"around 50"* this is done as the sum of all membership degrees:

   $$card_Y = \sum_{y \in Y} \mu_S(y)$$

6. Finally, the truth value T of the summary is determined by the grade of membership of $card_Y$ to Q computed on the basis of the membership function μ_Q of the quantifier, indicating the agreement of the quantifier Q with the cardinality $card_Y$:

   $$T = \mu_Q(card_Y)$$

For the generation of a type-I protoform-based linguistic summary using relative quantifiers, this whole process can be formalized as follows:

$$T = \mu_Q\left(\frac{\sum_{y \in Y} \mu_S(y)}{|Y|}\right) \tag{3.9}$$

For type-II protoforms, where a qualifier K is used, the cardinality of the referential is derived from an aggregation operation between properties K and S. This process can be formalized as follows:

$$T = \mu_Q\left(\frac{\sum_{y \in Y} (\mu_K(y) \wedge \mu_S(y))}{\sum_{y \in Y} \mu_K(y)}\right) \tag{3.10}$$

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Now, the cardinality of the referential represents the aggregated degree to which elements in Y simultaneously satisfy both the qualifier K and the summarizer S. In the equation, the denominator represents the aggregate membership degree of Y to the qualifier K, adjusting the resulting cardinality to reflect the proportion of elements from the referential Y that meet the criteria set by summarizer S among those that meet the criteria set by qualifier K (e.g. the proportion of students that are that are *"young"* among the students that are *"tall"*). Essentially, this equation adjusts the truth degree calculation to focus only on those elements of Y that are relevant under the context set by K.

Aggregation is achieved through the *conjunction* operation which, in fuzzy logic, is generalized as a *t-norm*. A t-norm, or triangular norm, is a function that takes two fuzzy membership values as input and returns a new membership value representing their conjunction. A t-norm is represented with the symbol $\otimes$, and many different t-norms have been proposed in the literature such as the minimum (or Gödel), product, drastic, nilpotent or Lukasiewicz [171]. In this thesis, the t-norm minimum — represented as $\wedge$ — will be used, as its application is straightforward and intuitively aligns with the classical logic interpretation of the *"and"* operator. Furthermore, using the minimum t-norm tends to yield conservative estimates in fuzzy logic inference. It ensures that the aggregation does not overestimate the degree of membership in the intersection of fuzzy sets. In contrast to t-norms, fuzzy logic also employs t-conorms (or s-norms), which are used for modeling the *disjunction*, or *"or"* operation. The most common t-conorm is the maximum, which, like the minimum t-norm, offers an intuitive parallel to classical logic. Represented symbolically as $\vee$, the maximum t-conorm is frequently chosen for its straightforward interpretation and ease of computation. This choice complements the minimum t-norm used for conjunctions, creating a cohesive system for handling both conjunctions and disjunctions.

Other quality measures

In addition to the truth degree, other criteria can be used when evaluating linguistic summaries. This is known as the quality framework [60], and for each particular application, establishing this framework is subject to the particularities of the domain and experts involved. Other measures used to evaluate the quality of a linguistic summary include [97, 172, 173]:

- **Relevance:** Concerns the quantity of elements in the dataset that need to satisfy the stipulated condition to be deemed significant.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- **Accuracy:** Evaluates how accurately the description reflects the actual data.
- **Length:** In certain domains, the brevity of the description may be crucial, sometimes at the cost of reduced precision.
- **Coverage:** This metric assesses the scope of the data coverage by the description and identifies data excluded from the description.
- **Specificity:** Determines the generality of a description, bearing in mind that a more generic description tends to be less informative.
- **Compatibility of Descriptions:** When dealing with multiple descriptions, it's vital to understand their degree of incongruity.
- **Ambiguity:** Even if statements are compatible, they can still be ambiguous, potentially leading to user confusion.

Yager proposes [97] the use of the *informativeness* degree of a summary for its evaluation. Informativeness strikes a balance between specificity and generality. A summary that is too general may not be very informative, even if it is true. Conversely, a very specific summary might not capture the broader trends or patterns in the data. It's a comprehensive measure that goes beyond just being a reflection of the data; it's about the value the summary adds in understanding the data.

3.1.6 Generating linguistic summaries

The generation of linguistic summaries for some particular data involves the generation of the most relevant sentences according to the data through the process of protoform instantiation. This process can be conceptualized as a search problem, where potential sentences are systematically generated and evaluated against the dataset under description. This search, which may be exhaustive or heuristic, is guided by quality measures, such as the truth value, that to quantify how well a protoform instance (sentence) summarizes the dataset.

The search space, or semantic space, is defined by the powerset of the instance space. The instance space is the set of all protoform instances that can be generated for a particular problem according to a knowledge base purpose-built for that given application domain. This knowledge base comprises the protoforms and their components, such as quantifiers and linguistic variables, which have been defined for the particular problem at hand. The construction of this knowledge

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Algorithm 3.1 Calculating linguistic summaries

```
Parameters
θ : [0...1] is a truth degree threshold
D is a dataset
Q is the set of defined quantifiers, e.g. Q = {most, ..., some}
A is the set of defined linguistic values used as summarizers and qualifiers, e.g. A = {long waiting time, tall, low, high, ...}

Generate all type-I protoforms
typeI = ∅
for q ∈ Q, a ∈ A do
    if T(q cases, had attribute a) > θ then
        typeI = typeI ∪ {q cases, had attribute a}
    end if
end for

Generate all type-II protoforms
B = A
typeII = ∅
for q ∈ Q, a ∈ A, b ∈ B do
    if T(q cases with attribute b, had attribute a) > θ then
        typeII = typeII ∪ {q cases with attribute b, had attribute a}
    end if
end for
```

base is a critical step, as it defines the scope and nature of the semantic space within which the search process occurs. The definition of the exact structure and parameters of the membership functions of linguistic values and quantifiers can vary significantly depending on the context of a particular problem. When talking about fuzzy systems, the definition of this knowledge base, in particular the definition of linguistic variables and membership functions, in terms of number, shape and position over the universe of discourse, is part of a process known as *system identification* [165] (see Section 4.2.3).

Once the knowledge base for a particular problem has been defined, the most basic generation process consists in an exhaustive search. In the process, all the possible instantiations of protoforms are generated i.e.: for all the defined protoforms, all possible combinations of values for the elements in its structure (quantifier, summarizer and qualifier in the case of type-II protoforms) are generated and evaluated against the dataset. Algorithm 3.1 delineates this process.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 3.1: Approaches for the generation of natural language descriptions of processes from the state of the art and the types of descriptions or summaries they support.

Authors	Protoform	Verbalized example
Cariñena et al. [174]	X was A in T	Temperature was high in the last minutes
	In T, X was A	Shortly after the increase in pressure, temperature was high
Castillo-Ortega et al. [149]	Q of D are A	Most days of the cold season patient inflow was high
Almeida et al. [68]	Q Y's are P Qt times	Most patients have high blood pressure most of the time.
	Q Y's with C are P	Most patients with disease X have low blood pressure.
Wilbik and Dijkman [127]	In Q cases there was P	In most cases there was a short throughput time
	In Q cases, when condition R was fulfilled there was P	In most cases when "Registration" was short, there was a short throughput time

3.2 Protoforms for the description of processes in natural language

As it was introduced in Chapter 2, the current application of LDD in process mining [127, 128, 139] is focused only on event log data, without the application of process mining techniques, the scope of the possible analysis of a process is greatly reduced. Furthermore, as discussed in Section 3.1, the limited efficacy of type-I and type-II quantified protoforms in complex domains highlights the need for more expressive alternatives.

One direction followed recently in order to improve this limited semantics is the extension of existing protoforms with additional elements such as the temporal dimension — due to great availability of time series data — or any other domain specific information [103]. For instance, in domains like healthcare [68] or process management [128], several extensions of classical quantified protoforms have been proposed. On a minor scale some authors propose new protoforms that deviate clearly from standard quantified sentences [103] (cf. [175, 176, 177]). Table 3.1 includes some examples of protoforms proposed in the literature different to classical type-I and type-II quantified sentences[3]. It is important to notice that, despite providing extensions or modifications to classical quantified propositions, most proposals are still based in quantified sentences due to their relevance for summarizing data in an understandable manner.

In this section we propose a series relevant natural language descriptions based on extensions and modifications to the type-I and type-II quantified protoforms for the description of processes in natural language. Table 3.2 shows a summary of the proposed extensions, their associated protoform, and an example of its instantiation. In a similar manner to how Figure 3.1 illustrates

[3] A more extensive review of protoforms and applications can be found in [103, 161]

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 3.2: Proposed protoforms for the generation of natural language descriptions of processes.

Extension	Protoform	Example
Temporal contextualization of features	In T_i, Q cases had feature P	In year 2020, many cases had a long waiting time
	In T_i, Q cases with feature C, case had feature P	In year 2020, most cases where resource MANAGER was involved had a short waiting time
Frequency and temporal relations between activities	In T_i, in Q cases, R	In the first semester of year 2021, in approximately half of cases, activity A takes place shortly after activity B
	In T_i, in Q cases with feature C, R	In the first semester of year 2021, in many cases where the wait time was long, activity C took place long after activity A
Compliance	*In T_i, P_1 is expected. However, in Q cases with feature C, had feature P_2.*	In year 2020, a wait time of 30 days between activity A and activity B is expected. However, in many cases where activity C is executed, wait time between activities A and B is higher than 30 days
Comparison between groups	*In T_i, Q_1 cases with feature C_1 had feature C_2. However, Q_2 cases with feature C_2 had feature P_2*	In year 2020, many cases where resource MANAGER was involved had a short waiting time between activities C and D. However, most cases where resource APPRENTICE was involved had a long waiting time between activities C and D.

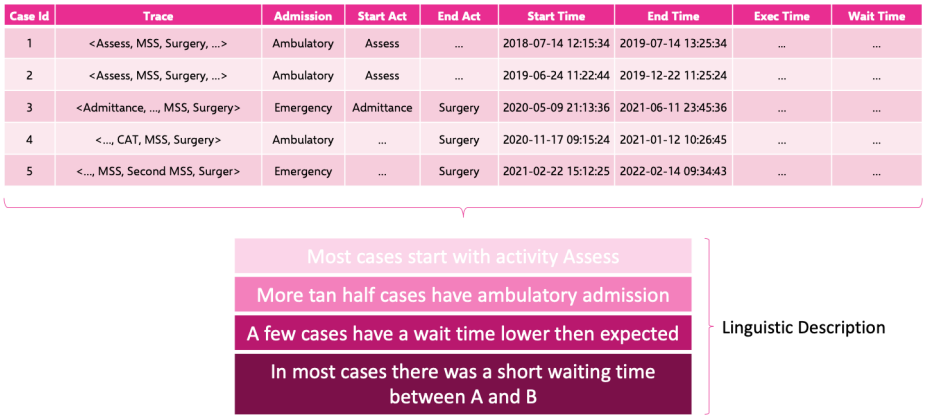


Figure 3.6: An example of a linguistic summary of a process based on an event log, similar to the example presented in Figure 3.1.

an examples of a linguistic summary of meteorological data, Figure 3.6 shows a linguistic summary of a process, based on an event log. In what follows, more detail is given on each of the proposed protoforms together with an illustrative example on what would be needed to properly instantiated.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

New features

Case Id	Trace	Admission	Start Act	End Act	Start Time	End Time	Exec Time	Wait Time
1	<Assess, MSS, Surgery, ...>	Ambulatory	Assess	...	2018-07-14 12:15:34	2019-07-14 13:25:34	...	...
2	<Assess, MSS, Surgery, ...>	Ambulatory	Assess	...	2019-06-24 11:22:44	2019-12-22 11:25:24	...	...
3	<Admittance, ..., MSS, Surgery>	Emergency	Admittance	Surgery	2020-05-09 21:13:36	2021-06-11 23:45:36	...	...
4	<..., CAT, MSS, Surgery>	Ambulatory	...	Surgery	2020-11-17 09:15:24	2021-01-12 10:26:45	...	...
5	<..., MSS, Second MSS, Surger>	Emergency	...	Surgery	2021-02-22 15:12:25	2022-02-14 09:34:43	...	...

Figure 3.7: The feature extraction process illustrated in a case-perspective event log. The attributes already present in the log, namely the case id, the trace, and an attribute called *admission*, are shown in grey. The newly computed attributes added to the log are highlighted in pink.

For the generation of linguistic summaries of a process we will focus on the case attributes. To do this, we create a view [31] of the process focused on cases and its attributes. These are attributes that refer to a case as a whole rather than an individual event (see Definition 3). Table 5.2 shows an example of the case attributes stored in an event log. This transformation reorganizes the data such that each row in the new case-perspective log represents an individual case from the original event log. This format is more conducive to LDD techniques, as it aligns the data structure with the requirements of linguistic summarization, enabling more effective extraction of meaningful patterns and insights from complex process data. For example, it becomes possible to generate summaries like *"Most cases have a short throughput time"* or *"Several cases with high number of choices have a high wait time"*.

Additionally, attributes to those originally present in the event log may be computed during a feature extraction process [12]. As any other data, event logs can be enriched with various additional attributes [31, 178]. Mainly, by deriving or computing additional event and case attributes based on the log itself, or by adding external data. For instance, the throughput time of a case, the waiting time between two particular activities, the number of times one activity triggers the execution of one another, etc., can be a computed using data from the log itself, while adding information on external elements like the price of gas at the time an event occurred is an example of including external data. New case attributes computed during this stage are added to the case-perspective event log previously defined. Figure 3.7 illustrates this with the event log used as a case study.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Before delving into the details of the proposed protoforms, a series of general considerations must be established:

- We understand the referential $Y = \{y_1, y_2, ...y_n\}$ of the proposed protoforms as the case-perspective representation of an event log of size n cases, where y represents each particular case.
- Case attributes will be modeled as linguistic variables and linguistic values to which the elements of the referential Y will be evaluated against (see Equation (3.7)) based on expert knowledge.
- We assume a summarizer S is a linguistic value for some case attribute V, (see Definition 5). We use $\mu_S(y)$ to denote the membership degree of a case $y \in Y$ to S. A qualifier K must also be a linguistic value of a case attribute V. Using these summarizers, we could, for example, speak of *short wait time* or *high number of choices*.
- Linguistic values are understood in a general way as a linguistic term associated with a membership function of any form, without restricting them to fuzzy sets. Since membership functions based on classical approaches (indicator functions) can be seen as particular cases of fuzzy sets [161]. Some approaches make a clear difference in this sense, by proposing different protoforms when dealing with linguistic values in a form different than fuzzy predicates, for example, when using crisp category data [68]. Here, for simplification and flexibility, protoforms are defined in a general way.
- As aggregation operators, the t-norm minimum operator $\wedge$ and t-conorm maximum $\vee$ are used. However, ny other pair of t-norm and t-conorms may be used [150].
- Zadeh's quantification model is used for the computation of the truth degree of the protoforms.

3.2.1 Temporal contextualization of attributes

The first question experts try to solve when analyzing a process is how cases characterize and how their features evolve during the execution of the process. This can be done by generating descriptions that relate the value of case features, and how features relate to each other, in different time intervals of the process. This gives a first understanding of what is happening in a process and helps on detecting groups of cases (grouped by the same feature or set of features), sections of the process or specific time intervals that should be analyzed in further detail.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

The first proposed protoform is focused on describing how attributes of cases behave during different stages of the process, for instance, based on the case study, sentences like *"In year 2020, most patients had emergency admittance"* can be generated. The extended protoform, also allows to get insights into relations between case attributes e.g. *"In year 2020, most patients who underwent a surgery were elderly"*. For the generation of these statements, protoforms (3.1) and (3.10) are extended with a temporal characterization, similar to the proposed in [175], resulting in the following protoforms respectively:

$$\text{In } T_r, Q\ Ys \text{ had attribute } S \tag{3.11}$$

$$\text{In } T_r, Q\ Ys \text{ with attribute } K, \text{ had attribute } S \tag{3.12}$$

Where, as in equation (3.1) and equation (3.10), Q is a quantifier S is a summarizer and K is a qualifier. S and K can make reference to any attribute or, through an aggregation process, a combination of attributes of a case. In cases where the summarizer or qualifier are linguistic values of a categorical attribute V we propose the use of crisp linguistic values. The membership of a value to these properties can be computed straightforward by taking $a = b$ and $c = d$ in expression (3.7). T_r is a time interval in which the attention on the attribute (or combination of attributes) described by S and K is focused.

The use of a fuzzy time reference Tr and a summarizer K allows for limiting the scope of the descriptions, in a similar manner to what *log filtering* does, a well-known and frequent data processing stage in process mining analysis [31] to reduce complexity or focus the analysis on a specific part of the dataset. This activity is often performed multiple times in an analysis iteration to obtain different perspectives on the event data.

With protoform (3.11) descriptions such as *"In year 2020, many cases had a long waiting time"* can be generated. Protoform (3.12) allows to check if any type of relation between features holds in a particular period of the process e.g., *"In year 2020, in most cases where resource MANAGER was involved, case had a short waiting time."*. It is important to note that S and Q can also make reference to temporal intervals that are a subset of the temporal interval originally denoted with Tr, for instance *"In year 2020, in most cases that took place in the first semester of the year, the waiting time was long"*. The truth value of protoform (3.11) can be directly calculated using [174] as:

$$T = \mu_Q \left(\frac{\sum_{y \in Y} (\mu_S(y) \wedge \mu_T(y))}{\sum_{y \in Y} \mu_T(y)} \right) \tag{3.13}$$

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

For the evaluation of protoform (3.12) we propose the following extension:

$$T = \mu_Q \left(\frac{\sum_{y \in Y} \left(\mu_S(y) \wedge \mu_{T_r}(y) \wedge \mu_K(y) \right)}{\sum_{y \in Y} \left(\mu_{T_r}(y) \wedge \mu_K(y) \right)} \right) \tag{3.14}$$

Through a process of double quantification (over time and over the summarizer), or triple, in the case of protoform (3.12) these protoforms allow both to contextualize relevant aspects of a process in time as well as to describe time and performance-related aspects of a process. This is done by aggregating the membership of each element in the referential (each case $y \in Y$) to each of the linguistic values that are part of the description through a t-norm. In this case, we propose the use of the t-norm minimum for this aggregation, represented with symbol $\wedge$.

3.2.2 Frequency and temporal relationship between events

The time perspective plays a crucial role in process mining, particularly in understanding and improving operational efficiencies such as wait times and control-flow dynamics. This perspective is instrumental in analyzing the relationships between activities within processes, offering insights into their frequency and duration. Protoforms that focus on frequency and temporal relationships are designed to provide insights into the control-flow perspective of a process by describing the paths that cases take as well as time-related information of these paths. Through the use of quantifiers, common and exceptional sequences of activities can be described. They help address key questions in process analysis, such as identifying prevalent paths, pinpointing bottlenecks, and understanding the factors contributing to these bottlenecks. This is particularly relevant in domains like healthcare, where process efficiency and timeliness can significantly impact outcomes.

Frequency and temporal relation protoforms aim to give a good understanding of the different paths cases follow (control-flow perspective), the frequency and wait times of activity relationships (frequency and time perspective) and how case features may influence case paths and wait times. These descriptions help to answer questions like “which are the most commonly followed paths and what exceptional paths are followed?”, “where are the bottlenecks in the process?” or “what may have caused the bottleneck?” which are some of the most frequent questions when applying process mining techniques. These are specially relevant highly time-sensitive domains, like the healthcare process presented as a case-study [152].

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

It is important to notice that, as opposed to [128, 139], were sequences of activities are extracted directly from the event log, here, process replay [12] is used (see Chapter 4). For example, a new attribute named "WaitingTime_A_B" could be added to each case row in the log, storing the calculated waiting time between Activities A and B for that case. This allows for the generation of descriptions following the proposed protoforms:

$$\text{In } T_r, Q\ Ys, R \tag{3.15}$$

$$\text{In } T_r, \text{in } Q\ Ys \text{ with attribute } K, R \tag{3.16}$$

Where T_r, Q and K are as defined for Equation (3.12) and R makes reference to a case attribute which states whether a transition between two activities in the process takes place in a case. The evaluation process is similar as before; for protoform (equation (3.15)) is computed as in Equation (3.13) and for protoform (3.16), as in Equation (3.12). Substituting S for R respectively in both cases.

Following protoform (3.15) descriptions as the following can be generated e.g. *"In the first semester of year 2021, in approximately half of cases, activity A takes place shortly after activity B"*. By following protoform (3.16) relations between features and activity relationships can be described e.g. *"In the first semester of year 2021, in many cases where activity C takes place long after activity A, the throughput time was long."*

3.2.3 Compliance

Compliance descriptions highlight deviances that happen in the process. By adding the expected value (the value originally defined by experts) of a feature to its description makes it easier for users to detect and understand deviances taking place in the process. By using a type-II protoform as in (3.12) descriptions of attributes influencing other attributes, or causing deviance over the expected value of some other attribute can be generated. For instance, describing how the participation of one particular resource in a case makes the wait time between two activities longer than initially expected. These descriptions are obtained by adding the expected value of some attribute (defined based on expert knowledge) and contrasting it with the actual value of said attribute in the process and is generated by following protoform (3.12). The protoform is structured as:

$$\begin{aligned}&\text{In } T_r, \text{ attribute } E \text{ was expected.}\\ &\text{However, } Q\ Ys \text{ with attribute } K \text{ had attribute } S.\end{aligned} \tag{3.17}$$

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

This extended protoform allows for the generation of descriptions such as *"In year 2020, a low wait between activities A and B is expected. However, in most cases where activity C is executed, wait time between activities A and B is high"*. This protoform is indeed a composite protoform [103], obtained by composing two protoforms through a semantic relation of contrast. A type-II temporal contextualization protoform (see Equation (3.12)) is composed with a protoform in the form of *"In Tr, attribute E was expected"*, where E is a linguistic value of a case attribute V. As the generation of this protoform is based on expert knowledge, its truth value Tr is always maximum. This protoform is inspired by the protoform *"X was A in Tr"* introduced in [174], where X is a linguistic variable that makes reference to an attribute V, A is a linguistic value of X and Tr is a time interval, with which sentences as *"Temperature was high throughout the last 30 minutes"* can be generated.

In order to generate coherent sentences with this protoform, it is important that both E and S must make reference to different linguistic values belonging to the same linguistic variable i.e.: both E and S might be linguistic values referring to the same case attribute V. For the same reason, the time intervals T referred by both constituent protoforms must be equal, so in the final protoform structure, only one reference to T is necessary. The truth degree of this protoform is derived from the evaluation of the underlying protoform (3.12).

3.2.4 Comparison between groups

Protoforms (3.12) and (3.16) reveal insights into how two attributes relate, describing how the attribute described by the qualifier K affects the attribute described by the summarizer S, where in protoform (3.16), the summarizer R specifically makes reference to the wait time between two activities. Comparison descriptions extend these protoforms in order to highlighting differences in the value of a particular case attribute for two groups of cases e.g.: *In year 2020, many cases where resource MANAGER was involved had a short waiting time between activities C and D. However, most cases where resource APPRENTICE was involved had a long waiting time between activities C and D.*

The grouping is done in the basis of another case attribute selected as a qualifier. This protoform allows to compare how an attribute selected as summarizer is influenced by different values of an attribute selected as a qualifier used to create groups of sentences. For instance, comparing if differences for the wait time between two activities exist based on the resource involved in the execution of that case. As the compliance protoforms previously described, this is a composite protoform. In this case, the composed protoforms are in the form of protoform

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

(3.12) or (3.16). The resulting composite protoform is defined as:

$$\begin{aligned} &\text{In } T_r,\ Q_1 \text{ cases with attribute } K_1, \text{ had attribute } S_1. \\ &\text{However, } Q_2 \text{ cases with attribute } K_2 \text{ had attribute } S_2. \end{aligned} \tag{3.18}$$

In this protoform, in order to define two comparable groups of cases, qualifiers K_1 and K_2 must represent different linguistic values of the same case attribute V, i.e., K_1 and K_2 make reference to the same linguistic variable (case attribute) but take different linguistic values. For instance, the case attribute *Resource*, indicating the person in charge of the execution of a particular case may be selected, then, two of the possible linguistic values defined for the attribute *Resource* are selected to define the groups to compare. Summarizers S_1 and S_2 work in a similar way, they convey different linguistic values of the linguistic variable under observation (summarizer) e.g., *low* vs. *high* for linguistic variable *Wait time*.

We refer to the first protoform "*In T_r, Q_1 cases with attribute K_1, had attribute S_1.*" as $P1$ and to the contrasting protoform "*However, Q_2 cases with attribute K_2 had attribute S_2.*" as $P2$. So the truth value of both P_1 and P_2 is computed as in (Equation (3.14)). The truth value for this protoform may be derived from the aggregation of the truth values of its constituents through any t-norm. For simplicity and consistency we propose the use of the t-norm minimum. If we refer to the general protoform as $S1$ and to the contrasting protoform as $S2$ the truth value of the group comparison protoform (3.18) can be computed as:

$$T = T(S1) \wedge T(S2) \tag{3.19}$$

Deviance

A particular case of this protoform is the one aim at putting in relevance attributes which may be exceptions over the predominant value of some attribute. For instance, a description as *"In year 2019, most patients had a normal waiting time between the MS session of the patient and its intervention. However, several patients with emergency admittance had a short waiting time"*. Can provide insights into how attribute *"Admittance"* causes attribute *"Wait time"* to have a value different than the one that is the norm in the process. This particular protoform is defined as follows:

$$\begin{aligned} &\text{In } T_r,\ Q_1 \text{ had attribute } S_1. \\ &\text{However, } Q_2 \text{ cases with attribute } K_2 \text{ had attribute } S_2. \end{aligned} \tag{3.20}$$

As before, the truth value is derived from the aggregation of the truth values of the constituents protoforms through any t-norm (we again propose the use of t-norm minimum).

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 3.3: Questions shown with each example as part of the evaluation questionnaire performed to assess the proposed protoforms.

Question
The information this natural language description provides is interesting.
Understanding the information this natural language description provides is easy for me.
I perceive information is expressed in a comprehensible way.
Which way to you find easier to understand the information here presented?

3.3 Evaluation of the proposed protoforms

In this section we present a preliminary evaluation of the proposed model in a case study in the healthcare domain. In particular, we use the process related to the management of patients in the Valvulopathy Unit of the Cardiology Department of the University Hospital of Santiago de Compostela [49]. A complete in-depth evaluation of the proposed protoforms, in the context of a complete Natural Language Generation system is provided in Section 4.4. This preliminary assessment was conducted by medical experts of this Cardiology Department of the University Hospital of Santiago de Compostela. Following the current evaluation standards of the NLG field [179], a short questionnaire based in the general ideas of the Technology Acceptance Model (TAM) [180] adapted to linguistic summarization (cf [128]) was used. Due to the lack of gold-standard corpora for comparison, human-based evaluation, particularly through expert manual assessment, is considered as the most suitable methodology for evaluating this case, where automatic metric-based evaluations cannot be performed.

In the proposed questionnaire, instances of each type of protoform — shown in Table 3.3 — were presented in order to assess the degree to which they provide useful information, and weather they achieve this in a comprehensible way. The corresponding model of the process from the use case was also included in the questionnaire, in order to assess whether natural language descriptions are preferable or not to graphical representations for process analysis, specially in the case of complex processes like those in the healthcare domain. Finally, general questions were asked in order to determine if the use of natural language descriptions of processes in the health-care domain is found useful. All questions are asked on a five-level Likert scale — the most common human-based evaluation methodology [179] — ranging from "strongly disagree" (1) to "strongly agree" (5) and "agree" as an average value (3) except the one question comparing natural language and graphical representations.

Results show most summaries are found to provide interesting information, except those

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 3.4: General questions asked to assess the general validity and usefulness of the proposal.

Question
Natural language descriptions would give me a better understanding of what happens in my job.
Natural language descriptions would allow me to complete tasks quicker.
Natural language descriptions would increase the quality of my work.
Natural language descriptions would increase my productivity.
Natural language descriptions would increase my effectiveness.
Natural language descriptions would make my work easier for me.
I would find natural language descriptions useful in my job.
I find natural language descriptions easy to understand.
I perceive it is easy to find the information that I need in the natural language descriptions.

cases where routine information was given (this was assessed looking at the scores for the corresponding question for individual summary instances). An average value of (3.28) out of 5 — being 5 the best possible value — was obtained regarding the interest of the information conveyed in the summaries. This illustrates that the proposed protoform taxonomy generally addresses relevant aspects of a process, in particular, in the healthcare domain. By generating summaries on the basis of the same protoforms with different data, we were able to recall that the perceived usefulness is only found lower when summaries convey information evaluators are already familiar with and not because the proposed protoforms address aspects of a process that are not relevant. When evaluators were not familiar with the information conveyed in the summaries, they were labelled interesting. In regard to the ease of understanding and the clarity of the summaries, average values of (3.85) and (3.14) were respectively obtained. All protoforms, except one case where the linguistic realization was not adapted to the necessities of the evaluators, were found clear and easy to understand. In the general test, descriptions were evaluated as easy to understand (5.00) and easy to locate relevant information in them (5.00). Finally, in all cases where natural language descriptions were confronted with graphical representations, a clear preference was shown for natural language descriptions.

On the basis of the general questions, evaluators stated that the proposed natural language descriptions and their underlying protoforms are useful (5.00) give them a better understanding of what is happening in the process (5.00), allow them to complete tasks quicker (4.00), increase the quality of their work (3.50), increase their effectiveness (3.00) and productivity (3.5).

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

3.4 Discussion and final remarks

Multiple protoform-based solutions have been described proposed for generating insights on data through linguistic summaries. Existing literature reveals that most systems are based on unary and binary fuzzy quantified protoforms with basic structures which despite proving beneficial in certain scenarios, the extent of information they offer is often quite restricted and inadequate. This is particularly evident in real-world applications, which typically demand more comprehensive and detailed information. To overcome this limitation, we presented a process-focused protoform-based model for the automatic generation of natural language descriptions of processes, extending classical fuzzy quantified protoforms with process-domain information, new semantic structures and a protoform composition framework, resulting in linguistic summaries with more complex and complete description structures.

The protoforms proposed in this chapter represent a significant advancement in the field of process description. By incorporating temporal contextualization of attributes, these protoforms enable the description of how case attributes behave during different stages of a process. Additionally, the protoforms allow for describing the frequency and temporal relationships between events, providing insights into event relations and wait times. This is particularly useful in complex domains like healthcare, where the timing of events can critically impact patient outcomes. By introducing the temporal dimension through imprecise descriptions of frequency and temporal characteristics of attributes and activities and through the composition of protoforms, the semantics and expressiveness of our proposal is greatly enhanced. Moreover, the protoforms support the description of compliance with expected values and highlight how certain attributes influence or deviate from the norm. This is crucial for identifying unusual patterns or deviations in a processes, which could indicate areas for improvement or the need for intervention.

The validation of these protoforms was conducted through a case study in a cardiac valvulopathy unit. Medical experts assessed the protoforms for their utility and comprehensibility. The results indicated that most protoforms provided useful and comprehensible information, especially when conveying unknown data to medical experts. The instances where protoforms conveyed routine information were perceived as less useful, underscoring the importance of context and relevance in the application of these protoforms.

In conclusion, the proposed protoform-based model significantly enhances the ability to describe process-related information in natural language. By integrating temporal information with case attributes, the model effectively addresses the intricacies of both case, control-flow

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

and time perspectives in process descriptions, something that had not been achieved yet in the literature. The successful validation in a real healthcare setting demonstrates the potential of these protoforms in providing valuable insights to domain experts, thereby facilitating better decision-making and process understanding.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

CHAPTER 4

A FRAMEWORK FOR THE AUTOMATIC DESCRIPTION OF PROCESSES IN NATURAL LANGUAGE

It is vital to remember that information– in the sense of raw data– is not knowledge, that knowledge is not wisdom, and that wisdom is not foresight. But information is the first essential step to all of these.

— Arthur C. Clarke

In Chapter 3 we introduce a series of natural language descriptions that address the most relevant aspects of a process supported by a protoform-based model. Multiple protoform-based solutions have been described in the literature for generating insights on data, however, an analysis of the state of the art reveals that most existing approaches propose overly simplified solutions that rely solely on unary and binary quantified sentences, which may not be sufficient in many application domains. To overcome this limitation, we propose a series of process-focused protoforms that take into account the temporal dimension and include domain specific information. Through a process of double quantification (over time and over the set of relevant properties of a process) these protoforms allow both to contextualize relevant aspects of a process in time as well as to describe control-flow and performance-related aspects of a

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

process. However, despite the proposed enhancements in expressiveness — syntactically and semantically through the proposal of new protoforms — the use of linguistic summaries as a direct output of a user-targeted system is still less than ideal.

To overcome this limitation, in this chapter, we present the Process-to-Text (P2T) framework. A framework for the generation of natural language descriptions of processes based on the most widely used D2T architecture [10]. An ontology-based knowledge representation formalism based on the protoforms introduced in Chapter 3 provides a mechanism to represent process-related information in a way that allows for reasoning and the inference of new knowledge. By being grounded in Fuzzy Logic, this formalism is capable of handling the inherent imprecision of natural language and represent the underlying semantics of the messages and descriptions that the system generates. This framework is applied to the Aortic Stenosis Integrated Care Process (AS ICP) implemented in the Cardiology Department of the University Hospital of Santiago de Compostela [49] as a use case. A user-based evaluation of the proposal is conducted on the basis of this use case.

In Section 4.1 the NLG and D2T architectures are presented, as they serve as the main backbone of the framework here proposed. Section 4.2 presents the proposed framework, including details on each of the stages and components. In particular the knowledge representation formalism designed for the generation of natural language descriptions of processes is introduced in Section 4.1.1. After illustrating the framework and its details, Section 4.3 includes the application of the proposed framework in the Aortic Stenosis Integrated Care Process (AS ICP) [49] together with the validation of the proposal by the medical experts involved in the process used and the evaluation of the obtained results in Section 4.4. In Section 4.5 an implementation of the P2T framework into an interactive system is presented. Finally Section 4.6 presents a discussion of the obtained results and some concluding remarks.

The contents of this chapter are extracted from the following publications:

Fontenla-Seco, Y.[a], Lama, M.[a], & Bugarín-Diz, A.[a] (2020). **Process-To-Text: A Framework for the Quantitative Description of Processes in Natural Language**. In *Trustworthy AI - Integrating Learning, Optimization and Reasoning (TAILOR 2020) at ECAI 2020*. Lecture Notes in Computer Science, vol. 12641 (pp. 212-219). Springer, Cham. ISBN: 978-3-030-73959-1 (online). doi: 10.1007/978-3-030-73959-1_1910.1007/978-3-030-73959-1_19

[a]Centro Singular de Investigación en Tecnoloxías Intelixentes, Universidade de Santiago de Compostela, Spain.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Fontenla-Seco, Y.[a], Lama, M.[a], González-Salvado, V.[b], Peña-Gil, C.[b], & Bugarín-Diz, A.[a] (2022). **A Framework For The Automatic Description Of Healthcare Processes In Natural Language: Application in an Aortic Stenosis Integrated Care Process**. *Journal of Biomedical Informatics*, 128, 104033. ISSN: 1532-0464. doi: 10.1016/j.jbi.2022.104033.

[a]Centro Singular de Investigación en Tecnoloxías Intelixentes, Universidade de Santiago de Compostela, Spain.
[b]Department of Cardiology, University Clinical Hospital of Santiago de Compostela, SERGAS IDIS CIBERCV, Santiago de Compostela, Spain.

4.1 The Data-to-Text (D2T) architecture

Natural Language Generation (NLG) systems are aimed to produce natural language texts from some underlying non-linguistic representation of information [56]. These systems use domain knowledge as well as linguistic knowledge in order to automatically generate texts or documents that summarize the most relevant aspects of some input data. They have been applied in multiple domains since their inception: from weather forecasting reports [71] to the generation of textual summaries from neonatal Intensive Care Unit data [67], and even to support the interaction of kids with complex communication needs [181].

As natural language is the inherent way of communicating for humans, NLG techniques do not rely on users capabilities to extract relevant information and patterns from visual analytics, the most frequent form of conveying information to users in data-intensive domains [182], which present a series of problems and inherent difficulties [46]. In this sense, it has been proved that in certain domains, experts can take better decisions when presented with textual summaries rather than when presented with graphical displays [52].

Different architectures and approaches have been proposed for NLG systems. The most widely used is the NLG pipeline defined by Reiter and Dale [56]. As depicted in Figure 4.1, this architecture defines 3 main modules, which are subdivided in a series of tasks:

1. **Document planning**: Focuses on the generation and organization of content i.e., "what to say".

    - **Content determination**: This stage involves the generation of messages or propositions from the input data and the determination of which messages should be included in the generated text. These messages are defined on the basis of a

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

knowledge representation formalism that defines the entities, attributes, concepts, and relationships of the input's data domain. The main activity in the content determination stage is then reasoning: selecting and summarizing portions of the input data in the form of messages. This stage should thus be seen as an expert system [56].

- **Document structuring**: This task involves determining the organization and grouping of content within a document, as well as establishing rhetorical connections between different chunks (messages, sections, etc.). This can be viewed as the structural component of document planning.

2. **Micro-planning**: Focuses on aggregating content and expressing it correctly, i.e., "how to say it".

   - **Aggregation**: This task involves determining the way structures created during document planning are transformed into linguistic formats like sentences and paragraphs. This can be regarded as the structural aspect of micro-planning.
   - **Lexicalization**: This task entails selecting the precise words or other linguistic tools, like specific syntactic structures, to articulate the content chosen by the content determination component. This process can be considered as the content-focused aspect of micro-planning.
   - **Referring expression generation**: This involves determining the appropriate expressions for referencing entities, and is another component of the content aspect of micro-planning.

3. **Realization**: Syntactic, morphological and orthographic rules are applied.

   - **Linguistic realization**: This task involves transforming abstract sentence representations into actual text, aligning with the content-focused aspect of the realization stage.
   - **Structure realization**: This task involves translating abstract elements like paragraphs and sections into markup symbols that are recognizable by the document presentation system. This process aligns with the structural aspect of surface realization.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Data-to-Text (D2T) systems are a class of NLG systems that aim to produce textual summaries from non-linguistic, typically numeric, data [10]. These systems perform two main tasks: a knowledge extraction process which can be defined as the "non-trivial acquisition of knowledge which is novel, potentially useful and easily understandable from data" [60]; and a linguistic expression process which appropriately turns the discovered knowledge into natural language texts. The main differentiating aspect between NLG systems and D2T systems is that while the formers usually take a knowledge base as input, the laters take numerical data, which needs to be analyzed and interpreted. This aspect also differentiates D2T systems from more conventional data analysis systems that focus solely in knowledge extraction, but do not address human communication issues [10, 161], highlighting the importance of both knowledge extraction and linguistic expression in D2T systems.

Building upon the foundations of the NLG architecture, the D2T architecture introduces several key extensions [10]. An initial *signal analysis* stage is included for those cases in which the input to the system is a numerical time series which needs to be processed and distilled into a series of relevant patterns and events. After this is done, the *data interpretation* stage is in charge of analyzing the identified patterns and events and infer a series of messages and relationships among them. Then, the *document planning* stage is in charge of selecting which messages to convey as well as the text rhetorical and document structure. These two stages represent the *knowledge extraction* task of a D2T system, where relevant knowledge is extracted from the input data. Finally, the *linguistic expression* task is represented by the *micro-planning* and *realization* stages of the NLG pipeline which, in the D2T architecture are fused together into a final stage in which the concepts and structures previously defined are expressed in the form of natural language. The combination of these tasks highlights how in D2T systems, the emphasis often lies in the relevance of the extracted knowledge, being the intricacy of its linguistic expression of secondary importance as users typically prefer the use of simpler language in this type of systems [10], making it even possible to subsume this tasks with the use of simple fill-in templates.

Figure 4.1 illustrates both the NLG and D2T architectures in parallel, and gives a look at their differences and similarities. An initial analysis reveals lots of similarities, this is due to the fact that in the original proposal by Reiter and Dale [56], NLG systems are actually understood as data-to-text systems — and not text-to-text systems — which generate natural language texts from non-linguistic input data. In this initial approach, authors establish a difference between NLG systems having as input a knowledge base, where the input data is already encoded in a

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

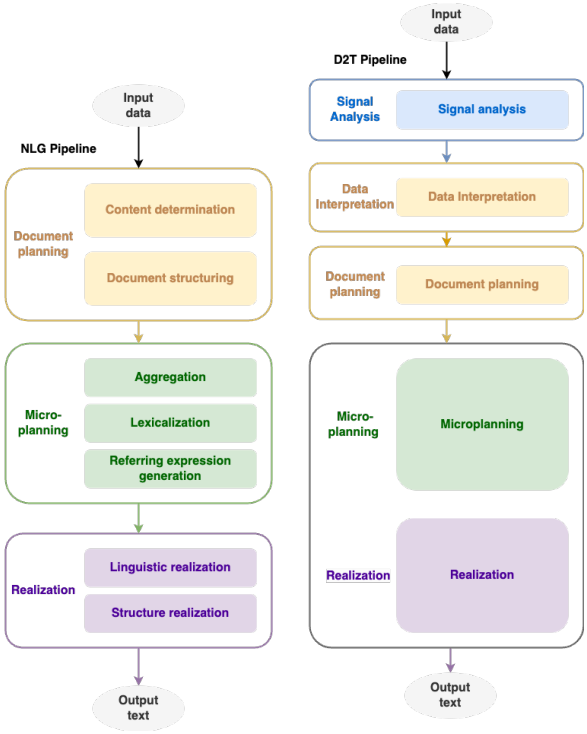


Figure 4.1: From left to right, the NLG and D2T architectures respectively. Colors are used to represent how tasks are matched in between both architectures.

semantic representation, and those having structured non-semantic data. This illustrates why the *data interpretation* and *document planning* stages of the D2T architecture correspond so closely to the *document planning* module from the NLG pipeline, with the only difference being that in the NLG architecture, both the generation and selection of messages is done in the *content determination* phase and the *document structuring* is just related with giving structure to the generated messages, while the D2T architecture divides the message generation and selection among the *data interpretation* and *document planning* stages respectively. The main differences then lays in the extension of the architecture with the inclusion of the *signal analysis* stage in the D2T architecture, which, puts more emphasis on the analysis of signal data. against structured data and how the NLG architecture gives more importance to a refined linguistic expression, whereas the D2T architecture defends that these systems produce simpler language because their users prefer it [10].

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

4.1.1 Representing knowledge

In D2T systems, the communication of relevant information is achieved through the extraction of knowledge from some input data and its transformation through a linguistic expression process which renders contextually relevant and comprehensible natural language texts. These texts convey the information extracted from the data, but the knowledge resides in their underlying semantics in the form of predications of properties and relationships of different entities or individuals — for example, that *many cases have a long wait time*. In order to extract knowledge from non-linguistic input data and represent it in the form of entities, properties and relationships so it can be expressed linguistically it becomes imperative to model the domain of the problem. The *domain modeling* task is therefore concerned with knowledge representation and the mapping of knowledge structures into semantic representations. This task involves defining and representing the domain of a particular problem in a way that not only captures the entities, attributes, concepts, and relationships pertinent to the domain but also facilitates reasoning and the inference of new knowledge. Usually, this is done by means of an ontology [56]. This semantic understanding is critical for the system to generate natural language texts that are contextually relevant and coherent.

The predication of relevant properties of entities and concepts and the assertion of relationships between entities on the domain are then conceptualized through messages or propositions. In the *message definition* task, the content and form of messages is defined on the basis of the defined domain model. Furthermore, categories or classes of messages are also defined based on the type of information they convey. Messages mediate between the underlying structure of the input data and the final output of a D2T system, imposing structure over the elements of the domain and serving as abstract representations of the linguistic fragments that will be eventually generated. They serve as semantic representations that capture the knowledge extracted from the data in a way that can be effectively communicated through natural language. In their most basic form, each message corresponds to an individual sentence in the final text, however, by using more abstract representations, messages can be further refined and combined in a variety of ways adapted to specific circumstances based on their semantics. Messages are then the basic unit of information that a D2T system manages.

Defining this formalism is especially important in applications where the input data needs to be reasoned with and summarized before it can be made use of [56], such as it happens in D2T systems.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

4.2 Process-to-Text (P2T): A framework for the automatic generation of natural language descriptions of processes

In this section, the Process-to-Text (P2T) framework is presented: a novel, process-mining-based approach, harnessing the strengths of symbolic reasoning and neural Natural Language Generation to generate accurate, rich and contextually-aware natural language descriptions of processes. The framework is proposed following a modular approach based on the D2T and NLG architectures adapted to the specific needs of the process domain. This architecture defines all the elements and stages necessary for the construction of a D2T system for the description of processes in a sufficiently general way as to be applicable to processes in any domain.

A knowledge representation formalism based on Fuzzy Logics serves as the cornerstone of the proposed framework. A unifying multi-layered ontology that provides a structured definition of the domain, where the central concept is that of a process instance and its attributes. These attributes are symbolically represented as linguistic terms, a representation grounded on Fuzzy Logics which allows for reasoning and the inference of new knowledge. In particular, through the linguistic summarization (see Chapter 3) of a process, fuzzy quantified protoforms are generated. These protoforms serve as intermediate representations of the natural language descriptions that the system will eventually generate. Thus, the definition of the form and content of messages in the P2T framework is based on the protoform-based model introduced in Chapter 3.

The proposed formalism serves as a unifying representation across all the different stages in the proposed architecture. From the symbolic representation of a process and its attributes and the use of these symbols to generate fuzzy quantified protoforms, to the combination of these into higher level messages and their linguistic expression as natural language descriptions. The P2T framework consists then of four different stages that can be clearly identified:

1. **Data interpretation:** Process mining techniques are applied to extract knowledge from the input event log and generate a case-perspective *enriched* event log (see Section 3.2). Through process model discovery and process replay, control-flow and time features are extracted from the process and added to the event log as new attributes (see Section 3.2). Any other additional attributes may be computed during this stage on the basis of expert knowledge.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

2. **Content determination:** Based on the defined domain model, the ontology gets populated giving place to a knowledge base which represents the semantics of process instances and their attributes. As a data intensive system, the input data is too fine-grained to be reported directly, it needs to be distilled and summarized before being expressed linguistically. For this reason, through a linguistic summarization process, the knowledge stored in the ontology is used for the construction of messages defined on the basis of the protoform-based model introduced in Chapter 3. Based on a quality framework, the propositions to be conveyed in the final text are selected.

3. **Micro-planning:** Higher-level messages are constructed through an inference process from the low-level protoforms generated during content determination on the basis of their semantics using both expert and induced knowledge from the input data. In this process, rhetorical connections between messages are stablished and protoforms are combined into richer expressions. The selection of these higher-level messages is done by the inference process on the basis of the quality of the low-level protoforms combined in each case.

4. **Realization:** A hybrid template-based realization approach is proposed. Based on the protoforms defined in Chapter 3, a series of fill-in templates are defined for the construction of semi-verbalized representations of the descriptions. Through these templates, messages are given a primitive linguistic expression. Despite the grammatical, syntactic, and semantic correction, their lack of linguistic refinement hinders their usability. For enhancing the expressiveness and linguistic refinement of the final descriptions, we propose a Large Language Model-augmented realization stage, where the semi-verbalized utterances are fed into a Large Language Model (LLM) resulting correct-by-design, context-aware and linguistically rich natural language descriptions of a process.

Figure 4.2 depicts the proposed architecture. Colors are used to distinguish between the different stages and elements of the framework. In blue, the data interpretation stage, in yellow, the content determination stage, in green, the micro-planning stage and in dark mauve, the final realization stage. In teal, the knowledge used across all the stages of the framework. A dotted gray line frames the whole knowledge representation formalism. The data interpretation and content determination stages are in charge of the *knowledge extraction* while the micro-planning and realization stages are in charge of the *linguistic expression* process.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

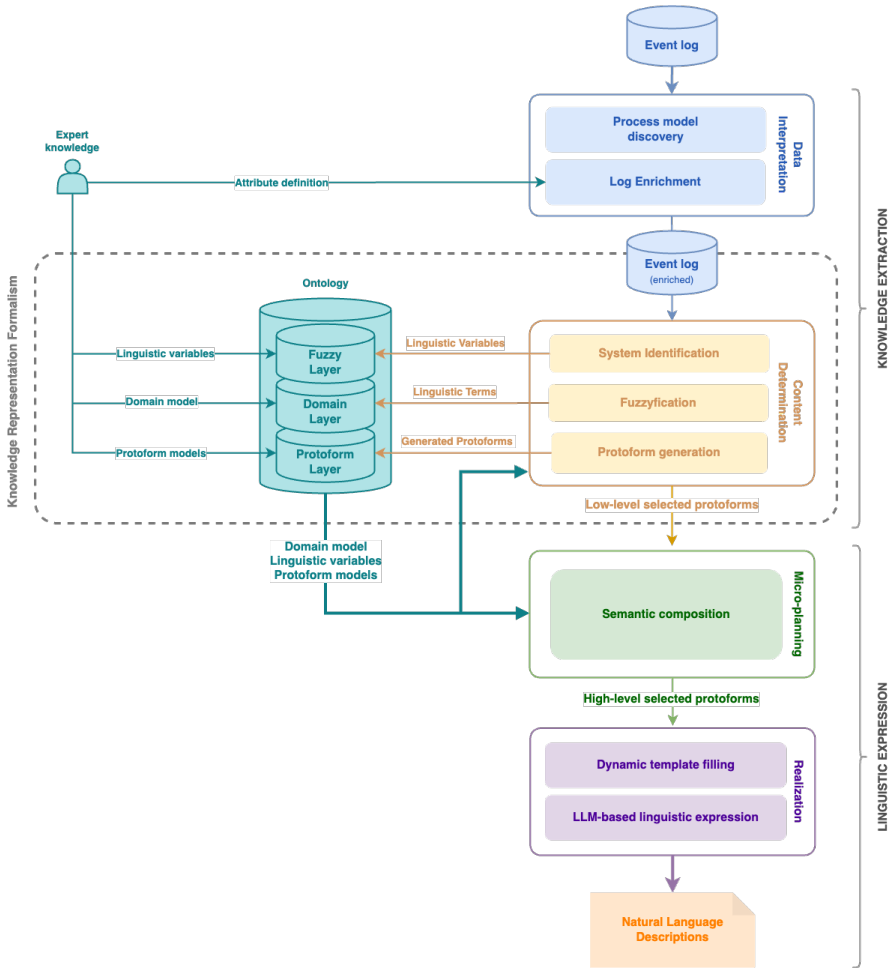


Figure 4.2: The Process-to-Text (P2T) framework. It can be seen how the proposed knowledge representation formalism represents the cornerstone of the proposal.

Before focusing into each of the stages of the architecture and the processing that takes place in each of them, the knowledge representation formalism is presented.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

4.2.1 A fuzzy logic-based knowledge representation formalism for the generation of natural language descriptions of processes

The first step for the definition of the knowledge representation formalism in a D2T system is the definition of a model that allows to represent the semantics of the entities, properties and relationships present in the domain. We define this model as an ontology, a semantic representation which organizes and structures information in a way that captures its meaning and relationships. An ontology lays out the relationships and hierarchies between different concepts in the domain, enabling the system to not only define the structure of the knowledge but also to store this knowledge, understand it and interpret it in a meaningful way. For this reason, the proposed ontology functions not only as a comprehensive domain model but also as a knowledge base which provides the core elements for the generation of messages. This semantic understanding is critical for the system to generate messages that result in natural language texts that are contextually relevant and coherent.

Fuzzy logic is chosen as a the underlying framework for the definition of the semantics of the linguistic terms that describe the process attributes for its well-known ability for linguistic concept modeling [101, 157] (see Chapter 3) and its use in system identification [183]. The semantic expressivity of fuzzy logic, using linguistic variables, and linguistic rules, is quite close to expert natural language. This fact reduces the effort of expert knowledge extraction. As a result fuzzy logic formalism has been widely used for building expert systems, which is the final objective of the content determination stage in a D2T system [56].

It is important to notice that the term *"knowledge representation formalism"* is not limited to a single ontology or knowledge base but refers to a comprehensive framework that can encompass multiple components, such as models and knowledge bases, rules, algorithms and reasoners, that contribute to the derivation of knowledge from the input data and its representation in the form of messages. In particular, the proposed knowledge representation formalism consists not just on the domain model but encompasses both the definition of knowledge structures together with the storage knowledge itself on a comprehensive layered ontology together with a series of inference systems required for the induction, inference and summarization of new knowledge based on the input event log. The overall knowledge representation formalism comprehends then a series of interconnected components:

- A unifying multi-layered ontology that provides a structured definition of the domain, linguistic variables, values and quantifiers and the proposed protoforms. This ontology

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

establishes the fundamental concepts, relationships, and entities necessary the linguistic representation of a process and its attribute and for the generation of natural language descriptions on the basis of fuzzy protoforms. Apart from modeling the underlying structure of the knowledge, this ontology stores this knowledge in a way that enables reasoning. This ontological definition is used by an external reasoner which operates independently but in conjunction with the ontology. It interprets the structure and knowledge defined in the ontology and uses it to conduct fuzzy logic computations. For this reason, a purpose-built reasoner equipped with the capability to handle fuzzy logic operations — as standard reasoners might not inherently support fuzzy computations — needs to be built.

- A **linguistic variables definition process** that, on the basis of the input data, linguistic variables that model and partitions the domain of case attributes into linguistic values are defined. In fuzzy systems this process is known as *system identification*. A process that integrates the induction of this linguistic variables from data together with expert knowledge to enhance the accuracy and relevance of the defined linguistic conceptualizations.

- A **fuzzy logic computational system** that interprets the input event log, based on the fuzzy ontology, in a *fuzzification* process. This system populates the domain ontology with a process, its cases and a series of linguistic terms characterizing each case. These linguistic terms are obtained by evaluating case attributes against their corresponding linguistic variables. The computational aspect of the system is highlighted, as the main task the system performs are calculations or computations based on fuzzy logic, such as evaluating membership functions, to transform crisp data into fuzzy linguistic terms.

- A **fuzzy inference system** that, based on the domain, fuzzy and protoform layers of the ontology generates a series of candidate protoforms and populates the protoform ontology. This same reasoner might apply additional logic over the protoform ontology's instances to derive higher-level messages. This system is defined as an inference system because not only processes data but also applies fuzzy logic rules to infer or deduce characteristics, patterns, or classifications from the data.

Different layers compose the proposed ontology, in particular, the fuzzy, domain and protoform layers are integrated together into a unifying ontology that effectively integrates all

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

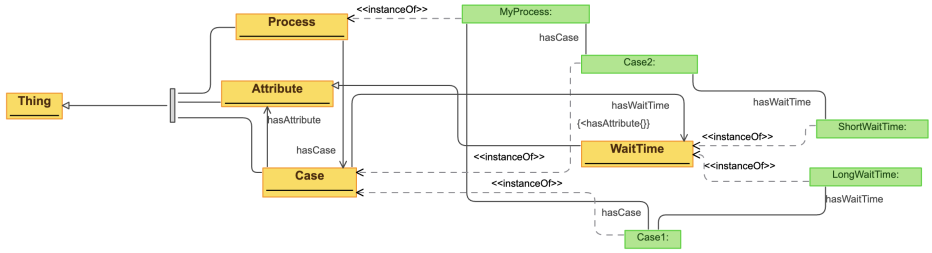


Figure 4.3: Graphical representation of the ontology-based domain model together with two case instances and their corresponding *wait time* values.

the knowledge modeling and storage required for the generation of natural language descriptions of processes in the form of protoforms grounded in fuzzy logics. In particular:

- The **domain layer** provides a structured definition of the domain, establishing the fundamental concepts, relationships, and entities of a process necessary in the P2T framework.
- The **fuzzy layer** models the fuzzy logic domain, including the concepts of linguistic variables and values, membership functions, etc. This ontology stores the corresponding entities that are defined from the data during the system identification stage. This linguistic variables and values are then used during the *fuzzification* process.
- The **protoform layer** is dedicated to modeling the proposed protoforms which serve as templates for the generation of messages during the content determination stage.

Domain layer

The XES meta-model [184] is taken as a *de facto* standard in almost any application or technique dealing with event logs. However, for our particular case, despite having an event log as input data, we focus on processes more at a conceptual level. For this reason, a much simpler model based on the definitions of process, case and attribute given in Chapter 1 can be defined. A visual representation of the proposed domain model is depicted in Figure 4.3.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Within the ontology, process attributes are modeled as linguistic terms whose semantics are grounded using fuzzy logic. While the ontology inherently provides a semantic structure, fuzzy logic introduces an additional layer of semantic richness by defining the meaning of linguistic terms such as *"high"*, *"low"*,*"long"*, or *"short"*. The linguistic attribute values that characterize each case are obtained through a *fuzzification* process. In this way, instead of numerical (or categorical) attributes, the ontology leverages semantically rich, human-like linguistic terms to characterize each process case. The integration of linguistic terms grounded on fuzzy logics enables the ontology to represent knowledge in a way that is much closer to human cognitive and linguistic capabilities by enabling reasoning and the inference of new knowledge based on linguistic terms (see Chapter 3). Furthermore, the integration of this linguistic terms into our ontological definition allows us to harness the capabilities of ontologies such as querying knowledge in ways closer to human cognition than when using non-symbolic forms of structured data.

In essence, both the ontology and fuzzy logic play integral roles in the knowledge representation formalism, while fuzzy logic offers a powerful tool for dealing with the inherent imprecision of natural language and reasoning, the ontology adds a structured, semantically rich layer of domain-specific knowledge representation. Together, they provide a more complete and effective knowledge representation formalism for systems dealing with natural language generation and understanding, particularly in complex or specialized domains.

Based on these premises, the domain model is defined as:

- **Classes**:
  - `Process`: A class representing the overall process under description.
  - `Case`: A class representing process instances. Each instance of this class is each of the process executions or cases recorded in the input case-perspective event log.
  - `Attribute`: A general class for all case attributes. Examples may be the wait time for the whole case *"wait time"*, the wait time between a particular pair of activities *"wait-time-A-B"*, the resource involved in the execution of the case *"resource"* or any other possible attribute.
  - For each possible attribute, a subclass of `Attribute` is defined. For example, attribute *"wait time"* is defined as $\texttt{WaitTime} \sqsubseteq \texttt{Attribute}$. Each possible value of the attribute is defined as a subclass of this class for which an instance for

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

each case characterized by that attribute is defined. For example, instance *"long wait time"* for attribute *"wait time"* is defined as `WaitTime(LongWaitTime)`.

- **Relations**
  - `hasCase`: An object property (relation) that links a `Process` to is `Cases`. This is defined as: `hasCase` $\subseteq$ `Process` $\times$ `Case`. This way a process is linked to all its corresponding cases.
  - `hasAttribute`: A relation that links a `Case` to instances of `Attributes`. This is defined as: `hasAttribute` $\subseteq$ `Case` $\times$ `Attribute`. For each subclass of `Attribute`, a specialized sub-property of this property is defined. For instance, the property `hasAttribute` for attribute `WaitTime` is formalized as `hasWaitTime` $\subseteq$ `Case` $\times$ `WaitTime`

This model — including the *wait time* attribute and its `long wait time` value used for illustrative purposes — can be formalized using Description Logics [], as:

Definition 6: ***Domain model***

- Classes:

$$\text{Process, Case, Attribute, WaitTime, LongWaitTime}$$

- Subclass Relationship:

$$\text{WaitTime} \sqsubseteq \text{Attribute} \qquad \text{LongWaitTime} \sqsubseteq \text{WaitTime}$$

- Object Properties:

$$\text{hasCase} \sqsubseteq \text{Process} \times \text{Case}$$
$$\text{hasAttribute} \sqsubseteq \text{Case} \times \text{Attribute}$$
$$\text{hasWaitTime} \sqsubseteq \text{Case} \times \text{WaitTime}$$

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Data Properties:

$$membershipDegree \subseteq \mathbb{R}$$

- Sub-property Relationship:

$$hasWaitTime \sqsubseteq hasAttribute$$

- Domain and Range Restrictions:

$\forall hasCase.Case$ (The range of $hasCase$ is $Case$)
$\forall hasAttribute.Attribute$ (The range of $hasAttribute$ is $Attribute$)
$Attribute \sqsubseteq \forall membershipDegree.\mathbb{R}$ (The range of $membershipDegree$ is $\mathbb{R}$)
$\forall hasWaitTime.WaitTime$ (The range of $hasWaitTime$ is $WaitTime$)

- Instances:

$$LongWaitTime(LongWaitTime_1)$$

Where we use inclusion ($\sqsubseteq$) to express subclass relationships and properties, and universal quantification ($\forall$) for property range restrictions. Through this ontological formalization, a process is related with a series of process executions in the form of cases, each of which is characterized by a series of linguistic terms that serve as semantic representations of the original attributes of a case.

Fuzzy layer

In fuzzy systems, a linguistic variable knowledge base is crucial for translating numerical data into a more qualitative, human-understandable form. This conversion is particularly important in contexts where decisions are based on subjective assessments rather than precise numerical thresholds. This knowledge base typically consists of a set of linguistic variables along with their corresponding linguistic values and the rules for their application. Each linguistic value is associated with a fuzzy set defined by its membership function that takes the form of a database and a rule base. The database contains the necessary definitions which are used

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

in the fuzzy reasoning process, including the definitions of linguistic variables and values and their membership functions. The rule base, on the other hand, contains a set of fuzzy rules that guide the inference process. In a fuzzy logic system, the knowledge base is used to evaluate inputs and make decisions. Rules in the knowledge base take linguistic forms, like "if temperature is high, then increase cooling". The system interprets these rules using the membership functions of the linguistic values to process numerical inputs and generate outputs. This approach allows fuzzy systems to mimic human reasoning by dealing with uncertainties and imprecision inherent in many real-world scenarios. It is particularly useful in areas like control systems, expert systems, and decision-making processes where binary logic falls short.

In practice, membership functions are stored as part of the fuzzy system's knowledge base, often in a digital format within the system's software. The parameters defining each membership function (e.g., the points of a triangle or the mean and standard deviation of a Gaussian curve) are stored as part of the system's configuration. Some systems use tabular data structures or arrays to store these parameters, especially when the system needs to be efficient in computation and memory usage. Advanced fuzzy systems might employ a database or an ontology-based approach for more complex or large-scale applications, where membership functions are a part of a broader knowledge representation. In a fuzzy logic controller or decision system, these membership functions are called upon to evaluate the degree to which a given input (e.g., a sensor reading) matches each linguistic term. The output of these functions is then used in the fuzzy inference process, where the fuzzy rules of the knowledge base are applied to derive a conclusion or action.

In the P2T framework only the definition of linguistic variables and values if of interest, an external reasoner is in charge of the inference process on the basis of the defined knowledge base. This knowledge base is structured according to an ontology that formalizes the definitions of linguistic variable, linguistic value and membership function as defined in Definition 5. In particular, the fuzzy model composes the fuzzy layer of the ontology. This formalization allows for a consistent definition of linguistic variables, values and membership functions across the framework, improving the general aspect of the framework by giving support to a standardized way of defining linguistic variables on any process domain. Furthermore, a taxonomy of membership functions based on the most common ones in the literature is defined, integrating the definition of membership functions in the ontology, facilitating the use of these functions in external reasoning systems. In particular, we define the following classes, relations and properties:

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- **Classes**:
  - `LinguisticVariable`: A class representing the general notion of linguistic variable
  - `LinguisticValue`: A class representing each of the linguistic values a linguistic variable may have. It has a `Linguistic term` and `Membership function` associated.
  - `MembershipFunction`: The membership function associated to a linguistic value. We propose a taxonomy of membership functions. In particular:
    * `CategoricalMembershipFunction` $\sqsubseteq$ `MembershipFunction`
    * `FuzzyMembershipFunction` $\sqsubseteq$ `MembershipFunction`
    * `TriangularMembershipFunction` $\sqsubseteq$ `FuzzyMembershipFunction`
    * `TrapezoidMembershipFunction` $\sqsubseteq$ `FuzzyMembershipFunction`
    * `GaussianMembershipFunction` $\sqsubseteq$ `FuzzyMembershipFunction`
    * `SingletonMembershipFunction` $\sqsubseteq$ `FuzzyMembershipFunction`
    * `SquareMembershipFunction` $\sqsubseteq$ `FuzzyMembershipFunction`, this is equivalent to a symmetric trapezoid membership function.

  Other membership functions can be added, like Imprecise sets, Singleton, Greater than or Lesser than numerical membership functions, etc.

- **Relations**:
  - `hasValue`: A relation that links a `LinguisticVariable` to its corresponding `LinguisticValue`. It is defined as: `hasValue` $\sqsubseteq$ `LinguisticVariable` $\times$ `LinguisticValue`.
  - `hasMembershipFunction`: A relation that links a `LinguisticValue` to its corresponding `MembershipFunction`. It is defined as: `hasMembership-Function` $\sqsubseteq$ `LinguisticValue` $\times$ `MembershipFunction`.

- **Properties**:
  - `LinguisticTerm` represents the linguistic term associated with a linguistic value. This term may be a single word or any other linguistic expression, whose semantics are modeled by the membership function associated with the linguistic value.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- `pointA, pointB, pointC, pointD` are data properties for a trapezoid membership function. In particular `pointA` and `pointD` represent the points where the trapezoidal function starts and ends its rise and fall. And `pointB` and `pointC` indicate the lower and upper bounds of the plateau of the trapezoid.
- `pointA, pointB, pointC` are data properties for a triangular membership function. More precisely, `pointB` indicates the point of maximum value and `pointA` and `pointC` indicate the lower and upper bounds of the function.
- `mean` and `standardDeviation` are the data properties for the `GaussianMembershipFunction`. The `mean` represents the central point (mean) of the Gaussian function and `standardDeviation` indicates the spread or standard deviation of the Gaussian curve.
- `category` is the data property of a categorical membership function. It represents the specific category or class for which the membership is defined. Based on the notion of crips sets and indicator functions, the membership degree assigned by this membership function is binary, i.e., 0 if an element is not compatible with the membership function, 1 if it is.

This can be formalized using Description Logics as follows:

Definition 7: ***Fuzzy model***

- Classes:

  LinguisticVariable, LinguisticValue, MembershipFunction

  FuzzyMembershipFunction $\sqsubseteq$ MembershipFunction

  CategoricalMembershipFunction $\sqsubseteq$ MembershipFunction

  TriangularMembershipFunction $\sqsubseteq$ FuzzyMembershipFunction

  TrapezoidMembershipFunction $\sqsubseteq$ FuzzyMembershipFunction

  GaussianMembershipFunction $\sqsubseteq$ FuzzyMembershipFunction

  SingletonMembershipFunction $\sqsubseteq$ FuzzyMembershipFunction

  SquareMembershipFunction $\sqsubseteq$ TrapezoidMembershipFunction

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Relations:

  $\text{hasValue} \subseteq \text{LinguisticVariable} \times \text{LinguisticValue}$

  $\text{hasMembershipFunction} \subseteq \text{LinguisticValue} \times \text{MembershipFunction}$

- Properties:

  $\text{pointA} \sqsubseteq \mathbb{R}$

  $\text{pointB} \sqsubseteq \mathbb{R}$

  $\text{pointC} \sqsubseteq \mathbb{R}$

  $\text{pointD} \sqsubseteq \mathbb{R}$

  $\text{category} \sqsubseteq \mathbb{S}$

- Domain and Range Restrictions:

  $\text{TrapezoidMembershipFunction} \sqsubseteq \forall pointA.\mathbb{R} \sqcap pointB.\mathbb{R} \sqcap pointC.\mathbb{R} \sqcap pointD.\mathbb{R}$

  $\text{TriangularMembershipFunction} \sqsubseteq \forall pointA.\mathbb{R} \sqcap pointB.\mathbb{R} \sqcap pointC.\mathbb{R}$

  $\text{GaussianMembershipFunction} \sqsubseteq \forall mean.\mathbb{R} \sqcap standardDeviation.\mathbb{R}$

  $\text{SingletonMembershipFunction} \sqsubseteq \forall pointA.\mathbb{R}$

  $\text{CategoricalMembershipFunction} \sqsubseteq \forall category.\mathbb{S}$

  $\forall \text{hasValue.LinguisticValue}$

  $\exists \text{hasValue}^{-1}\text{.LinguisticVariable}$

  $\forall \text{hasMembershipFunction.MembershipFunction}$

  $\exists \text{hasMembershipFunction}^{-1}\text{.LinguisticValue}$

  Note: *In the implementation of SquareMembershipFunction, ensure the following:*

  1. *The values of pointA and pointB are equal.*
  2. *The values of pointC and pointD are equal.*

Where we use inclusion ($\sqsubseteq$) to express subclass relationships and properties, logical operator "and" ($\sqcap$) to denote the intersection of classes and conjunction of properties and universal quantification ($\forall$) for property range restrictions. The dot symbol "." is used in property

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

restrictions to denote the range of a property and the symbol $\mathbb{S}$ is used to denote any string. The existential symbol ($\exists$) expresses that there exists at least one instance of the property whose value falls within the specified range. Together with the exponential notation ($^{-1}$) this serves to define inverse properties.

Quantifiers

In addition to linguistic variables that describe the attributes of a process, it is necessary to define a series of fuzzy quantifiers. Fuzzy quantifiers are used to express quantities in a vague or imprecise manner and are essential for creating meaningful and interpretable summaries of large data sets in natural language form (see Section 3.1.3). They allow for the representation of aggregate information in a way that is more aligned with human language and cognition, making the summaries more intuitive and easier to understand for decision-makers and analysts. Quantifiers may be absolute such as *"approximately 5"* and *"around 10"* or relative *"most"* or *"many"*. In this framework, both fuzzy numbers and fuzzy quantifiers are treated as linguistic values (see Section 3.1.3): the membership function associated with each linguistic value describes the degree to which a particular value (which could be a number, a term, or a phrase) belongs to the set defined by the variable. Based on this notion we define the following classes and properties in the ontology:

Definition 8: ***Fuzzy Quantifiers Model***

- Classes:

$$\text{AbsoluteQuantifier} \sqsubseteq \text{LinguisticValue}$$
$$\text{RelativeQuantifier} \sqsubseteq \text{LinguisticValue}$$

- Domain and Range Restrictions:

$$\text{AbsoluteQuantifier} \sqsubseteq \forall \text{hasMembershipFunction.FuzzyMembershipFunction}$$
$$\text{RelativeQuantifier} \sqsubseteq \forall \text{hasMembershipFunction.FuzzyMembershipFunction}$$

Note: *In the implementation, ensure that the FuzzyMembershipFunction for RelativeQuantifier is defined in* $\in [0, 1]$

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

With this formalization we define quantifiers as linguistic values which have inherently associated a membership function, in this case, restricted to a fuzzy membership function and a linguistic term expressing an imprecise absolute or relative quantity.

Time

Apart from being used as quantifiers, in practical applications, fuzzy numbers are often used in scenarios where precise numerical data is either unavailable or unnecessary. For example, in estimating the time required to complete a task, one might use a fuzzy number to represent *"about 30 minutes"*. Knowing that the time aspect is of high relevance when dealing with processes (see Section 3.2) as it is precisely the control-flow and temporal aspects that difference the analysis of processes from other type of structured data. Using the notion of fuzzy temporal reference introduced in Section 3.1.4 and based on the fuzzy temporal model proposed in [61], we propose a fuzzy time layer in the ontology to represent and handle the use of imprecise temporal expressions in natural language descriptions of processes. This model is based on the concept `TemporalReference`, a class representing instances of time references as *"in year 2018"*, or *"before April"* whose semantics are defined based on fuzzy logics through a membership function. In particular, we propose the following classes:

Definition 9: ***Fuzzy Time Intervals Model***

- Classes:

$$\begin{aligned}
&\text{TemporalReference} \sqsubseteq \text{LinguisticValue}\\
&\text{FuzzyInterval} \sqsubseteq \text{TemporalInterval}\\
&\text{ImpreciseInterval} \sqsubseteq \text{TemporalInterval}\\
&\text{Instant} \sqsubseteq \text{TemporalInterval}\\
&\text{BeforeImpreciseInterval} \sqsubseteq \text{ImpreciseInterval}\\
&\text{AfterImpreciseInterval} \sqsubseteq \text{ImpreciseInterval}
\end{aligned}$$

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Domain and Range Restrictions:

  TemporalInterval $\sqsubseteq \forall$hasMembershipFunction.FuzzyMembershipFunction

  Instant $\sqsubseteq \forall$hasMembershipFunction.SingletonMembershipFunction

  ImpreciseInterval $\sqsubseteq \forall$hasMembershipFunction.SquareMembershipFunction

  Note:

  1. *For the MembershipFunction associated with BeforeImpreciseInterval:*
     The value of property pointA is intended to be $-\infty$.
  2. *For the MembershipFunction associated with AfterImpreciseInterval:*
     The value of property pointD is intended to be ∞.

With this formalization, we define a fuzzy temporal model capable of managing uncertainty in the temporal dimension. The linguistic term associated with each fuzzy time interval effectively conveys the underlying temporal notion of that interval. This flexibility in representation allows the same period to be expressed in different ways, accommodating various linguistic styles and contexts. For instance, a specific time span can be referred to as "the first semester of the year" or alternatively as "between January and June," both conveying the same time frame but in distinct linguistic forms. This versatility is key in adapting to diverse communication needs and preferences, providing a range of descriptive options for the same temporal interval.

Message definition: Protoform layer

Once the domain and fuzzy models are defined, messages that organize the content and predicate the entities and their properties and relationships can be defined. The definition of messages is based on the protoform-based model introduced in Chapter 3. This way, protoforms serve as intermediate representations of the natural language descriptions that the system will eventually generate. For this reason, an ontology layer is defined for modeling the proposed protoforms and their semantics.

The use of protoforms as messages grounds the aggregation and referring expression generation processes into their semantics and those of its constituents, allowing for the generation of higher-level, richer specifications by inference through a semantic composition framework [103]. The proposed model categorizes different types of protoforms which are

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

used as templates for message generation including all the components defined in the previous layers of the ontology, in particular, quantifiers (e.g., 'most', 'many'), qualifiers (conditions or restrictions), summarizers (descriptive terms), and fuzzy temporal references. Based on the protoforms introduced in Section 3.2, the following model is proposed:

Definition 10: ***Protoforms Model***

- Classes

$$\text{Protoform} \sqsubseteq \text{Thing}$$
$$\text{TypeIProtoform} \sqsubseteq \text{Protoform}$$
$$\text{TypeIIProtoform} \sqsubseteq \text{Protoform}$$
$$\text{QualitativeProtoform} \sqsubseteq \text{Protoform}$$
$$\text{QuantitativeProtoform} \sqsubseteq \text{Protoform}$$
$$\text{TemporalProtoform} \sqsubseteq \text{Protoform}$$
$$\text{QualitativeTypeI} \sqsubseteq \text{TypeIProtoform}$$
$$\text{QualitativeTypeII} \sqsubseteq \text{TypeIIProtoform}$$
$$\text{QuantitativeTypeI} \sqsubseteq \text{TypeIProtoform} \sqcap \text{QuantitativeProtoform}$$
$$\text{QuantitativeTypeII} \sqsubseteq \text{TypeIIProtoform} \sqcap \text{QuantitativeProtoform}$$
$$\text{TemporalTypeI} \sqsubseteq \text{TypeIProtoform} \sqcap \text{TemporalProtoform}$$
$$\text{TemporalTypeII} \sqsubseteq \text{TypeIIProtoform} \sqcap \text{TemporalProtoform}$$
$$\text{QuantitativeProtoform} \sqsubseteq \forall \text{hasQuantifier.Quantifier}$$
$$\text{TemporalProtoform} \sqsubseteq \forall \text{hasTemporalReference.TemporalReference}$$

With this model, we formalize the protoforms proposed in Chapter 3 in a general way that allows to model not only the exact protoforms introduced in Section 3.2 — and thus their corresponding instances — but all their possible variations. Through the inclusion, or not, of temporal intervals, quantifiers or qualifiers, as well as the use of composite qualifiers or summarizers for describing more than one attribute a broad spectrum of process descriptions is supported. Examples may include a temporal, Type-I, composite, qualitative protoform like

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

"In the first semester of the year, the wait time is around average and the throughput time is long", which was defined as "temporal contextualization of attributes" in Chapter 3, a Type-II, quantified sentence such as *"In most occasions where cases have a high number of exceptions, throughput time is long"* or a Type-I absolute quantified, temporal frequency, sentence such as *"In approximately 50 cases, activity CAT happens shortly after activity RX"*, which was defined as a "frequency and relations between events protoform" in Chapter 3.

Generalization capabilities

In addition to the already mentioned advantages, this knowledge representation formalism represents a notion general enough to model processes in any domain. When dealing with a process in a new domain in which any particular group of attributes may be of special interest their inclusion in the descriptions just requires the definition of their corresponding linguistic variables and values as a series of rules on how to semantically represent this attribute on the basis of the input data during the fuzzification process. For example, adding the year in which a case starts allows to add temporal referenced to summaries. The same thing happens for the inclusion of new elements into existing protoform models or the inclusion of new protoform models int the message ontology. The only requirement for these new message types to be generated is to define their structure together with a series of rules on the conditions under which each new proposed protoform model may be instantiated.

4.2.2 Stage 1: Data interpretation

Preprocessing

As being really ad-hoc, processes (from any domain) are recorded in many different ways by the information systems in which they are implemented, so no standard way of extracting information from processes exists. However, process mining defines a model on how process elements relate to one another, this way, parting from the raw data, and based on this model, it is possible to generate event logs to which process mining techniques can be applied. The input to this stage is the original process log. Manipulation by a process expert is required. We will use the AS ICP as an illustrative example, as commonly some preprocessing is needed in order to apply process mining techniques to healthcare data [152]. An outline of the original registry of clinical data from the AS ICP is shown in Table 4.1 (only some columns are shown, as the original number of variables is too large for including all). As it can be seen it does not look as

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 4.1: Extract of the original registry of clinical data from the Aortic Stenosis Integrated Care Process.

ID	Origin	Date_UA	Date_CT	Date_RX	Frailty	Date_MS	Interv	Date_Interv	Date_Rel	Sex	Birthdate
1	1	-	13/01/2018	24/11/2017	0	26/01/2018	0	07/06/2018	12/06/2018	1	04/04/1938
2	1	-	22/08/2017	04/10/2017	-	18/12/2017	0	26/06/2018	24/08/2018	1	12/04/1956
3	1	31/05/2018	15/03/2018	-	0	16/03/2018	1	03/07/2018	02/10/2018	0	13/09/1936
4	1	14/08/2018	10/04/2018	03/10/2016	1	27/04/2018	1	31/07/2018	10/10/2018	0	22/01/1934
5	1	-	09/01/2018	26/09/2017	1	02/11/2017	1	19/04/2018	30/10/2018	0	15/04/1936
6	1	04/02/2018	-	28/06/2017	1	05/12/2017	0	01/06/2018	24/09/2018	1	11/03/1942
7	1	06/02/2020	27/03/2018	28/07/2014	0	06/04/2018	0	05/09/2018	17/10/2018	1	06/05/1935
8	1	04/03/2019	15/07/2018	15/01/2018	0	06/07/2018	0	06/11/2018	19/12/2018	1	18/04/1935

a classical event log used in process mining analysis, but rather as a data table (suitable for data mining), in which each tuple represents a patient, and each column one variable defined for the process. These variables include patient characteristics, such as origin of admission (Origin), results of fragility test (Fragilty), type of intervention (Interv), or sex of the patient (Sex), the dates of execution of diagnostic and therapeutic interventions, such as date of the last unscheduled admission (Date_UA), date on which a CT scan was performed on the patient (Date_CT), date of the heart team meeting of a patient (Date_MS), or date of its intervention (Date_Interv), among others.

As this registry is not suitable, as is, for applying process mining techniques to it, some preprocessing is necessary. An event log is constructed using the *patient id* as the case identifier, i.e. each patient will represent a case in the process, and the sequence of diagnostic and therapeutic interventions it goes through will represent the trace it follows. Thus, each diagnostic or therapeutic intervention will represent an event in the process. The distinct set of all events in the process represents the possible activities that can be executed, these are: the diagnostic or therapeutic interventions that can be performed on patients, patient management activities, and unexpected events. All attributes will be considered case attributes (apart from the executed activity and the timestamp of an event), as in the original log attributes refer to each patient, and as each patient is considered a case, all attributes are understood as case attributes. The output of this stage is an event log in the correct format for applying process mining techniques. Table 4.2 shows an excerpt of the event log constructed from the AS ICP original registry.

Process model discovery

Once an event log properly defined from the input data, the next step is to discover the underlying process model. Figure 1.3 depicts the model of the process used as a case study in this proposal. This is done by applying a discovery algorithm. Although many have been proposed in the

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 4.2: Preprocessed event log extracted from the original AS ICP registry from Table 4.1. Columns starting with *"event_"* refer to event attributes, and columns starting with *"case_"* refer to case attributes.

case_id	event_activity	event_time	case_sex	case_age	case_frag
1	consultation	2013-06-04 09:00	Male	84	1
1	special-consultation	2012-06-14 09:00	Male	84	1
1	echocardiogram	2012-06-21 09:00	Male	84	1
1	consultation	2012-06-21 10:00	Male	84	1
⋮	⋮	⋮	⋮	⋮	⋮
21657	echocardiogram	2012-10-25 09:00	Female	76	0
21657	consultation	2012-10-25 10:00	Female	76	0

literature [185], here we use the inductive miner algorithm as it provides sound models with fitness one [186]. Fitness assesses how well a model supports the behavior seen in the event log i.e. the proportion of behavior recorded in the event log possible according to the model. In order to be able to replay all the behavior found in the event log, it is necessary to have a model with fitness one. Without this requirement, any other algorithm proposed in the literature could have been used [18, 187, 188].

Case-perspective representation and event log enrichment

Event logs are usually represented in a tabular manner as shown in Table 4.2. Individual rows make reference to each event in a trace, holding information of the event itself (activity and other attributes) and the case it belongs to. For particular questions it is possible to convert an event log into a case-perspective log. This process is called feature extraction or feature encoding, and encodes a log in a way that can be used for any sort of data-driven technique such as Machine Learning or LDD. In Chapter 3 we introduced the concepts of case-perspective log and event log enrichment. Figure 4.4 illustrates these first stages of the pipeline.

For the generation of linguistic summaries of a process we will focus on the case attributes. To do this, we create a view [31] of the process focused on cases and its attributes. In a case-perspective log, each individual row characterizes one particular instance of the process (case) instead of a single event: parting from the original event log, all events referring to the same case are grouped. A new feature called "trace" is created, representing the sequence of activities executed in a particular case and encoded as an ordered list. For the rest of attributes, multiple values are possible during the execution of the process. In case this happens, if the attribute is categorical, we keep the most recent value of attribute. If it is numerical, the average

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

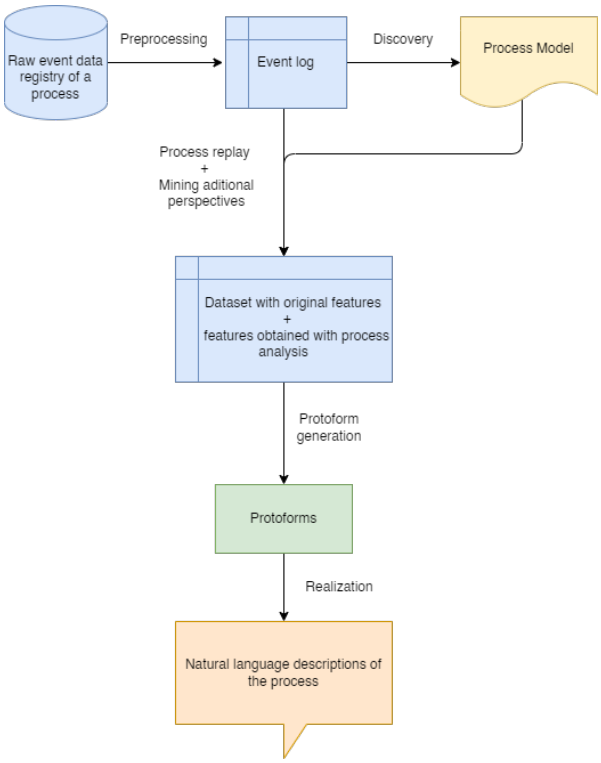


Figure 4.4: Data transformations that take place within the Process-to-Text pipeline. The original raw log is converted into a preprocessed event log, after applying a discovery algorithm a process model is obtained. With the process model and the event log, process replay and mining of additional perspectives can happen.

value is computed. Some loss of information is possible during this process, however, this problem is out of the scope of this work.

These are attributes that refer to a case as a whole rather than an individual event (see Definition 3). Table 5.2 shows an example of the case attributes stored in an event log. This format is more conducive to the generation of linguistic summaries, as it aligns the data structure with the requirements of linguistic summarization, enabling more effective extraction of meaningful insights from complex process data. For example, it becomes possible to generate summaries like *"Most cases have a short throughput time"* or *"Several cases with high number of choices have a high wait time"*.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Event Log Enrichment

Additionally, attributes to those originally present in the event log may be computed during a feature extraction process [12]. As any other data, event logs can be enriched with various additional attributes [31, 178]. Mainly, by deriving or computing additional attributes based on the log itself, or by adding external data. New case attributes computed during this stage are added to the case-perspective event log previously defined.

One particular process used to define temporal attributes is process replay [12]. Process replay involves simulating the sequence of events in the log according to the paths dictated by the process model. Using the original event log and a process model as inputs, events in each trace are fired over the model following the allowed behavior (supported by the model) from start to finish, checking whether each event in the log aligns with the expected sequence and timing as per the model. Replay establishes a strong relation between a process model and reality (as in the form of the event log the process model is extracted from) by relating events in the log to activities in the process. Figure 4.5 shows an example of how replay works.

Replay allows for different purposes [186]: discrepancies between the log and the model can be found and quantified (conformance checking), give operational support to partial traces that are still running (detecting possible deviations or suggesting future actions) or, in this particular case, process replay is used to extend the event log with control-flow and temporal features. While replaying each case, the system records the timestamp of each event. By comparing the timestamps of consecutive (as by the model, not the log) activities, the waiting time between them can be calculated. Once waiting times are calculated during the replay, these values are added to the case-perspective log as new attributes. For example, a new attribute named "WaitingTime_A_B" could be added to each case row in the log, storing the calculated waiting time between Activities A and B for that case. This allows for the generation of descriptions following the proposed protoforms:

During replay, the events in the log get related to activities in the model, allowing to identify relationships between activities in each case i.e. which activity precedes and succeeds each other, how often this happens and what is the wait time between them. This allows characterize each case from the control-flow and time perspectives through a series of new features, namely: the start and end times of each case, the waiting time between each pair of activities (if a transition between two activities happens more than once in a case, the average waiting time is computed), the throughput time and total waiting time, among many others. The reader is directed to [186] to get more insights in replay works and its possible uses.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

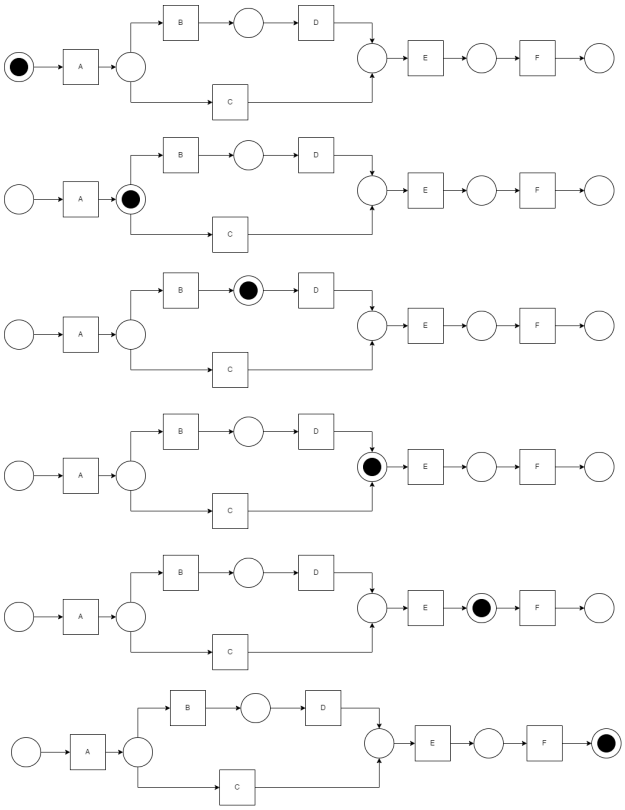


Figure 4.5: Replaying of trace = $\langle a, b, d, e, f \rangle$ on top of its corresponding model (represented as a Petri Net).

At this stage, any other possible attribute (based on the data present in the event log) could be computed based on expert knowledge. Future work includes extending the proposed ontology with knowledge to support the definition of new features in a similar way to the Key Performance Indicator Ontology defined in [189]. The output of this stage is an enriched case-perspective event log, which now contains as new case attributes, the collected frequency and timing information, among any other computed attribute.

Table 4.3 shows an excerpt of the output of this stage for the AS ICP case. The log is now enhanced with the collection of durations for the transition between activities *heart team meeting* and *CT scan* and the duration of the stage between the *heart team meeting* and the *intervention* of each case. The corresponding case-perspective view of this event log is provided in Table 4.4.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 4.3: Enhanced event log with the wait time (expressed in days) between the heart team meeting session and the intervention of a patient and the heart team meeting and the CT scanning of a patient extracted through replaying. Duration is positive if origin event precedes destination event and negative if destination event precedes origin event. Columns starting with *"event_"* refer to event attributes, and columns starting with *"case_"* refer to case attributes.

case_id	event_activity	event_time	case_sex	case_age	case_fragility	case_MS-intervention	case_MS-CTscan
1	consultation	2013-06-04 09:00	Male	84	1	46	-13
1	special-consultation	2012-06-14 09:00	Male	84	1	46	-13
1	echocardiogram	2012-06-21 09:00	Male	84	1	46	-13
1	consultation	2012-06-21 10:00	Male	84	1	46	-13
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
21657	echocardiogram	2012-10-25 09:00	Female	76	0	23	-118
21657	consultation	2012-10-25 10:00	Female	76	0	23	-118

Table 4.4: Dataset corresponding to AS ICP event log with its original attributes as well as the newly calculated with process mining techniques.

case_id	trace	admittance	age	sex	int_year	num_events	num_tests	HTM_INT	CT_HTM
1	$\langle incl, valor, ergo, coro, ... \rangle$	ambulatory	82	male	2018	0	3	46	-13
2	$\langle incl, valor, coro, CT, ... \rangle$	ambulatory	64	male	2017	0	2	23	-118
3	$\langle emerg, emerg_admi, incl, valor, ... \rangle$	ambulatory	84	female	2018	1	2	24	-1
4	$\langle emerg, emerg_admi, incl, valor, ... \rangle$	ambulatory	86	female	2018	1	3	32	-10
5	$\langle incl, valor, coro, CT, ... \rangle$	ambulatory	84	female	2017	1	4	41	68
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
21655	$\langle incl, valor, coro, CT, ... \rangle$	ambulatory	78	male	2017	0	2	48	75
21656	$\langle emerg_admi, incl, valor, coro, ... \rangle$	ambulatory	85	male	2018	0	2	46	-7
21657	$\langle emerg_admi, incl, valor, coro, ... \rangle$	ambulatory	85	male	2018	0	3	67	49

4.2.3 Content Determination

System Identification: Linguistic Variable Definition

At this stage, the semantics of the process are extracted from the case-perspective log in a way they can be later used in the protoform generation, micro-planning and realization stages. Through the use of protoforms as a knowledge representation formalism, its semantics and those of its components support the meaning of the explanations generated by the system. For this reason, a comprehensive knowledge base of linguistic variables and values, along with their respective membership functions, plays a pivotal role in the P2T architecture, as this knowledge base contains both the protoforms and their composing elements necessary for the generation process. As it was introduced in Chapter 3, when talking about fuzzy systems, the definition of this knowledge base, in particular the definition of membership functions, in terms of number, shape and position over the universe of discourse, is part of a process known as *system identification* [165]. In practical applications, the design of these components can be either established based on standard criteria, customized according to user input or expert knowledge or learnt through an inductive process that is driven by the analysis of the input data.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Domain knowledge significantly contributes to the definition of these linguistic variables [147]. Experts in the domain of the process can provide insights into how different terms (like *"high"*, *"medium"*, *"low"* in the context of wait time or blood pressure) should be represented as fuzzy sets. This involves defining the shape (triangular, trapezoidal, Gaussian, etc.) and the range of each term. For this reason, when available, expert knowledge should be used for the definition and refinement of linguistic variables and values, further enhancing the quality of the *fuzzification* process and bringing it closer to the own reasoning an expert on the domain of the process would perform. When no specific domain or expert knowledge is available or when standard knowledge or data notions apply to the data of the problem, the system identification process can be automated by deriving membership functions from data. This involves analyzing the data to identify patterns or distributions, and then using this information to define membership functions that best represent the linguistic terms in the context of the data. This induction method ensures that the linguistic values accurately reflect the variations and nuances present in the actual data. Standard criteria for the definition of fuzzy partitions on the basis of data are typically used.

Many different linguistic variables represent generic concepts like *"high"* or *"low"*, that can be adapted for different situations, so using some standard criteria fine-tuned for a particular problem, may be the best approach. For example, a fuzzy partition — this is, dividing the universe of discourse into a series of fuzzy sets each associated with a linguistic value — can be defined over a general interval to be later adapted to particular discourse universes through scaling or normalization functions. For instance, Figure 4.6 illustrates the definition of linguistic variable *Wait Time* on the basis of standard criteria. When the system identification process is done by learning the parameters from data, historically, genetic algorithms have been the most explored in the literature[1] [190], but other alternatives, like Neural Networks, may be used [190] (cf. [191, 192]).

In our particular case, during the *system identification* process, for each attribute of the process, a corresponding linguistic variable is defined. These linguistic variables are then partitioned into several linguistic values, together with its corresponding membership function. In particular, a combination of standard criteria and expert knowledge is proposed. The following series of rules for the definition of linguistic variables is proposed:

- For numerical features (continue or discrete): 5 linguistic values are defined as trapezoid

[1] The use of genetic or evolutionary algorithms to design fuzzy systems constitutes the branch of the Soft Computing paradigm called genetic fuzzy systems (GFSs)

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

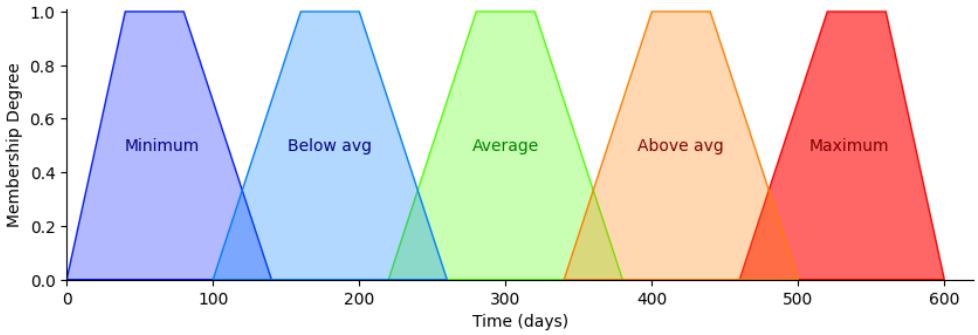


Figure 4.6: Graphic representation of the five linguistic values (*"minimum", "low", ..., "really high"*) defined as fuzzy sets on the basis of standard criteria for the linguistic variable *"Wait time"*.

fuzzy sets using the midpoint and range of the original domain to define each membership function. A neutral value labeled as *"average"* is defined centered in the midpoint of the domain; *"below average"* and *"above average"* are defined right next, and extreme values extremely above average and *"extremely below average"* get defined at the extremes of the domain. Figure 4.6 illustrates this with an example [2].

- For categorical discrete features with less than 5 values: a linguistic value gets defined for each possible value of the feature with a membership function defined as a boolean function.

For numerical variables, an initial definition of the linguistic values and their corresponding membership functions is done through a fuzzy partition based on the domain of the variable. For instance, Figure 4.6 illustrates the definition of linguistic variable *Wait Time* which is originally defined in $[0, 600]$ on the basis of standard criteria. A fuzzy partition is defined over the discourse universe based on the domain of the original numerical variable. In particular, the fuzzy sets are defined using trapezoidal membership functions, which are a common choice in fuzzy logic for representing linguistic terms (see Section 3.2). For each numerical attribute, the following elements are defined:

- **Min:** The minimum value of the variable in the dataset, denoted as min.
- **Max:** The maximum value of the variable in the dataset, represented as max.

[2] Data quality is really important during this process, as incorrect data, missing or extreme values, can affect the definition of linguistic values.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- **Range:** The difference between the maximum and minimum values of the variable, calculated as range = max − min.
- **Width:** This is a subdivision of the range, dividing it into five equal parts, defined as width = range/5. This allows for the definition of 5 equal partitions.
- **Midpoint:** The midpoint of the domain of the numerical variable in the dataset, serving as a central reference point for defining the fuzzy sets, computed as midpoint = range/2.
- **Base:** A further subdivision of the width, used to define the boundaries of the trapezoidal functions, given as base = width/2.
- **Core:** A smaller subdivision of the width, specifically half of a third of the width, used to fine-tune the trapezoidal membership functions, calculated as core = (width/3)/2.

With these elements, the following partitions are defined:

- **Average**[3]: Centered around the midpoint, it spans from (median - base - core) to (median + base + core). This set is intended to capture values that are close to the median of the data.
- **Great Below Average**: Focuses on values significantly lower than the median. It starts from the minimum value of the variable and extends up to (min +2 × base + core).
- **Great Above Average**: Concentrates on values significantly higher than the median. It begins from (max −2 × base − core) and stretches up to slightly beyond the maximum value.
- **Above Average**: Designed to cover values that are higher than average but not excessively so. It starts from (median + base − core) and extends to (median + 3 × base + core).
- **Below Average**: Caters to values slightly below the average, ranging from (median − 3 × base − core) to (median − base + core).

At the end of this process, a linguistic value has been defined for each feature and with a series of corresponding linguistic values on the basis of general knowledge or expert knowledge. The extracted semantics get embedded into the ontology, so that during the fuzzification stage, each case is associated with its corresponding attributes and semantics.

[3]Despite using the midpoint of the domain, linguistic term *"average"* is used as it gives a simpler or more understandable notion than talking about the median, for non-expert users

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Fuzzification: Fuzzy Inference and Ontology Population

Populating the ontology in accordance with the domain model involves a systematic process of data extraction, transformation, and integration. Initially, cases and attributes are extracted from the event log. Following extraction, these elements undergo a transformation process to align them with the ontology's structure and terminologies. In particular, case attributes get aligned with the domain model through a *fuzzification* process, in which each case attribute gets represented as a linguistic term that the original attribute value in a linguistic manner. This step ensures that the data is not only consistent with the ontology's schema but also semantically aligned with its conceptual framework. Subsequently, the transformed data is integrated into the ontology, effectively populating it with instances that represent cases as recorded in the log. An instance representing the particular process under analysis is added to the ontology, together with instances for each of the cases together with their corresponding `hasCase` relationship. This integration enables the enriched ontology to serve as a comprehensive knowledge base, representing in a linguistic manner knowledge about a process while adhering to the predefined domain model.

In the fuzzification process, each numerical (or categorical) attribute of a case is evaluated against a linguistic variable defined for said attribute (see Chapter 3). These variables are defined during the so-called *system identification* process. For instance, for a numerical attribute *numerical wait time*, a linguistic variable *WaitTime* is defined, with linguistic values *"long"*, *"short"* and *"average"*, each associated with a membership function in the form of a fuzzy set. During the fuzzification process, for each case, the value of *numerical wait time* is assessed against these linguistic values. The linguistic term associated to the linguistic value that best represents the attribute — based on the highest degree of membership — is then associated with the case. Once the fuzzification process is complete, the case instance gets related to the corresponding linguistic terms into the ontology. In Figure 4.3 together with the depiction of the domain model, it can be seen how an instance of a case (Case1) is related with the linguistic term *"LongWaitTime"* through property *hasWaitTime*. This process transforms raw data into a format that is more intuitive and semantically meaningful, aligning the numerical or categorical attributes with human-like linguistic categorizations.

An overview of the fuzzification process together with the system identification is given in Figure 4.7. Inference processes are depicted in color green and with round corners. Classes and entities of the respective fuzzy ontology and domain ontology are depicted as orange boxes with squared corners. And in blue, the event log that is used as the underlying source of data.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

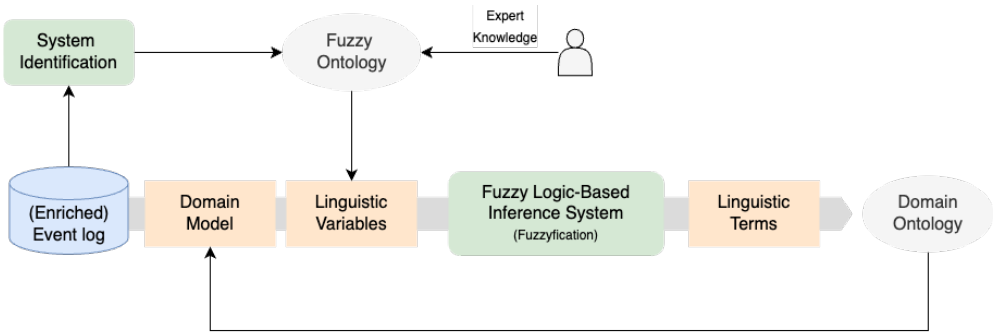


Figure 4.7: The system identification and fuzzy inference processes for the derivation of case attribute semantics and their conversion into linguistic terms that are finally included into the domain ontology.

An expert user is depicted next to the fuzzy ontology, as expert knowledge can be used during the system identification process for the definition of the semantics of linguistic terms.

In the context of an ontology-based system, fuzzification functions effectively as a semantic annotation process [193], enriching numerical data with linguistically meaningful terms. The linguistic terms resulting from the fuzzification process not only provide a richer semantic layer to the data but also facilitate a more intuitive and human-like understanding of the information. By integrating these fuzzified terms into the ontology, they act as semantic annotations, enhancing the ontology's ability to depict real-world complexities in a nuanced and interpretable manner. This semantic annotation, enables advanced operations like semantic search and reasoning based on the qualitative aspects of data. For instance, a SPAQRL[194] query like the one in Figure 4.8 would allow us to retrieve the average membership value of case instances with a *long waiting time*. It should be noticed, that through the taxonomy of attributes defined based on classes, the `hasAttribute` property can be used to make reference to any attribute, and only the precise linguistic term depicting the attribute is needed for retrieving case with a particular attribute value.

Due to the inherent limitations of ontology languages and reasoning engines in handling certain types of logic computations, the fuzzification process is carried out by an automated, external inference system based on the principles of fuzzy logic. This reasoner[4] is specifically designed to handle the evaluation of linguistic variables and the application of fuzzy rules. This external system functions as an intermediary, bridging the gap between the raw numerical data

[4]We will use the term inference to refer to the abstract process of deriving new knowledge and the term reasoner to refer to a specific element in the formalism that performs this task.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

```
PREFIX ex: <http://example.org/ontology#>
SELECT (AVG(?membershipDegree) AS ?averageMembershipDegree)
WHERE {
?case ex:hasAttribute ?attribute .
?attribute a ex:LongWaitTime .
?attribute ex:membershipDegree ?membershipDegree .
}
```

Figure 4.8: A SPARQL query to compute the average value of the membershipDegree for instances with an attribute of class LongWaitTime.

and the semantically rich environment of the ontology. The process of associating each term with a linguistic value and its characteristic membership function occurs externally. Within the ontology, the focus is on utilizing these linguistically intuitive terms, which have been semantically grounded through fuzzy logic, to represent the data, keeping the representation straightforward yet semantically meaningful. This approach allows the ontology to reflect a more flexible and realistic interpretation of knowledge, particularly beneficial in domains where precision intersects with qualitative analysis.

4.2.4 Protoform generation

Once the semantics have been extracted protoforms can be generated. For the generation of messages in the form of protoforms, an engine able to interpret fuzzy classes and properties, making inferences based on degrees of truth rather than binary true/false values is needed. As reasoning systems implemented classically within ontological frameworks like Hermit [195] or Pellet [196], we need to define a an inference mechanism based on the proposed ontology. This inference process is in essence the process of generating linguistic summaries on the basis of a knowledge base defined for a particular problem, with the particularity that the knowledge base takes the form of an ontology, which can be modeled as a search problem over the semantic space defined by the elements in the domain ontology and the message ontology. This process is explained in detail in Chapter 3.

Instancing a protoform involves assigning values to its elements (quantifier, qualifier and summarizer) and computing its truth degree $T \in [0, 1]$, i.e. the degree to which the protoform

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

conveys true information about a referential. The closer it is to one the more truthful the summary is. To compute the truth degree of a protoform, the membership degree of each element of the referential to the property enunciated by the summarizer gets computed. A quantification model allows to aggregate all these membership degrees into a single value (called cardinality) and evaluate this value against a fuzzy quantifier. Resulting in a single truth degree that relates how representative a protoform is to its underlying data. In our approach, Zadeh's quantification model is used [169, 170], although many other quantification models could be used on its behalf [102, 150].

This way, an instanced protoform is characterized by the elements that are used to fill its structure (quantifier, summarizer, qualifier and the underlying referential) and its corresponding truth degree. The challenge now is to automatically extract (instance) true protoforms from the case-perspective log. Being the most naive strategy an exhaustive search over the semantic space of the process (given by the linguistic variables and quantifiers extracted in the previous stage) i.e.: generating all possible combinations protoform structures, quantifiers and linguistic variables (as summarizers and qualifiers), compute the truth degree, and retain those protoforms that are true (or that its truth value is above certain threshold).

Based on the proposed knowledge representation formalism, generating the proposed protoforms is a mater of combining all protoform classes defined in the protoform layer of the ontology together with all the protoform components defined in the domain and fuzzy layers of the ontology (time reference, quantifier, linguistic value) and compute the truth value for any given instanced protoform. As in the domain ontology each element of the referential we are going to define i.e. the cases of a process, are already characterized in a linguistic manner, as the corresponding linguistic values for each case have been assigned during the fuzzification process. The computation of truth degrees, following Zadeh's quantification model (see Chapter 3) is achieved by computing using the Σcount i.e., the average or sum (depending on wether an absolute or relative quantifier is in use) of the membership degrees of cases which show the attribute or combination of attributes (including temporal intervals) enunciated in a protoform. Thanks to the proposed ontological representation, semantic search capabilities could be harnessed for this computation, as illustrated in Figure 4.8.

A series of improvements can be done to reduce the search space and optimize the generation process. Resulting in a non-exhaustive search and reducing the number of generated protoforms in a similar way as the proposed in [128]. In particular, Algorithm 3.1 is modified as shown in Algorithm 4.1.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Algorithm 4.1 Generating Linguistic Summaries

```
Initial Parameters
θ : [0, 1] is a truth value threshold
linguisticVariables is the set of linguistic variables
variablesToUse and qualifiers are subsets of linguistic variables for filtering
quantifiers is the set of defined quantifiers

Filtering Linguistic Variables
filteredVariables = Filter linguisticVariables by variablesToUse
filteredQualifiers = Filter linguisticVariables by qualifiers

Generating Type-I Sentences
type1 = ∅
for variable in filteredVariables do
    for value in variable.values do
        s = Create Type-I Sentence(variable, value)
        if s.truthValue > θ then
            type1 = type1 ∪ {s}
            type1 = type1 ∪ GenerateQuantifiedSentences(s, quantifiers, θ)
            type1 = type1 ∪ GenerateTemporalSentences(s, quantifiers, θ)
        end if
    end for
end for

Generating Type-II Sentences
type2 = ∅
for variable in filteredQualifiers do
    for value in variable.values do
        s2 = Create Type-II Sentence(variable, value, type1)
        if s2.truthValue >= θ then
            type2 = type2 ∪ {s2}
            type2 = type2 ∪ GenerateQuantifiedTemporalSentences(s2, quantifiers, θ)
        end if
    end for
end for
```

This algorithm introduces a series of improvements in generating linguistic summaries, enhancing both the quality and efficiency of the process. The selective filtering of linguistic variables, using subsets *variablesToUse* and *qualifiers*, narrows down the semantic space to be explored. This approach reduces computational complexity by avoiding the consideration of all variables, focusing instead on those most relevant. The inclusion of a truth value threshold θ further refines the process, ensuring that only sentences with a significant truth value are processed, thereby enhancing the relevance of summaries and excluding less truthful descriptions. Additionally, the algorithm integrates quantifiers and temporal elements, adding depth to the summaries. Unlike basic algorithms that focus on static descriptions, this feature allows for the representation of dynamic and quantitative data aspects, offering a more

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

comprehensive view. The iterative and combinatorial construction of sentences, where type-II sentences build upon type-I sentences, leverages the computations done in previous steps and results in richer, more contextually layered summaries generated in a computationally-efficient way.

4.2.5 Micro-planning

Once protoforms have been generated, it is necessary to select which ones are going to be conveyed to the user. This is part of the so-called quality framework. In the previous step, the truth value has already been used as a quality measure to discard all those protoforms that do not convey true enough information. Furthermore, through the proposed improvements the number of generated summaries is reduced by bounding the search space. So, in practice, the pruning task of the micro-planning stage is subsumed by the document planning stage for optimization reasons: pruning the search space during the generation process allows to discard irrelevant explanations before they are generated.

One shortcoming of using protoforms as a knowledge representation formalism is their limited semantic expressiveness. Trying to improve this lack of expressiveness, many different protoforms have been proposed in the literature adding temporal information, domain specific elements, double quantification, etc. as a way to enrich their semantics. Despite these improvements, the expressive power of protoform-based approaches is still quite limited. As most approaches in the literature consider protoforms as individual primitives — i.e., protoform instances correspond in a one-to-one manner to the linguistic summaries generated — the resulting texts are still limited to collections of isolated single-sentences summaries.

In an attempt to overcome this limitation, based on Rhetorical Structure Theory (RST) [197], a framework to model semantic relations (e.g. motivation, antithesis, background, evidence, condition, etc.) between protoforms is proposed in [103]. A composite protoform P^r is defined as the composition of two or more protoforms, one acting as nucleus P^N and one or more as satellites P^S. When defining a new composite protoform, constraints on the nucleus, satellites, and its combination are used to specify the type of protoforms that be used as constituents, and more detailed conditions that should be fulfilled. The truth value of a composite protoform is computed by aggregating the truth values of the constituent protoforms (see Section 3.2.4).

An example definition of a composite protoform can be seen in Table 4.5. With semantic composition, protoforms are no longer individual primitives, but building blocks of larger and

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Algorithm 4.2 Generating Complex Linguistic Protoforms

```
Initial Parameters
θ : [0, 1] is a truth value threshold
σ : [0, 1] is a strength value threshold
type1, type2 are sets of Type-I and Type-II sentences
validQuantified is a set of valid quantified sentences

contrastSentences, evidenceSentences, emphasisSentences = ∅
for nucleus in validQuantified do
    sameSummarizerSentences = Filter sentences with same summarizer as nucleus
    for satellite in sameSummarizerSentences do
        contrastSentence = Create Contrast Protoform(nucleus, satellite)
        if ValidSentence(contrastSentence, θ, σ) then
            contrastSentences = contrastSentences ∪ {contrastSentence}
        end if
        if IsTypeII(satellite) then
            evidenceSentence = Create Evidence Protoform(nucleus, satellite)
            if ValidSentence(evidenceSentence, θ, σ) then
                evidenceSentences = evidenceSentences ∪ {evidenceSentence}
            end if
        end if
    end for
    potentialSatellites = Filter for Emphasis Protoform(nucleus, type1)
    for combination in Powerset(potentialSatellites) do
        emphasisSentence = Create Emphasis Protoform(nucleus, combination)
        if ValidSentence(emphasisSentence, θ, σ) then
            emphasisSentences = emphasisSentences ∪ {emphasisSentence}
        end if
    end for
end for
```

richer texts (explanations), greatly enhancing the expressiveness of the system. Based on the composition framework proposed in [103] and the proposed protoform ontology a taxonomy of composite protoforms for processes can be built. Examples of these composite protoforms are the compliance, comparison and deviance protoforms presented in Chapter 3. Together with these, evidence and emphasis relationships are defined in [103].

The basic algorithm for generating the proposed composite protoforms is to iterate over every type of composite protoform and generate all possible combinations of type-I and type-II protoforms that passed the previous filters and that conform to the constraints imposed by each composite protoform definition. To reduce the number of generated explanations, only higher-level descriptions (those based on composite descriptions) are given to the user, unless no higher-level descriptions exist for a particular process element. Furthermore, once a protoform is used as a constituent element of a composite protoform, it gets discarded to prevent the

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 4.5: Specification of the *Contrast* extended protoform.

Relation: Contrast
Protoform: P^N but, P^S
Semi-verbalized: *Q Xs are A,* ***but*** *Q Bs are A', Q Cs are A"...*
Constraints on P^N: P^N is a type-I quantified sentence
Constraints on P^S: P^S is a type-II quantified sentence
Constraints on $P^N + P^S$: $Bs, Cs, ... \subset Xs$
Example: In most cases, attribute waiting time is long, but in many cases where attribute origin is emergency admittance, attribute waiting time is very short.
Verbalized example: Most patients have a long waiting time between its TAC and intervention, however, patients admitted under emergency conditions are intervened immediately.

repetition of information, as conveying both the composition and the individual descriptions would result in redundancy. This process is illustrated in Algorithm 4.2.

A new quality measure called *"relation strength"* is defined. This measure indicates the intensity of the semantic relation between the constituent protoforms of a composition. It is modeled generally by means of a function μ_r that determines the intensity of the semantic relation between a nucleus and a satellite protoform — the constituent protoforms of a composition — for a specific semantic relation r. These functions are usually defined based on expert knowledge. However, as linguistic values, standard criteria can be applied during the system identification process in order to derive them in an automatic manner.

4.2.6 Realization

The final stage of the pipeline is the linguistic realization phase. By defining messages on the basis of protoforms, we bring the definition of a message in our framework closer to the notion of *proto-phrase* introduced in [56]. A proto-phrase is the intermediate representation used in the micro-planning stage of a NLG or D2T system, where the abstract, domain-based structure of a message is transformed into a linguistic-focused representation. A proto-phrase is still an abstract representation of a natural language description, but in a lower abstraction level than a message. In a proto-phrase, the syntactic structure of a description is already defined and the lexicalization process has already taken place, meaning the specific words that will be used in the final description have been already chosen — in a basic form.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

For each of the protoforms proposed in the framework, a proto-phrase — as semi-verbalized base structure — is defined. This way, the basic syntactic structure of each type of description is fixed, but the terms used in it (summarizer, qualifier, quantifier) are selected and accordingly modified (number, tense, negation, etc.) for each of the natural language descriptions generated. Some additional logic exists in the templates; for example, when using an aggregation value as a feature summary (e.g. using the average value of a certain feature to give a quick insight of it instead of using a quantifier), the part of the template that relates "In Q cases..." gets replaced to "Cases...". Also, if multiple properties are used as a summarizer or qualifier, certain modifications need to happen, as using the correct connectives in each place is necessary for the correctness of the description. These semi-verbalized utterances can then be further processed in the next stages of the realization stage.

Once the generated explanations have been grouped and turned into more refined semi-verbalized utterances at the micro-planning stage, they are ready for the realization stage. For this task we will use an End-To-End approach in which the semi-verbalized utterances are fed into a LLM to get properly expressed i.e. remove redundancies and repetitions by introducing referring expressions, correct grammatical or syntactical errors, introduce some variation in the texts, etc. Compared to rule-based realization approaches [10, 56], End-To-End approaches have the advantage of producing more fluent and natural texts. Their main general disadvantage is the existence of hallucinations and omissions, which in our case will not occur due to the hybrid design strategy we have followed. From an auditing and replicability standpoint the most desirable LLMs are those built under open-source initiatives like BLOOM [198], and Open-Llama [199]. However, due to computing infrastructure constraints, GPT-3.5 [90] was used. Using an open-source model for this task and comparing the results to the current ones is part of future planned work.

A crucial aspect in this stage is the prompt given to the LLM, i.e. how to specify to the LLM the task it must perform. The prompt used for the linguistic realization of composite sentences, on the basis of their constituents, can be seen in Table 4.6. Also, a mapping of the different verbalizations of each process element is extracted from the ontology and given as input to the LLM so it properly replaces each entity with its corresponding verbalization. It is also important to set adequate values for the rest of the parameters of the model, so we prevent hallucinations, repetitions or omissions. The temperature parameter is set to 0.15. As lower as this parameter gets the model is less "inventive" and so less prone to hallucinations but more prone to repetitions. In an environment where accuracy is more important than

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 4.6: Prompt used for the linguistic realization task in the P2T framework.

Perform referring expression generation and combine the following sentences into a better written text, don't use lists or enumerations, write a rich and clear text. It is really important that the generated text conveys exactly the same information, but it is just better written. If you notice correlation, relationships, or influence of variables, please, highlight it. Case = patient, activity A = medical surgical session, activity B = surgery, for attribute origen 1 = emergency, 0 = ambulatory, waiting_time = waiting time between the medical surgical session and the surgery.

Table 4.7: Example of a query and the result obtained from the system.

Query: How does the admittance origin of a patient affect the waiting time between activities A and B? **Semi-verbalized utterance**: In most cases where attribute origin is ambulatory, attribute waiting time is > 30, **but** in many cases where attribute origin is emergency, attribute waiting time is < 30 **Realized result (LLM)**: The origin of a patient has a significant impact on their wait time between medical surgical sessions (activity A) and surgery (activity B). Specifically, when patients are admitted ambulatorily, most cases experience a wait time of over 30 days. On the other hand, when patients are admitted through emergency services, many cases experience a wait time of less than 30 days.

linguistic refinement (not that the later is irrelevant), some repetitions are more desirable than hallucinations. Anyhow, in this approach we ensure the accuracy and reliability of the realization process by design. As reliable content validated *a priori* is used as a base and only a linguistic realization task (not the content determination) is entrusted to the LLM, the possibility of hallucinations occurring is greatly reduced. Additionally to this prompt, the LLM is fed the semi-verbalized utterance resulting from the previous step. And as output, a linguistically enhanced text is returned. An example of this process can be seen in Table 4.7.

By deriving the content conveyed in the descriptions on the basis of symbolic reasoning processes, this LLM-extended approach allows to generate context-aware, accurate and reliable by-design descriptions which are linguistically enhanced on the basis of LLMs. Grounding the generation process in reasoning and inference becomes specially relevant as, in the context of D2T, research shows that texts generated by neural models may show discrepancies with the input data [94].

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

4.3 A use case in the healthcare domain

As a case study, a process from the healthcare domain is used to illustrate the types of descriptions that can be generated through the proposed protoforms. In particular, we use the process related to the management of patients in the Valvulopathy Unit of the Cardiology Department of the University Hospital of Santiago de Compostela [49]. In this unit, consultations and medical examinations are performed to patients with aortic stenosis [200] in order to decide their treatment. Patient information and management activities are also recorded in the event log. As heart diseases (in particular ischemic cardiopathy) are the most frequent cause of death worldwide [201], analysis of cardiology processes is of great interest, specially for practitioners in the healthcare domain. This use case has been developed in collaboration with medical experts from this unit, whom have helped in the identification of relevant aspects of the process and whose expert knowledge was incorporated into the generation process.

Aortic stenosis (AS) is a chronic disease and the most prevalent valvular heart disease, especially in elderly patients [202]. In the absence of treatments, valve replacement is currently the only intervention with the potential to improve the prognosis of AS patients [203]. However, when intervention is indicated, it is generally not performed immediately. As the paradigm of care for the patient with AS involves different professionals who at different times and places provide care that is fragmented, with little continuity of care and a high risk of lack of coordination. In contrast, Integrated Care Processes (ICP) seek to ensure the effectiveness of clinical actions through greater coordination and a guarantee of continuity of care [204]. In January 2018, an ICP of AS was implemented in the Valvulopathy Service of the Cardiology Department of the University Hospital of Santiago de Compostela, with the aim of managing the care of patients requiring intervention and guaranteeing the correct and coordinated functioning of each one of its phases[49].

As this case study was carried out before the introduction of the LLM-extended linguistic realization stage in the P2T framework (see Section 4.2.6), process descriptions were generated following the conventional rule-based D2T methodology with a template-based approach. This approach has less flexibility and linguistic variety, but provides greater reliability as the content of the descriptions is controlled. The SimpleNLG-ES [205] realization engine, (the Spanish version of the original SimpleNLG engine [206]) was used for realizing the templates into syntactical and grammatically correct texts.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

4.3.1 Process definition

The AS ICP covers all diagnostic and therapeutic interventions, from the heart team meeting[5] (formal inclusion of the patient in the process) to the patient return to normal activity (patient exit from the process). The following stages are distinguished:

1. **Stage 0**: Prior to the inclusion of the patient in the process, all the necessary tests will be requested and evaluated for presentation at the heart team meeting with the objective of identifying severe AS patients amenable of intervention.
2. **Stage 1**: From the decision to intervene in the heart team meeting to the valve replacement procedure.
3. **Stage 2**: Surgical or percutaneous valve replacement procedure.
4. **Stage 3**: Rehabilitation and follow-up after intervention in order to facilitate the patient's recovery after surgery.

In order to monitor the results of the AS ICP, a registry of clinical data of the patients included in the process has been established. Information about the execution of all diagnostic and therapeutic interventions as well as patient management activities are recorded. Monitoring of unexpected events (e.g. mortality, unscheduled admission due to cardiovascular causes, emergency room attendance due to cardiovascular causes) is available in the registry as well.

4.3.2 Problem definition

Process and outcome quality indicators are defined to analyze safety, identify funnels, bottlenecks and futile actions, and implement improvements. The aim is to reduce unexpected events and unscheduled admissions, reduce overall delays in patient care, improve adherence to scientific recommendations and care protocol, and implement continuous quality improvement initiatives.

Multiple key process indicators (KPIs) are defined to evaluate the results of the AS ICP. Medical experts in the Valvulopathy Service show genuine interest in applying process mining techniques to extract valuable knowledge and relationships between the different KPIs defined. The KPIs of interest for the AS ICP, and that will be described in this work are the following: proportion of patients with emergency admittance, compliance with the standard of wait time

[5]The heart team meeting is a multidisciplinary clinical session in which cardiac surgeons, vascular surgeons, clinical cardiologists, and cardiac imaging experts evaluate patients with heart disease amenable to surgical treatment.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

between the heart team meeting and the intervention of a patient, relation between the type of intervention and the wait time for a patient, delay between the heart team meeting and the intervention caused by CT scanning of a patient and influence of the type of admittance in the wait time.

Due to the complexity of the process and the numerous variables and scopes defined for its evaluation, a problem arises when conveying the process mining results to the medical experts in the Valvulopathy service, as severe difficulties arise in its comprehension. The lack of good visualization techniques for process mining analysis results in the state-of-the-art, emphasizes the need of proposing novel ways of conveying the results of process mining analysis to users in a clear, and comprehensible way. At this stage, medical experts show interest in natural language descriptions of the AS ICP and its process mining analysis results, which can help in the understanding the process and the implementation of improvements. As the elements under description are common to any other healthcare process, and to any process outside the healthcare domain: relating case attributes to KPIs, analyzing the control-flow perspective (paths followed by cases in the process), getting insights about frequency and timing of events, reducing unexpected or undesired behavior (reduce unexpected events), reduce wait times (overall delays in cases), among may others. We propose a general model, able to encompass all these process elements, and therefore applicable to processes in any domain.

Following the extended protoforms proposed in the previous section, we show how these descriptions are generated with the proposed framework, what information is used in each case and what knowledge or techniques are used for guiding the generation process of the descriptions. More specifically descriptions about temporal contextualization of case (patient) features, frequency and temporal relations between activities, compliance with clinical guidelines and comparison between groups of patients, are generated.

4.3.3 Temporal contextualization of features

One of the AS ICP concerns in respect of temporal contextualization of features is how many complementary tests are performed on a patient in Stage 1[6]. The following descriptions are generated:

[6]Stage 1 spans from the decision to intervene a patient in the heart team meeting, to the actual valve replacement procedure

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- In year 2018, an average of 3 complementary tests are performed to a patient between the heart team meeting and its intervention.
- In year 2019, an average of 2 complementary test are performed to a patient between the heart team meeting and its intervention.

The generation process of these descriptions is as follows:

1. Feature *number of complementary tests* is created and computed by summing the number of complementary test performed between activities *heart team meeting* and *intervention* for each patient.
2. Years *2018* and *2019* are defined as crisp time intervals T_r and summarizer P is defined as the new computed feature *number of complementary tests*. Interval T_r works as a qualifier, defining a subset of the referential that will be evaluated against the summarizer P. This way, only cases starting in year 2018 are selected for the first description, and cases starting in year 2019 are selected for the second description.
3. The average number of tests for each subset defined by T_r is computed.
4. As an average value is given, no quantifier Q is needed and as this value summarizes each entire subset, therefore no evaluation is needed, the truth value of both protoforms is 1.

Experts are also interested in analyzing how the COVID-19 pandemic has affected patient characterization. By defining different years for T_r, experts can easily grasp if any difference in proportion of patients for some feature exists. The decision of generating these protoforms is based on the statistical analysis performed over feature *admittance*. By applying a proportion test on the data, comparing the proportion of *emergency admittance* patients in years 2019 and 2020, a statistically significant difference in proportion was found. The following descriptions are generated:

- In year 2019, in some cases (38.8%), patients had emergency admittance.
- In year 2020, in approximately half of cases (50.6%) patients had emergency admittance.

These descriptions allow experts to get some first insights and even see how the feature they are interested on, evolves during time. The generation process is as follows:

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

1. A set of proportional quantifiers is defined as in Figure 3.3.
2. As before, T_r is defined as a crisp time interval and acts a qualifier. All cases are evaluated against qualifier T_r.
3. In this case, summarizer P makes reference to the type of admittance patients have in the process, this is feature *admittance* of cases (patients).
4. Feature *admittance* of all cases is selected and evaluated against membership function of summarizer P. As feature *admittance* only has two possible categorical values, a membership function for each category (each linguistic value) is defined as in Equation (3.6). Linguistic value *emergency admittance* is taken for the evaluation.
5. Conjunction of sets of membership degrees to T_r and C is computed and aggregated into a membership proportion.
6. Membership proportion is taken and evaluated against the selected quantifier.
7. Protoform is generated.

4.3.4 Frequency and temporal relations between activities

On the AS ICP, attention is mainly focused on analyzing and reducing wait times in Stage 1. So medical experts show interest in descriptions that address activity relationships in this stage, mostly, how the CT scanning of a patient temporally relates to the heart team meeting and, if the CT scanning takes place after the heart team meeting, how much of a bottleneck represents for the intervention of the patient[7]. Medical expert knowledge is used to determine which protoforms to generate based on the interest in analyzing particular KPIs of the process. The following descriptions are generated:

- In approximately half of cases (52.72%), patient had a wait time between its CT scanning and its intervention lower than 30 days.
- In approximately half of cases (56.28%) where patients were intervened via TAVI, a CT scan is performed after its corresponding heart team meeting.

[7]Performing a CT scan is a requirement when percutaneous prosthesis implantation (TAVI) is performed, so the performance of the CT scan entails the main possible bottleneck between the decision to intervene a patient and its actual intervention.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- In approximately half of cases (51.57%) where patients were intervened via TAVI, patient had a wait time between its CT scanning and its intervention lower than 30 days, with a median value of 35 days.
- In most cases (90%) where patients were intervened via TAVI and to whom a CT scan was performed after its corresponding heart team meeting, patient had a wait time between its heart team meeting and its CT scanning lower that 30 days.

For the generation of these protoforms, the required columns of the dataset are: *intervention* (type of intervention performed on a patient), *CT_HTM* (wait time between the CT scanning and the heart team meeting of a patient) and *CT_INT* (wait time between the CT scanning and the intervention of a patient). These will be used as both qualifier C and summarizer R depending on the particular protoform instance. Linguistic values *before* and *after* are defined as crisp linguistic values with membership functions:

$$\begin{aligned} \mu_{\text{before}}(x) &= 1, \text{ if } x < 0 \\ &= 0, \text{ otherwise} \end{aligned} \tag{4.1}$$

$$\begin{aligned} \mu_{\text{after}}(x) &= 1, \text{ if } x > 0 \\ &= 0, \text{ otherwise} \end{aligned} \tag{4.2}$$

Where x is each of the elements of the newly computed attribute *CT_HTM* (wait time between the CT scanning and the heart team meeting of a patient) or *CT_INT* (wait time between the CT scanning and the intervention of a patient)[8]. In a similar way, summarizer *wait time between CT scanning and its intervention lower than 30 days* is defined as a crisp linguistic value with membership function:

$$\begin{aligned} \mu_{\text{wait time}}(x) &= 1, \text{ if } x < 30 \\ &= 0, \text{ otherwise} \end{aligned} \tag{4.3}$$

Value *30 days* is used as it is established as the maximum number of days of wait time between the heart team meeting and the intervention of a patient for providing quality care on the AS ICP clinical guideline.

[8]Functions *before* and *after* are defined the same way for any temporal distance attribute computed, here is exemplified with attributes *CT_HTM* and *CT_INT*.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Additional to the use of fuzzy quantifiers, a specific quantity is given in parentheses, this way, the granularity of the summary is chosen by the user, allowing him/her to grasp a general idea by focusing on the fuzzy quantifier, or getting the exact proportion of patients by focusing on the percentage. Same idea is followed with the median value of the wait time between the CT scanning and the intervention of the patient. With these elements protoforms are generated and evaluated as following:

1. A set of proportional quantifiers is defined as in Figure 3.3.
2. Even time interval T_r can be used, the generated protoforms address the whole process time-span, so no need for defining or evaluating time intervals.
3. For type-II protoforms (all except the first), feature *intervention* of all cases is taken and used as qualifier. This feature is a categorical variable so, when defined as a linguistic variable, a membership function for each category (linguistic value) is defined as in equation (3.6). Linguistic value *intervened via TAVI* is selected.
4. Feature *intervention* of all cases is selected and evaluated against the membership function associated to qualifier C.
5. The corresponding feature used in each protoform (summarizer) is selected for all cases and evaluated against the membership function of the selected linguistic value of the summarizer P (wait time or temporal relation between activities).
6. Conjunction between the sets of membership degrees for qualifier C and summarizer P is computed.
7. The resulting set of membership degrees from the conjunction is aggregated and the proportion of membership is computed.
8. Membership proportion is evaluated against all possible quantifiers defined in step 1. Selecting the one with a higher truth degree.
9. Protoform is generated.

As quantity (e.g. *approximately half of cases*), temporal reasoning (e.g. *after*) and duration (e.g. *less than 30 days*) of a relationship between two activities (e.g. intervention and CT scanning) is given, this descriptions provide revealing insights to medical experts about where

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

and how frequently bottlenecks in the process occur. Also, as the extended protoforms include a qualifier (e.g. *patients were intervened via TAVI*) relating other features of a case (patient) to the frequency and timing perspective, this allows to further analyze if any relation holds between patient characteristics (e.g. type of admittance, age of the patient, year of inclusion in the process, etc.) or volume variables (e.g. number of events suffered, number of complementary tests performed, etc.) with process variables (compliance with care protocol standards related with timing, delays and bottlenecks and paths followed by patients), thus helping answer the question *"what caused the bottleneck?"*.

4.3.5 Compliance

In the AS ICP interest is focused again on the wait time for Stage 1. By adding the expected wait time for this stage, descriptions help on detecting more easily compliance (or not) with guidelines. Also, medical experts from the Cardiology Department, but not involved in the AS ICP may need to participate in the process at some point, as healthcare processes are highly multidisciplinary [4, 48, 152]. In those cases, new participating experts may not know the clinical guidelines that support the process, thus, basic frequency and temporal descriptions, without further context, may not convey all the necessary information this experts need. By adding this context, all the necessary information for grasping whether a deviance in behavior or timing exists. The following description is generated:

- Wait time between the heart team meeting of a patient and its intervention is expected to be lower than 30 days. However, in approximately half of cases (59.30%) patient had a wait time between its heart team meeting and its intervention higher than 30 days.

Medical expert knowledge is used to determine which protoforms to generate based on the interest in analyzing particular variables of the process. Also, statistical testing was used to assess if statistically significant differences exist between the real and expected values of feature *wait time between heart team meeting and intervention*. In particular, a one-sample t-test was used to discover that statistically significant differences exist between the expected and real value of the feature. This knowledge also determines the usage of the semantic relation "However", as the relevant aspect experts are trying to analyze is if deviances respect to guidelines or expected behaviors are taking place.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

4.3.6 Comparison between groups of patients

In the AS ICP great interest is shown in analyzing if patient characterization affects output or process variables, specially how patient admittance, type of intervention and year of admittance affect wait times. The following descriptions are generated:

- In many cases (76.70%) where patient had ambulatory admittance, patient had a wait time between its heart team meeting and its intervention higher than 30 days. However, in many cases (64.64%) where patient had emergency admittance, patient had a wait time between its heart team meeting and its intervention lower than 30 days.
- In approximately half of cases (42.25%) where patient was intervened in year 2019, patient had a wait time between its heart team meeting and its intervention lower than 30 days. However, in many cases (77.50%) where patient was intervened in year 2020, patient had a wait time between its hear team meeting and its intervention lower than 30 days.
- Cases where patient was intervened via valve replacement had a wait time between its heart team meeting and its intervention of 83 days in average. However, cases where patient was intervened via TAVI have a wait time between its heart team meeting and its intervention of 49 days in average.

By comparing wait times between different years it is possible to analyze how the process has evolved and, if changes made during those years are fruitful and reducing wait times. The generation process is as follows:

1. A set of proportional quantifiers is defined as in Figure 3.3.
2. Even time intervals Ti can be used, the generated examples refer to the whole process time-span. In the example were years 2019 and 2020 are compared, time intervals are used directly as qualifiers referring to a case feature $intervention_year$.
3. Feature used as qualifiers C_1 and C_2 (e.g. *intervention_year*) is taken for $S1$ and $S2$ respectively, and cases (patients) are classified and grouped based on the selected linguistic values (e.g. *2019* as C_1 and *2020* as C_2). This is done by evaluating the selected feature of all cases against the membership function of qualifier C_1 for S_1 and C_2 for S_2 respectively.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

4. The feature used in each protoform as summarizer (e.g. *wait time between its heart team meeting and its intervention*) is taken for all cases and evaluated against the membership function of the selected linguistic values P_1 and P_2 (e.g. *lower than 30 days* and *higher than 30 days*).

5. Conjunction between the sets of membership degrees for qualifier C and summarizer P is computed for each protoform S_1 and S_2.

6. The resulting set of membership degrees from the conjunction is aggregated and the proportion of membership is computed for S_1 and S_2 respectively.

7. Membership proportion is evaluated against all possible quantifiers defined in step 1. Selecting the one with a higher truth degree for each S_1 and S_2.

8. Semantic relation "*however*" is used, as this protoform conveys differences between groups of patients.

Statistical testing is used to check if statistically significant differences exist between groups and decide which descriptions to generate. For these protoforms, proportions tests where used to assess if differences in the number of cases with a waiting time lower than 30 days was statistically significant between patients with different admittance origin and between patients intervened in years 2018 and 2019. A Mann–Whitney–Wilcoxon test was used to determine if statistically significant differences exist for the average waiting time of patients intervened via valve replacement and via TAVI. Based on medical expert hypotheses and using statistical testing accurate descriptions of the process are generated.

4.4 Evaluation of the P2T framework

In this section we present the evaluation of our proposal in a real healthcare environment, the AS ICP. Following the current standards of manual human expert evaluation of the NLG field [179], this evaluation was conducted by 16 medical experts of the Cardiology Department of the University Hospital of Santiago de Compostela who did not participate in the definition of the system requirements. Manual evaluation by human experts is the appropriate evaluation methodology for this case, since automatic metric-based cannot be performed due to the lack of corpus or available datasets and to the vivid discussion on the NLG community about the validity of the metrics used. Comparison to other approaches cannot be performed since, to

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

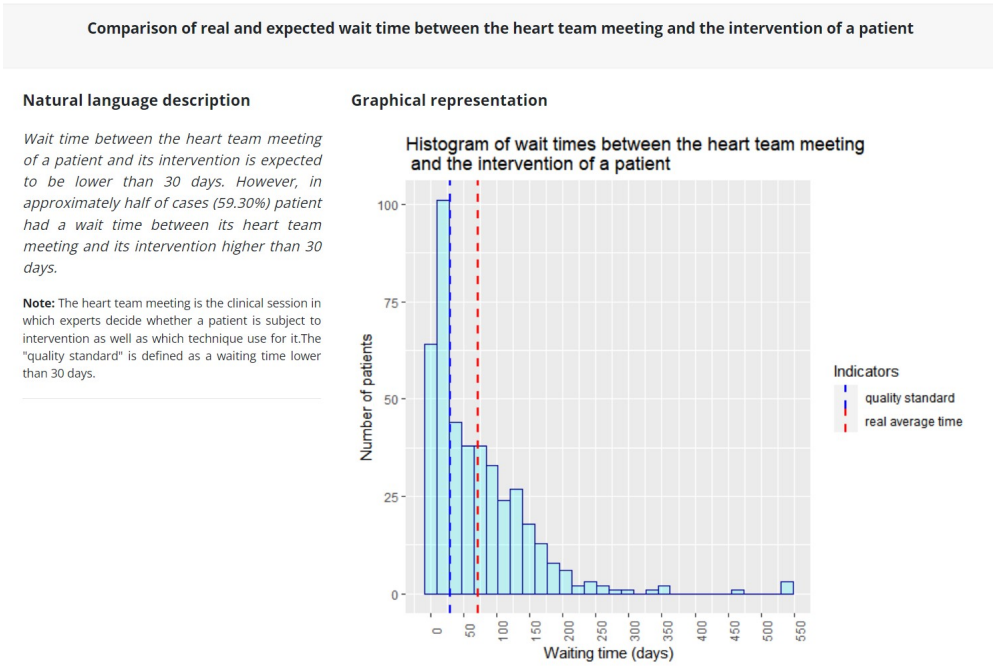


Figure 4.9: One of the scenarios presented in the evaluation questionnaire of the pipeline over the AS ICP (originally in Spanish). In this scenario a natural language description regarding the expected and real wait times between the heart team meeting and the intervention of a patient is shown next to the histogram representing the wait times (in days) for the patients between said activities on the process.

the best of our knowledge, this is the first system which deals with the problem of describing healthcare processes using NLG and particularly the AS ICP.

4.4.1 Set-up

The goal of human expert validation of a NLG system is, in general, to assess to what extent the generated natural language descriptions are understandable for its users and provide truthful and relevant information. Additionally, for the AS ICP, we are also interested in assessing if the descriptions are helpful in order to improve users work quality and, ultimately, if they give answer to any of the questions medical experts may have when dealing with and analyzing the AS ICP, or other possible healthcare processes. Following the best practices of human expert evaluation in NLG systems [179], we created a Likert-scale questionnaire where the

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

dimensions to be assessed are asked to the human evaluators over different cases. Availability of human experts is one of the critical issues when performing manual validation in NLG, since medical experts, who are the human experts for this application, have very limited quality time for performing these tasks. This general problem is solved by focusing the assessment items and the cases included in the questionnaire on the most critical issues to be evaluated, avoiding creating a very long questionnaire involving too many different questions. As described in what follows, each human evaluator answered 9 questions about 6 different cases or scenarios, and 6 more general questions about the approach as a whole. We found this as a very good compromise between the evaluation needs and the limited availability of the medical experts.

The proposed questionnaire consists of 6 cases or scenarios regarding different elements of the AS ICP medical experts have identified as relevant and on which they have interest in get insights of. For each scenario, the natural language description generated with the framework here proposed is presented next to the corresponding graphical representation that would be used in classical process mining analysis. Figure 4.9 illustrates a scenario in which a natural language description regarding the expected and real wait times between the heart team meeting and the intervention of a patient is shown next to the histogram representing the wait times (in days) for the patients between said activities on the process. For each scenario, 9 questions were asked, on a five-level Likert scale ranging from "strongly agree" to "strongly disagree". As a hybrid template-based realization approach is used, form aspects regarding linguistic quality of the descriptions (i.e., spelling and grammatical correctness) do not need to be assessed. Therefore, the evaluation process is mainly focused on content-related aspects, aiming to evaluate if natural language descriptions help on the understanding of process mining analysis results, its relevance and usefulness for the users, and if the descriptions provide interesting information for medical experts. Table 4.8 shows the questions asked for each scenario.

Questions were grouped based on the aspect they evaluate: first two questions (*I1-I2*) assess the content presented on the scenario, helping understand if the information presented is interesting and novel for the medical experts. The second group of questions (*R1-R4*) focuses on the preferred modality of conveying information for medical experts (natural language vs. graphical displays). They aim at evaluating if natural language descriptions are preferred over graphical representations of the same information, by directly asking which representation of the information, or if the combination of both (and in which order of preference) was the most efficient in conveying the information of the scenario. It is relevant to emphasize that this is one of the main objectives of this evaluation, *assess whether medical experts can better understand*

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

process related information based on natural language descriptions rather than on graphical representations. The next two questions (*C1-C2*) assess the quality of the natural language descriptions by asking medical experts to evaluate the ease of understanding the information presented in the natural language description and whether the information was conveyed in a comprehensible way. The last question (*F1*) regards the use of fuzzy quantification and fuzzy terms in the descriptions, by asking if the degree of concreteness in which the information is conveyed in the natural language description (by the use of average values, fuzzy quantifiers, etc.) is adequate and not too succinct to not be informative or too detailed to not give a general enough description of what is happening in the process.

For facilitating and reducing the amount of time users spend when answering of the test, question *R2* is shown only whenever the medical expert has selected a value "Neutral" or worse for *R1*. Similarly, *R3* and *R4*, are only shown when previous questions are answered with values "Neutral" or worse. The answer for all hidden questions in this category was set to "Strongly Disagree", as by agreeing to the previous questions user is implicitly disagreeing with them. This was done after a revision of a preliminary version of the questionnaire, since the AS ICP coordinators highlighted that users would probably consider answering to all four questions as redundant and repetitive, and would not fully complete the questionnaire because of their very limited time availability. This emphasizes the availability of human experts as a critical issue when evaluating NLG systems, as pointed out earlier. Under these challenging circumstances, having fifteen human experts is a rather unusually high number of participants, which makes the results conclusive for this application.

In a similar way to the validation test used for the evaluation of the proposed protoforms in Chapter 3, a version of the Technology Acceptance Model [180] adapted to linguistic summarization [128] is used as a set of general questions asked at the end of the evaluation process in order to evaluate the natural language description of healthcare processes as a whole. This way, once evaluated whether experts prefer natural language or graphical displays, and assessing the quality of the descriptions, direct questions about perceived usefulness and intention of use are asked. Table 4.9 shows this set of questions. Questions *G1-G4* evaluate usefulness and intention of use and question *P1* assesses how familiar the medical expert who is performing the evaluation is with the AS ICP. Through this question, conclusions can be drawn about to what extent the descriptions are found more/less useful or more difficult/easier to understand based on the different medical expert profiles.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 4.8: Questions asked for each scenario on the proposed evaluation test

Id	Question
I1	The information this scenario provides is interesting for me.
I2	The information this scenario provides is new for me (I was unaware of it until now).
R1	The representation which provided me with information most efficiently has been the natural language description.
R2	The representation which provided me with information most efficiently has been the graphical representation.
R3	Natural language description coupled with the graphical representation has provided me with information in the most efficient way. Without the natural language description I would not have understood the scenario.
R4	Graphical representation coupled with the natural language description has provided me with information in the most efficient way. Without the graphical representation I would not have understood the scenario.
C1	Understanding the information this natural language description provides is easy for me
C2	I perceive information is expressed in a comprehensible way.
F1	The degree of concreteness in which the information is expressed in this natural language description is adequate (neither too detailed nor too succinct).

Table 4.9: General questions asked at the end of the evaluation test.

Id	Question
P1	I am familiar with the AS ICP
G1	Natural language descriptions would provide me a better understanding of what happens in my job.
G2	Natural language descriptions would allow me to complete tasks quicker.
G3	Natural language descriptions would increase the quality of my work.
G4	I would find natural language descriptions useful in my job.
G5	I find natural language descriptions easy to understand.

The questionnaire was deployed as a web page with two introductory screens followed by the six scenarios created. The first screen related the objective of the evaluation process, the second one showed instructions regarding what information is shown in each scenario, what each question is trying to assess and how to answer the questions.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

4.4.2 Results

Table 4.10 shows the validation results for each of the proposed questions. Results are shown in a 5-level numerical scale derived from the Likert-scale used in the questionnaire, where 1 equals "Strongly disagree" and 5 equals "Strongly agree". The table shows the average value and its confidence interval at a 95% confidence (CI) (computed using the t-student distribution as the population standard deviation is not known), standard deviation (SD), mode, median and interquartile range (IQR) for each of the asked questions.

As we can see with questions *I1-I2*, in average, results show most descriptions are found to provide really interesting (4.11/5.00) and novel information (3.69/5.00). This is specially relevant, considering that 13 out of the 16 participants (81.25%) were already familiar with the AS ICP. Questions *R1-R4*, in particular question *R1*, show that natural language descriptions are (by far) the preferred way of conveying information for almost all users, who show a very clear preference for natural language descriptions (4.28/5.00) over graphical displays (1.30/5.00). Questions *C1-C2* prove information was found to be quite easy to understand (4.40/5.00) and no user has had any trouble understanding the proposed descriptions because the information was expressed in a comprehensible way (4.38/5.00). Finally, the degree in which information was presented, evaluated with question *F1* is found correct with a high agreement (4.22/5.00). So, not only natural language descriptions convey information in a more convenient way than graphical displays but also users that interact on a daily basis with the AS ICP find novel information on the descriptions. This proves the techniques proposed in this framework are helpful on extracting and conveying process mining analysis results to medical experts in a comprehensible and easy to understand way, utilizing fuzzy terms that help on the summarization of numerical data. Figure 4.10 shows the histograms of response values for each of the asked questions.

The general questions regarding the whole approach also show very positive results, with an average value higher than "agree" (4.07/5.00) for all questions. Table 4.11 shows the average value and its confidence interval (CI) (computed using the t-student distribution as the population standard deviation is not known), standard deviation (SD), mode, median and interquartile range (IQR) for each of the asked questions. Results prove that users are keen on incorporating natural language descriptions into their daily workflow as they find them useful (4.12/5.00) and easy to understand (4.44/5.00). Users also state that natural language descriptions provide a better understanding of what happens at their job (4.06/5.00), allow

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 4.10: Expert validation results for the AS ICP descriptions when taking the whole answer set (6 scenarios per 15 users). Results are shown in a 5-level numerical scale derived from the Likert-scale used in the questionnaire (1 equals "Strongly disagree" and 5 equals "Strongly agree"). Average value and its confidence interval at a 95% confidence (CI), standard deviation (SD), mode, median and interquartile range (IQR) are shown for each of the asked questions.

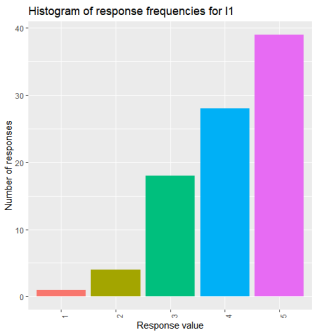
	Average	CI	SD	Mode	Median	IQR
I1	4.11	[3.94, 4.27]	0.95	5	4	1.75
I2	3.69	[3.48, 3.89]	1.16	4	4	2.00
R1	4.28	[4.13, 4.41]	0.79	5	4	1.00
R2	1.30	[1.15, 1.44]	0.80	1	1	0.00
R3	1.26	[1.22, 1.38]	0.76	1	1	0.00
R4	1.12	[1.03, 1.20]	0.47	1	1	0.00
C1	4.40	[4.26, 4.53]	0.75	5	5	1.00
C2	4.38	[4.23, 4.52]	0.82	5	5	1.00
F1	4.22	[4.06, 4.35]	0.78	4	4	1.00

Table 4.11: Expert validation results for the general questions of the AS ICP. Average value and its confidence interval (CI), standard deviation (SD), mode, median and interquartile range (IQR) are shown for each of the asked questions.

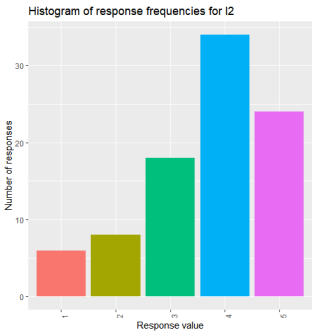
	Average	CI	SD	Mode	Median	IQR
G1	4.06	[3.72, 4.40]	0.77	4	4	1.25
G2	4.06	[3.72, 4.40]	0.77	4	4	1.25
G3	3.69	[3.34, 4.04]	0.79	4	4	1.00
G4	4.13	[3.81, 4.44]	0.72	4	4	1.00
G5	4.44	[4.16, 4.71]	0.63	5	5	1.00

them to complete tasks quicker (4.06/5.00) and increase the quality of their work (3.69/5.00). Thus, reinforcing the conclusions extracted from the results obtained in the previous part of the evaluation test. Question *P1* is not used nor shown on Table 4.11 as all evaluators except three were already familiar with the process, thus, not allowing us to extract any valuable information from this question. Figure 4.11 shows the histograms of response values for each of the asked questions.

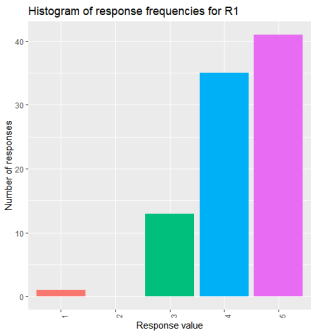
USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



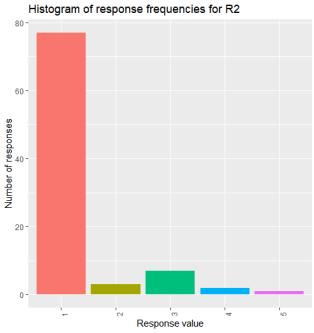
(a) Histogram of response frequencies for question *I1*.



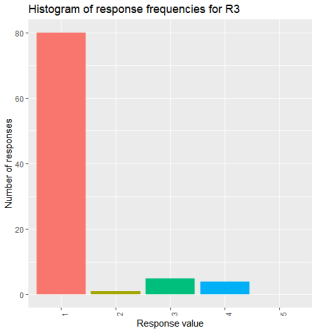
(b) Histogram of response frequencies for question *I2*.



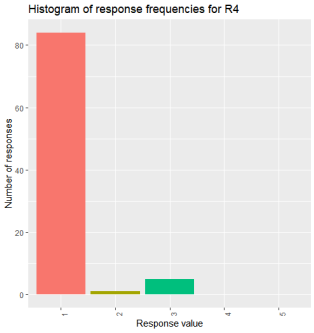
(c) Histogram of response frequencies for question *R1*.



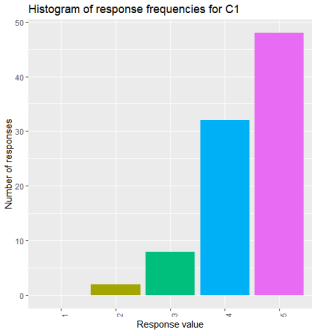
(d) Histogram of response frequencies for question *R2*.



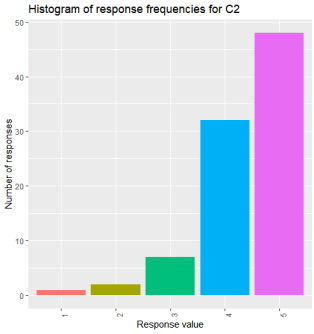
(e) Histogram of response frequencies for question *R3*.



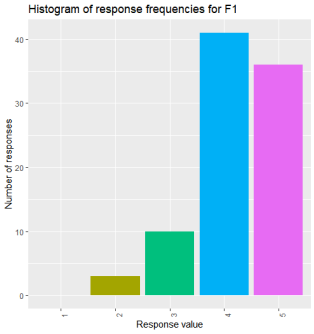
(f) Histogram of response frequencies for question *R4*.



(g) Histogram of response frequencies for question *C1*.



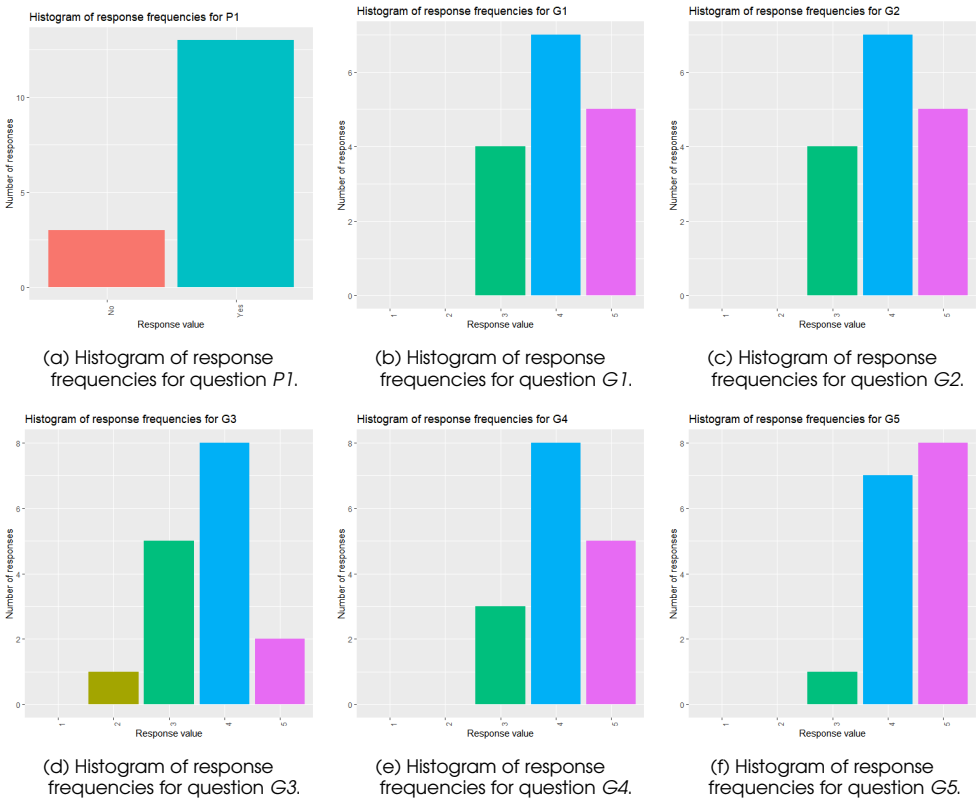
(h) Histogram of response frequencies for question *C2*.



(i) Histogram of response frequencies for question *F1*.

Figure 4.10: Histograms of response frequencies for the questions asked in each scenario.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



(a) Histogram of response frequencies for question *P1*.

(b) Histogram of response frequencies for question *G1*.

(c) Histogram of response frequencies for question *G2*.

(d) Histogram of response frequencies for question *G3*.

(e) Histogram of response frequencies for question *G4*.

(f) Histogram of response frequencies for question *G5*.

Figure 4.11: Histograms of response frequencies for the questions asked in each scenario.

4.5 The Natural Language Generation for Process Mining (NLG4PM) system

In this section we propose a Natural Language Generation system for Process Mining (NLG4PM), which harnesses the P2T framework for its underlying functioning and provides a transparent interface for users to, on the basis of an event log, obtain a series of relevant, truthful by-design and contextually-aware descriptions of a process. The NLG4PM system is implemented in Python and follows a Model-View-Controller [207] or Presentation-Abstraction-Controller [208] architecture. This architecture improves separation of concerns and modularity in software applications by separating business logic, user interface, and interaction control: The *view* component is responsible for presenting information to the user and represents the

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

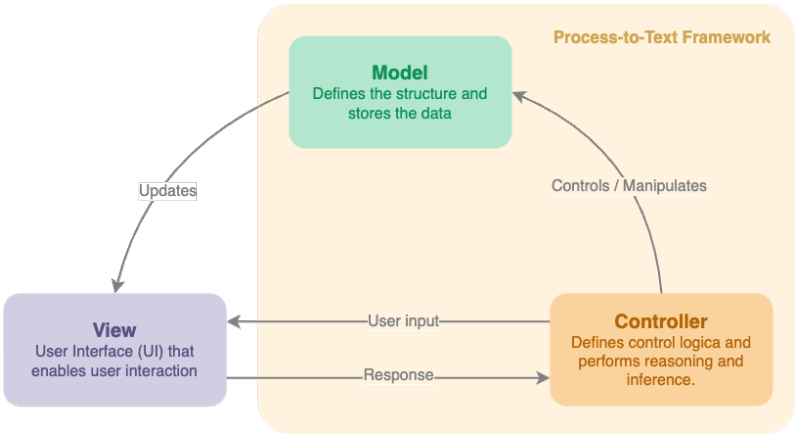


Figure 4.12: Overview of the basic architecture of the NLG4PM system.

concrete syntax of the application, while the *controller* handles user interactions and defines the logic that modifies the model and view in reaction to the input and the data in the model. The *model*, on the other hand, encompasses both the modeling of the structure of the data and its storage and embodies the semantics of the application. The controller is then responsible for maintaining the mapping and consistency between the abstract entities in the model and their representation in the view. An overview of the basic architecture of NLG4PM is depicted in Figure 4.12. It is important to notice that the controller and the model represent indeed the P2T framework, where the ontology serves as the model, and the rest of the knowledge representation formalism and stages of the pipeline serve as the controller. Together with this backend, an interactive interface is designed and implemented.

A series of libraries are used in the definition of the core elements that compose the backbone of the NLG4PM system. These libraries provide the necessary tools and methods required to implement the P2T framework in practice. In particular, for the pre-processing and handling of event logs, the PM4Py[9] library [209] is used. This library provides with the necessary methods for managing event logs, discovering process models, performing process replay, and the computation of relevant attributes. The proposed ontology is expressed in the Web Ontology Language [210] (in particular, version 2 of OWL [211] (OWL 2[10]) is used)

[9] pm4py.fit.fraunhofer.de
[10] w3.org/TR/owl2-overview

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

and implemented in Python through the Owlready2[11] [212] library. Data corresponding to the ontology are stored in a graph database called *quadstore*, that is made persistent in disk as a SQLite3 relational database. The ontology-to-database mapping is automatically done by Owlready2 and automatically updated when new classes, entities or relations are added to the ontology. Entities are loaded on RAM and to Python when in use, and automatically saved when modified. The design of the implemented ontology, due to limitations in the libraries and technologies used for its implementation and for practicality and efficiency, differs slightly from the one proposed in Section 4.1.1. However, this design and implementation follow the same principals of the previously proposed ontology.

For the linguistic realization of the explanations, the OpenAI library[12] is used. This library provides access to the OpenAI API, where the GPT model, used for linguistic realization, can be used. Through this integration, the complete pipeline of the P2T framework forms the backend of the NLG4PM system. This backed provides all the necessary methods for the generation of natural language descriptions of processes guided by user interaction.

The view or interface of NLG4PM is developed based on Dash[13]. A framework for building interactive web applications with rich user interfaces, extensively used in research (cf. [213]). Dash is based on the Flask, Plotly and ReactJS libraries. Leveraging Flask for server-side operations, Plotly for creating interactive visualizations, and ReactJS for building user interfaces, facilitates the development of highly responsive and visually appealing applications. In the context of NLG4PM, the use of Dash enables the creation of a user interface that can dynamically present natural language generation outputs, manage user inputs, and integrate visual elements for an enhanced user experience. This combination of technologies ensures that the view of NLG4PM is not only functional but also intuitive and engaging for users.

4.5.1 Running example

In this section, a running example based on a series of screen captures of the NLG4PM interface is presented to illustrate the generation process of the natural language descriptions. As the underlying functioning of NLG4PM is that of the Process-to-Text framework, the explanation of the implemented model, reasoning processes and generation of descriptions has been provided in the previous sections. For illustration purposes, a running example is provided showing the

[11] github.com/pwin/owlready2
[12] github.com/openai/openai-python
[13] dash.plotly.com

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

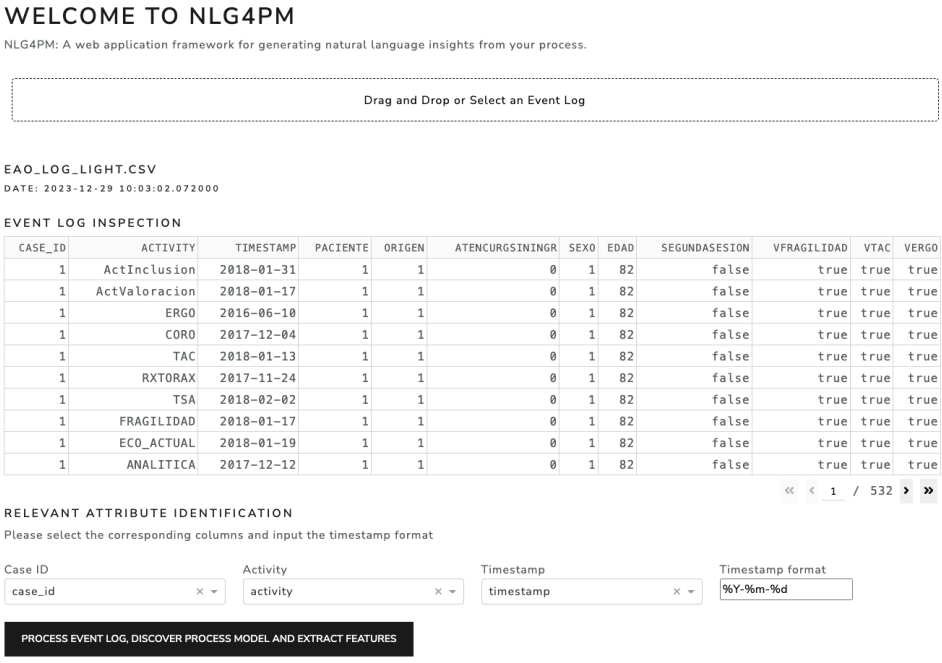


Figure 4.13: Initial screen of the NLG4PM system. Here the user can upload an event log and do an exploratory analysis.

typical interaction of a user with the system. For this example, a simplified version of event log corresponding to the AS ICP process will be used. Please note that, as Spanish is the original language of the process, activity names and attribute values are in Spanish.

When starting the NLG4PM system, the user with a mostly empty screen in which an event log can be uploaded for its analysis and the generation of natural language descriptions. Once a log is uploaded, it gets processed and stored, the interface is updated with a tabular visualization of the log. Based on this visualization, the user must select which column in the log represents the case identifiers, identifies activities and registers the timestamps of events (together with its format). This initial screen is shown in Figure 4.13.

Once identified, the log can be processed with the PM4Py library. During this step, the process model is discovered and process replay is done. Furthermore, a log enrichment process takes place and a series of pre-defined additional attributes are computed. As this new attributes are computed on the basis of the mandatory attributes that any event log must contain, the

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

PROCESS INSPECTION, DATA VISUALIZATION AND NLG PROCESSING

CASE ORIENTED EVENT LOG INSPECTION

Below you can inspect the case oriented event log extracted from the original log, now grouped by case_id. Automatic feature extraction has been performed, and new relevant features are now part of the log. If you wish to define new or extract more features, please press button "DEFINE NEW FEATURES".

DEFINE NEW FEATURES

Click a cell on the table to further inspect its value

TRACE	PACIENTE	WAITING_TIME	START_ACTIVITY	END_ACTIVITY	START_DATE	END_DATE
ERGO,RXTORAX,CO…	1	132	ERGO	ActRevisado	2016-06-10T00:0…	2020-04-03T00:0…
ERGO,RXTORAX,Ac…	10	29	ERGO	ActRevisado	2017-10-06T00:0…	2020-04-03T00:0…
ActValoracion,R…	100	134	ActValoracion	ActRevisado	2018-06-08T00:0…	2020-04-02T00:0…
ActValoracion,C…	102	274	ActValoracion	ActRevisado	2017-10-31T00:0…	2020-04-02T00:0…
RXTORAX,CORO,Ac…	103	37	RXTORAX	ActRevisado	2018-06-21T00:0…	2020-04-02T00:0…
ActValoracion,A…	107	14	ActValoracion	ActRevisado	2018-06-28T00:0…	2020-04-02T00:0…
RXTORAX,ANALITI…	109	184	RXTORAX	ActRevisado	2018-01-18T00:0…	2020-04-02T00:0…
ANALITICA,CORO,…	110	45	ANALITICA	ActRevisado	2018-04-09T00:0…	2020-04-02T00:0…
ANALITICA,ActIn…	111	110	ANALITICA	ActRevisado	2018-05-11T00:0…	2020-04-02T00:0…
ECO_ACTUAL,ANAL…	112	480	ECO_ACTUAL	ActRevisado	2017-10-18T00:0…	2020-04-02T00:0…
RXTORAX,ECO_ACT…	113	-1	RXTORAX	ActRevisado	2017-10-25T00:0…	2018-12-07T00:0…
ECO_ACTUAL,ActV…	114	61	ECO_ACTUAL	ActRevisado	2018-04-10T00:0…	2020-04-02T00:0…
RXTORAX,ECO_ACT…	115	-1	RXTORAX	ActRevisado	2015-12-09T00:0…	2019-12-30T00:0…
RXTORAX,ECO_ACT…	116	254	RXTORAX	ActRevisado	2017-10-24T00:0…	2020-04-06T00:0…
RXTORAX,ActValo…	117	170	RXTORAX	ActRevisado	2011-02-17T00:0…	2020-04-06T00:0…
ECO_ACTUAL,ANAL…	118	26	ECO_ACTUAL	ActRevisado	2018-06-28T00:0…	2020-04-06T00:0…
RXTORAX,ANALITI…	119	90	RXTORAX	ActRevisado	2018-01-05T00:0…	2020-04-06T00:0…
FRAGILIDAD,CORO…	12	52	FRAGILIDAD	ActRevisado	2018-01-16T00:0…	2020-04-03T00:0…
RXTORAX,ECO_ACT…	120	71	RXTORAX	ActRevisado	2018-07-25T00:0…	2018-12-20T00:0…
RXTORAX,ANALITI…	121	64	RXTORAX	ActRevisado	2018-06-23T00:0…	2020-04-06T00:0…

« ‹ 1 / 26 › »

Figure 4.14: The case-perspective, enriched event log is shown to the user.

proposed log enrichment process is general and applicable to processes in any domain. At this stage, a case-perspective view of the log is generated, so the generation of the descriptions focuses on process instances and case attributes. Figure 4.14 depicts this stage, in which, the case-perspective event log, with the newly computed attributes (those whose name is in English) is shown to the user.

At this stage, targeted towards more experienced users, the discovered process model is shown as a DFG, together with an interactive chart in which two process attributes can be selected to explore how they relate. The inclusion of this visualizations comes motivated by the use case in which the P2T framework was evaluated. Here, domain experts show particular interest in the temporal perspective and are somewhat used to these kind of visual representations. Through this chart, a high-level idea of the distribution of the data is provided, which is later further analyzed through the generated descriptions. This screen is shown in

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

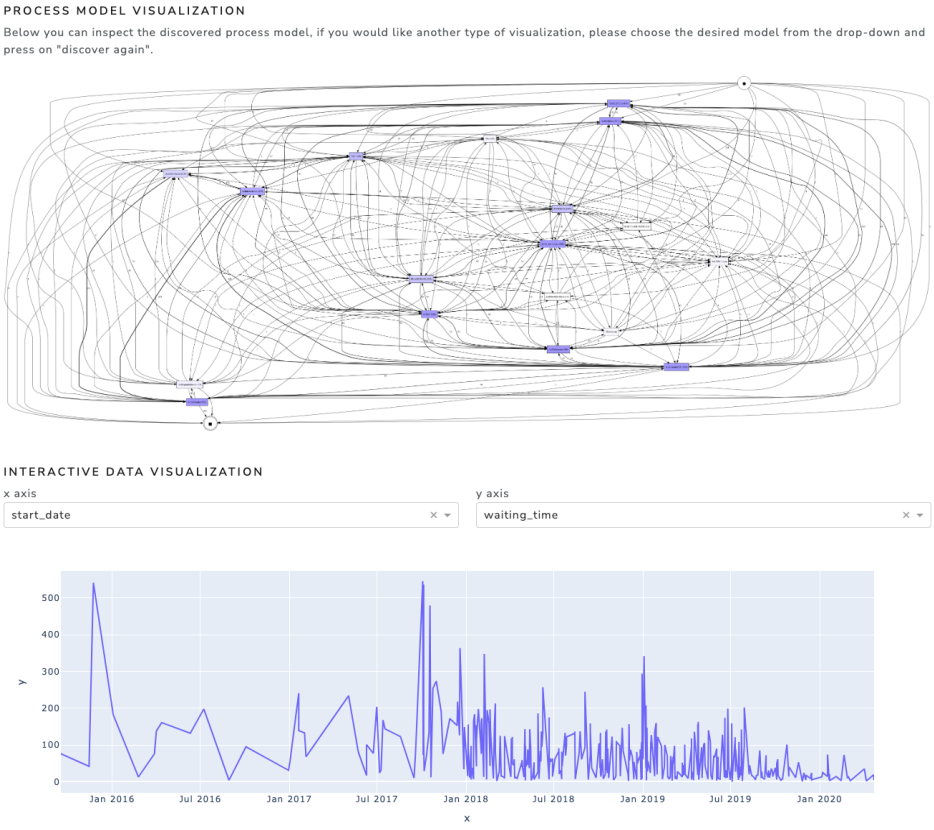


Figure 4.15: The case-perspective, enriched event log is shown to the user.

Figure 4.15, where it can be clearly seen how the discovered process model depicts an spaghetti process, which renders completely useless this visualization, further motivating the use of natural language descriptions for getting insights into the process.

After this is done, the *system identification* and *fuzzification* processes take place and linguistic variables and values are defined on the basis of the event log used as input based on standard criteria see (Section 4.2.3). This is also done for temporal references, based on the temporal span of the process. By default, the temporal range of the process gets subdivided in years and semesters. However, users can specify other temporal references in which they show interest. With these elements, the protoform generation stage can take place. Figure 4.16

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

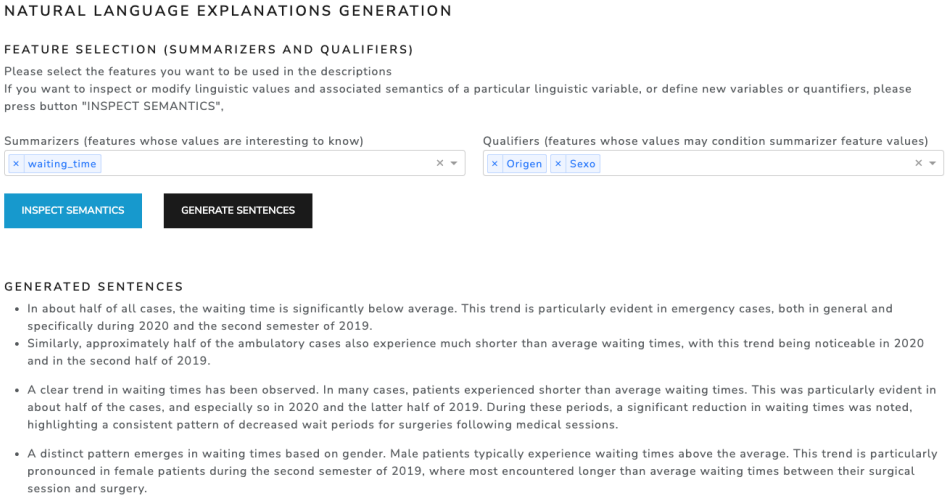


Figure 4.16: The case-perspective, enriched event log is shown to the user.

shows the final screen of NLG4PM in which users can select which attributes should be used as summarizers and which should be used as qualifiers for the generation of the descriptions as part of the content determination process. With the selected elements, protoforms are generated (see Section 4.2.4) and selected based on a quality framework. After this initial generation, the micro-planning stage combines low-level protoforms into more complex descriptions on the basis of their semantics through an inference process. The quality of these protoforms is computed on the basis of the quality of its constituents. Finally, in the realization stage, protoforms are given an intermediate verbalization and get realized into a final text through the use of a LLM. This allows to generate context-aware, accurate and truthful descriptions by-design, as the content conveyed in the descriptions is derived through inference provided by a symbolic reasoning process.

4.6 Discussion and final remarks

In this section we have presented a framework for the generation of qualitative and quantitative natural language descriptions of healthcare processes addressed to medical experts. The framework is complete, since it is able to handle all stages of the generation, from the preprocessing of clinical registries to event logs, to the final generation of the natural language

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

texts. It is based on the most widely used Data-To-Text (D2T) pipeline [10], on the usage of process mining techniques, and fuzzy quantification techniques which allow to model uncertain terms in the natural language descriptions.

The framework is able to handle relevant healthcare process data such as events and its attributes, temporal relations between events, patient attributes, and quantify them during process life-span, recall temporal relations and waiting times between events and its possible causes and compare patients attributes between groups, among other features.

A real application of the framework was presented and validated, over the Aortic Stenosis Integrated Care Process of the University Hospital of Santiago de Compostela. Following the usual standards of D2T systems, manual human validation was conducted for the generated textual descriptions by 16 medical experts in Cardiology. Validation results are very positive, since from a quantitative point of view a global average of 4.07/5.00 was obtained for the general questions related to the understandability and usefulness of the natural language descriptions as well as the capability of medical experts (those who validated the approach) to complete tasks easier and increase the quality of their work. Furthermore, numerical assessment for question *R1* (the representation which provided the users with information most efficiently was the natural language descriptions) was 4.28/5.00, which is much better than the assessment for graphical representations or combinations of both (1.30/5.00, 1.26/5.00, and 1.12/5.00, respectively). More specifically, they show *i)* that the modality which conveyed the information most efficiently about the process was natural language; *ii)* a very clear preference of texts over the usual graphic representation of processes information as the way for conveying information to experts; and *iii)* natural language descriptions provided relevant and useful information about the process, providing ways for its improvement.

Through the P2T framework and its implementation in the NLG4PM system, contextually-aware, precise and factual by-design natural language descriptions of a process can be generated. Through interaction, the NLG4PM system allows users to guide the generation process, resulting in descriptions that specifically target users necessities. As the derivation of knowledge from the input process and the generation of the protoforms that serve as intermediate representations of the descriptions that will be eventually generated is based on symbolic reasoning and inference processes, the truthfulness of the generated descriptions is provided by design, as these reflect knowledge extracted from the data. The use of protoforms allows for a standardized way of inferring and expressing knowledge in the P2T framework. This uniformity ensures that information is presented in a recognizable and predictable manner, making it easier

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

to understand and process, both by humans and computational systems. On this base, the realization stage of the P2T framework is extended with a LLM linguistic refinement process, which, based on semi-verbalized representations of the descriptions, generates syntactic, semantic and grammatically correct natural language descriptions that better resemble the way human experts convey information.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Chapter 5

C-4PM: A Conversational Agent for Declarative Process Mining

These aren't the droids you're looking for.

— Ben Obi-Wan Kenobi *(Star Wars)*

In the preceding chapters, our proposals in the topic of process analysis and natural language were presented. Chapter 3 delved into a formal model supported by fuzzy protoforms, providing a structured approach to the linguistic summarization of processes. This model, while innovative, faced limitations in its adaptability to varying user requirements due to its inherent rigidity.

Chapter 4 expanded on this by presenting a Data-to-Text platform that integrates advanced data and process mining techniques with the protoform-based model. This platform significantly enhances the richness and comprehensiveness of process descriptions by transforming enriched event logs into nuanced linguistic summaries. Despite these advancements, both the formal model and the Data-to-Text platform show a lack of interactive capabilities, restricting their flexibility and application in diverse user-driven scenarios.

Also in Chapter 4 we emphasized the critical role of interactivity in process analysis. This chapter explored the integration of interactive elements into the P2T framework, highlighting the necessity for tools to adapt to the varying needs of users. This focus on interactivity is

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

crucial, as it allows for personalized analysis and understanding of process data, making the analysis more effective and user-friendly. It is this emphasis on interactivity and user-centric design that forms the basis for the development of the current chapter.

In this chapter, we introduce C-4PM as the first conversational agent for declarative process mining. We propose C-4PM as an innovative solution to the challenge of presenting declarative process models to non-expert users in an understandable and queryable format. Unlike procedural models, which can be complex and less suited for linguistic presentation, C-4PM employs declarative models, particularly using the Declare specification, to facilitate various tasks such as consistency, conformance, and model checking through a conversational-driven approach. This novel system not only makes declarative process mining accessible to a broader range of users but also enhances the understanding of process models, mitigating misinterpretations and promoting accurate insights, especially in complex environments like healthcare. C-4PM's ability to interact with users via a text interface, understand natural language queries, and provide responses in natural language makes it a significant leap forward in the field, showcasing the potential of integrating conversational agents and process mining for more effective and user-centric process analysis.

The contents of this chapter are extracted from the following publication:

Fontenla-Seco Y.[a], Winkler S.[b], Gianola A.[b], Montali M.[b], Lama M.[a], Bugarín-Diz A.[a] (2023). **The Droid You're Looking For: C-4PM, a Conversational Agent for Declarative Process Mining**. In *Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2023 co-located with 21st International Conference on Business Process Management (BPM)* (pp. 112-116). Utrecht, The Netherlands. Available at ceur-ws.org/Vol-3469/paper-20.pdf.

[a]Centro Singular de Investigación en Tecnoloxías Intelixentes, Universidade de Santiago de Compostela, Spain.
[b]Free University of Bozen-Bolzano, Bolzano, Italy

5.1 A conversational agent for declarative process mining

Process models constitute a formal description of the steps, activities, and rules that govern a process. A model outlines how tasks are structured, their sequence, and their interdependencies within a process. The challenge of comprehending and interpreting these models is recognized as an open challenge within the Process Mining community [5, 42, 45, 214]. Understanding process models requires highly technical knowledge, a necessity that becomes more evident in

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

more complex and dynamic domains such as healthcare [5, 6, 7]. Commonly, process models are depicted visually using imperative (procedural) styles like Petri nets [20], Workflow nets [21] or Directly-Follows Graphs (DFGs) [12]. However, these traditional methods often fall short in terms of flexibility and clarity, to the point that experts indicate that DFGs, the most used technique for modeling processes [38] often lead to misleading insights [28, 38].

Declarative process modeling, especially when implemented through the Declare language [215], offers a valuable alternative to traditional process modeling approaches. It characterizes processes through a set of rules and constraints on the permissible and prohibited action patterns, without explicitly indicating how process instances (cases) should be routed i.e.: it defines a series of constraints on the temporal evolution of a process. This approach is particularly appealing due to its much more concise and flexible specifications than those resulting from imperative models [215]. The Declare language is structured around a series of templates, each representing a specific type of constraint or rule. These templates are grounded in Linear Temporal Logic on Finite Traces (LTL_f), providing a solid logical foundation that allows for the expression of temporal relationships and constraints in a process. Furthermore, LTL_f gives support to a series of reasoning tasks that allow a more profound analysis and understanding of a process. Yet, while the use of Declare results in more concise specifications, it also introduces a layer of complexity in its analysis. The hidden dependencies [216] that arise from the interplay of the constraints in a specification can make it difficult to discern which process traces are acceptable or not in a declarative specification [217].

To make the benefits of declarative process mining more accessible, especially to those users without deep technical expertise in process mining, a growing need for innovative solutions that simplify the understanding of these models, emerges. In domains like healthcare, practitioners often have the necessary domain knowledge to ask relevant questions about their processes but lack the technical background in process mining to able to do so with the current tools and techniques. Addressing this gap, we introduce C-4PM, the first conversational agent designed for declarative process mining. The motivation to use declarative process specifications in C-4PM comes not only from the capacity of expressing Declare patterns directly in natural language but also because these patterns cover the majority of relationships between activities that are important in practice. C-4PM leverages event logs, to uncover the underlying declarative specification of a process, effectively enabling users to perform multiple reasoning tasks, such as consistency, conformance, and model checking, though natural language queries. This initiative marks a significant step towards bridging the knowledge gap in process mining, making it more

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

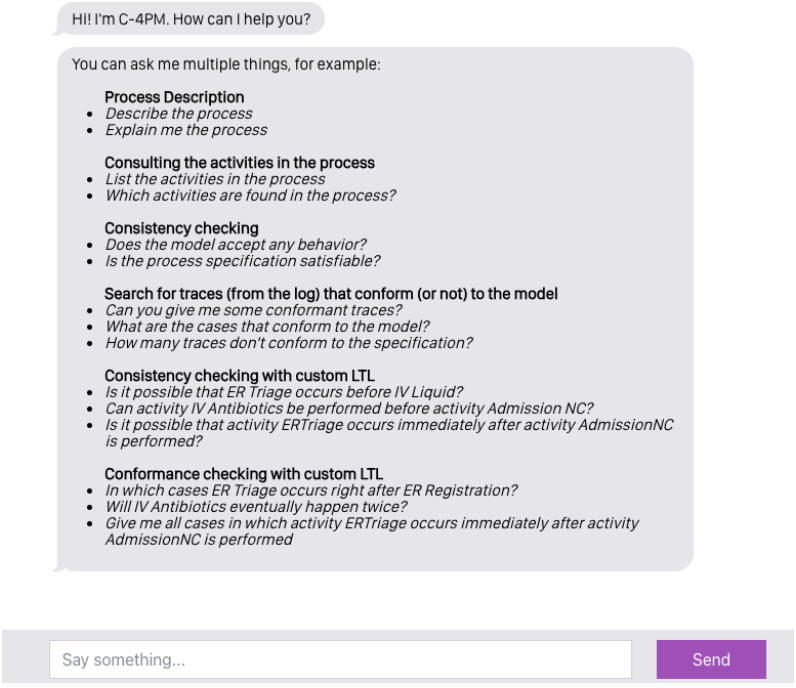


Figure 5.1: Chat-like interface of C-4PM. A text field is used to provide the assistant with a query and a main window shows the conversation (real screenshot from the demo available at tec.citius.usc.es/c-4pm).

user-friendly and accessible to a broader audience, particularly in sectors where process mining can yield critical insights for operational efficiency and effectiveness. A repository with the code, documentation for its deployment and modification, and a demonstration can be accessed at github.com/Yagouus/c-4pm. Furthermore, C-4PM is accessible at tec.citius.usc.es/c-4pm.

Our approach builds on a conversational agent architecture. This architecture allows for a more natural interaction than one-way Natural Language Understanding (NLU) approaches, as a designated module is in charge of handling the conversation depending on the input of the user and the current state of the agent. An initial module in charge of understanding the input from the user is specifically designed for identifying its intention and extracting the relevant entities from the input query through a machine-learning based approach. To align the extracted entities to its corresponding elements in a process, a rule-based technique is used.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

We propose a data-augmentation technique that allows for great flexibility in the recognition and linking of process entities in natural language texts. A few-shot-trained Large Language Model (LLM) is exploited for translating parts of the input into an intermediate abstract logical representation (LTL_f) which can be appropriately used for reasoning. C-4PM's interface, depicted in Figure 5.1, is designed as a chat interface with a text field and a main window where the conversation unfolds. On start-up, C-4PM proactively introduces itself, outlining the various tasks it can perform. This user-friendly approach not only aids in understanding the capabilities of C-4PM but also guides users in interacting with the system, making it accessible even for those with limited technical expertise.

A Natural Language Generation (NLG) component is integrated to be able to communicate with the user in a two-way natural language manner. This component operates through a sophisticated template-based mechanism in which each potential user intent is associated with a predefined template. When generating a response, a template is dynamically populated with information extracted from the reasoning task executed. This template then undergoes a LLM-based linguistic refinement process, where a LLM is specially prompted for the task of refining these semi-structured text responses, enhancing their grammatical accuracy and syntactic coherence. In particular, the GPT-4 [90] model is used for this task. With this hybrid template-based approach the correctness and relevancy of the responses is ensured by-design.

5.2 Understanding natural language inputs in C-4PM

In order to correctly interpret the input of a user and translate it into a machine-readable representation which can be used to perform a series of different tasks a series of inherent challenges need to be addressed. For example, if we take the Sepsis event log [218], answering correctly to the input query *"Does it ever happen that emergency room sorting takes place after intravenous antibiotics have been given to a patient?"*, requires a series of steps to be carried out: *i)* the first step involves the recognition that this input corresponds to the intention of evaluating if the behavior *"emergency room sorting"* happening after *"intravenous antibiotics"* is allowed by the process specification (details on how this is done are given in the following sections) and, *ii)* recognize that expressions *"emergency room sorting"* and *"intravenous antibiotics"* make reference to activities *"ER Triage"* and *"IV Antibiotics"* respectively, and that *"takes place after"* refers to Declare template *"response"*. In this section we illustrate the principal challenges associated with the transformation and understanding of natural language

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

inputs from users into formal representations that can be used for logical inference.

The first step for an effective conversation is to understand the intention of a user and identify the entities that are being mentioned in a query. To tackle this task, C-4PM follows an intent based approach; a method used in conversational agents to understand and categorize the goal of a user in a conversation (cf. [219]). In addition, a complete understanding of a query requires recognizing relevant entities from the input and mapping them to their corresponding elements, in this case, within the process. Figure 5.3 illustrates this process. This is achieved through the following tasks:

- **Intent recognition:** Involves correctly identifying the purpose of a user's input. Misinterpretation of intent can lead to irrelevant or incorrect responses, impacting the effectiveness of the tool. For example, if a user asks, *"Is it possible for a patient to be readmitted immediately after discharge?"*, the system needs to discern whether the user is inquiring about the possibility within the model or seeking specific instances from the log.

- **Entity recognition:** Identifying and correctly interpreting the relevant entities in user queries. These entities could range from specific activities to broader process concepts. For instance, in the same query *"Is it possible for a patient to be readmitted immediately after discharge?"*, the system must recognize *"readmitted"* and *"discharge"* as entities in the process.

- **Entity linking:** Refers to the ability of the system to link identified entities to corresponding elements within the process models or event logs. Incorrect linking can lead to erroneous analysis or feedback. For example, in the Sepsis log [218] linking the terms *"readmitted"* and *"discharge"* to its corresponding activities *"Admission NC"* and *"Release A"* is essential for accurate processing of the query.

The examples shown in Table 5.1 highlight the complexities involved in developing a conversational agent like C-4PM. In the table, these challenges are presented as distinct entities; however, in practice, intent recognition (IR), entity recognition (ER), and entity linking (EL) are deeply intertwined and must be addressed collectively to ensure accurate and effective natural language understanding in conversational agents like C-4PM.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 5.1: Classification of the different challenges C-4PM tackles when interpreting user input queries. ER means "Entity Recognition", IR means "Intent Recognition" and EL means "Entity Linking". The meaning of each of these tasks is explained in the text.

User Query	Challenge	Explanation
Can you show me the model for patient admissions?	**ER, EL**	Different users might refer to the same process aspect (e.g., "patient admissions") using varied terminology.
Tell me about the constraints involving IV Antibiotics.	**ER**	The system must correctly identify 'IV Antibiotics' as an entity. Linking it to the relevant process activity is direct as this is the proper name in the process.
What are the rules for handling emergency cases?	**IR, EL**	Understanding the intent (asking about rules) and linking 'emergency cases' to the correct model elements.
Is it possible for a patient to be readmitted immediately after discharge?	**IR, ER, EL**	The system needs to discern whether the user is inquiring about the possibility within the model or seeking specific instances from the log and correctly identify and link entities to the corresponding process elements.
How often do we run the preliminary diagnosis step?	**IR, ER, EL**	The system needs to classify the intent of the user and identifying 'preliminary diagnosis' as an entity and linking it accurately to the corresponding process step is crucial.

5.3 Reasoning with declarative process specifications

In C-4PM, reasoning and process analysis are supported by the Declare4Py [220] framework. Declare4Py is a Python library based on the Declare language that is designed to analyze processes using declarative, constraint-based process specifications. Backed by Declare4Py, C-4PM has the ability to automatically discover the declarative specification of a process based on event log data with the use of a discovery algorithm (cf. [221]) as well as a broad spectrum of tasks that are divided in two categories: *analysis* and *reasoning*. Analysis tasks range from explaining a process model specification in natural language to retrieving the number of distinct activities in a process. Reasoning tasks include complex reasoning like consistency, conformance, and model checking, which leverage the declarative nature of C-4PM. A detailed view on all the tasks supported by C-4PM is given in Table 5.2.

As introduced before, Declare is a declarative process modeling language based on Linear Temporal Logic on Finite Traces (LTL_f). LTL_f allows the expression of temporal properties over finite traces, effectively enabling the modeling of constraints in processes. Building upon the foundational concepts of Declare, a declarative process specification $\mathcal{DS}$ represents a process using a set of constraints that define allowed and forbidden patterns of behavior within

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

said process expressed in Declare language [215]. This form of process specification is not only more flexible but also more intuitive for representing complex, non-linear processes where multiple paths or sequences of activities may achieve the same outcome, reflecting the inherent flexibility of real-world processes. This nature of declarative specifications is what provides the necessary framework for the analysis and reasoning tasks supported in C-4PM.

Definition 11: *Declarative Process Specification*

Given the universe of activity names U_α, a declarative process specification $\mathcal{DS}$ over U_α is defined as a set of constraints $\{d_1, d_2, ..., d_n\}$, where each constraint d_i is a rule that specifies permissible or prohibited sequences of activities represented by their names in U_α. Formally, this can be represented as:

$$\mathcal{DS} = \{d_1, d_2, ..., d_n\}$$

where:

- U_α represents the universe of all activity names in the process.
- Each d_i is a declarative rule (constraint), expressed in an appropriate formal language, that describes a specific aspect of the process behavior.

Consistency checking is a type of reasoning task which ensures that declarative rules or constraints that define a process do not contradict each other. A declarative process specification $\mathcal{DS}$ is consistent if there is at least one sequence of activities — trace — that satisfies all of the process constraints simultaneously. This implies that the model is logically coherent and able to support real-world processes and behaviors.

Definition 12: *Consistency Checking*

Let $\mathcal{DS}$ be a declarative process model consisting of a set of constraints $\{d_1, d_2, \ldots, d_n\}$. These constraints represent rules that must be adhered to by sequences of events (or traces) in the process.

The consistency of $\mathcal{DS}$ is defined as the existence of at least one finite trace that satisfies all the constraints in $\mathcal{DS}$. Formally, this can be expressed as:

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

$$\text{Consistent}(\mathcal{DS}) \iff \exists t \in U_\alpha^* : \bigwedge_{i=1}^{n} t \models d_i \tag{5.1}$$

Where:

- U_α^* represents the set of all possible finite sequences (traces) of activity names.
- t is a specific trace in U_α^*.
- $t \models d_i$ indicates that trace t satisfies the constraint d_i.
- The model $\mathcal{DS}$ is consistent if there exists at least one trace t such that this condition holds true for all constraints d_i in $\mathcal{DS}$.

Conformance checking is a process used to verify if the observed behavior of a process, typically represented as a sequence of events or traces, adheres to a given process model. In the context of a declarative process specification $\mathcal{DS}$, conformance checking involves comparing each trace in an event log with the constraints defined in $\mathcal{DS}$ to determine if the traces comply with these constraints. It allows to assess the alignment of real-world process executions — as recorder in an event log — agains a process model, enabling the identification of deviations, non-compliances, and areas for process improvement.

Definition 13: *Conformance Checking*

Given a declarative process specification $\mathcal{DS}$ consisting of a set of constraints $\{d_1, d_2, ..., d_n\}$ and an event log L containing a set of traces $\{t_1, t_2, ..., t_m\}$, conformance checking can be formally defined as:

$$\text{Conforms}(t, \mathcal{DS}) \iff \bigwedge_{d_i \in \mathcal{DS}} t \models d_i \tag{5.2}$$

Where:

- t represents a trace from the event log L.
- $t \models d_i$ denotes that trace t satisfies the constraint d_i in $\mathcal{DS}$.
- A trace t is said to conform to $\mathcal{DS}$ if it satisfies all the constraints in $\mathcal{DS}$.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Model checking is other reasoning task that is made possible thanks to the use of declarative process specifications. This task is critical for ensuring that the introduction of new constraints does not make the process model infeasible or contradictory when considered alongside the existing specification. This is useful when asking questions about what behavior is allowed — explicitly or implicitly — by a process model, giving users insights about the behavior supported by the process.

Definition 14: *Model Checking*

Given an existing Declare Specification $\mathcal{DS}$, represented as a set of constraints $\{d_1, d_2, ..., d_n\}$, and a new set of constraints $\mathcal{N} = \{n_1, n_2, ..., n_m\}$ to be checked for compatibility with $\mathcal{DS}$, model checking in this context aims to determine if the combined set of constraints $\mathcal{DS} \cup \mathcal{N}$ remains consistent. Formally, this is defined as:

$$\text{Consistent}(\mathcal{DS} \cup \mathcal{N}) \iff \exists t : \bigwedge_{d_i \in \mathcal{DS}} (t \models d_i) \wedge \bigwedge_{n_j \in \mathcal{N}} (t \models n_j) \tag{5.3}$$

Where:

- t represents a possible trace (sequence of events).
- $t \models d_i$ and $t \models n_j$ denote that trace t satisfies constraint d_i from DS and constraint n_j from the new set $\mathcal{N}$, respectively.
- The combined model $\mathcal{DS} \cup \mathcal{N}$ is consistent if there is at least one trace t that satisfies all constraints from both $\mathcal{DS}$ and $\mathcal{N}$.

Each of these tasks is facilitated by the system's ability to translate user queries into corresponding logical representations, in particular, LTL_f, which then can be used to reason about the model and the traces in an event log. Apart from reasoning, C-4PM gives support to other tasks that rather rely on more classical process mining analysis techniques. These are the tasks grouped under the *analysis* class:

- **Describing a Declare Specification in Natural Language:** This task involves interpreting and translating the formal Declare specification into a more understandable natural language description, making it accessible to users who may not be familiar with formal process modeling languages.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Table 5.2: Classification of the different proposals for the use of NLU and NLG techniques in BPM, process mining, workflow management, event data and related realms.

Task	Type	Answer	Question
Specification description in NL	*Analysis*	Description of the process in NL	*Can you describe the process?* *Explain the process specification*
List activities	*Analysis*	Counts and lists all possible activities in the process	*List the activities in the process* *What activities can be executed in the process?*
Consistency checking	*Reasoning*	Consistency result and explanation	*Does the model accept any behavior?* *Is there any possible trace that conforms to the model?*
Conformance checking	*Reasoning*	Examples of conformant traces	*Can you give me some conformant traces?* *What are the cases that conform to the model?*
Number of conformant traces	*Analysis*	Conformant traces count	*How many conformant traces can be found in the log?* *Give me the number of conformant traces*
Non-conformance checking	*Reasoning*	Number and examples of non-conformant traces	*How many traces don't conform to the specification?* *Are there any non conformant traces?*
Model checking	*Reasoning*	Boolean model checking and examples of that behavior in the log	*Is it possible that ER Triage occurs before IV Liquids?* *If Admission NC has not happened yet, can activity Release A happen?*
Restricted conformance checking	*Reasoning*	Examples of traces in which that behavior happens	*Find traces in which IV Antibiotics occurs right after LacticAcid is performed* *In which cases ER Triage occurs right after ER Registration?*
Query checking (restricted model checking)	*Reasoning*	Examples of elements of the process that fulfill the proposed constraint	*What constraints hold between ER Registration and ER Triage?* *Which activities are allowed after IV Antibiotics?*

- **Counting Distinct Activities in a Process:** C-4PM can analyze a process model or event log to identify and count the unique activities that occur within the process, providing insights into the complexity and diversity of the process steps.

- **Listing Distinct Activities:** Beyond just counting, this task involves listing all the distinct activities identified in the process. This can help users to get a comprehensive view of what the process entails at a granular level.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- **Counting the Number of Conformant Traces:** This involves assessing how many traces (or sequences of activities) within an event log conform to the given Declare specification. It's a measure of how well the actual process executions align with the defined model.

5.4 C-4PM architecture

In Figure 5.2 an overview of our approach is depicted. C-4PM is based on the Rasa Framework [222] and is divided into two main components: the Rasa Stack (or Rasa Open Source) and the Action Server (based on the Rasa SDK). In the Rasa Stack, the NLU module handles user intent classification and entity extraction and linking from the user input; the Rasa Core module is in charge of the dialogue management, i.e.: deciding the next action in the conversation based on the context and user input. Once an action is decided by the dialogue manager, the Action Server runs the appropriate reasoning tasks. All components work together to enable effective communication and interaction in the conversational system.

A NLG component is integrated in the action server to generate natural language response which enable a two-way natural language-based interaction with the user. This component functions using a dynamic fill-in template-based method, where every possible user intent is linked to a specific pre-defined template. During response generation, a template is dynamically filled with data derived from the completed reasoning task. Subsequently, this template is processed through a LLM-based linguistic refinement process. In this phase, a LLM, particularly prompted for refining these semi-structured text responses, improves their grammatical precision and syntactic coherence. Specifically, the GPT-4 [90] model is used. It is important to highlight that, through this hybrid approach, the resulting natural language responses are truthful and consistent by-design. As the knowledge used to fill the pre-defined templates is obtained through a series of reasoning and inference processes based on the underlying data.

This component operates through a sophisticated template-based mechanism in which each potential user intent is associated with a predefined template. When generating a response, a template is dynamically populated with information extracted from the reasoning task executed. This template then undergoes a LLM-based linguistic refinement process, where a LLM is specially prompted for the task of refining these semi-structured text responses, enhancing their grammatical accuracy and syntactic coherence. In particular, the GPT-4 [90] model is used for this task.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

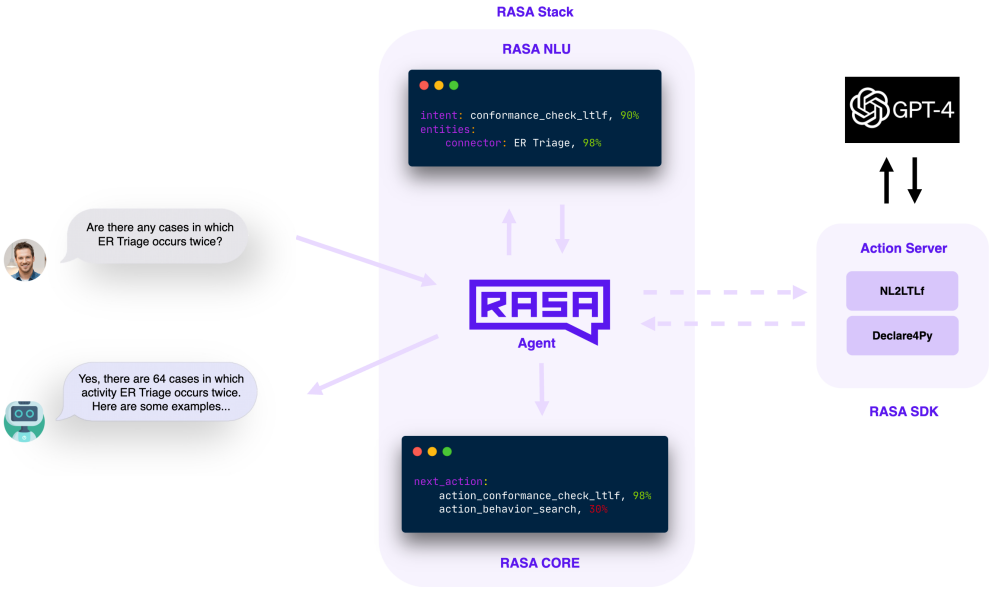


Figure 5.2: Architecture of C-4PM. Shadowed in purple are the two main components: the Rasa Stack, in charge of NLU and dialogue management; and the action server, where the reasoning tasks and response generation are carried out.

The use of the Rasa framework comes motivated by its advanced capabilities for identifying user intents and extracting relevant entities thanks to the transformer-based DIETClassifier [223]. This, coupled with its comprehensive suite of conversation management tools, streamlines the development process. Additionally, Rasa benefits from strong community support and a range of pre-built components, facilitating the creation and implementation of conversational interfaces. In addition, Rasa provides scalability, crucial for managing growth in complexity, making it a reliable choice for developing future-proof systems.

The scope of C-4PM is limited to the tasks identified in the previous sections. An input query should correspond to one of the two types of tasks: *reasoning*, or *analysis*, and one of the nine identified intents *specification description, list activities, consistency checking, conformance checking, non-conformance checking, number of conformant traces, model checking, restricted conformance checking* and *query checking*. In addition, in the entity recognition and linking stages, activity names (e.g., *ER Triage*) and Declare templates (the different types of restrictions that can be stablished in a Declare-based declarative process specification) (such as *response* or in an implicit manner, such as *takes place after*) are recognized.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

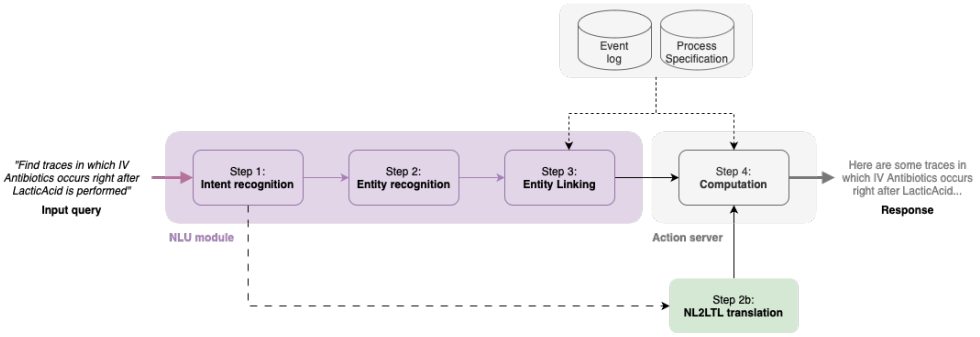


Figure 5.3: Detailed view of the NLU pipeline. In purple, the main steps of the pipeline: intent and entity recognition and entity linking. In some cases, entity recognition and linking are subsumed by the NL2LTL module. In gray, the action server, where the computation and response generation are carried out.

5.4.1 NLU module

The NLU module, depicted in detail in Figure 5.3, comprises several steps to ensure accurate processing of user input. Initially, the module performs *intent recognition* to understand the user's purpose or goal in their query. Following this, *entity recognition* is conducted to identify and extract relevant entities, such as specific process elements or constraints, from the user's input. The subsequent step of *entity linking* involves associating these recognized entities with their corresponding elements in the process models or event logs. This step is vital for ensuring accurate context and reference in the system's responses. Finally, the *computation* step takes place, where C-4PM processes the identified intent and entities to execute the appropriate reasoning tasks, such as consistency or model checking. This computation is performed in the action server, ant its outcome of this computation is then formulated into a coherent response, leveraging both predefined response templates and enrichment from Large Language Models for enhanced naturalness and contextual relevance.

In this first step, C-4PM takes a natural language query from a user as input and identifies the intent and relevant entities necessary to perform the reasoning or analysis task identified. Intent and entity recognition are foundational to C-4PM's ability to understand and process user queries accurately. Despite being depicted as two separate tasks, both tasks are carried at the same time, as in C-4PM the NLU module is based on the DIET classifier [223]. The DIET — which stands for Dual Intent and Entity Transformer — classifier is a machine learning model that combines the effectiveness of transformer-based architectures [224] with a Conditional

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Random Field (CRF) [225] layer, enhancing its capabilities for understanding nuances in natural language.

The DIET classifier in C-4PM plays a crucial role in the NLU module. It is a powerful component of the RASA framework as it is responsible for the simultaneous classification of user intents and extraction of relevant entities from the user input. However, in order to achieve good results in these tasks, it becomes necessary to train the model accordingly. Further details are provided in the following sections. The dual functionality of this classifier is key to understanding and accurately responding to user queries in the context of process mining. The integration of the transformer architecture in DIET allows for the leveraging of contextual information in text, a feature that is particularly advantageous in understanding complex queries in process mining.

Intent Recognition

Intent classification in the DIET Classifier starts with the processing of user input. The input sentence is first tokenized into words or subwords, and a special classification token ($__CLS__$) is appended. Each token is then transformed into feature vectors, combining both sparse features, like one-hot encodings, and dense features derived from pre-trained word embeddings. In the next step, the initial features are fed into the transformer layers of DIET, where they are processed to encode the contextual relationships between words. Although tokens are already initially represented as embeddings, the transformer layer, with its attention mechanism, refines these embeddings by considering each word in the context of the entire sentence. The Transformer architecture, comprehends the entire input sentence, considering each word in the context of the others, a crucial aspect for understanding the general intent behind the user words.

After processing through the transformer, the output embedding corresponding to the $__CLS__$ token is specifically used for intent classification. This token contains aggregated information about the entire input sequence, making it suitable for sentence-level tasks like intent classification. DIET then compares this sentence embedding against known intent embeddings, using a dot-product loss function. This step essentially matches the user input with the most likely intent label, based on the learned embeddings from the training data. The result is the identification of the user's intent, which could range from a simple greeting to a complex query about a process model. Intents can be understood as *classes* in which to categorize the possible intentions of a user based on their inputs to the system. In particular, an

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

intent, or *class* is defined for each of the tasks identified in Table 5.2 as follows:

- **Intent: List activities** - This intent is about listing or inquiring about activities within a process, e.g., *"List the activities"*.
- **Intent: Specification description** - Here, users seek a description of the process in natural language, like *"Can you describe the process?"*.
- **Intent: Consistency checking** - It involves verifying whether there is any possible trace that conforms to the model, ensuring the model's consistency.
- **Intent: Conformance checking** - It involves checking for traces that conform to a given model, such as *"Can you give me all conformant traces?"*.
- **Intent: Model checking** - It involves evaluating whether some specific behavior is allowed by the process specification, e.g., *"Can an emergency room triage occur before the patient gets an IV drip?"*.
- **Intent: Non-conformance checking** - This intent is for identifying traces that do not conform to the model, e.g., *"Are there any non conformant traces?"*.
- **Intent: Number of conformant traces** - This intent counts the number of traces that conform to the model, e.g., *"How many conformant traces can be found in the log?"*.
- **Intent: Restricted conformance checking** - It encompasses searches for specific behaviors or patterns within the process model, like *"Is normal care admission ever executed in two repeated occasions?"*.
- **Intent: Query checking** - It covers queries about possible actions after specific activities or the relationships between different activities.

Some additional minor intents are defined to handle unrecognized or out-of-scope queries, greet the user and address other miscellaneous aspects of a conversation. Together with the use of RASA entity storage, this allows for more fluent conversations in which the user can make reference to entities recognized in previous messages. This gives place to a more natural interaction than the one a bare NLU system would provide.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- **Intent: Greet** - This intent captures greetings from the user. Examples include "Hi", "Good morning", and other variations. It is crucial for initiating a conversation and establishing a user-friendly interaction.

- **Intent: Goodbye** - It involves recognizing parting phrases like "Goodbye" or "See you soon". This intent is vital for gracefully concluding interactions with users.

- **Intent: Challenge** - This intent is used when users inquire about the nature of the agent, with questions like "are you a bot?". It is essential for clarifying the AI-driven nature of C-4PM.

- **Intent: Out-of-scope** - It deals with user inputs that fall outside the domain of process mining, such as "Who's the US President?". Handling such queries is important for maintaining contextually appropriate dialogues.

Entity Recognition

On top of the intents or *classes*, our approach aims at recognizing or extracting activity names such as *"ER Triage"* or *"CRP"* and Declare templates such as *"succesion"* or *"takes place after"*. Entity recognition in the DIET Classifier begins with the same tokenization and feature vectorization phases as intent classification. These feature vectors are then processed by the transformer layers in the same way as before. However, unlike intent classification, entity recognition focuses on the output corresponding to each individual token, rather than the $__CLS__$ token. The transformer's output for each token is enriched with contextual information from the entire sentence, allowing the model to understand each word in relation to its surrounding context.

The subsequent step involves a Conditional Random Field (CRF) layer. The CRF layer uses the transformer's output to predict a sequence of entity labels. This layer is adept at capturing the dependencies between adjacent tokens, making it effective for tasks like entity recognition where context is key. For example, in a user query about a process step, the CRF layer helps identify not just isolated tokens as entities but also understands their relation in the context of the entire query.

The final output is a sequence of entity tags corresponding to each token in the user input. These tags indicate which tokens are entities and what type of entities they are, such as process names, activity labels, or other terminologies relevant to the conversational agent's domain.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

This capability is crucial for C-4PM to extract meaningful information from user inputs, which then informs the system's subsequent actions or responses.

Training and Data Augmentation

Training the DIET classifier involves a comprehensive annotated training dataset comprising various user intents and entities. An initial collection of annotated examples was constructed based on insights from prior research [226] and expert knowledge. This dataset was hand-annotated by the authors, identifying both intents and entities relevant to the problem domain. Given the complexity and diversity of natural language interactions, especially when dealing with a multitude of intents and entity types, a considerable amount of training data is required [41]. To address this challenge, a data augmentation process is carried out. We employed state-of-the-art Large Language Models (LLMs), specifically GPT-4, to build a big enough dataset to achieve the desired level of accuracy and comprehensiveness for the DIET classifier. This approach allowed us to significantly expand the original dataset with new, automatically generated examples that mirrored the examples from the original training dataset. These newly generated examples were meticulously reviewed and annotated, ensuring the reliability and relevance of the data. This process resulted in a robust and extensive dataset, providing the DIET classifier with a diverse range of training examples. Such a dataset is essential to handle the flexibility of user inputs and maintain high accuracy in intent and entity recognition, thereby enhancing the overall performance of C-4PM.

Entity Linking

Once intents and entities are identified, the next critical step is entity linking, where the system associates the extracted entities with their corresponding real-world process elements. To address this challenge, we employ a rule-based entity linking approach, utilizing the look-up and synonym tables provided by the Rasa framework. This method ensures the effective mapping of user-specified terms to corresponding process elements.

Given a tagged sentence as the output of the intent and entity recognition stages, recognized entity will be used to find a value for an element in the process under analysis. Look-up tables in C-4PM serve as predefined lists of entity names. For example, in the Sepsis [218] process, these tables might include terms like *"Emergency Room Triage"*, *"Leucocytes"*, or *"Admission NC"*. These tables act as a primary reference during the entity linking process, enabling the

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

system to scan and identify specific terms or phrases from the user input that match the listed entities. The advantage of using look-up tables lies in their straightforward and efficient nature, providing a reliable mechanism to recognize a wide array of domain-specific terminologies.

Complementing the look-up tables, synonym tables play a crucial role in giving more flexibility to the entity recognition process. This tables collect colloquial or alternative expressions that refer to the same entity, effectively addressing the linguistic variability inherent in natural language. For instance, the term *"Emergency Room Registration"* might have several synonyms like *"Emergency Room Sign-In"* or *"Hospital Emergency Room Admission"*. The generation of synonym tables is aligned with the data augmentation process employed to expand the original training dataset. This approach is crucial for enhancing the system's understanding of natural language variations and ensuring accurate entity recognition across diverse user inputs.

This rule-based approach to entity linking proves to be highly effective in the context of C-4PM. It allows for the consistent and accurate identification of entities, regardless of the varied linguistic representations used by different users. Furthermore, this method contributes significantly to the robustness and flexibility of the conversational agent, enabling it to comprehend and interact effectively within its specific domain.

For intents in which an LTL_f formula is given as an input, namely model and query checking and the restricted form of conformance checking, NL2LTL is in charge of translating the natural language input of the user to an appropriate LTL_f formula compatible with Declare4Py. When this is the case, after the NLU module identifies that the user intent is to perform a reasoning task in which some LTL_f formula has to be extracted as an entity from the input, the NL2LTL module is called. This module can operate with two different engines: a RASA-based implementation and LLM-based one, in particular, using the GPT model. In C-4PM, the GPT engine is used. Due to the extension of the number of LTL_f formulas supported by the NL2LTL framework a re-training of the model in charge of translating the natural language formulas into LTL_f was needed. When using an LLM-based engine, this process is trivial, as through few-shot training, only a small set of tagged examples are needed to obtain good results. Furthermore, after some testing, it was found that with the extended catalogue of supported formulas, adapting the prompt used with the GPT engine yields more accurate results that those obtained by modifying and re-training the original RASA-based engine.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

5.4.2 Reasoning and response generation

In the final stage of the C-4PM framework, the system executes the tasks identified in the earlier stages and generates responses to user queries. During this phase, the Declare4Py framework is used for performing the analysis and reasoning tasks specified in Section 5.3 like conformance checking, consistency checking, and model checking. This reasoning is conducted based on the outcomes of the NLU module: the identified intent defines the task to perform and the identified entities are used for performing the task. The use of Declare4Py is crucial, as by being grounded in Declare enables the system to assess whether the input of the system is consistent with the process specification, whether there are any cases that comply with the behavior specified in the input or if the discovered process specification accepts any behavior at all, among many other tasks.

Once the analysis or reasoning process is completed, the system generates a natural language response conveying the result through a hybrid approach. C-4PM utilizes a template-based approach for the initial response generation. These templates are designed to align with specific intents and provide a structured format for responses. They provide a basic syntactic and semantic structure similar to protoforms for the generation of the natural language responses. These templates serve as a scaffold for the responses, ensuring all responses maintain a coherent and structured format. The content to fill these templates is derived from the results of the reasoning process carried out by Declare4Py.

To further refine the responses, C-4PM integrates the template-based response generation with LLMs. In particular, the GPT engine. This step is critical in refining the responses, adding layers of naturalness, detail, and fluidity that are characteristic of human-like language. LLMs advanced capabilities in language understanding and generation allow it to enrich the template-based responses, making them more engaging, informative, and contextually relevant to the user's query. The combination of fill-in templates with an LLM-based linguistic refinement process results in accurate-by-design responses ensuring that the user receives a coherent and contextually relevant answer.

In summary, C-4PM's final output to users is a blend of structured, accurate reasoning and rich, natural language. This combination showcases C-4PM's advanced ability to deliver clear, user-friendly explanations in process mining, making complex data more accessible and understandable.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

5.4.3 Running example

To better illustrate how C-4PM operates we will use an example based on the *Sepsis* event log [218]. This log contains 1050 traces describing the treatment process of sepsis patients in a Dutch hospital. For instance, users can ask:[1]

- *"Explain the model of the process"*, for the system to explain the discovered Declare specification in natural language to grasp a general idea of it (specification description).
- If the discovered (or given as input) Declare specification allows for any trace at all (consistency checking) as a way of checking if the specification is correct and aligns with the event log e.g.: *"Does the model allow for any behavior?"*.
- Examples of traces that conform (or not) to the specification to get a general idea of the frequent behavior in the process can be retrieved by asking *"Can you give me traces from the log that conform to the model?"*, i.e., for conformance checking.
- To know whether some behavior is supported by the model, e.g., whether it is possible that patients experience first admission to normal care, (activity *Admission NC*), and later on to intensive care, (activity *Admission IC*) one may ask *"Is it possible that activity Admission IC occurs after activity Admission NC has taken place?"*. This constitutes a form of model checking.
- Knowing that cases in which the above behavior happens are considered problematic [227, p.299], a user might ask *"Are there cases where Admission IC happens after Admission NC?"*. Furthermore, as C-4PM holds account of past messages, the user is able to simply ask *"Give me some cases in which this takes place."* and, the system recalls the relevant entities (activity *Admission NC*, activity *Admission IC* and their relationship) and performs restricted conformance checking to retrieve any traces in which this behavior occurs. A similar example is illustrated in Figure 5.4.

In the latter two cases, after identifying the intent and recognizing the, C-4PM translates the user input to LTL_f using the NL2LTL framework. This LTL_f formula is subsequently translated to its corresponding Declare constraint $response(AdmissionNC, AdmissionIC)$. Once this is done, C-4PM checks in the former case whether this formula is consistent with the model, and in the latter uses it for conformance checking.

[1] All tasks with exemplary queries can be found in the project's repository: `https://github.com/Yagous/c-4pm`

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

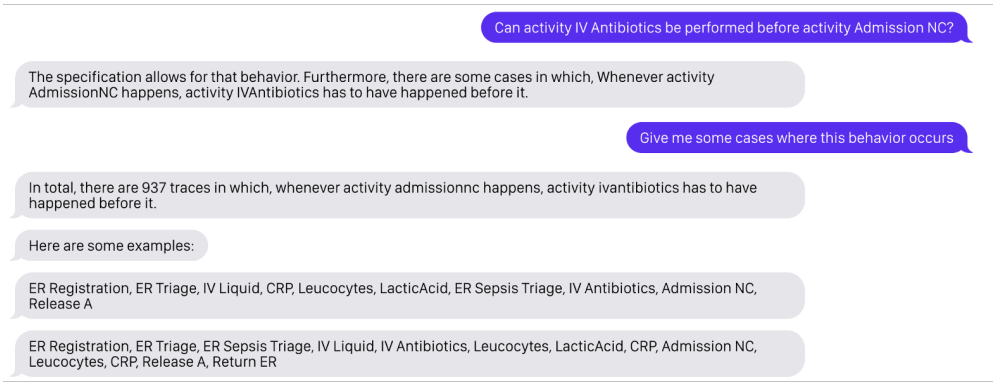


Figure 5.4: Fragment of a conversation in which the model checking and restricted conformance checking tasks are performed. It can be seen how the system is able of recalling entities mentioned in previous messages.

5.5 Discussion and final remarks

In this chapter C-4PM was introduced as the first conversational agent for declarative process mining. It is evident that this innovative approach has the potential to significantly enhance user interaction with declarative process specifications and process mining in general. By fusing conversational interfaces with case and control-flow analysis, C-4PM not only simplifies the interaction with complex process data but also broadens the accessibility of process insights to a more diverse audience. This user-centric interaction paradigm encourages a more natural and engaging user experience, breaking down technical barriers and enabling users of varying expertise levels to effectively query and comprehend intricate process data.

As C-4PM is a proof-of-concept implementation, extensive experiments are left for future work. However, all the described functionality can already be used and tested. A preliminary user validation has been conducted with process mining experts to evaluate its performance. The performed tests have yielded positive results, indicating the potential effectiveness of the tool in addressing the challenges of presenting process models and facilitating the interaction of non-expert users with a declarative process specification; effectively allowing them to perform multiple reasoning tasks in a transparent and understandable manner. While further investigation and evaluation are necessary to fully assess the maturity and reliability of the tool, the initial feedback provides promising indications of its viability and potential. For this reason, immediate research efforts should focus on conducting a more comprehensive evaluation to assess the maturity and reliability of the system in a wider range of contexts. By conducting a

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

comprehensive user validation study involving a diverse range of users and scenarios more valuable insights into the usability and performance of C-4PM, guiding further improvements and refinements, can be obtained.

Future work for C-4PM also involves the development of an advanced bootstrapping pipeline. This pipeline will autonomously process an event log and perform the data augmentation process, ensuring that C-4PM works in an optimal manner independently of the event log and process under analysis. This pipeline would initially digest the entire event log, assimilating all entities in the process, automatically extract the underlying declarative process model and identifying and understanding the underlying patterns and relationships within the process. After all entities have been recognized, based on a series or rules and prompts specifically designed, data-augmentation is carried out and the NLU module gets specifically trained for that particular process.

Finally, future work is also focused on enhancing the user experience by exploring new NLU and NLG techniques, potentially migrating from an intent-based approach towards a more powerful LLM-based NLU system that can provide a deeper, more flexible and context-aware understanding of user queries, for example, by understanding more complex restrictions from the users input. Furthermore, expanding the scope of the reasoning tasks that the system can handle, such as complex decision-making and logical inference, would contribute to its overall effectiveness.

Despite these challenges, C-4PM marks a significant step forward in making (declarative) process mining more accessible and interactive, paving the way for its innovative use across multiple domains. The conversational interface of C-4PM simplifies the interaction with data, making it more approachable for non-specialists while still retaining the depth needed by experts. This broadens the scope of declarative process mining, extending its benefits beyond traditional process mining.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Chapter 6

Conclusions

Trying is the first step towards failure.

— Homer Simpson

This dissertation has focused on identifying and addressing the limitations of the current methods used for analyzing and conveying process related information, specially for non-expert users. Contemporary process mining tools and methodologies, despite their sophistication, present challenges in terms of accessibility and user-friendliness, particularly for those who are most likely to benefit from their insights. Due to the ubiquity of processes across organizations in any domain, the undeniable utility and potential of process mining is often hindered by its perceived complexity, highlighting the pressing need for tools and representations that are more intuitive and approachable. Equipping users with methods that facilitate accurate interpretation and effective interaction with process analyses has become a critical imperative. In this context, Natural Language Generation (NLG) systems are developed to bridge this gap, effectively translating intricate information into a format that is more accessible and user-friendly. These systems have shown their worth across various domains. For this reason, the hypothesis of this thesis was stated as:

> *Natural language descriptions of processes represent a better way of conveying process related information. Allowing non-expert users to interact with process mining techniques and derive insights from a process in an understandable manner and without the need of process mining practitioners.*

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Our comprehensive review of the state of the art — presented in Chapter 2 — reveals a series of shortcomings in the current natural language-based approaches to process mining. Specifically, when focusing on the generation of process descriptions in natural language, research currently navigates between two distinct methods: generating text descriptions of process models without using execution data, and creating linguistic summaries from event logs without incorporating the process models. The first method offers a static, idealized view, omitting valuable insights from real process executions. The second focused on the linguistic summarization of event logs effectively describes execution data but misses out on structural aspects defined by process models. This division leads to analysis that might not capture the full complexity and interplay of real-world processes. Furthermore, despite advancements in natural language generation, determining the key information to convey remains a challenge. In the process mining domain this is aggravated by the variable relevance of process aspects across different domains and user needs. This review underscores the existing challenges in process model understandability and interpretability, particularly in complex environments, and points to the necessity for a unified approach that combines the structured perspective of process models with the real execution insights from event logs for providing a more comprehensive and actionable process understanding on the basis of natural language descriptions. Building on these findings, the necessity for interactive elements in process mining becomes evident. An evolution towards an interactive approach is vital for the practical application and effectiveness of NLG in process mining, ensuring that insights are both accurate and contextually relevant. The shortcomings highlighted in the literature have been addressed as the main contributions of this thesis.

In particular, in Chapter 3, a series of relevant natural language descriptions were proposed encompassing qualitative and quantitative information, including temporal and frequency details, as well as the interrelations between activities and domain-specific process information. These descriptions are proposed in a protoform-based model which, harnessing fuzzy logics, allows not only for the definition of the semantics of these descriptions but also provides methods for the automatic generation of linguistic summaries of a process on the basis of the proposed protoforms. Results of a human expert-based evaluation process show that the proposed descriptions address relevant aspects of a process and provide useful information in a comprehensible way. Descriptions in which routine information was conveyed were perceived as less useful, underscoring the importance of context and relevance in the application of these protoforms in practice. The proposed model incorporates both qualitative and quantitative

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

information derived from process execution data, alongside structural information obtained through process discovery, and offers greater syntactic and semantic richness compared to the existing natural language descriptions of processes in the literature.

- In Section 3.1 and Section 3.1.1, a background on LDD and Fuzzy Logic — its underlying formalism — is provided to contextualize the proposed solution. In this section, it is explained how protoforms serve not only as the underlying structure of relevant natural language descriptions, but how LDD provides methods for the automatic generation of linguistic summaries on the basis of protoforms and some underlying data.
- The state of the art presented in Chapter 2 revealed that most existing approaches propose overly simplified solutions that rely solely on unary and binary quantified sentences, which may not be sufficient in many application domains.
- To overcome this limitation, in Section 3.2 a series of process-focused protoforms, which take into account the temporal dimension and include domain specific information, were proposed. These protoforms allow both to contextualize relevant aspects of a process in time as well as to describe control-flow and performance-related aspects of a process.
- In Section 3.3, an evaluation of the proposed descriptions and protoform-based model was presented. The results of this preliminary evaluation show that the proposed descriptions address relevant aspects of a process and provide useful information in a comprehensible way.

Despite the proposed enhancements in expressiveness — syntactically and semantically through the proposal of new protoforms — the use of linguistic summaries as a direct output of a user-targeted system is still less than ideal. For this reason, in Chapter 4, building on the previously defined formal model, a comprehensive framework for process mining, grounded in the Data-To-Text pipeline, was developed for generating natural language descriptions of processes. A fuzzy logic-based knowledge representation formalism defined on the basis of the previously defined protoforms serves as the background of this framework. This formalism allows for the linguistic representation of process information by grounding the semantics of linguistic terms in fuzzy logics, and for the inference of new knowledge and the generation of natural language descriptions of processes. This framework was then evaluated using a real-world process to confirm the effectiveness and validity of the proposed formal model and the generation framework.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Section 4.1 presents the most extended NLG and Data-to-Text architectures in the literature. These modular architectures define a series of modules and tasks that structure and streamline the process of generating natural language descriptions from non-linguistic input data.
- Our Process-to-Text (P2T) framework is presented in Section 4.2. One of the most relevant capabilities of D2T systems is to accurately interpret the underlying meaning of their input data in order to generate correct output texts. The objective of the initial data analysis stage is to identify and abstract the most relevant features of the input data into an initial collection of messages that formalize the extracted knowledge. For this purpose, a knowledge representation formalism based on Fuzzy Logics is presented in Section 4.1.1. This formalism serves as the cornerstone of the proposed framework through a unifying multi-layered ontology that provides a structured definition of the domain, linguistic variables, values and quantifiers and the proposed protoforms used for the generation of the natural language descriptions of a process. Together with the proposed ontology, a linguistic variables definition process, a fuzzy logic computational system and a fuzzy inference system were introduced for the extraction of the semantics from a process, the *fuzzification* of numerical process information and its linguistic representation and for the generation of linguistic summaries respectively.
- Section 4.3 presents the application of the P2T framework in a real use case in the healthcare domain. Based on the proposed framework, a series of natural language descriptions targeted towards medical experts are generated. On the basis of this use case, a manual human-based validation was conducted, following the best practices for the evaluation of NLG systems, by 16 medical experts in Cardiology. Section 4.4 introduces the validation process. The obtained results are very positive, since from a quantitative point of view a global average of 4.07/5.00 was obtained for the general questions related to the understandability and usefulness of the natural language descriptions as well as the capability of medical experts (those who validated the approach) to complete tasks easier and increase the quality of their work. In particular, results show *i)* that natural language emerged as the most efficient modality for conveying process-related information; *ii)* a marked preference for textual descriptions over conventional graphical representations for communicating process information to experts; and *iii)* natural language descriptions yielded pertinent and valuable insights about the process, offering avenues for its enhancement.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Based on the proposed P2T framework, in Section 4.5, the Natural Language Generation for Process Mining (NLG4PM) system is presented. In this section, the Natural Language Generation for Process Mining (NLG4PM) system is presented. This system implements the P2T framework described in the previous sections of Chapter 5. In particular, the NLG4PM system takes a human-in-the loop approach where, through an straightforward web-based interface, users can perform an exploratory analysis of a process and select those elements of the process in which the generated descriptions should focus. As introduced in [56], introducing user preferences in the content determination stage prevents that an NLG system makes incorrect assumptions about the information needed by the user, as it is able to directly (or indirectly) specify the content of interest that should be conveyed in the natural language descriptions. NLG4PM represents a comprehensive system capable of generating natural language descriptions of a process that address the particular interests of a user.

Furthermore, deeper investigating the interactive generation of natural language descriptions of processes, Chapter 5 introduces C-4PM, which is, to the best of our knowledge, the first conversational agent for declarative process mining. The use of declarative process specifications facilitates various tasks such as consistency, conformance, and model checking through a conversational-driven approach. C-4PM represents an innovative proposal to the challenge of presenting declarative process models to non-expert users in an understandable way. This novel system not only makes declarative process mining accessible to a broader range of users but also enhances the understanding of process models, mitigating misinterpretations and promoting accurate insights, especially in complex environments.

- Section 5.1 gives a first look into the rationale behind C-4PM, the concept of a conversational agent as well as a general characterization of this type of agents where, not just Natural Language Understanding (NLU) techniques are necessary to correctly interpret the query of a user, but reasoning, conversation state management and NLG are needed in order to provide accurate and contextually aware responses to these queries.

- In Section 5.2 the principal challenges when interpreting natural language queries in C-4PM are presented. These are the challenges that need to be addressed for understanding the intention of a user. In particular, the classification of the user input into a series of pre-defined categories (intent recognition), the extraction of the relevant entities that are mentioned in the input text (entity recognition) and, linking them to entities in the process that is being analyzed (entity linking).

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Section 5.3 gives an introduction to declarative process mining, specifically, to the Declare language. This language is used as the underlying formalization over which declarative process specifications (models) are defined. Through the Declare language, a series of reasoning tasks can be performed on the basis of a process specification, such as consistency checking, model checking or conformance checking.
- In Section 5.4 a detailed description of the C-4PM architecture and implementation is provided. From the design of the NLU modules and the processing they do, to how the reasoning and response generation is done based on a hybrid template approach harnessing LLMs for linguistic realization. C-4PM's ability to interact with users via a text interface, understand natural language queries, and provide responses in natural language makes it a significant leap forward in the field, showcasing the potential of integrating conversational agents and process mining for more effective and user-centric process analysis.

As future work, we plan to face the following limitations and areas for potential enhancement we have identified in our proposal.

- *Extension of the proposed model*: As it was pointed in Chapter 1, a plethora of process mining methodologies have been developed to support organizations in their process mining endeavors. However, understanding how process mining is practically implemented remains an open challenge. This comes from the lack of consensus and a unified framework guiding the actual analysis and execution of process mining. It is crucial to investigate how practitioners conduct process mining analysis and understand the organizational context of process analysis. This understanding is essential to fully leverage the potential of process mining. Future work should focus on assessing how process analysis is performed and which are the crucial elements of a process that need to be described. In this context, the construction of an expert-based corpus of question-answer pairs would allow to determine these key aspects and serve as a basis for the evaluation of systems that aim to generate natural language descriptions of processes. Furthermore, on the basis of this corpus, the proposed taxonomy of natural language descriptions of process can be greatly extended, serving as a standard for the future development of solutions similar to the ones proposed in this thesis.
- *Improve the interactivity capabilities of NLG4PM*: After a first proposal of a system for the automatic generation of natural language descriptions of processes, the inclusion

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

of interactivity becomes evident. Future work for NLG4PM should aim to develop a comprehensive NLP system, similar to the one proposed in C-4PM, encompassing both NLU and NLG in a comprehensive system designed to interpret and classify user intents, specifically concerning the type of process description they require. This system should have the capability to extract relevant entities mentioned in the input and map these to the corresponding entities within the process under analysis. Once the intent is classified, the system will generate natural language explanations about the identified process entities. This integrated approach aims to create a user-centric system that not only understands and processes user inputs effectively but also communicates complex process information in an understandable and useful format as NLG4PM is capable of doing at its current state. In this same context, future research for C-4PM will aim to enhance the user experience by investigating new NLU NLG techniques. This includes a potential shift towards a more advanced NLU system based on large-scale pre-trained models, such as LLMs, which can offer deeper and more context-sensitive conversations. Additionally, research efforts should focus on extending the range of reasoning tasks the system can manage, such as intricate decision-making and logical inference processes, to improve its overall effectiveness.

- *Further in-depth evaluation*: In Chapters 3 and 4, following the current standards of system evaluation on the NLG field, human-based evaluations using the Likert scales were proposed. Human-based evaluation, particularly through expert manual assessment, is considered as the most suitable methodology for evaluating cases like these, where automatic metric-based evaluations cannot be performed due to the lack of gold-standard corpora for comparison and comparisons with other systems are not feasible. However, despite its popularity, the Likert scale method has faced criticism for its lack of consistency and reliability [228]. In contrast to intrinsic methods, extrinsic evaluation methods measure the success of the system in achieving specific goals, such as influencing behavior [229], or aiding in decision-making processes [67]. Future work should delve into these evaluation challenges, exploring best practices for NLG system assessment as proposed in the literature [179, 230], and addressing the current limitations in both intrinsic and extrinsic evaluation methodologies. Following the same idea, a deeper evaluation on the use of LLMs for linguistic realization in the framework of a D2T system is needed. Despite a preliminary qualitative evaluation shows that the output of this process is consistent with the abstract representations of messages generated by means of

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

inference, a quantitative evaluation of the possible hallucinations and omissions typical in neural Natural Language Generation (cf [94]) shall be performed. This evaluation would enable us to rigorously measure the reliability and accuracy of the system's outputs, ensuring that the generated texts not only appear coherent but also faithfully represent the underlying data without distortions or significant information gaps. It would also provide valuable insights into the strengths and limitations of LLMs in practical NLG applications, guiding future improvements and optimizations.

In summary, this dissertation marks substantial advancements in applying natural language processing techniques for generating natural language descriptions and facilitating the interaction with processes and process mining techniques. The current stage of this research presents multiple paths for future work to further evolve the proposed solutions and extend the use of natural language in the process mining realm. These paths include the definition of an standard corpus and the exploration of new natural language descriptions of processes, the extension of the proposed NLG4PM system to better handle interaction with users for the generation of contextually-aware descriptions and the exhaustive evaluation of the proposed solutions. By addressing these challenges, an interactive, complete and robust natural language-based solution for process mining can be conceived, helping on making process mining ubiquitous and accessible to users in any domain despite their expertise. The possibilities for further research and improvement appear promising and hold enough potential to represent a long-term significant contribution in the process mining realm.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

APPENDIX A

LIST OF PUBLICATIONS INCLUDED IN THIS PH.D. DISSERTATION

This Ph.D. dissertation reproduces the contents of the following publications:

- Chapter 3 reproduces the contents of Fontenla-Seco, Y., Bugarín-Diz, A., & Lama, M. (2021). **Fuzzy Temporal Protoforms For The Quantitative Description Of Processes In Natural Language**. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–6). Luxembourg, Luxembourg. ISBN: 978-1-6654-4407-1. doi: 10.1109/FUZZ45933.2021.9494444.
  Quality indicators: Conference ranking (GRIN-SCIE): B, Class 3.
  Author's contribution: research of the topic, conceptualization, design and implementation of the algorithms, design and conduction of the experiments, and writing of the article.
  Reproduction authorization: IEEE allows to reuse accepted papers in a Ph.D. dissertation without express permission (see `https://journals.ieeeauthorcenter.ieee.org/choose-a-publishing-agreement/avoid-infringement-upon-ieee-copyright/`).

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Universidade de Santiago's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Chapter 4 reproduces the contents of Fontenla-Seco, Y., Lama, M., & Bugarín-Diz, A. (2020). **Process-To-Text: A Framework for the Quantitative Description of Processes in Natural Language**. In *Trustworthy AI - Integrating Learning, Optimization and Reasoning (TAILOR 2020) at ECAI 2020*. Lecture Notes in Computer Science, vol. 12641 (pp. 212-219). Springer, Cham. ISBN: 978-3-030-73959-1 (online). doi: 10.1007/978-3-030-73959-1_19.

  Quality indicators: Conference ranking (GRIN-SCIE): A, Class 2.

  Author's contribution: research of the topic, conceptualization, design and implementation of the algorithms, design and conduction of the experiments, and writing of the article.

  Reproduction authorization: Authors have the right to reuse their article's Version of Record, in whole or in part, in their own thesis. Additionally, they may reproduce and make available their thesis, including Springer Nature content, as required by their awarding academic institution. Authors must properly cite the published manuscript in their thesis according to current citation standards and include the following acknowledgement: 'Reproduced with permission from Springer Nature' (see `https://www.springer.com/gp/rights-permissions/obtaining-permissions/882`).

- Chapter 3 and Chapter 4 reproduce the contents of Fontenla-Seco, Y., Lama, M., González-Salvado, V., Peña-Gil, C., & Bugarín-Diz, A. (2022). **A Framework For The Automatic Description Of Healthcare Processes In Natural Language: Application in an Aortic Stenosis Integrated Care Process**. *Journal of Biomedical Informatics*, 128, 104033. ISSN: 1532-0464. doi: 10.1016/j.jbi.2022.104033.

  Quality indicators: Journal Impact Factor: 4.500 (JCR 2022), 1.083 (SJR 2022). Journal ranked in JCR 2022, in Computer Science, Interdisciplinary applications (Q2, 39/110); in Medical Informatics (Q2, 12/31).

  Author's contribution: research of the topic, conceptualization, design and implementation of the algorithms, design and conduction of the experiments, and writing of the article.

  Reproduction authorization: This article has been published as open access, therefore, Elsevier allows its inclusion as part of doctoral theses without express permission (see `www.elsevier.com/about/policies/copyright#Author-rights`).

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

- Chapter 5 reproduces the contents of Fontenla-Seco, Y., Winkler, S., Gianola, A., Montali, M., Lama, M., & Bugarín-Diz, A. (2023). **The Droid You're Looking For: C-4PM, a Conversational Agent for Declarative Process Mining**. In *Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2023 co-located with 21st International Conference on Business Process Management (BPM)* (pp. 112-116). Utrecht, The Netherlands. Available at ceur-ws.org/Vol-3469/paper-20.pdf.

  Quality indicators: Conference ranking (GRIN-SCIE): A, Class 2.

  Author's contribution: research of the topic, conceptualization, design and implementation of the algorithms, design and conduction of the experiments, and writing of the article.

  Reproduction authorization: PhD students, who published their papers in a volume at CEUR-WS, retain the copyright to their papers. Kindly include a reference to the original publication at CEUR-WS. There is no need to ask us for permission. (see `https://ceur-ws.org/HOWTOSUBMIT.html#FAQ`).

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Bibliography

[1] Borja Vázquez-Barreiros, David Chapela, Manuel Mucientes, Manuel Lama, and Diego Berea. Process mining in IT service management: A case study. In Wil M. P. van der Aalst, Robin Bergenthum, and Josep Carmona, editors, *Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data 2016 Satellite event of the conferences: 37th International Conference on Application and Theory of Petri Nets and Concurrency Petri Nets 2016 and 16th International Conference on Application of Concurrency to System Design ACSD 2016, Torun, Poland, June 20-21, 2016*, volume 1592 of *CEUR Workshop Proceedings*, pages 16–30. CEUR-WS.org, 2016. URL `https://ceur-ws.org/Vol-1592/paper02.pdf`.

[2] Pavlina Kröckel and Freimut Bodendorf. Process mining of football event data: A novel approach for tactical insights into the game. *Frontiers Artif. Intell.*, 3:47, 2020. doi: 10.3389/frai.2020.00047. URL `https://doi.org/10.3389/frai.2020.00047`.

[3] R. S. Mans, M. H. Schonenberg, M. Song, W. M. P. van der Aalst, and P. J. M. Bakker. Application of process mining in healthcare – a case study in a dutch hospital. In Ana Fred, Joaquim Filipe, and Hugo Gamboa, editors, *Biomedical Engineering Systems and Technologies*, pages 425–438, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-540-92219-3.

[4] Álvaro Rebuge and Diogo R. Ferreira. Business process analysis in healthcare environments: A methodology based on process mining. *Information Systems*, 37(2):99–116, 2012. ISSN 0306-4379. doi: https://doi.org/10.1016/j.is.2011.01.003. Management and Engineering of Process-Aware Information Systems.

[5] Eric Rojas, Jorge Munoz-Gama, Marcos Sepúlveda, and Daniel Capurro. Process mining in healthcare: A literature review. *J. Biomed. Informatics*, 61:224–236, 2016. doi:

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

10.1016/j.jbi.2016.04.007. URL https://doi.org/10.1016/j.jbi.2016.04.007.

[6] Ronny Mans, Wil M. P. van der Aalst, and Rob J. B. Vanwersch. *Process Mining in Healthcare - Evaluating and Exploiting Operational Healthcare Processes*. Springer Briefs in Business Process Management. Springer, 2015. ISBN 978-3-319-16070-2. doi: 10.1007/978-3-319-16071-9. URL https://doi.org/10.1007/978-3-319-16071-9.

[7] Edgar Batista and Agusti Solanas. Process mining in healthcare: A systematic review. In Nikolaos G. Bourbakis, George A. Tsihrintzis, and Maria Virvou, editors, *9th International Conference on Information, Intelligence, Systems and Applications, IISA 2018, Zakynthos, Greece, July 23-25, 2018*, pages 1–6. IEEE Computer Society, 2018. doi: 10.1109/IISA.2018.8633608. URL https://doi.org/10.1109/IISA.2018.8633608.

[8] Artini M. Lemos, Caio C. Sabino, Ricardo Massa Ferreira Lima, and César A. L. de Oliveira. Using process mining in software development process management: A case study. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Anchorage, Alaska, USA, October 9-12, 2011*, pages 1181–1186. IEEE, 2011. doi: 10.1109/ICSMC.2011.6083858. URL https://doi.org/10.1109/ICSMC.2011.6083858.

[9] Vladimir A. Rubin, Alexey A. Mitsyuk, Irina A. Lomazova, and Wil M. P. van der Aalst. Process mining can be applied to software too! In Maurizio Morisio, Tore Dybå, and Marco Torchiano, editors, *2014 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14, Torino, Italy, September 18-19, 2014*, pages 57:1–57:8. ACM, 2014. doi: 10.1145/2652524.2652583. URL https://doi.org/10.1145/2652524.2652583.

[10] Ehud Reiter. An Architecture for Data-to-Text Systems. In *Proc. 11th European Workshop on Natural Language Generation*, ENLG '07, pages 97–104, USA, 2007. ACL.

[11] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. *Fundamentals of Business Process Management, Second Edition*. Springer, 2018. doi: 10.1007/978-3-662-56509-4.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[12] Wil M. P. van der Aalst. *Process Mining - Data Science in Action, Second Edition*. Springer, 2016. doi: 10.1007/978-3-662-49851-4.

[13] Wil M. P. van der Aalst. Business process simulation revisited. In Joseph Barjis, editor, *Enterprise and Organizational Modeling and Simulation - 6th International Workshop, EOMAS 2010, held at CAiSE 2010, Hammamet, Tunisia, June 7-8, 2010. Selected Papers*, volume 63 of *Lecture Notes in Business Information Processing*, pages 1–14. Springer, 2010. doi: 10.1007/978-3-642-15723-3_1. URL https://doi.org/10.1007/978-3-642-15723-3_1.

[14] Wil M. P. van der Aalst, Ton Weijters, and Laura Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16:1128–1142, 2004. doi: 10.1109/TKDE.2004.47.

[15] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Discovering block-structured process models from event logs - A constructive approach. In José Manuel Colom and Jörg Desel, editors, *Application and Theory of Petri Nets and Concurrency - 34th International Conference, PETRI NETS 2013, Milan, Italy, June 24-28, 2013. Proceedings*, volume 7927 of *Lecture Notes in Computer Science*, pages 311–329. Springer, 2013. doi: 10.1007/978-3-642-38697-8_17. URL https://doi.org/10.1007/978-3-642-38697-8_17.

[16] A. J. M. M. Weijters and Wil M. P. van der Aalst. Rediscovering workflow models from event-based data using little thumb. *Integr. Comput. Aided Eng.*, 10(2):151–162, 2003. doi: 10.3233/ica-2003-10205. URL https://doi.org/10.3233/ica-2003-10205.

[17] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, and Artem Polyvyanyy. Split miner: automated discovery of accurate and simple business process models from event logs. *Knowl. Inf. Syst.*, 59(2):251–284, 2019. doi: 10.1007/s10115-018-1214-x. URL https://doi.org/10.1007/s10115-018-1214-x.

[18] Borja Vázquez-Barreiros, Manuel Mucientes, and Manuel Lama. Prodigen: Mining complete, precise and minimal structure process models with a genetic algorithm. *Information Sciences*, 294:315–333, 2015.

[19] Object Management Group. *Business Process Model and Notation (BPMN)*, 2010.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[20] James L. Peterson. Petri nets. *ACM Comput. Surv.*, 9(3):223–252, 1977. doi: 10.1145/356698.356702. URL https://doi.org/10.1145/356698.356702.

[21] Wil M. P. van der Aalst. The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems and Computers*, 8:21–66, 1998. doi: 10.1142/S0218126698000043.

[22] Josep Carmona, Boudewijn F. van Dongen, Andreas Solti, and Matthias Weidlich. *Conformance Checking - Relating Processes and Models*. Springer, 2018. ISBN 978-3-319-99413-0. doi: 10.1007/978-3-319-99414-7. URL https://doi.org/10.1007/978-3-319-99414-7.

[23] Dirk Fahland and Wil M. P. van der Aalst. Model repair - aligning process models to reality. *Inf. Syst.*, 47:220–243, 2015. doi: 10.1016/j.is.2013.12.007. URL https://doi.org/10.1016/j.is.2013.12.007.

[24] Wil M. P. van der Aalst, M. H. Schonenberg, and Minseok Song. Time Prediction Based on Process Mining. *Information Systems*, 36:450–475, 2011. doi: 10.1016/j.is.2010.09.001.

[25] Juan Antonio Caballero-Hernández, Juan Manuel Dodero, Iván Ruiz-Rube, Manuel Palomo-Duarte, José Fidel Argudo, and Juan José Domínguez-Jiménez. Discovering Bottlenecks in a Computer Science Degree through Process Mining Techniques. In *Proceedings of the 2018 International Symposium on Computers in Education (SIIE 2018)*, pages 1–6, 2018. doi: 10.1109/SIIE.2018.8586680.

[26] Fábio de Lima Bezerra and Jacques Wainer. Algorithms for anomaly detection of traces in logs of process aware information systems. *Inf. Syst.*, 38(1):33–44, 2013. doi: 10.1016/j.is.2012.04.004. URL https://doi.org/10.1016/j.is.2012.04.004.

[27] Thomas Grisold, Jan Mendling, Markus Otto, and Jan vom Brocke. Adoption, use and management of process mining in practice. *Bus. Process. Manag. J.*, 27(2):369–387, 2021. doi: 10.1108/BPMJ-03-2020-0112. URL https://doi.org/10.1108/BPMJ-03-2020-0112.

[28] Francesca Zerbato, Pnina Soffer, and Barbara Weber. Process mining practices: Evidence from interviews. In Claudio Di Ciccio, Remco M. Dijkman, Adela del-Río-Ortega, and Stefanie Rinderle-Ma, editors, *Business Process Management - 20th International Conference, BPM 2022, Münster, Germany, September 11-16, 2022, Proceedings*,

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

volume 13420 of *Lecture Notes in Computer Science*, pages 268–285. Springer, 2022. doi: 10.1007/978-3-031-16103-2_19. URL https://doi.org/10.1007/978-3-031-16103-2_19.

[29] Jan vom Brocke, Mieke Jans, Jan Mendling, and Hajo A. Reijers. A five-level framework for research on process mining. *Bus. Inf. Syst. Eng.*, 63(5):483–490, 2021. doi: 10.1007/s12599-021-00718-8. URL https://doi.org/10.1007/s12599-021-00718-8.

[30] Melike Bozkaya, Joost Gabriels, and Jan Martijn E. M. van der Werf. Process diagnostics: A method based on process mining. In Andrew Kusiak and Sang-goo Lee, editors, *International Conference on Information, Process, and Knowledge Management, eKNOW 2009, Cancun, Mexico, February 1-7, 2009*, pages 22–27. IEEE Computer Society, 2009. doi: 10.1109/eKNOW.2009.29. URL https://doi.org/10.1109/eKNOW.2009.29.

[31] Maikel L. van Eck, Xixi Lu, Sander J. J. Leemans, and Wil M. P. van der Aalst. PM ^2 : A process mining project methodology. In Jelena Zdravkovic, Marite Kirikova, and Paul Johannesson, editors, *Advanced Information Systems Engineering - 27th International Conference, CAiSE 2015, Stockholm, Sweden, June 8-12, 2015, Proceedings*, volume 9097 of *Lecture Notes in Computer Science*, pages 297–313. Springer, 2015. doi: 10.1007/978-3-319-19069-3_19. URL https://doi.org/10.1007/978-3-319-19069-3_19.

[32] Carlos Capitán-Agudo, María Salas-Urbano, Cristina Cabanillas, and Manuel Resinas. Analyzing how process mining reports answer time performance questions. In Claudio Di Ciccio, Remco M. Dijkman, Adela del-Río-Ortega, and Stefanie Rinderle-Ma, editors, *Business Process Management - 20th International Conference, BPM 2022, Münster, Germany, September 11-16, 2022, Proceedings*, volume 13420 of *Lecture Notes in Computer Science*, pages 234–250. Springer, 2022. doi: 10.1007/978-3-031-16103-2_17. URL https://doi.org/10.1007/978-3-031-16103-2_17.

[33] Gregor Kipping, Djordje Djurica, Sandro Franzoi, Thomas Grisold, Laura Marcus, Sebastian Johannes Schmid, Jan vom Brocke, Jan Mendling, and Maximilian Röglinger. How to leverage process mining in organizations - towards process mining capabilities. In Claudio Di Ciccio, Remco M. Dijkman, Adela del-Río-Ortega, and Stefanie Rinderle-Ma,

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

editors, *Business Process Management - 20th International Conference, BPM 2022, Münster, Germany, September 11-16, 2022, Proceedings*, volume 13420 of *Lecture Notes in Computer Science*, pages 40–46. Springer, 2022. doi: 10.1007/978-3-031-16103-2_5. URL https://doi.org/10.1007/978-3-031-16103-2_5.

[34] Francesca Zerbato, Pnina Soffer, and Barbara Weber. Initial insights into exploratory process mining practices. In Artem Polyvyanyy, Moe Thandar Wynn, Amy Van Looy, and Manfred Reichert, editors, *Business Process Management Forum - BPM Forum 2021, Rome, Italy, September 06-10, 2021, Proceedings*, volume 427 of *Lecture Notes in Business Information Processing*, pages 145–161. Springer, 2021. doi: 10.1007/978-3-030-85440-9_9. URL https://doi.org/10.1007/978-3-030-85440-9_9.

[35] Niels Martin, Dominik Andreas Fischer, Georgi Dimov Kerpedzhiev, Kanika Goel, Sander J. J. Leemans, Maximilian Röglinger, Wil M. P. van der Aalst, Marlon Dumas, Marcello La Rosa, and Moe Thandar Wynn. Opportunities and challenges for process mining in organizations: Results of a delphi study. *Bus. Inf. Syst. Eng.*, 63(5):511–527, 2021. doi: 10.1007/s12599-021-00720-0. URL https://doi.org/10.1007/s12599-021-00720-0.

[36] Thomas Grisold, Jan vom Brocke, Steven Groß, Jan Mendling, Maximilian Röglinger, and Katharina Stelzl. Digital innovation and business process management: Opportunities and challenges as perceived by practitioners. *Commun. Assoc. Inf. Syst.*, 49:27, 2021. doi: 10.17705/1cais.04927. URL https://doi.org/10.17705/1cais.04927.

[37] Boudewijn F. van Dongen, Ana Karla A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and Wil M. P. van der Aalst. The prom framework: A new era in process mining tool support. In Gianfranco Ciardo and Philippe Darondeau, editors, *Applications and Theory of Petri Nets 2005, 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005, Proceedings*, volume 3536 of *Lecture Notes in Computer Science*, pages 444–454. Springer, 2005. doi: 10.1007/11494744_25. URL https://doi.org/10.1007/11494744_25.

[38] Wil M. P. van der Aalst. A practitioner's guide to process mining: Limitations of the directly-follows graph. In Maria Manuela Cruz-Cunha, Ricardo Martinho, Rui Rijo, Emanuel Peres, and Dulce Domingos, editors, *CENTERIS 2019 - International*

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Conference on ENTERprise Information Systems / ProjMAN 2019 - International Conference on Project MANagement / HCist 2019 - International Conference on Health and Social Care Information Systems and Technologies 2019, Sousse, Tunisia, volume 164 of *Procedia Computer Science*, pages 321–328. Elsevier, 2019. doi: 10.1016/j.procs.2019.12.189. URL https://doi.org/10.1016/j.procs.2019.12.189.

[39] Christopher Klinkmüller, Richard Müller, and Ingo Weber. Mining process mining practices: An exploratory characterization of information needs in process analytics. In Thomas T. Hildebrandt, Boudewijn F. van Dongen, Maximilian Röglinger, and Jan Mendling, editors, *Business Process Management - 17th International Conference, BPM 2019, Vienna, Austria, September 1-6, 2019, Proceedings*, volume 11675 of *Lecture Notes in Computer Science*, pages 322–337. Springer, 2019. doi: 10.1007/978-3-030-26619-6_21. URL https://doi.org/10.1007/978-3-030-26619-6_21.

[40] M. Song and W.M.P. van der Aalst. Supporting process mining by showing events at a glance. In K. Chari and A. Kumar, editors, *Proceedings of 17th Annual Workshop on Information Technologies and Systems (WITS 2007)*, pages 139–145, Montreal, Canada, 2007.

[41] Manuel Resinas, Adela del-Río-Ortega, and Han van der Aa. From text to performance measurement: Automatically computing process performance using textual descriptions and event logs. In Chiara Di Francescomarino, Andrea Burattin, Christian Janiesch, and Shazia Sadiq, editors, *Business Process Management - 21st International Conference, BPM 2023, Utrecht, The Netherlands, September 11-15, 2023, Proceedings*, volume 14159 of *Lecture Notes in Computer Science*, pages 266–283. Springer, 2023. doi: 10.1007/978-3-031-41620-0_16. URL https://doi.org/10.1007/978-3-031-41620-0_16.

[42] Felix Mannhardt. Responsible process mining. In Wil M. P. van der Aalst and Josep Carmona, editors, *Process Mining Handbook*, volume 448 of *Lecture Notes in Business Information Processing*, pages 373–401. Springer, 2022. doi: 10.1007/978-3-031-08848-3_12. URL https://doi.org/10.1007/978-3-031-08848-3_12.

[43] Iris Beerepoot, Claudio Di Ciccio, Hajo A. Reijers, Stefanie Rinderle-Ma, Wasana Bandara, Andrea Burattin, Diego Calvanese, Tianwa Chen, Izack Cohen, Benoît Depaire, Gemma Di Federico, Marlon Dumas, Christopher G. J. van Dun, Tobias Fehrer,

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Dominik Andreas Fischer, Avigdor Gal, Marta Indulska, Vatche Isahagian, Christopher Klinkmüller, Wolfgang Kratsch, Henrik Leopold, Amy Van Looy, Hugo A. López, Sanja Lukumbuzya, Jan Mendling, Lara Meyers, Linda Moder, Marco Montali, Vinod Muthusamy, Manfred Reichert, Yara Rizk, Michael Rosemann, Maximilian Röglinger, Shazia Sadiq, Ronny Seiger, Tijs Slaats, Mantas Simkus, Ida Asadi Someh, Barbara Weber, Ingo Weber, Mathias Weske, and Francesca Zerbato. The biggest business process management problems to solve before we die. *Comput. Ind.*, 146:103837, 2023. doi: 10.1016/j.compind.2022.103837. URL https://doi.org/10.1016/j.compind.2022.103837.

[44] Han van der Aa, Henrik Leopold, Adela del-Río-Ortega, Manuel Resinas, and Hajo A. Reijers. Transforming unstructured natural language descriptions into measurable process performance indicators using hidden markov models. *Inf. Syst.*, 71:27–39, 2017. doi: 10.1016/j.is.2017.06.005. URL https://doi.org/10.1016/j.is.2017.06.005.

[45] Luciana Barbieri, Edmundo Roberto Mauro Madeira, Kleber Stroeh, and Wil M. P. van der Aalst. Towards a natural language conversational interface for process mining. In Jorge Munoz-Gama and Xixi Lu, editors, *Process Mining Workshops - ICPM 2021 International Workshops, Eindhoven, The Netherlands, October 31 - November 4, 2021, Revised Selected Papers*, volume 433 of *Lecture Notes in Business Information Processing*, pages 268–280. Springer, 2021. doi: 10.1007/978-3-030-98581-3_20. URL https://doi.org/10.1007/978-3-030-98581-3_20.

[46] Marian Petre. Why looking isn't always seeing: Readership skills and graphical programming. *Commun. ACM*, 38(6):33–44, 1995. doi: 10.1145/203241.203251. URL https://doi.org/10.1145/203241.203251.

[47] P. Homayounfar. Process mining challenges in hospital information systems. *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1135–1140, 2012.

[48] Wil M. P. van der Aalst. *Process Mining: Data Science in Action*. Springer, 2016. ISBN 978-3-662-49850-7.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[49] Violeta González, Carlos Peña, Carmen Neiro, and Diego López. Proceso asistencial integrado de estenosis aórtica. Technical report, Servivio de Cardiología del Complejo Hospitalario Clínico Universitario de Santiago, 2021.

[50] Suriadi Suriadi, Moe Thandar Wynn, Chun Ouyang, Arthur H. M. ter Hofstede, and Nienke J. van Dijk. Understanding process behaviours in a large insurance company in australia: A case study. In Camille Salinesi, Moira C. Norrie, and Oscar Pastor, editors, *Advanced Information Systems Engineering - 25th International Conference, CAiSE 2013, Valencia, Spain, June 17-21, 2013. Proceedings*, volume 7908 of *Lecture Notes in Computer Science*, pages 449–464. Springer, 2013. doi: 10.1007/978-3-642-38709-8_29. URL https://doi.org/10.1007/978-3-642-38709-8_29.

[51] Diptikalyan Saha, Avrilia Floratou, Karthik Sankaranarayanan, Umar Farooq Minhas, Ashish R. Mittal, and Fatma Özcan. Athena: An ontology-driven system for natural language querying over relational data stores. *Proc. VLDB Endow.*, 9(12):1209–1220, aug 2016.

[52] A. S. Law, Y. Freer, Jim Hunter, R. Logie, N. McIntosh, and J. Quinn. A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit. *Jour. Clinical Monitoring Computing*, 19: 183–194, 2005.

[53] Marian Petre. Why looking isn't always seeing: Readership skills and graphical programming. *Commun. ACM*, 38:33–44, 06 1995.

[54] Patricia A. Carpenter and Priti Shah. A model of the perceptual and conceptual processes in graph comprehension. *Journal of Experimental Psychology: Applied*, 4:75–100, 1998. URL https://api.semanticscholar.org/CorpusID:38142300.

[55] Ehud Reiter, Somayajulu Sripada, and Roma Robertson. Acquiring correct knowledge for natural language generation. *J. Artif. Intell. Res.*, 18:491–516, 2003. doi: 10.1613/JAIR.1176. URL https://doi.org/10.1613/jair.1176.

[56] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. Cambridge University Press, USA, 2000. ISBN 0521620368.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[57] S. Dwivedi, Z. Kulpa, and M. Sobolewski. Graphical and natural language interface with a knowledge-based concurrent engineering environment. *Computers in Industry*, 23:175–184, 1993. doi: 10.1016/0166-3615(93)90004-K.

[58] K. McKeown. Language generation: Applications, issues, and approaches. *Proceedings of the IEEE*, 74:961–968, 1986. doi: 10.1109/PROC.1986.13575.

[59] Zhao Jiaojia. Visualization method of association rules based on natural language generation. *Computer Engineering and Applications*, 2014.

[60] Nicolás Marín and Daniel Sánchez. On generating linguistic descriptions of time series. *Fuzzy Sets Syst.*, 285:6–30, 2016.

[61] Andrea Cascallar-Fuentes, Javier Gallego-Fernández, Alejandro Ramos-Soto, Anthony Saunders-Estévez, and Alberto Bugarín-Diz. Automatic generation of textual descriptions in data-to-text systems using a fuzzy temporal ontology: Application in air quality index data series. *Applied Soft Computing*, 119:108612, 2022. ISSN 1568-4946.

[62] François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816, 2009. ISSN 00043702.

[63] Ehud Reiter and Robert Dale. Building applied natural language generation systems. *Nat. Lang. Eng.*, 3(1):57–87, 1997. doi: 10.1017/S1351324997001502. URL https://doi.org/10.1017/S1351324997001502.

[64] Ehud Reiter. Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation, INLG 1994, Kennebunkport, Maine, USA, June 21-24, 1994*, 1994. URL https://aclanthology.org/W94-0319/.

[65] Yago Fontenla-Seco, Sarah Winkler, Alessandro Gianola, Marco Montali, Manuel Lama Penín, and Alberto José Bugarín Diz. The droid you're looking for: C-4pm, a conversational agent for declarative process mining. In Dirk Fahland, Andrés Jiménez-Ramírez, Akhil Kumar, Jan Mendling, Brian T. Pentland, Stefanie Rinderle-Ma, Tijs Slaats, Johan Versendaal, Barbara Weber, Mathias Weske, and Karolin Winter, editors, *Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration &*

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Resources Forum at BPM 2023 co-located with 21st International Conference on Business Process Management (BPM 2023), Utrecht, The Netherlands, September 11th to 15th, 2023, volume 3469 of *CEUR Workshop Proceedings*, pages 112–116. CEUR-WS.org, 2023. URL https://ceur-ws.org/Vol-3469/paper-20.pdf.

[66] Donald Gillies. *Artificial Intelligence and Scientific Method*. Oxford University Press, Oxford and New York, 1996.

[67] François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816, 2009. ISSN 00043702.

[68] Rui Jorge Almeida, Marie-Jeanne Lesot, Bernadette Bouchon-Meunier, Uzay Kaymak, and Gilles Moyse. Linguistic summaries of categorical time series for septic shock patient data. In *Proc. 2013 IEEE Int. Conf. Fuzzy Systems*, pages 1–8, 2013. doi: 10.1109/FUZZ-IEEE.2013.6622581.

[69] David L. Chen and Raymond J. Mooney. Learning to sportscast: a test of grounded language acquisition. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 128–135. ACM, 2008. doi: 10.1145/1390156.1390173. URL https://doi.org/10.1145/1390156.1390173.

[70] Vassilis Plachouras, Charese Smiley, Hiroko Bretz, Ola Taylor, Jochen L. Leidner, Dezhao Song, and Frank Schilder. Interacting with financial data using natural language. In Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel, editors, *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 1121–1124. ACM, 2016. doi: 10.1145/2911451.2911457. URL https://doi.org/10.1145/2911451.2911457.

[71] A. Ramos-Soto, A. J. Bugarín, S. Barro, and J. Taboada. Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Transactions on Fuzzy Systems*, 23(1):44–57, 2015. doi: 10.1109/TFUZZ.2014.2328011.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[72] Eli Goldberg, Norbert Driedger, and Richard I. Kittredge. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53, 1994. doi: 10.1109/64.294135. URL https://doi.org/10.1109/64.294135.

[73] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008. ISBN 978-0-521-86571-5. doi: 10.1017/CBO9780511809071. URL https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf.

[74] Philipp Cimiano, Christina Unger, and John McCrae. *Ontology-Based Interpretation for Question Answering*, pages 131–140. Springer International Publishing, Cham, 2014. ISBN 978-3-031-02154-1. doi: 10.1007/978-3-031-02154-1_9. URL https://doi.org/10.1007/978-3-031-02154-1_9.

[75] Mika V. Mäntylä, Daniel Graziotin, and Miikka Kuutila. The evolution of sentiment analysis - A review of research topics, venues, and top cited papers. *Comput. Sci. Rev.*, 27:16–32, 2018. doi: 10.1016/J.COSREV.2017.10.002. URL https://doi.org/10.1016/j.cosrev.2017.10.002.

[76] Marcos Fernández-Pichel, David E. Losada, and Juan C. Pichel. A multistage retrieval system for health-related misinformation detection. *Engineering Applications of Artificial Intelligence*, 115:105211, 2022. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2022.105211. URL https://www.sciencedirect.com/science/article/pii/S0952197622002950.

[77] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Comput. Linguistics*, 29(1):19–51, 2003. doi: 10.1162/089120103321337421. URL https://doi.org/10.1162/089120103321337421.

[78] Tina Klüwer. From chatbots to dialog systems. In *Conversational agents and natural language interaction: Techniques and Effective Practices*, pages 1–22. IGI Global, 2011.

[79] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, jan 1966. ISSN 0001-0782. doi: 10.1145/365153.365168. URL https://doi.org/10.1145/365153.365168.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[80] Karen Kukich. Design of a knowledge-based report generator. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, ACL '83, page 145–150, USA, 1983. Association for Computational Linguistics. doi: 10.3115/981311.981340. URL https://doi.org/10.3115/981311.981340.

[81] Kathleen R. McKeown, Karen Kukich, and James Shaw. Practical issues in automatic documentation generation. In *4th Applied Natural Language Processing Conference, ANLP 1994, Stuttgart, Germany, October 13-15, 1994*, pages 7–14. ACL, 1994. doi: 10.3115/974358.974361. URL https://aclanthology.org/A94-1002/.

[82] Michael White and Ted Caldwell. EXEMPLARS: A practical, extensible framework for dynamic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation, INLG 1998, Niagara-on-the-Lake, Ontario, Canada, August 5-7, 1998*, 1998. URL https://aclanthology.org/W98-1428/.

[83] Lidija Iordanskaja, Richard Kittredge, and Alain Polguère. *Lexical Selection and Paraphrase in a Meaning-Text Generation Model*, pages 293–312. Springer US, Boston, MA, 1991. ISBN 978-1-4757-5945-7. doi: 10.1007/978-1-4757-5945-7_11. URL https://doi.org/10.1007/978-1-4757-5945-7_11.

[84] Benoit Lavoie, Owen Rambow, and Ehud Reiter. The modelexplainer. In *Eighth International Natural Language Generation Workshop, INLG 1996, Herstmonceux Castle, Sussex, UK, June 12-15, 1996 - Posters and Demonstrations*, 1996. URL https://aclanthology.org/W96-0503/.

[85] Farid Meziane, Nikos Athanasakis, and Sophia Ananiadou. Generating natural language specifications from UML class diagrams. *Requir. Eng.*, 13(1):1–18, 2008. doi: 10.1007/S00766-007-0054-0. URL https://doi.org/10.1007/s00766-007-0054-0.

[86] Albert Gatt and Emiel Krahmer. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018. URL https://doi.org/10.1613%2Fjair.5477.

[87] James Clarke and Mirella Lapata. Discourse constraints for document compression. *Comput. Linguistics*, 36(3):411–441, 2010. doi: 10.1162/COLI_A_00004. URL https://doi.org/10.1162/coli_a_00004.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[88] Hugo Gonçalo Oliveira. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In José Maria Alonso, Alberto Bugarín, and Ehud Reiter, editors, *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, pages 11–20. Association for Computational Linguistics, 2017. doi: 10.18653/V1/W17-3502. URL https://doi.org/10.18653/v1/w17-3502.

[89] Robert Dale, Ilya Anisimoff, and George Narroway. HOO 2012: A report on the preposition and determiner error correction shared task. In Joel R. Tetreault, Jill Burstein, and Claudia Leacock, editors, *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, BEA@NAACL-HLT 2012, June 7, 2012, Montréal, Canada*, pages 54–62. The Association for Computer Linguistics, 2012. URL https://aclanthology.org/W12-2006/.

[90] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell et al. Language models are few-shot learners, 2020.

[91] Van-Khanh Tran and Le-Minh Nguyen. Natural language generation for spoken dialogue system using rnn encoder-decoder networks. *ArXiv*, abs/1706.00139, 2017. doi: 10.18653/v1/K17-1044.

[92] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *CoRR*, abs/2303.18223, 2023. doi: 10.48550/ARXIV.2303.18223. URL https://doi.org/10.48550/arXiv.2303.18223.

[93] Juliette Faille, Albert Gatt, and Claire Gardent. The natural language pipeline, neural text generation and explainability. In *2nd Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence*, pages 16–21, Dublin, Ireland, November 2020. Association for Computational Linguistics.

[94] Javier González Corbelle, Alberto Bugarín Diz, Jose Alonso-Moral, and Juan Taboada. Dealing with hallucination and omission in neural natural language generation: A use

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

case on meteorology. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 121–130, 2022.

[95] Lotfi A. Zadeh. Generalized theory of uncertainty (GTU) - principal concepts and ideas. *Comput. Stat. Data Anal.*, 51(1):15–46, 2006. doi: 10.1016/j.csda.2006.04.029. URL https://doi.org/10.1016/j.csda.2006.04.029.

[96] Lotfi A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications*, 9(1):149–184, 1983. ISSN 0898-1221. doi: https://doi.org/10.1016/0898-1221(83)90013-5. URL https://www.sciencedirect.com/science/article/pii/0898122183900135.

[97] Ronald R. Yager. A new approach to the summarization of data. *Information Sciences*, 28(1):69–86, 1982.

[98] Janusz Kacprzyk and Sławomir Zadrożny. Data mining via linguistic summaries of databases: An interactive approach. 03 2001.

[99] Janusz Kacprzyk and Sławomir Zadrożny. Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools. *Information Sciences*, 173(4):281–304, 2005. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2005.03.002. Dealing with Uncertainty in Data Mining and Information Extraction.

[100] L A Zadeh. A prototype-centered approach to adding deduction capability to search engines-the concept of protoform. In *Proc. NAFIPS-FLINT 2002*, pages 523–525, 2002.

[101] L A Zadeh. A prototype-centered approach to adding deduction capability to search engines-the concept of protoform. In *Proc. NAFIPS-FLINT 2002*, pages 523–525, 2002.

[102] Andrea Cascallar Fuentes, Alejandro Ramos-Soto, and Alberto José Bugarín Diz. An experimental study on the behaviour of fuzzy quantification models. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *ECAI 2020*, volume 325, pages 267–274. IOS Press, 2020.

[103] Alejandro Ramos-Soto and Patricia Martín-Rodilla. Enriching linguistic descriptions of data: A framework for composite protoforms. *Fuzzy Sets Syst.*, 407:1–26, 2021.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[104] A Ramos-Soto, A Bugarín, and S Barro. On the role of linguistic descriptions of data in the building of natural language generation systems. *Fuzzy Sets and Systems*, 285:31–51, 2016.

[105] Beatriz Blanco-Filgueira, Daniel García-Lesta, Mauro Fernández-Sanjurjo, Víctor M. Brea, and Paula López. Deep learning-based multiple object visual tracking on embedded system for iot and mobile edge computing applications. *IEEE Internet Things J.*, 6(3):5423–5431, 2019. doi: 10.1109/JIOT.2019.2902141. URL https://doi.org/10.1109/JIOT.2019.2902141.

[106] Amir Gandomi and Murtaza Haider. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137–144, 2015. ISSN 0268-4012. doi: https://doi.org/10.1016/j.ijinfomgt.2014.10.007. URL https://www.sciencedirect.com/science/article/pii/S0268401214001066.

[107] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nat.*, 521(7553): 436–444, 2015. doi: 10.1038/NATURE14539. URL https://doi.org/10.1038/nature14539.

[108] Lakshmikar R. Polamreddy and Youshan Zhang. Laksnet: an end-to-end deep learning model for self-driving cars in udacity simulator. *CoRR*, abs/2310.16103, 2023. doi: 10.48550/ARXIV.2310.16103. URL https://doi.org/10.48550/arXiv.2310.16103.

[109] Lorenzo Vaquero, Victor M Brea, and Manuel Mucientes. Tracking more than 100 arbitrary objects at 25 fps through deep learning. *Pattern Recognition*, 121:108205, 2022.

[110] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[111] Aleksandra Revina and Ünal Aksu. An approach for analyzing business process execution complexity based on textual data and event log. *Inf. Syst.*, 114:102184, 2023. doi: 10.1016/J.IS.2023.102184. URL https://doi.org/10.1016/j.is.2023.102184.

[112] Maxim Vidgof, Stefan Bachhofner, and Jan Mendling. Large language models for business process management: Opportunities and challenges. In Chiara Di Francescomarino, Andrea Burattin, Christian Janiesch, and Shazia W. Sadiq, editors, *Business Process Management Forum - BPM 2023 Forum, Utrecht, The Netherlands, September 11-15, 2023, Proceedings*, volume 490 of *Lecture Notes in Business Information Processing*, pages 107–123. Springer, 2023. doi: 10.1007/978-3-031-41623-1_7. URL https://doi.org/10.1007/978-3-031-41623-1_7.

[113] Nataliia Klievtsova, Janik-Vasily Benzin, Timotheus Kampik, Juergen Mangler, and Stefanie Rinderle-Ma. Conversational process modelling: State of the art, applications, and implications in practice. In Chiara Di Francescomarino, Andrea Burattin, Christian Janiesch, and Shazia W. Sadiq, editors, *Business Process Management Forum - BPM 2023 Forum, Utrecht, The Netherlands, September 11-15, 2023, Proceedings*, volume 490 of *Lecture Notes in Business Information Processing*, pages 319–336. Springer, 2023. doi: 10.1007/978-3-031-41623-1_19. URL https://doi.org/10.1007/978-3-031-41623-1_19.

[114] Michael Grohs, Luka Abb, Nourhan Elsayed, and Jana-Rebecca Rehse. Large language models can accomplish business process management tasks. *CoRR*, abs/2307.09923, 2023. doi: 10.48550/ARXIV.2307.09923. URL https://doi.org/10.48550/arXiv.2307.09923.

[115] Urszula Jessen, Michal Sroka, and Dirk Fahland. Chit-chat or deep talk: Prompt engineering for process mining. *CoRR*, abs/2307.09909, 2023. doi: 10.48550/ARXIV.2307.09909. URL https://doi.org/10.48550/arXiv.2307.09909.

[116] Fabian Friedrich, Jan Mendling, and Frank Puhlmann. Process model generation from natural language text. In Haralambos Mouratidis and Colette Rolland, editors, *Advanced Information Systems Engineering - 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings*, volume 6741 of *Lecture Notes in Computer*

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Science, pages 482–496. Springer, 2011. doi: 10.1007/978-3-642-21640-4_36. URL https://doi.org/10.1007/978-3-642-21640-4_36.

[117] Henrik Leopold, Sergey Smirnov, and Jan Mendling. Recognising activity labeling styles in business process models. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, 6(1):16–29, 2011.

[118] Henrik Leopold, Jan Mendling, and Hajo A. Reijers. On the automatic labeling of process models. In Haralambos Mouratidis and Colette Rolland, editors, *Advanced Information Systems Engineering*, pages 512–520, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21640-4.

[119] Pedro Gamallo-Fernandez, Juan C Vidal, and Manuel Lama. Learning context-based representations of events in complex processes. In *2023 IEEE International Conference on Web Services (ICWS)*, pages 214–223. IEEE Computer Society, 2023.

[120] Henrik Leopold, Han van der Aa, Jelmer Offenberg, and Hajo A. Reijers. Using hidden markov models for the accurate linguistic analysis of process model activity labels. *Inf. Syst.*, 83:30–39, 2019. doi: 10.1016/J.IS.2019.02.005. URL https://doi.org/10.1016/j.is.2019.02.005.

[121] Henrik Leopold, Sergey Smirnov, and Jan Mendling. On the refactoring of activity labels in business process models. *Information Systems*, 37(5):443–459, 2012.

[122] Henrik Leopold, Jan Mendling, and Artem Polyvyanyy. Generating natural language texts from business process models. In Jolita Ralyté, Xavier Franch, Sjaak Brinkkemper, and Stanislaw Wrycza, editors, *Advanced Information Systems Engineering - 24th International Conference, CAiSE 2012, Gdansk, Poland, June 25-29, 2012. Proceedings*, volume 7328 of *Lecture Notes in Computer Science*, pages 64–79. Springer, 2012. doi: 10.1007/978-3-642-31095-9_5. URL https://doi.org/10.1007/978-3-642-31095-9_5.

[123] Henrik Leopold, Jan Mendling, and Artem Polyvyanyy. Supporting process model validation through natural language generation. *IEEE Trans. Software Eng.*, 40(8):818–840, 2014. doi: 10.1109/TSE.2014.2327044. URL https://doi.org/10.1109/TSE.2014.2327044.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[124] Henrik Leopold, Rami-Habib Eid-Sabbagh, Jan Mendling, Leonardo Guerreiro Azevedo, and Fernanda Araujo Baião. Detection of naming convention violations in process models for different languages. *Decis. Support Syst.*, 56:310–325, 2013. doi: 10.1016/J.DSS.2013.06.014. URL https://doi.org/10.1016/j.dss.2013.06.014.

[125] Han van der Aa, Henrik Leopold, and Hajo A. Reijers. Detecting inconsistencies between process models and textual descriptions. In Hamid Reza Motahari-Nezhad, Jan Recker, and Matthias Weidlich, editors, *Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*, volume 9253 of *Lecture Notes in Computer Science*, pages 90–105. Springer, 2015. doi: 10.1007/978-3-319-23063-4_6. URL https://doi.org/10.1007/978-3-319-23063-4_6.

[126] Lars Ackermann, Stefan Schönig, Michael Zeising, and Stefan Jablonski. Natural language generation for declarative process models. In Joseph Barjis, Robert Pergl, and Eduard Babkin, editors, *Enterprise and Organizational Modeling and Simulation - 11th International Workshop, EOMAS 2015, Held at CAiSE 2015, Stockholm, Sweden, June 8-9, 2015, Selected Papers*, volume 231 of *Lecture Notes in Business Information Processing*, pages 3–19. Springer, 2015. doi: 10.1007/978-3-319-24626-0_1. URL https://doi.org/10.1007/978-3-319-24626-0_1.

[127] Anna Wilbik and Remco M. Dijkman. Linguistic summaries of process data. *2015 IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*, pages 1–7, 2015.

[128] Remco M. Dijkman and Anna Wilbik. Linguistic summarization of event logs - A practical approach. *Inf. Syst.*, 67:114–125, 2017.

[129] Chen Qian, Lijie Wen, Jianmin Wang, Akhil Kumar, and Haoran Li. Structural descriptions of process models based on goal-oriented unfolding. In Eric Dubois and Klaus Pohl, editors, *Advanced Information Systems Engineering - 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings*, volume 10253 of *Lecture Notes in Computer Science*, pages 397–412. Springer, 2017. doi: 10.1007/978-3-319-59536-8_25. URL https://doi.org/10.1007/978-3-319-59536-8_25.

[130] Josep Sànchez-Ferreres, Josep Carmona, and Lluís Padró. Aligning textual and graphical descriptions of processes through ILP techniques. In Eric Dubois and Klaus Pohl, editors,

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Advanced Information Systems Engineering - 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings, volume 10253 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2017. doi: 10.1007/978-3-319-59536-8_26. URL https://doi.org/10.1007/978-3-319-59536-8_26.

[131] Josep Sànchez-Ferreres, Han van der Aa, Josep Carmona, and Lluís Padró. Aligning textual and model-based process descriptions. *Data Knowl. Eng.*, 118:25–40, 2018. doi: 10.1016/J.DATAK.2018.09.001. URL https://doi.org/10.1016/j.datak.2018.09.001.

[132] X. Han et al. Bootstrapping natural language querying on process automation data. In *2020 IEEE International Conference on Services Computing (SCC)*, pages 170–177, 2020.

[133] Anti Alman, Karl Johannes Balder, Fabrizio Maria Maggi, and Han van der Aa. Declo: A chatbot for user-friendly specification of declarative process models. In Wil M. P. van der Aalst, Jan vom Brocke, Marco Comuzzi, Claudio Di Ciccio, Félix García, Akhil Kumar, Jan Mendling, Brian T. Pentland, Luise Pufahl, Manfred Reichert, and Mathias Weske, editors, *Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2020 co-located with the 18th International Conference on Business Process Management (BPM 2020), Sevilla, Spain, September 13-18, 2020*, volume 2673 of *CEUR Workshop Proceedings*, pages 122–126. CEUR-WS.org, 2020. URL https://ceur-ws.org/Vol-2673/paperDR12.pdf.

[134] Meriana Kobeissi, Nour Assy, Walid Gaaloul, Bruno Defude, and Bassem Haidar. An intent-based natural language interface for querying process execution data. In Claudio Di Ciccio, Chiara Di Francescomarino, and Pnina Soffer, editors, *3rd International Conference on Process Mining, ICPM 2021, Eindhoven, The Netherlands, October 31 - Nov. 4, 2021*, pages 152–159. IEEE, 2021. doi: 10.1109/ICPM53251.2021.9576850. URL https://doi.org/10.1109/ICPM53251.2021.9576850.

[135] Luciana Barbieri, Edmundo R. M. Madeira, Kleber Stroeh, and Wil M. P. van der Aalst. A natural language querying interface for process mining. *J. Intell. Inf. Syst.*, 61(1): 113–142, 2023. doi: 10.1007/S10844-022-00759-9. URL https://doi.org/10.1007/s10844-022-00759-9.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[136] Sudhanshu Chouhan, Anna Wilbik, and Remco Dijkman. Explanation of anomalies in business process event logs with linguistic summaries. In *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–7, 2022. doi: 10.1109/FUZZ-IEEE55066.2022.9882673.

[137] Daniel Barón-Espitia, Marlon Dumas, and Oscar González Rojas. Coral: Conversational what-if process analysis (extended abstract). In Marwan Hassani, Agnes Koschmider, Marco Comuzzi, Fabrizio Maria Maggi, and Luise Pufahl, editors, *Proceedings of the ICPM Doctoral Consortium and Demo Track 2022 co-located with 4th International Conference on Process Mining (ICPM 2022), Bolzano, Italy, October, 2022*, volume 3299 of *CEUR Workshop Proceedings*, pages 118–122. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3299/Paper25.pdf.

[138] Julian Caspary, Adrian Rebmann, and Han van der Aa. Does this make sense? machine learning-based detection of semantic anomalies in business processes. In Chiara Di Francescomarino, Andrea Burattin, Christian Janiesch, and Shazia Sadiq, editors, *Business Process Management - 21st International Conference, BPM 2023, Utrecht, The Netherlands, September 11-15, 2023, Proceedings*, volume 14159 of *Lecture Notes in Computer Science*, pages 163–179. Springer, 2023. doi: 10.1007/978-3-031-41620-0_10. URL https://doi.org/10.1007/978-3-031-41620-0_10.

[139] Anna Wilbik and Remco M. Dijkman. On the generation of useful linguistic summaries of sequences. In *2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016, Vancouver, BC, Canada, July 24-29, 2016*, pages 555–562. IEEE, 2016. doi: 10.1109/FUZZ-IEEE.2016.7737736. URL https://doi.org/10.1109/FUZZ-IEEE.2016.7737736.

[140] Luis Delicado, Josep Sànchez-Ferreres, Josep Carmona, and Lluís Padró. NLP4BPM - natural language processing tools for business process management. In Robert Clarisó, Henrik Leopold, Jan Mendling, Wil M. P. van der Aalst, Akhil Kumar, Brian T. Pentland, and Mathias Weske, editors, *Proceedings of the BPM Demo Track and BPM Dissertation Award co-located with 15th International Conference on Business Process Modeling (BPM 2017), Barcelona, Spain, September 13, 2017*, volume 1920 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017. URL https://ceur-ws.org/Vol-1920/BPM_2017_paper_164.pdf.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[141] A.M. Wilbik and U. Kaymak. Linguistic summarization of processes : a research agenda. In Jose M. Alonso, Humberto Bustince, and Marek Reformat, editors, *International Joint Conference IFSA-EUSFLAT : 16th World Congress of the International Fuzzy Systems Association (IFSA) and 9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), 30 June - 3 July 2015, Gijon, Spain*, pages 1636–1643, Netherlands, 2015. Atlantis Press.

[142] Han van der Aa, Josep Carmona, Henrik Leopold, Jan Mendling, and Lluís Padró. Challenges and opportunities of applying natural language processing in business process management. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2791–2801. Association for Computational Linguistics, 2018. URL https://aclanthology.org/C18-1236/.

[143] Henrik Leopold. Natural language in business process models. In *Lecture Notes in Business Information Processing*, 2013.

[144] Adela del-Río-Ortega, Manuel Resinas, Amador Durán, Beatriz Bernárdez, Antonio Ruiz-Cortés, and Miguel Toro. Visual ppinot: A graphical notation for process performance indicators. *Bus. Inf. Syst. Eng.*, 61(2):137–161, 2019. doi: 10.1007/S12599-017-0483-3. URL https://doi.org/10.1007/s12599-017-0483-3.

[145] Jaydeep Sen, Chuan Lei, Abdul Quamar, Fatma Özcan, Vasilis Efthymiou, Ayushi Dalmia, Greg Stager, Ashish R. Mittal, Diptikalyan Saha, and Karthik Sankaranarayanan. ATHENA++: natural language querying for complex nested SQL queries. *Proc. VLDB Endow.*, 13(11):2747–2759, 2020. URL http://www.vldb.org/pvldb/vol13/p2747-sen.pdf.

[146] Stefan Schönig, Cristina Cabanillas, Stefan Jablonski, and Jan Mendling. Mining the organisational perspective in agile business processes. In Khaled Gaaloul, Rainer Schmidt, Selmin Nurcan, Sérgio Guerreiro, and Qin Ma, editors, *Enterprise, Business-Process and Information Systems Modeling - 16th International Conference, BPMDS 2015, 20th International Conference, EMMSAD 2015, Held at CAiSE 2015, Stockholm, Sweden, June 8-9, 2015, Proceedings*, volume 214 of *Lecture Notes in Business Information*

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Processing, pages 37–52. Springer, 2015. doi: 10.1007/978-3-319-19237-6_3. URL https://doi.org/10.1007/978-3-319-19237-6_3.

[147] Janusz Kacprzyk and Slawomir Zadrozny. Computing with words is an implementable paradigm: Fuzzy queries, linguistic data summaries, and natural-language generation. *IEEE Transactions on Fuzzy Systems*, 18(3):461–472, 2010.

[148] Janusz Kacprzyk, Ronald R. Yager, and Slawomir Zadrozny. *Fuzzy Linguistic Summaries of Databases for an Efficient Business Data Analysis and Decision Support*, pages 129–152. Springer, 2002.

[149] Rita Castillo-Ortega, Nicolás Marín, and Daniel Sánchez. A fuzzy approach to the linguistic summarization of time series. *Multiple-Valued Logic and Soft Computing*, 17 (2-3):157–182, 2011.

[150] Miguel Delgado, M. Dolores Ruiz, Daniel Sánchez, and M. Amparo Vila. Fuzzy quantification: a state of the art. *Fuzzy Sets and Systems*, 242:1–30, 2014.

[151] David Chapela-Campa, Manuel Mucientes, and Manuel Lama. Discovering infrequent behavioral patterns in process models. In Josep Carmona, Gregor Engels, and Akhil Kumar, editors, *Business Process Management*, Cham, 2017. Springer International Publishing.

[152] Eric Rojas, Jorge Munoz-Gama, Marcos Sepúlveda, and Daniel Capurro. Process mining in healthcare: A literature review. *Journal of Biomedical Informatics*, 61:224–236, 2016. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2016.04.007.

[153] Ronny S. Mans, Wil M. P. van der Aalst, Rob J. B. Vanwersch, and Arnold J. Moleman. Process mining in healthcare: Data challenges when answering frequently posed questions. In Richard Lenz, Silvia Miksch, Mor Peleg, Manfred Reichert, David Riaño, and Annette ten Teije, editors, *Process Support and Knowledge Representation in Health Care*, pages 140–153, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-36438-9.

[154] Jorge Munoz-Gama, Niels Martin, Carlos Fernández-Llatas, Owen A. Johnson, and Marcos Sepúlveda. Innovative informatics methods for process mining in health care. *J. Biomed. Informatics*, 134:104203, 2022. doi: 10.1016/J.JBI.2022.104203. URL https://doi.org/10.1016/j.jbi.2022.104203.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[155] Jorge Munoz-Gama, Niels Martin, Carlos Fernández-Llatas, Owen A. Johnson, Marcos Sepúlveda, Emmanuel Helm, Victor Galvez-Yanjari, Eric Rojas, Antonio Martinez-Millana, Davide Aloini, Ilaria Angela Amantea, Robert Andrews, Michael Arias, Iris Beerepoot, Elisabetta Benevento, Andrea Burattin, Daniel Capurro, Josep Carmona, Marco Comuzzi, Benjamin Dalmas, Rene de la Fuente, Chiara Di Francescomarino, Claudio Di Ciccio, Roberto Gatta, Chiara Ghidini, Fernanda Gonzalez-Lopez, Gema Ibáñez-Sánchez, Hilda B. Klasky, Angelina Prima Kurniati, Xixi Lu, Felix Mannhardt, Ronny Mans, Mar Marcos, Renata Medeiros de Carvalho, Marco Pegoraro, Simon K. Poon, Luise Pufahl, Hajo A. Reijers, Simon Remy, Stefanie Rinderle-Ma, Lucia Sacchi, Fernando Seoane, Minseok Song, Alessandro Stefanini, Emilio Sulis, Arthur H. M. ter Hofstede, Pieter J. Toussaint, Vicente Traver, Zoe Valero-Ramon, Inge van de Weerd, Wil M. P. van der Aalst, Rob J. B. Vanwersch, Mathias Weske, Moe Thandar Wynn, and Francesca Zerbato. Process mining for healthcare: Characteristics and challenges. *J. Biomed. Informatics*, 127:103994, 2022. doi: 10.1016/J.JBI.2022.103994. URL https://doi.org/10.1016/j.jbi.2022.103994.

[156] Emmelien De Roock and Niels Martin. Process mining in healthcare - an updated perspective on the state of the art. *J. Biomed. Informatics*, 127:103995, 2022. doi: 10.1016/J.JBI.2022.103995. URL https://doi.org/10.1016/j.jbi.2022.103995.

[157] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[158] Fatih Emre Boran, Diyar Akay, and Ronald R. Yager. An overview of methods for linguistic summarization with fuzzy sets. *Expert Syst. Appl.*, 61:356–377, 2016. doi: 10.1016/J.ESWA.2016.05.044. URL https://doi.org/10.1016/j.eswa.2016.05.044.

[159] Somayajulu Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. Generating english summaries of time series data using the gricean maxims. In Lise Getoor, Ted E. Senator, Pedro M. Domingos, and Christos Faloutsos, editors, *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pages 187–196. ACM, 2003. doi: 10.1145/956750.956774. URL https://doi.org/10.1145/956750.956774.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[160] A Ramos-Soto, A Bugarín, and S Barro. On the role of linguistic descriptions of data in the building of natural language generation systems. *Fuzzy Sets and Systems*, 285:31–51, 2016. ISSN 0165-0114. doi: https://doi.org/10.1016/j.fss.2015.06.019.

[161] Nicolás Marín and Daniel Sánchez. On generating linguistic descriptions of time series. *Fuzzy Sets Syst.*, 285:6–30, 2016.

[162] Miguel Delgado, M. Dolores Ruiz, Daniel Sánchez, and M. Amparo Vila. Fuzzy quantification: a state of the art. *Fuzzy Sets and Systems*, 242:1–30, 2014.

[163] Janusz Kacprzyk and Slawomir Zadrozny. Fuzzy linguistic summaries in text categorization for human-consistent document-driven decision support systems. In Bernd Reusch, editor, *Computational Intelligence, Theory and Applications, International Conference 8th Fuzzy Days, Dortmund, Germany, Sept. 29 - Oct. 01, 2004*, volume 33 of *Advances in Soft Computing*, pages 271–279. Springer, 2004. doi: 10.1007/3-540-31182-3_24. URL https://doi.org/10.1007/3-540-31182-3_24.

[164] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning-iii. *Information Sciences*, 9(1):43–80, 1975.

[165] Robert Babuska. Fuzzy systems, modeling and identification. 1998. URL https://api.semanticscholar.org/CorpusID:5314384.

[166] Ingo Glöckner and Alois Knoll. A formal theory of fuzzy natural language quantification and its role in granular computing. 2001. URL https://api.semanticscholar.org/CorpusID:118584741.

[167] Yaxin Liu and Etienne E. Kerre. An overview of fuzzy quantifiers. (i). interpretations. *Fuzzy Sets and Systems*, 95(1):1–21, 1998. ISSN 0165-0114. doi: https://doi.org/10.1016/S0165-0114(97)00254-6. URL https://www.sciencedirect.com/science/article/pii/S0165011497002546.

[168] S. Barro, A.J. Bugarin, P. Carinena, and F. Diaz-Hermida. A framework for fuzzy quantification models analysis. *IEEE Transactions on Fuzzy Systems*, 11(1):89–99, 2003. doi: 10.1109/TFUZZ.2002.806319.

[169] Lotfi A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications*, 9(1):149–184, 1983.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[170] Lotfi A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 9(2):111–127, 1983.

[171] J. Fodor. Contrapositive symmetry of fuzzy implications. *Fuzzy Sets and Systems*, 69: 141–156, 1995. doi: 10.1016/0165-0114(94)00210-X.

[172] Rita Castillo-Ortega, Nicolás Marín, Daniel Sánchez, and Andrea Tettamanzi. Quality assessment in linguistic summaries of data. In Salvatore Greco, Bernadette Bouchon-Meunier, Giulianella Coletti, Mario Fedrizzi, Benedetto Matarazzo, and Ronald R. Yager, editors, *Advances on Computational Intelligence - 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2012, Catania, Italy, July 9-13, 2012. Proceedings, Part II*, volume 298 of *Communications in Computer and Information Science*, pages 285–294. Springer, 2012. doi: 10.1007/978-3-642-31715-6_31. URL https://doi.org/10.1007/978-3-642-31715-6_31.

[173] Alberto Bugarín, Nicolás Marín, Daniel Sánchez, and Gracián Triviño. Aspects of quality evaluation in linguistic descriptions of data. In Adnan Yazici, Nikhil R. Pal, Uzay Kaymak, Trevor Martin, Hisao Ishibuchi, Chin-Teng Lin, João M. C. Sousa, and Bülent Tütmez, editors, *2015 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2015, Istanbul, Turkey, August 2-5, 2015*, pages 1–8. IEEE, 2015. doi: 10.1109/FUZZ-IEEE.2015.7338055. URL https://doi.org/10.1109/FUZZ-IEEE.2015.7338055.

[174] Purificación Cariñena, Alberto Bugarín, Manuel Mucientes, and Senén Barro. A language for expressing expert knowledge using fuzzy temporal rules. In *Proceedings of the EUSFLAT-ESTYLF Joint Conference*, pages 171–174, 1999.

[175] P. Cariñena. *A model of Fuzzy Temporal Rules for reasoning on dynamic systems.* PhD thesis, Universidade de Santiago de Compostela, 2003.

[176] Félix Díaz-Hermida, Purificación Cariñena, and Alberto Bugarín. Efficient evaluation of similarity quantified expressions in the temporal domain. In Jonathan Lawry, Enrique Miranda, Alberto Bugarín, Shoumei Li, María Angeles Gil, Przemyslaw Grzegorzewski, and Olgierd Hryniewicz, editors, *Soft Methods for Integrated Uncertainty Modelling, Proceedings of the 2006 International Workshop on Soft Methods in Probability and*

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Statistics, SMPS 2006, Bristol, UK, 5-7 September 2006, volume 37 of *Advances in Soft Computing*, pages 191–198. Springer, 2006. doi: 10.1007/3-540-34777-1_24. URL https://doi.org/10.1007/3-540-34777-1_24.

[177] Anna Wilbik and Uzay Kaymak. Gradual linguistic summaries. In Anne Laurent, Olivier Strauss, Bernadette Bouchon-Meunier, and Ronald R. Yager, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 15th International Conference, IPMU 2014, Montpellier, France, July 15-19, 2014, Proceedings, Part II*, volume 443 of *Communications in Computer and Information Science*, pages 405–413. Springer, 2014. doi: 10.1007/978-3-319-08855-6_41. URL https://doi.org/10.1007/978-3-319-08855-6_41.

[178] Massimiliano de Leoni, Wil M. P. van der Aalst, and Marcus Dees. A general framework for correlating business process characteristics. In Shazia Wasim Sadiq, Pnina Soffer, and Hagen Völzer, editors, *Business Process Management - 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings*, volume 8659 of *Lecture Notes in Computer Science*, pages 250–266. Springer, 2014. doi: 10.1007/978-3-319-10172-9_16. URL https://doi.org/10.1007/978-3-319-10172-9_16.

[179] Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368. Association for Computational Linguistics, October 2019.

[180] Fred D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340, 1989.

[181] Nava Tintarev, Ehud Reiter, Rolf Black, Annalu Waller, and Joe Reddington. Personal storytelling: Using natural language generation for children with complex communication needs, in the wild... *International Journal of Human Computer Studies*, 92-93:1–16, September 2016. ISSN 1071-5819. doi: 10.1016/j.ijhcs.2016.04.005.

[182] Daniel A. Keim, Gennady L. Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual analytics: Definition, process, and challenges. In Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors, *Information Visualization - Human-Centered Issues and Perspectives*, volume 4950 of

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Lecture Notes in Computer Science, pages 154–175. Springer, 2008. doi: 10.1007/978-3-540-70956-5_7. URL https://doi.org/10.1007/978-3-540-70956-5_7.

[183] José Maria Alonso and Luis Magdalena. HILK++: an interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based classifiers. *Soft Comput.*, 15(10):1959–1980, 2011. doi: 10.1007/S00500-010-0628-5. URL https://doi.org/10.1007/s00500-010-0628-5.

[184] IEEE standard for extensible event stream (XES) for achieving interoperability in event logs and event streams. *IEEE Std 1849-2016*, pages 1–50, 2016.

[185] A. Augusto, R. Conforti, M. Dumas, M. L. Rosa, F. M. Maggi, A. Marrella, M. Mecella, and A. Soo. Automated discovery of process models from event logs: Review and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 31(4):686–705, 2019.

[186] Wil M. P. van der Aalst. *Process Mining: Data Science in Action*. Springer, 2016. ISBN 978-3-662-49850-7.

[187] A. Weijters and Wil Aalst. Rediscovering workflow models from event-based data using little thumb. *Integrated Computer-Aided Engineering*, 10:151–162, 07 2003.

[188] Sander J. J. Leemans, Dirk Fahland, and Wil M.P. van der Aalst. Discovering block-structured process models from event logs - a constructive approach. In *Application and Theory of Petri Nets and Concurrency*, pages 311–329. Springer, 2013. ISBN 978-3-642-38697-8.

[189] Adela del Río-Ortega, Manuel Resinas, and Antonio Ruiz-Cortés. Defining process performance indicators: An ontological approach. In Robert Meersman, Tharam Dillon, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems: OTM 2010*, pages 555–572, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-16934-2.

[190] Oscar Cordón, Francisco Herrera, Frank Hoffmann, and Luis Magdalena. *Genetic Fuzzy Systems - Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, volume 19 of *Advances in Fuzzy Systems - Applications and Theory*. World Scientific, 2001. ISBN 978-981-02-4016-5. doi: 10.1142/4177. URL https://doi.org/10.1142/4177.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[191] J.-S.R. Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993. doi: 10.1109/21.256541.

[192] Ilija Svalina, Vjekoslav Galzina, Roberto Lujic, and Goran Simunovic. An adaptive network-based fuzzy inference system (ANFIS) for the forecasting: The case of close price indices. *Expert Syst. Appl.*, 40(15):6055–6063, 2013. doi: 10.1016/J.ESWA.2013.05.029. URL https://doi.org/10.1016/j.eswa.2013.05.029.

[193] Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2(1):49–79, 2004. ISSN 1570-8268. doi: https://doi.org/10.1016/j.websem.2004.07.005. URL https://www.sciencedirect.com/science/article/pii/S1570826804000162.

[194] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3C Recommendation, 2008. URL http://www.w3.org/TR/rdf-sparql-query/.

[195] Birte Glimm, Ian Horrocks, B. Motik, G. Stoilos, and Zhe Wang. Hermit: An owl 2 reasoner. *Journal of Automated Reasoning*, 53:245–269, 2014. doi: 10.1007/s10817-014-9305-1.

[196] E. Sirin, B. Parsia, B. C. Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *J. Web Semant.*, 5:51–53, 2007. doi: 10.1016/j.websem.2007.03.004.

[197] William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text & Talk*, 8:243 – 281, 1988. URL https://api.semanticscholar.org/CorpusID:60514661.

[198] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, and Matthias Gallé et al. Bloom: A 176b-parameter open-access multilingual language model, 2023.

[199] Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023. URL https://github.com/openlm-research/open_llama.

[200] Blase A Carabello and Walter J Paulus. Aortic stenosis. *The Lancet*, 373(9667):956–966, 2009. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(09)60211-7.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[201] World Health Organization. https://www.who.int/health-topics/cardiovascular-diseases, 2021. [Online; accessed in August 2021].

[202] Gry Eveborn, Henrik Schirmer, Geir Heggelund, Per Lunde, and Knut Rasmussen. The evolving epidemiology of valvular aortic stenosis. the tromso study. *Heart (British Cardiac Society)*, 99, 09 2012. doi: 10.1136/heartjnl-2012-302265.

[203] Martin B. Leon, Craig R. Smith, Michael Mack, D. Craig Miller, Jeffrey W. Moses, Lars G. Svensson, E. Murat Tuzcu, John G. Webb, Gregory P. Fontana, Raj R. Makkar, David L. Brown, Peter C. Block, Robert A. Guyton, Augusto D. Pichard, Joseph E. Bavaria, Howard C. Herrmann, Pamela S. Douglas, John L. Petersen, Jodi J. Akin, William N. Anderson, Duolao Wang, and Stuart Pocock. Transcatheter aortic-valve implantation for aortic stenosis in patients who cannot undergo surgery. *New England Journal of Medicine*, 363(17):1597–1607, 2010.

[204] R. Gomis, M. Mata Cases, D. Mauricio Puente, S. Artola Menéndez, J. Ena Muñoz, J. J. Mediavilla Bravo, C. Miranda Fernández-Santos, D. Orozco Beltrán, L. Rodríguez Mañas, C. Sánchez Villalba, and J. A. Martínez. Aspectos metodológicos de los procesos asistenciales integrados (PAI). *Revista de Calidad Asistencial*, 32(4):234–239, 2017.

[205] A. Ramos-Soto, J. Janeiro-Gallardo, and Alberto Bugarín. Adapting SimpleNLG to spanish. In *10th International Conference on Natural Language Generation*, pages 142–146. Association for Computational Linguistics, 2017. ISBN 978-1-945626-52-4.

[206] Albert Gatt and Ehud Reiter. SimpleNLG: A Realisation Engine for Practical Applications. In *ENLG 2009 - Proceedings of the 12th European Workshop on Natural Language Generation, March 30-31, 2009, Athens, Greece*, pages 90–93. The Association for Computer Linguistics, 2009.

[207] A. Leff and J.T. Rayfield. Web-application development using the model/view/controller design pattern. In *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*, pages 118–127, 2001. doi: 10.1109/EDOC.2001.950428.

[208] Joëlle Coutaz. Pac, an object oriented model for dialog design. In *Human–Computer Interaction–INTERACT'87*, pages 431–436. Elsevier, 1987.

[209] Alessandro Berti, Sebastiaan van Zelst, and Daniel Schuster. Pm4py: A process mining library for python. *Software Impacts*, 17:100556, 2023. ISSN 2665-9638. doi:

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

https://doi.org/10.1016/j.simpa.2023.100556. URL https://www.sciencedirect.com/science/article/pii/S2665963823000933.

[210] Grigoris Antoniou and Frank van Harmelen. Web ontology language: Owl. *Handbook on ontologies*, pages 91–110, 2009.

[211] C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, and M. Smith. OWL 2 Web Ontology Language. W3C recommendation, 2012.

[212] Jean-Baptiste Lamy. Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. *Artificial intelligence in medicine*, 80:11–28, 2017.

[213] Shammamah Hossain. Visualization of bioinformatics data with dash bio. pages 126–133, 2019. doi: 10.25080/MAJORA-7DDC1DD1-012.

[214] Iris Beerepoot et al. The biggest business process management problems to solve before we die. *Computers in Industry*, 146:103837, 2023. ISSN 0166-3615.

[215] Claudio Di Ciccio and Marco Montali. Declarative process specifications: Reasoning, discovery, monitoring. In Wil M. P. van der Aalst and Josep Carmona, editors, *Process Mining Handbook*, volume 448 of *Lecture Notes in Business Information Processing*, pages 108–152. Springer, 2022. doi: 10.1007/978-3-031-08848-3_4. URL https://doi.org/10.1007/978-3-031-08848-3_4.

[216] Marco Montali et al. Declarative specification and verification of service choreographies. *ACM Trans. Web*, 4(1):3:1–3:62, 2010.

[217] Cornelia Haisjackl et al. Understanding declare models: strategies, pitfalls, empirical results. *Softw. Syst. Model.*, 15(2):325–352, 2016.

[218] Felix Mannhardt. Sepsis cases: Event log, 2016. Eindhoven University of Technology.

[219] Qian Chen, Zhu Zhuo, and Wen Wang. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*, 2019.

[220] Ivan Donadello et al. Declare4py: A python library for declarative process mining. In *BPM (PhD/Demos)*, CEUR Workshop Proceedings, pages 117–121, 2022.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

[221] Fabrizio Maria Maggi et al. Efficient discovery of understandable declarative process models from event logs. In *CAiSE 2012*, pages 270–285, 2012.

[222] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. Rasa: Open source language understanding and dialogue management, 2017.

[223] Tanja Bunk, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol. DIET: lightweight language understanding for dialogue systems. *CoRR*, abs/2004.09936, 2020. URL https://arxiv.org/abs/2004.09936.

[224] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[225] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann, 2001.

[226] Francesco Fuggitti and Tathagata Chakraborti. NL2LTL – a python package for converting natural language (NL) instructions to linear temporal logic (LTL) formulas. In *AAAI*, pages 16428–16430, 2023.

[227] Felix Mannhardt. *Multi-perspective Process Mining*. PhD thesis, Technical University of Eindhoven, 2018.

[228] Dominic Espinosa, Rajakrishnan Rajkumar, Michael White, and Shoshana Berleant. Further meta-evaluation of broad-coverage surface realization. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of*

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

SIGDAT, a Special Interest Group of the ACL, pages 564–574. ACL, 2010. URL https://aclanthology.org/D10-1055/.

[229] Ehud Reiter, Roma Robertson, and Liesl Osman. Lessons from a failure: Generating tailored smoking cessation letters. *Artif. Intell.*, 144(1-2):41–58, 2003. doi: 10.1016/S0004-3702(02)00370-3. URL https://doi.org/10.1016/S0004-3702(02)00370-3.

[230] Craig Thomson and Ehud Reiter. A gold standard methodology for evaluating accuracy in data-to-text systems. In Brian Davis, Yvette Graham, John D. Kelleher, and Yaji Sripada, editors, *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020, Dublin, Ireland, December 15-18, 2020*, pages 158–168. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.inlg-1.22/.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

List of Figures

Fig. 1.1 Business Process Management framework lifecycle (adapted from [11]). . . 2

Fig. 1.2 Process Mining framework (adapted from [12]). 4

Fig. 1.3 Example of a process model using the Petri Net formalism corresponding to the event log in 1.1. 6

Fig. 1.4 DFGs generated with Disco and Celonis (extracted from [12]). 10

Fig. 1.5 Example of a dot chart in which all cases start at time zero. The chart is used to analyze variation in flow times i.e. why some cases are handled in a few days whereas others take almost a year (extracted from [12]. 11

Fig. 1.6 Screenshot of ProM 5.2. interface (extracted from [12]). It can clearly be seen how visualization techniques conform the core aspect for process analysis with this tool. 12

Fig. 1.7 Complex process model (spaghetti model) of an Aortic Stenosis Integrated Care Process in a Spanish hospital [49]. Generated with ProM [37]. 14

Fig. 2.1 Example of a process description generated by the tool (extracted from [123]) 37

Fig. 2.2 Exemplary natural language report generated with the DPIL approach (extracted from [126]). 40

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Fig. 3.1 The straightforward nature of this example facilitates a clear comprehension of the compositional relationship between a linguistic description and its constituent sentences (or instantiated protoforms). This distinction is crucial, as the literature sometimes blurs these concepts by using them interchangeably, leading to confusion. (extracted from [103]). 52
Fig. 3.2 Graphic representation (as fuzzy sets) of the five linguistic values (*"minimum"*, *"low"*, ..., *"really high"*) of the linguistic variable *"Wait time"*. 58
Fig. 3.3 Graphic representation (as fuzzy sets) of relative quantifiers. 60
Fig. 3.4 Monotonically, non-decreasing relative quantifiers. 61
Fig. 3.5 Fuzzy temporal references defined from left to right as an instant, an imprecise time interval and a fuzzy time interval (extracted from [61]). 63
Fig. 3.6 An example of a linguistic summary of a process based on an event log, similar to the example presented in Figure 3.1. 70
Fig. 3.7 The feature extraction process illustrated in a case-perspective event log. The attributes already present in the log, namely the case id, the trace, and an attribute called *admission*, are shown in grey. The newly computed attributes added to the log are highlighted in pink. 71

Fig. 4.1 From left to right, the NLG and D2T architectures respectively. Colors are used to represent how tasks are matched in between both architectures. . . . 88
Fig. 4.2 The Process-to-Text (P2T) framework. It can be seen how the proposed knowledge representation formalism represents the cornerstone of the proposal. 92
Fig. 4.3 Graphical representation of the ontology-based domain model together with two case instances and their corresponding *wait time* values. 95
Fig. 4.4 Data transformations that take place within the Process-to-Text pipeline. The original raw log is converted into a preprocessed event log, after applying a discovery algorithm a process model is obtained. With the process model and the event log, process replay and mining of additional perspectives can happen. 110
Fig. 4.5 Replaying of trace = $\langle a, b, d, e, f \rangle$ on top of its corresponding model (represented as a Petri Net). 112
Fig. 4.6 Graphic representation of the five linguistic values (*"minimum"*, *"low"*, ..., *"really high"*) defined as fuzzy sets on the basis of standard criteria for the linguistic variable *"Wait time"*. 115

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Fig. 4.7 The system identification and fuzzy inference processes for the derivation of case attribute semantics and their conversion into linguistic terms that are finally included into the domain ontology. 118
Fig. 4.8 A SPARQL query to compute the average value of the membershipDegree for instances with an attribute of class LongWaitTime. 119
Fig. 4.9 One of the scenarios presented in the evaluation questionnaire of the pipeline over the AS ICP (originally in Spanish). In this scenario a natural language description regarding the expected and real wait times between the heart team meeting and the intervention of a patient is shown next to the histogram representing the wait times (in days) for the patients between said activities on the process. 137
Fig. 4.10 Histograms of response frequencies for the questions asked in each scenario. 143
Fig. 4.11 Histograms of response frequencies for the questions asked in each scenario. 144
Fig. 4.12 Overview of the basic architecture of the NLG4PM system. 145
Fig. 4.13 Initial screen of the NLG4PM system. Here the user can upload an event log and do an exploratory analysis. 147
Fig. 4.14 The case-perspective, enriched event log is shown to the user. 148
Fig. 4.15 The case-perspective, enriched event log is shown to the user. 149
Fig. 4.16 The case-perspective, enriched event log is shown to the user. 150

Fig. 5.1 Chat-like interface of C-4PM. A text field is used to provide the assistant with a query and a main window shows the conversation (real screenshot from the demo available at tec.citius.usc.es/c-4pm). 156
Fig. 5.2 Architecture of C-4PM. Shadowed in purple are the two main components: the Rasa Stack, in charge of NLU and dialogue management; and the action server, where the reasoning tasks and response generation are carried out. . . 165
Fig. 5.3 Detailed view of the NLU pipeline. In purple, the main steps of the pipeline: intent and entity recognition and entity linking. In some cases, entity recognition and linking are subsumed by the NL2LTL module. In gray, the action server, where the computation and response generation are carried out. 166
Fig. 5.4 Fragment of a conversation in which the model checking and restricted conformance checking tasks are performed. It can be seen how the system is able of recalling entities mentioned in previous messages. 174

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

List of Tables

Tab. 1.1 Example of a log with 15 events belonging to 3 different traces. It records the execution of 5 different activities by 6 resources in a fictional company. . . . 6

Tab. 2.1 Classification of the different proposals for the use of NLU and NLG techniques in BPM, process mining, workflow management, event data and related realms. 29
Tab. 2.2 Approaches for the generation of natural language descriptions of processes from the state of the art and the types of descriptions or summaries they support. 44

Tab. 3.1 Approaches for the generation of natural language descriptions of processes from the state of the art and the types of descriptions or summaries they support. 69
Tab. 3.2 Proposed protoforms for the generation of natural language descriptions of processes. 70
Tab. 3.3 Questions shown with each example as part of the evaluation questionnaire performed to assess the proposed protoforms. 78
Tab. 3.4 General questions asked to assess the general validity and usefulness of the proposal. 79

Tab. 4.1 Extract of the original registry of clinical data from the Aortic Stenosis Integrated Care Process. 108

USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Tab. 4.2 Preprocessed event log extracted from the original AS ICP registry from Table 4.1. Columns starting with *"event_"* refer to event attributes, and columns starting with *"case_"* refer to case attributes. 109
Tab. 4.3 Enhanced event log with the wait time (expressed in days) between the heart team meeting session and the intervention of a patient and the heart team meeting and the CT scanning of a patient extracted through replaying. Duration is positive if origin event precedes destination event and negative if destination event precedes origin event. Columns starting with *"event_"* refer to event attributes, and columns starting with *"case_"* refer to case attributes. 113
Tab. 4.4 Dataset corresponding to AS ICP event log with its original attributes as well as the newly calculated with process mining techniques. 113
Tab. 4.5 Specification of the *Contrast* extended protoform. 124
Tab. 4.6 Prompt used for the linguistic realization task in the P2T framework. 126
Tab. 4.7 Example of a query and the result obtained from the system. 126
Tab. 4.8 Questions asked for each scenario on the proposed evaluation test 140
Tab. 4.9 General questions asked at the end of the evaluation test. 140
Tab. 4.10 Expert validation results for the AS ICP descriptions when taking the whole answer set (6 scenarios per 15 users). Results are shown in a 5-level numerical scale derived from the Likert-scale used in the questionnaire (1 equals "Strongly disagree" and 5 equals "Strongly agree"). Average value and its confidence interval at a 95% confidence (CI), standard deviation (SD), mode, median and interquartile range (IQR) are shown for each of the asked questions. 142
Tab. 4.11 Expert validation results for the general questions of the AS ICP. Average value and its confidence interval (CI), standard deviation (SD), mode, median and interquartile range (IQR) are shown for each of the asked questions. . . 142

Tab. 5.1 Classification of the different challenges C-4PM tackles when interpreting user input queries. ER means "Entity Recognition", IR means "Intent Recognition" and EL means "Entity Linking". The meaning of each of these tasks is explained in the text. 159
Tab. 5.2 Classification of the different proposals for the use of NLU and NLG techniques in BPM, process mining, workflow management, event data and related realms. 163

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Effectively presenting and comprehending processes is a challenging task. This thesis introduces different solutions for communicating process knowledge via natural language. We propose a taxonomy of relevant process descriptions based on a fuzzy protoform model, validated by medical experts. We introduce the Process-to-Text framework, building on the Data-To-Text architecture, leverages process mining techniques in an ontology-driven system using fuzzy logic, for the generation of accurate and contextually-aware process descriptions. Additionally, we introduce C-4PM, a conversational agent for declarative process mining, enhancing interaction and knowledge inference through natural language. This work signifies a stride towards making process mining accessible and understandable for a broader audience.