

Predicciones en Procesos de Negocio Declarativos

Andrés Jiménez-Ramírez¹, Irene Barba¹, Juan Fernández-Olivares², Carmelo Del Valle¹, and Barbara Weber³

¹ Universidad de Sevilla, Sevilla, España
{ajramirez,irenebr,carmelo}@us.es

² Universidad de Granada, Granada, España
{faro@decsai.ugr.es}

³ Technical University of Denmark, Kongens Lyngby, Denmark
{bweber@dtu.dk}

Resumen La generación de predicciones sobre instancias de procesos de negocio permite anticipar problemas, evitar el incumplimiento de restricciones de una manera proactiva, y tomar decisiones sobre prioridades y restricciones al enfrentarse a eventos inesperados, e.g., retrasos. Sin embargo, elaborar una predicción es una tarea compleja en la mayoría de los casos ya que se deben tener en cuenta múltiples instancias y recursos, es necesario adaptar dichas predicciones a circunstancias cambiantes, y hay que tener en cuenta distintas dimensiones, no sólo el tiempo. En este contexto, el presente trabajo propone una propuesta novedosa para generar predicciones sobre un conjunto de instancias en ejecución relacionadas con un modelo declarativo de un proceso de negocio. Dicha propuesta consiste en generar un plan de ejecución optimizado a partir del modelo declarativo y del estado de las instancias en ejecución. Tras ello, la predicción se genera evaluando la función que se desea predecir sobre el plan de ejecución generado. La presente propuesta ha sido evaluada utilizando un modelo de proceso de un escenario real que incluye restricciones temporales, de datos, de recursos y de control-flow que lo dotan de una alta complejidad. Los prometedores resultados obtenidos alientan a continuar los trabajos en escenarios con características diferentes que permitan extender la validez de la propuesta.

Keywords: Flexible process-aware information systems, Time prediction, Constraint programming, Planning and scheduling, Constraint-based process models

1. Introducción

Hoy en día los negocios muestran un interés creciente en la mejora de la calidad y la eficiencia de los procesos de negocio así como en alinear sus sistemas de información a una perspectiva centrada en procesos [10]. De esta manera los sistemas de información orientado a procesos (PAIS de sus siglas en inglés)[1] surgen para dotar de dinamismo y flexibilidad al soporte de los procesos de

negocio (BP de sus siglas en inglés), i.e., conjunto de actividades que son realizadas de forma coordinada en el entorno técnico de una organización y que, conjuntamente, persiguen un objetivo común [15].

Este soporte que ofrecen los PAISs se mejora enormemente cuando éstos pueden generar predicciones ya que constituyen una herramienta de trabajo muy valorada al gestionar BPs [9] y, por lo tanto, la generación de predicciones se considera crucial para cualquier PAIS [13]. En concreto, las predicciones permiten (1) anticipar problemas, (2) evitar el incumplimiento de restricciones de una manera proactiva, y (3) tomar decisiones sobre prioridades y restricciones al enfrentarse a eventos inesperados, e.g., retrasos. [2].

Por un lado, de acuerdo a [11], las predicciones han de cumplir una serie de requisitos básicos: (1) deben ser altamente precisas, (2) tienen que poder generarse casi instantáneamente, (3) su funcionalidad debe ser fácil de usar, y (4) la generación de las predicciones no debe influir en la eficiencia del PAIS. Por otro lado, existe un conjunto de características deseables que hacen que la generación de predicciones sea aún más compleja: (1) no deben basarse en una sola instancia del proceso, sino que debe considerar múltiples instancias [12,11], (2) no debe sólo predecir medidas temporales, sino que debe extenderse a las múltiples dimensiones de análisis de un BP, e.g., uso de recursos, actividades críticas, etc., y (3) el sistema de predicción debe considerar la flexibilidad de los procesos que existe en entornos cambiantes, e.g., que un recurso no esté disponible inesperadamente, una actividad tarde más de lo esperado, etc. [10]

Cuando se trabaja en entornos altamente cambiantes, los modelos declarativos de BPs son los más convenientes permitiendo especificar *qué* tiene que hacerse en vez de *cómo*. Sin embargo, es en estos entornos donde menos soporte se ofrece para la generación de predicciones [13].

Aunque existen muchas propuestas en el campo de la predicción en BPs [14,13,12,9], no se encuentra ninguno que cubra las características anteriormente mencionadas. Es por ello que el presente trabajo propone un método para generar predicciones para las instancias en ejecución de BPs los cuales han sido especificados de manera declarativa considerando diferentes perspectivas, i.e., tiempo, recursos, datos y control-flow. Para generar estas predicciones, se considera que las decisiones sobre cómo ejecutar el proceso declarativo se toman en base a la optimización de una función objetivo, e.g., ¿cuál es la mejor forma de ejecutar el modelo para que todas las instancias acaben lo antes posible?.

Para ello, en primer lugar, se considera un modelo declarativo de BP multi-perspectiva junto con la especificación de alguna función objetivo. Cuando se solicita una predicción, se recupera el estado de las instancias de dicho proceso que estén en ejecución. A partir de estos elementos, se genera un plan de ejecución optimizado. A partir de este plan, la predicción se calcula evaluando la función deseada a predecir sobre dicho plan.

Esta propuesta presenta varias ventajas:

1. La generación de planes optimizados considera múltiples instancias de procesos así como la asignación de recursos, que son factores de alta importancia para generar predicciones precisas [12,11].

2. El sistema puede generar predicciones relacionadas con diferentes dimensiones, no sólo el tiempo.
3. Ya que la generación de predicciones está basada en el estado actual de las instancias en ejecución, es posible hacer frente a eventos inesperados [11].
4. Se considera la especificación declarativa de BPs como punto de partida otorgando, por tanto, aún más flexibilidad al usuario final.

En la Secc. 2 se presentan algunos conceptos básicos necesarios para entender la propuesta. La Secc. 3 detalla la propuesta para generar predicciones. La Secc. 4 muestra una evaluación de la propuesta. Finalmente, en la Secc. 5 se concluye la propuesta y se muestran las líneas que se plantean como trabajos futuros.

2. Antecedentes

Existen diferentes paradigmas para el modelado de BPs, e.g., imperativo y declarativo. Mientras que los modelos imperativos especifican exactamente *cómo* deben ejecutarse los procesos, los modelos declarativos se centran en *qué* debe hacerse permitiendo la creación de modelos muchos más flexibles.

En esta propuesta se considera el lenguaje declarativo Declare⁴[7,8] para la especificación de los BPs. Los modelos en Declare se componen de actividades y restricciones sobre éstas que restringen el comportamiento deseado.

Como ejemplo, la Fig. 1 muestra un modelo Declare consistente en 4 actividades (i.e., A, B, C y D) donde aparecen, entre otras, las siguientes restricciones: *Exactly(A, 1)*, que indica que la actividad A debe ejecutarse 1 sola vez; *Precedence(C, D)*, que implica que si D se ejecuta, C tiene que haberse ejecutado previamente; *NotResponse(B, D)*, que implica que D no puede ejecutarse después de B. Además, las actividades de dicho modelo Declare van acompañadas de atributos, i.e., *dur* y *res* que indican la duración y los recursos que pueden ejecutar cada actividad.⁵

A la hora de ejecutar un modelo declarativo existen muchas alternativas comparado con un modelo imperativo. A dichas alternativas se les llaman planes de ejecución (cf. Def. 1), cada una de las cuales puede caracterizarse por el valor que alcanza en una determinada función objetivo, e.g., el tiempo de ejecución del plan, el beneficio obtenido, etc.

Definition 1. *Un plan de ejecución $PE = (pId, Acts)$ está compuesto por un identificador (i.e., pId) y un conjunto de actividades (i.e., $Acts$) que se ejecutan sin interrupción. Cada actividad $act \in Acts$ es una tupla $\langle actId, st, et, res \rangle$ donde $actId$ es un identificador único de la actividad, st y et establecen el tiempo de inicio y fin de la actividad en el plan respectivamente, y res identifica el recurso que ejecuta la actividad.*

⁴ Declare es uno de los lenguajes para modelos de procesos declarativos más usados y citados.

⁵ La capacidad de incluir atributos como la duración o la consideración de recursos no está contemplada en la versión inicial de Declare. Sin embargo, existen extensiones del lenguaje que permiten su modelado [4,6].

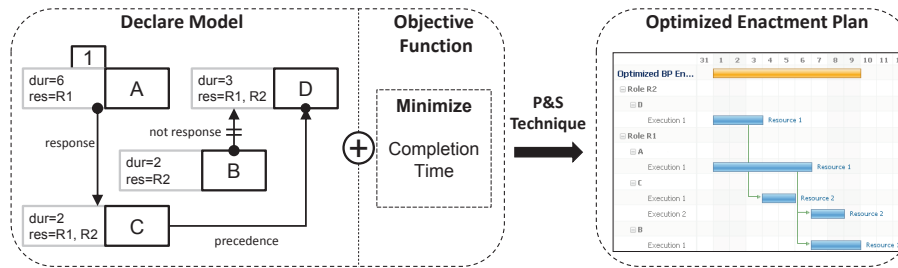


Figura 1. Plan de ejecución optimizado obtenido a partir de un modelo Declare y una función objetivo.

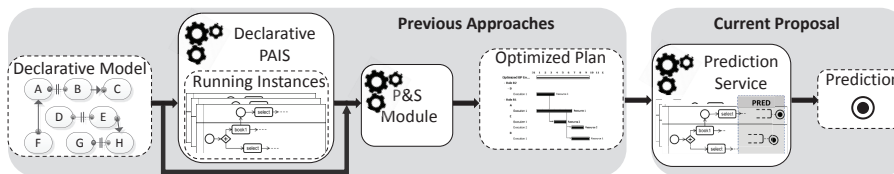


Figura 2. Visión general de la presente propuesta para la generación de predicciones.

Para ser capaces de obtener diferentes planes de ejecución a partir de un modelo declarativo, en nuestra propuesta se aplican técnicas de Planning & Scheduling (P&S). En concreto, se usa P&S para encontrar un plan de ejecución que satisfaga las restricciones existentes a la vez que intenta optimizar alguna función objetivo[5,6]. Los planes de ejecución son comúnmente representados como un diagrama de Gantt[3].

En la Fig. 1 se muestra un plan de ejecución optimizado generado usando técnicas de P&S utilizando como entrada un modelo Declare y una función objetivo, e.g., minimizar el tiempo de completitud del proceso.

3. Generación de Predicciones para Modelos Declarativos

Esta sección describe el método propuesto para la generación de predicciones (cf. Fig. 2) cuyos componentes principales son los siguientes:

- El **modelo declarativo del BP** es un artefacto consistente en un modelo basado en restricciones que es comúnmente creado por un analista tras analizar el contexto del negocio. Dicho modelo incluye las restricciones de control-flow, tiempo, recursos, estimaciones, etc.
- El **PAIS declarativo** es la herramienta encargada de ejecutar las diferentes **instancias del BP** anterior.
- El **módulo de P&S** es una herramienta basada en restricciones que se encarga de generar los artefactos denominados **planes de ejecución optimizados**. Dichos planes son generados considerando el modelo declarativo anterior y el estado de las instancias de procesos que hay en ejecución. De

esta manera se asegura que, por un lado, el plan es conforme a las restricciones existentes en el modelo y, por otro, las instancias en ejecución son reproducibles en el plan.

- El **servicio de predicción** permite evaluar una **predicción** sobre el estado actual de las instancias en ejecución. Para ello, este servicio hace uso de los planes de ejecución previamente generados.

Como se muestra en la Fig. 2, parte de la propuesta ha sido descrita en trabajos anteriores. En concreto, los detalles acerca del lenguaje de modelado declarativo de BPs así como los detalles del módulo de P&S para la generación de planes optimizados se han descrito en [6,5]. La contribución actual se centra en la generación de predicciones a partir de planes de ejecución optimizados.

Con el objetivo de evaluar una predicción sobre las instancias en ejecución del proceso, la herramienta propuesta utiliza: (1) el plan de ejecución (cf. Def. 1) optimizado que es generado por el módulo de P&S y (2) una función de medida (cf. Def. 2) que indica las magnitudes a extraer del plan.

Definition 2. *Sea P un plan de ejecución y T un instante de tiempo. Entonces, una **función de medida** $f(P, T)$ es aquella capaz de producir uno o varios valores relacionados con P en el instante T .*

De manera similar a [13], esta definición puede ser usada para obtener valores tanto predictivos como no predictivos. Por un lado, los valores no predictivos son aquellos en los que sólo se requiere información anterior a T , e.g., el beneficio acumulado hasta dicho instante. Por otro lado, los valores predictivos son aquellos dónde la función mira más allá del instante T , e.g., el beneficio que se obtiene desde dicho instante, el tiempo restante para finalizar, etc.

Cada vez que se solicita una predicción al servicio de predicción, éste recupera el plan de ejecución optimizado más reciente.⁶ A continuación, el servicio de predicción evalúa la función de medida — asociada a la predicción solicitada — sobre el plan recuperado considerando el instante T indicado. El valor devuelto por dicha función es el valor de predicción que devuelve el servicio.

A modo de ejemplo, la función de medida TR calcula el valor predictivo del tiempo restante del plan de ejecución P a partir de T .

$$TR(P, T) = \text{Max}_{act \in P} (et(act)) - T,$$

donde $act \in P$ se refiere a todas las actividades que hay en el plan de ejecución P y $et(act)$ indica el tiempo en el que la actividad act acaba según P .

De la misma manera, la función de medida BA calcula el valor no predictivo del beneficio acumulado en el plan P hasta el tiempo T .

$$BA(P, T) = \sum_{act \in P | et(act) < T} (act.benefit),$$

donde $act.benefit$ indica el valor de un atributo de act llamado *benefit* que estaría definido en el escenario de aplicación.

⁶ Dado que las instancias de procesos están en ejecución, el módulo de P&S tiene que ir actualizando los planes de ejecución a medida que dichas instancias avanzan.

```

1 public interface MeasurementFunction<T>{
2     String getName ();
3     T eval (EnactmetPlan p, float time);
4 }

```

Figura 3. Interfaz para las funciones de medida.

```

...
5:<1,101,4>,<2,102,0>,<3,132,0>,<4,132,3>, ...
6:<1,132,4>,<2,133,0>,<5,146,2>,<6,231,1>, ...
7:<1,212,3>,<2,213,0>,<3,213,0>,<7,340,1>, ...
8:<1,234,4>,<2,235,0>,<3,235,0>,<4,235,3>, ...
...

```

Figura 4. Extracto de la representación textual de un plan de ejecución.

3.1. Detalles de Implementación

En este apartado se dan algunos detalles de implementación del servicio de predicción para ayudar a entender su funcionamiento.

El servicio de predicción se ha implementado en lenguaje Java e incluye tres funciones principales:

1. Evaluar una función de medida que recibe como parámetros (1) un plan de ejecución y (2) una marca de tiempo. El sistema desarrollado contempla un conjunto de funciones de medida fácilmente ampliable. Cada una de estas funciones sigue una interfaz común (cf. Fig. 3) que permite definir un nombre y un comportamiento (cf. métodos *getName* y *eval* respectivamente). Además, el tipo del valor devuelto por la función de medida es configurable (cf. tipo genérico *T* en la interfaz) para cada función de medida.
2. Recuperar un plan de ejecución (cf. Def. 1) a partir de un fichero de texto que incluya la información serializada de éste (cf. Fig. 4). En concreto, estos ficheros guardan, para cada instancia del proceso, una lista de 3-tuplas con (1) el identificador de la actividad, (2) el tiempo de inicio de dicha actividad⁷ y (3) el identificador del recurso que la ejecuta. A modo de ejemplo, el extracto mostrado en la Fig. 4 muestra varias instancias identificadas por 5, 6, 7 y 8 respectivamente. Para la instancia 5, puede verse que empieza con una actividad identificada por 1 en el instante 101 y el recurso encargado de su ejecución es el identificado por 4. Con objetivo de independizar el servicio de predicción de la herramienta de generación de planes —cuyos detalles pueden consultarse en [5]—, esta última está configurada para generar los planes en el formato anterior y almacenarlos en un lugar conocido (e.g., base de datos o sistema de archivos).
3. Ofrecer un servicio RESTful para poder integrar el servicio de predicción en otras sistemas. En concreto, se permite consumir el servicio de predicción usando como parámetros un identificador de la función de medida (i.e., ha de

⁷ Dado que se considera que cada actividad tiene una duración concreta, el tiempo de fin se calcula directamente sumando la duración de la actividad al tiempo de inicio de ésta.



Figura 5. Proceso general seguido en el salón de belleza.

C: 0	T: 9.00	S: Dye
C: 1	T: 9.00	S: CleanAndCut, Dye, Manicure
C: 2	T: 10.30	S: CleanAndCut, Dye, Facial
C: 3	T: 12.15	S: Dye
C: 4	T: 12.15	S: Facial
C: 5	T: 13.00	S: Dye, Facial
C: 6	T: 14.00	S: Manicure, Dye, Facial
C: 7	T: 15.30	S: Facial, Manicure, Dye
C: 8	T: 17.20	S: Manicure
C: 9	T: 17.20	S: CleanAndCut

Figura 6. Ejemplo de fichero de definición de uno de los problemas para la evaluación.

coincidir con el devuelto por *getName* en la interfaz) y la marca de tiempo a considerar en dicha función de medida. La respuesta del servicio incluirá el valor devuelto por el método *eval* de la función de medida seleccionada cuando es aplicada sobre el plan de ejecución más reciente recuperado del almacén.

4. Evaluación

Para evaluar si la propuesta es apropiada para la generación de predicciones, se ha considerado el proceso de atención al público que se sigue dentro de un salón de belleza. En dicho proceso se modela el comportamiento que siguen los usuarios al ser atendidos por los trabajadores del salón cuando se les realizan diferentes servicios. En concreto, en la Fig. 5 se observa la secuencia general donde, en primer lugar, se recibe al cliente, posteriormente se le aplican los servicios deseados y, finalmente, se realiza el cobro de dichos servicios. Dentro de la actividad *Services* se consideran alrededor de 20 actividades que están relacionadas entre ellas y que son relativas a los diferentes servicios ofertados en el salón de belleza, cuyos detalles y su modelado completo en Declare pueden consultarse en [6]. Además, por un lado, el salón considera varios recursos diferentes, cada uno de los cuales está habilitado para realizar algunas de las actividades. Por otro lado, la hora de llegada al salón y los servicios reservados por cada cliente son conocidos de antemano (i.e., los clientes piden las citas con un día de antelación). Cada instancia de este proceso corresponde a un solo cliente.

En total, se han generado 90 problemas diferentes para cada número de clientes (i.e., NC). A modo de ejemplo, en la Fig. 6 se muestra un fichero con un problema consistente en 10 clientes, cada uno con un tiempo de llegada y una lista de servicios deseados⁸. Dado que el tiempo de espera de los clientes

⁸ El conjunto total de los diferentes casos de prueba puede consultarse en <http://azarias.lsi.us.es/Predict/ObjectsBeautySalon.zip>

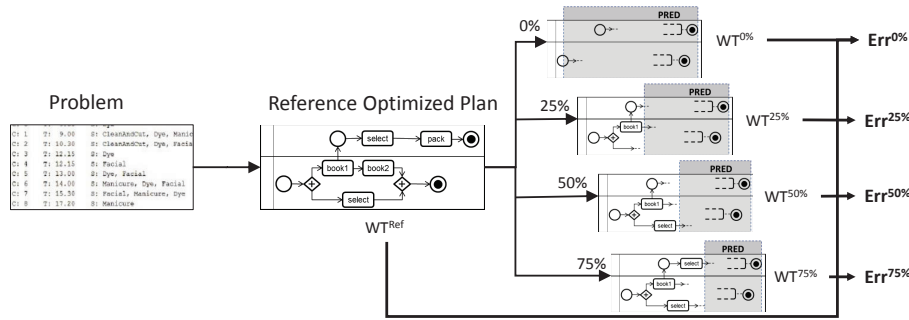


Figura 7. Proceso de evaluación.

es una función crítica en este escenario, se ha tomado dicho valor como función objetivo y como valor a predecir, i.e., la función de medida (cf. Def. 2) que se usará es WT que calcula el valor predictivo del tiempo de espera total en el plan P incluyendo tanto lo anterior como lo posterior a T , i.e., es independiente de T . Dicho tiempo de espera se calcula como:

$$WT(P, T) = \frac{\sqrt{\sum_{c \in C} ((P^{et(c)} - c.T) - (\sum_{b \in c.S} b.estimate))^2}}{C.size},$$

donde C son los clientes, P es el plan de ejecución, $P^{et(c)}$ es cuando el cliente c finaliza en el plan P , $c.T$ es la hora de llegada de c , $c.S$ son los servicios que se aplican a c , and $b.estimate$ es el tiempo estimado para el servicio b — constante para cada servicio—. Como puede verse, en esta función de medida el parámetro T es irrelevante.

La Fig. 7 muestra el proceso de evaluación seguido para cada uno de los problemas. Inicialmente se calcula utilizando el módulo de P&S un plan de ejecución optimizado⁹ que se usará a modo de referencia. El valor del tiempo de espera se calcula para este plan de referencia (i.e., WT^{Ref}). A continuación, se evalúa la predicción (cf. Sec. 3) considerando el estado de las instancias en ejecución al 0%, 25%, 50% y 75% de dicho plan de referencia, i.e., (1) se considera únicamente el porcentaje inicial indicado del plan de referencia, (2) el módulo de P&S calcula el plan de ejecución restante, y (3) el servicio de predicción evalúa la función de medida sobre dicho plan. Obteniéndose así las predicciones $WT^{0\%}$, $WT^{25\%}$, $WT^{50\%}$ y $WT^{75\%}$.

Finalmente, se ha comprobado el error ($Err^X\%$) obtenido en cada una de las predicciones. Este error se ha medido como sigue:

$$Err^X\% = \frac{|WT^X\% - WT^{Ref}|}{Rango},$$

donde $WT^X\%$ es la predicción obtenida para el tiempo de espera, WT^{Ref} es el valor de referencia que se conoce para el valor del tiempo de espera, y $Rango$ es el rango de posibles valores que puede tomar el tiempo de espera, i.e., la diferencia entre el máximo y el mínimo valor¹⁰.

⁹ En <http://azarias.lsi.us.es/Predict/PlansBeautySalon.pdf> pueden observarse dos planes de ejecución para dos problemas diferentes

¹⁰ El valor del $Rango$ se calcula de forma aproximada.

Cuadro 1. Resultados de la evaluación.

NC	Err ^{0%}	Err ^{25%}	Err ^{50%}	Err ^{75%}
10	4,0%	2,8%	0,7%	0,1%
15	14,8%	9,9%	2,9%	0,1%
20	34,4%	26,7%	9,7%	0,7%

La Tabla 1 muestra las medias de los errores que se han obtenido para los conjuntos de problemas de 10, 15 y 20 clientes. Como puede observarse el error obtenido por las predicciones aumenta al incrementarse la complejidad del problema. En el peor de los casos (i.e., $NC = 20$ con el 0% de ejecución) el error cometido es inferior al 35% que, aún no siendo un buen resultado, es asumible por el hecho de que la ejecución del proceso se considera no empezada. No obstante, para la mayoría de los casos el error es inferior al 10% que puede ser considerado como una buena precisión para las predicciones.

5. Conclusiones y Trabajo Futuro

Este trabajo presenta un sistema que permite generar predicciones para instancias de procesos correspondientes a un modelo declarativo de proceso que considera múltiples perspectivas (i.e., tiempo, recurso, datos y control-flow). Para realizar dichas predicciones, el sistema primero genera planes de ejecución optimizados teniendo en cuenta el modelo de proceso, el estado de las instancias que están parcialmente ejecutadas y una función objetivo. A continuación el sistema evalúa sobre el plan de ejecución generado la función de medida asociada a la predicción que se requiere. Este enfoque presenta diversas ventajas en comparación con los trabajos existentes: (1) se consideran múltiples instancias de procesos además de la asignación de recursos, (2) permite adaptarse a circunstancias cambiantes del modelo declarativo, y (3) además de poder predecir el tiempo restante, permite predecir otras medidas relevantes. Sin embargo, dada la naturaleza de la propuesta, ésta es aplicable principalmente en escenarios en donde el comportamiento del sistema persigue la optimización de alguna función objetivo conocida.

La presente propuesta ha sido evaluada sobre instancias de un BP correspondiente a un escenario real. A pesar de la complejidad de los problemas que se consideran en dicho escenario, los resultados indican que el enfoque tomado produce resultados satisfactorios sobre todo para los problemas en los que la ejecución del proceso ya ha empezado. No obstante, la evaluación realizada permite una validación limitada de la propuesta ya que el diseño de la evaluación está deliberadamente orientado al escenario que se ha considerado.

Como trabajo futuro, se propone realizar una validación formal del algoritmo de predicción incluyendo medidas estadísticas que permitan comparar esta propuesta con otras similares. Además, para extender su validez, se han identificado otros escenarios similares donde aplicar este enfoque y analizar sus resultados,

e.g., guías médicas del entorno sanitario. Finalmente, se plantea considerar los datos de ejecuciones pasadas de los procesos como entrada del sistema para poder ajustar el algoritmo de predicción y generar predicciones más precisas.

Agradecimientos: Este trabajo ha sido financiado por el proyecto Pololas (TIN2016-76956-C3-2-R) y la Red SoftPLM (TIN2015-71938-REDT) del Ministerio de Economía y Competitividad.

Referencias

1. M. Dumas, W.M.P. van der Aalst, and A.H. ter Hofstede, editors. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley-Interscience, Hoboken, NJ, 2005.
2. J. Eder and H. Pichler. Duration Histograms for Workflow Systems. In *Proc. IFIP TC8/WG8.1 Working Conference on Engineering Information Systems in the Internet Context*, pages 239–253, 2002.
3. H.L. Gantt. *Work, wages, and profits*. Engineering Magazine Co., 1913.
4. A. Jimenez-Ramirez, I. Barba, C. del Valle, and B. Weber. Generating multi-objective optimized business process enactment plans. In *Advanced Information Systems Engineering*, volume 7908 of *LNC3*, pages 99–115. Springer Berlin Heidelberg, 2013.
5. A. Jimenez-Ramirez, I. Barba, C. Del Valle, and B. Weber. OptBPPlanner: Automatic Generation of Optimized Business Process Enactment Plans. In *Proc. ISD*, pages 429–442. Springer US, 2013.
6. A. Jimenez-Ramirez, I. Barba, Weber, B., and C. Del Valle. Generating optimized configurable business process models in scenarios subject to uncertainty. *Information Software Technology*, 57:571–594, 2015.
7. M. Pestic. *Constraint-Based Workflow Management Systems: Shifting Control to Users*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2008.
8. M. Pestic, M.H. Schonenberg, N. Sidorova, and W.M.P. van der Aalst. Constraint-Based Workflow Models: Change Made Easy. In *OTM Conferences (1)*, pages 77–94, 2007.
9. Mirko Polato, Alessandro Sperduti, Andrea Burattin, and Massimiliano de Leonni. Data-aware remaining time prediction of business process instances. In *Proc. IJCNN*, pages 816–823, 2014.
10. Manfred Reichert and Barbara Weber. *Enabling Flexibility in Process-Aware Information Systems*. Springer Berlin Heidelberg, 2012.
11. H. Reijers. Case prediction in bpm systems: a research challenge. *Journal of the Korean Institute of Industrial Engineers*, 33:1–10, 2006.
12. B. Schellekens. Cycle Time Prediction in Staffware. In *Master's thesis, University of Technology, Eindhoven*, 2009.
13. W.M.P. van der Aalst, M.H. Schonenberg, and M. Song. Time prediction based on process mining. *Inform Syst*, 36(2):450–475, 2011.
14. B.F. van Dongen, R.A. Crooy, and W.M.P. van der Aalst. Cycle Time Prediction: When Will This Case Finally Be Finished? In *Proc. CoopIS 2008*, volume 5331(I), pages 319–336, 2008.
15. Mathias Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007.