

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Redes de Ingeniería

<http://revistas.udistrital.edu.co/ojs/index.php/redes/index>
<http://dx.doi.org/10.14483/udistrital.jour.redes.2016.1.a09>

REDES DE INGENIERÍA
[Rompiendo las barreras del conocimiento]

ARTÍCULO DE INVESTIGACIÓN CIENTÍFICA

Modelo metodológico para programación de tareas en sistemas de servicios: un enfoque de ingeniería de software

Methodologic model to scheduling on service systems: a software engineering approach

Eduyn Ramiro López-Santana¹ Sandro Javier Bolaños Castro² Germán Andrés Méndez Giraldo³

Para citar este artículo: López-Santana, E., Bolaños, S. y Méndez, G. (2016). Modelo metodológico para programación de tareas en sistemas de servicios: un enfoque de ingeniería de software. *Revista Redes de Ingeniería*. 7(1), 93-104. Doi: 10.14483/udistrital.jour.redes.2016.1.a09

Recibido: 12-noviembre-2015 / Aprobado: 21-junio-2016

Resumen

Se presenta una estructura de Ingeniería de Software (IS) para un proyecto de investigación relacionada con la construcción de un prototipo de Sistema Experto (SE) para la programación de tareas en sistemas de servicios (PTSS), empleando metodologías y procesos del desarrollo de software. Se utiliza la metodología de desarrollo de software adaptable para el desarrollo de la arquitectura de software, partiendo de la descripción como un metaproceso de software que caracterizará el proceso de la investigación. Se emplean diagramas de UML (Unified Modeling Language) para proporcionar un modelado visual que describe la metodología del trabajo de investigación, que permite establecer los actores, elementos e interacciones en el proceso de investigación.

Palabras clave: desarrollo de software adaptativo, ingeniería de software, investigación, programación de tareas, sistemas de servicios.

Abstract

This paper presents an approach of software engineering to a research proposal to make an Expert System to scheduling on service systems using methodologies and processes of software development. We use the adaptive software development as methodology for the software architecture based on the description as a software metaprocess that characterizes the research process. We make UML's diagrams (Unified Modeling Language) to provide a visual modeling that describes the research methodology in order to identify the actors, elements and interactions in the research process.

Keywords: adaptive software development, research, scheduling, service systems, software engineering.

-
1. Ingeniero industrial y Especialista en Gestión de Proyectos de Ingeniería, Universidad Distrital Francisco José de Caldas; magíster en Ingeniería Industrial, Universidad de los Andes; estudios de Doctorado en Ingeniería, Universidad Distrital Francisco José de Caldas; docente asistente, Universidad Distrital Francisco José de Caldas. Correo electrónico: erlopezs@udistrital.edu.co
 2. Ingeniero de sistemas y magister en Teleinformática, Universidad Distrital Francisco José de Caldas; PhD. en Informática, Universidad Pontificia de Salamanca; docente asociado, Universidad Distrital Francisco José de Caldas. Correo electrónico: sbolanos@udistrital.edu.co
 3. Ingeniero industrial y especialista en Informática Industrial, Universidad Distrital Francisco José de Caldas; magister en Ingeniería Industrial, Universidad de los Andes; Ph.D. en Ciencias Técnicas, Universidad Central de las Villas. Docente Titular, Universidad Distrital Francisco José de Caldas. Correo electrónico: gmendez@udistrital.edu.co

INTRODUCCIÓN

En un proceso de investigación es necesario definir la manera en que se abordará el problema de interés, los actores que intervienen, los elementos que lo constituyen y las interrelaciones que se generan entre ellos. Todas estas y su comportamiento emergente, permiten llegar a la solución del problema en una investigación exitosa. Sin embargo, este proceso es abordado desde diferentes posturas metodológicas, dependiendo del enfoque, las variables y los fines de la investigación [1]–[4]. Las nuevas tecnologías han incidido de manera significativa en los procesos de investigación, aportando nuevas metodologías y nuevas formas de investigación. Una de estas, es la Ingeniería de Software (IS), vista como un enfoque de modelado genérico que analiza conceptos mediante lenguajes específicos y que emplea los principios de un sistema modular, en especial de procesos, metodologías, patrones, diagramas UML, entre otros, para modelar la investigación como un metaproceso de software [5]–[7].

En este trabajo se va a presentar una estructura de IS para la construcción de un modelo metodológico aplicado a un proyecto de investigación relacionada con la construcción de un prototipo de Sistema Experto (SE) para la PTSS. La PTSS [8]–[12] consiste en determinar en qué orden se deben ejecutar un conjunto de tareas para cumplir con la prestación de un servicio y qué recursos se deben asignar para la ejecución de dichas tareas, buscando satisfacer una medida de desempeño definida; por ejemplo, el tiempo de servicio o de respuesta.

Este artículo está organizado de la siguiente manera: para empezar se presentan los antecedentes relacionados la PTSS; en la siguiente sección se muestra la definición sistema y vocabulario empleado para la construcción del metaproceso; acto seguido se expone la construcción del modelo metodológico; después se muestra el desarrollo de los diagramas de casos de uso, actividades, estados y clases que soportan el metaproceso y se

describen sus interrelaciones. Finalmente se presentan las conclusiones y trabajos futuros de esta investigación.

PROGRAMACIÓN DE TAREAS (PT) EN SISTEMAS DE SERVICIOS (SS)

La PT la define Pinedo [11] como un proceso de toma de decisiones para la asignación de recursos en la ejecución de tareas en un horizonte de tiempo definido buscando optimizar uno o varios criterios de desempeño. Otros autores, comprende la PT como las actividades de control de entrada-salida, análisis de carga, secuenciación y despacho de órdenes de trabajo [13], [12], [9], [10].

En un ambiente de manufactura la PT juega un rol crucial desde que las órdenes son liberadas para ser fabricadas, y por tanto se deben sincronizar todos los recursos necesarios para entregarlas en las fechas establecidas, hasta la ordenación de actividades en una estación de trabajo teniendo en cuenta las decisiones de largo, mediano y corto plazo [14], [8], [10]. En los sistemas de servicios la PT también juega un rol crucial y se puede evidenciar en una gran variedad de problemas, por ejemplo, la asignación de puertas a los aviones o la reserva de sitios de trabajos, equipos u otras instalaciones, mantenimiento, sistemas de salud [15]–[20], [14].

Un servicio se puede definir como el conjunto de actividades realizadas por recursos (máquinas y personas) para satisfacer las necesidades o deseos de personas, mediante una función que transforma el estado inicial de alguno de los recursos del cliente [21]–[27]. Existen varias diferencias de los SS respecto a los sistemas de manufactura [28], [21], [22], [26], [27], como: los servicios son intangibles y raramente almacenables; la demanda no puede posponerse (no hay posibilidad de espera o diferir a periodos futuros); la utilización de equipos para la prestación de servicios suele ser menos importante que el empleo de personas; en los servicios no se transfiere la propiedad del producto, mientras

que en manufactura sí se transfiere; la demanda es difícil de prever, por lo que está sujeta a variabilidad y condiciones dinámicas que implica reprogramación y ajustes; y, son necesarios recursos adicionales al proceso para completar un servicio, por ejemplo transporte disponible para entregar un servicio o para el manejo de los equipos necesarios para el mismo.

Por otro lado, es muy frecuente ver en los SS (aunque también en algunos sistemas de manufactura) la dependencia de la experticia y conocimiento de las personas, donde las organizaciones suelen delegar las actividades de planeación y programación (proceso de toma de decisiones) en sistemas de alta complejidad a personal con experiencia, confiando en su intuición y conocimiento. Sin embargo, esto se ve limitado cuando se tiene gran cantidad de información y relaciones que hacen complejo el entendimiento del sistema. Además, a menudo se necesita un tiempo largo para generar (casi siempre de forma manual) un plan y programa factible. Por tanto, es necesario que el SS utilice de una manera ordenada y sistemática el conocimiento individual de su personal experimentado para resolver de manera colectiva un problema complejo como lo es la PT.

En la literatura se evidencian problemas relacionados con la PT relacionados con talento humano y la utilización de conocimiento representado en la experticia del personal como los presentados en la tabla 1. En los cuales se han empleado técnicas

de programación matemática, técnicas heurísticas y metaheurísticas, y algunas de inteligencia artificial, aunque con menos desarrollo respecto a los sistemas de manufactura, y trabajos que emplean modelos basados en inteligencia individual y colectiva para ayudar a resolver el problema de PT, generando impactos significativos en las medidas de desempeño de los sistemas o en el funcionamiento de los procedimientos diseñados para resolver el problema.

DEFINICIÓN DEL PROBLEMA Y VOCABULARIO

Problema de investigación

Dadas las características de los sistemas de servicios mencionadas en la sección anterior y que los métodos tradicionales que se han planteado para resolver el problema de PT en manufactura no responden a todas estas, surge la necesidad de establecer métodos y herramientas para la función de PTSS motivada por cuatro aspectos fundamentales:

1. la necesidad de tratar sistemas complejos como lo constituyen los propios del sector de servicios por sus características ya mencionadas,
2. la necesidad de desarrollar métodos de PT robustos y reactivos, es decir que puedan nueva información al sistema y responder de manera eficiente a estos cambios (característica dinámica de los SS),

Tabla 1. Aplicaciones de PTSS.

Problema	Autores
Programación de horarios de trabajo	[29], [30]
Reserva y asignación de salas, y programación de itinerarios.	[17], [31], [32]
programación de operaciones de mantenimiento	[33]–[36]
Inteligencia artificial para PT en sistemas restringidos	[37]
Sistemas Cooperativos entre máquina-hombre para PT en manufactura	[38]
Análisis de aspectos humanos, tecnológicos y organizacionales en la PT	[39]
Métodos para la PT en un ambiente dinámico basados en inteligencia social y colectiva	[40]–[42]

Fuente: elaboración propia.

3. integrar estos métodos de PT con el conocimiento del sistema a través del desarrollo de modelos basados en inteligencia que permitan utilizar la experticia individual de los actores en el sistema y resolver el problema de manera colectiva, y
4. la necesidad de incorporar múltiples criterios de decisión para el proceso de toma de decisiones en la PTSS.

Este problema plantea dos elementos distintivos (figura 1): el análisis de los sistemas de servicios enmarcado en la ciencia de los servicios definido por el problema de uso de conocimiento en sistemas de servicios como los presentados por [23], [43], [44] y la programación de tareas (Problema de PT), que es un problema de optimización combinatoria enmarcado dentro de los problemas NP-hard [8], [10], [11]. Con estos dos elementos se pretende estudiar el problema de PTSS mediante métodos basados en modelación matemática, gestión del conocimiento e inteligencia (Integración de Enfoques).

Definición de vocabulario

A continuación se expone la notación que se empleará para definir un SS basados en [22] quienes proponen una notación (R, S, I, F) , donde R es el conjunto de recursos que son relevantes para recibir el servicio (tanto por el cliente como por el

proveedor), S es el conjunto de estados al que un recurso perteneciente a R puede entrar, I es una función que describe el estado inicial del sistema servicio, y F una función que describe el estado final del SS. Con esta aproximación se intenta establecer una definición formal de un SS como una meta-proceso para facilitar el tratamiento de la información.

De esta forma, se podría realizar la abstracción de un SS simple con dos recursos: un cliente (C) y un proveedor (P), entonces el conjunto $R = \{C\}$, quien es el que recibe el servicio. El conjunto de estados $S = \{s_1, s_2, \dots, s_n\}$ de todos los posibles estados en los que puede estar y entrar el recurso C . El conjunto de estados iniciales $I = \{\emptyset\}$ significando que el cliente no se ha atendido, y el conjunto de estados finales $F = \{\text{Satisfecho}, \text{No Satisfecho}\}$ con respecto a una medida de desempeño d . Esta medida puede ser, por ejemplo, el tiempo de respuesta medida en la duración del servicio más el tiempo de espera. La variable d puede tener varias dimensiones, es decir, varias medidas de desempeño. Para la PT existen varias medidas de desempeño trabajadas en la literatura, algunas de estas se pueden ver en la tabla 2. De esta manera se extendió esta notación, involucrando un parámetro (α, d) referente al método de PTSS. La tabla 3 presenta un compendio de posibles técnicas para la PT. La notación se describe en (1), en la cual se tiene un $SS = (R, S, I)$ al cual se le aplica un

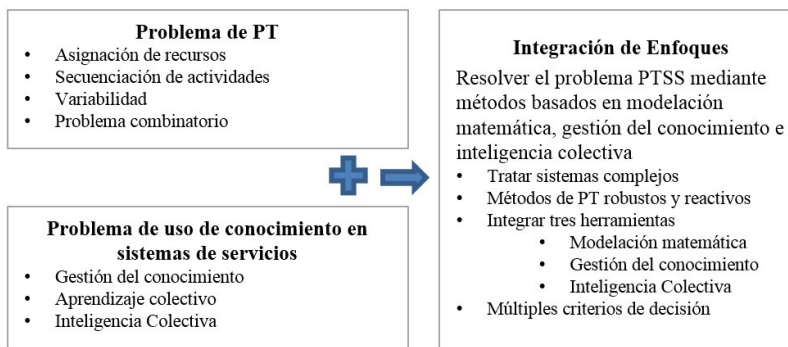


Figura 1. Estructura del problema de PTSS.

método de PT α buscando satisfacer una medida (o medidas) de desempeño d . para pasar a un estado final $SS = (R, S, F)$.

$$(R, S, I) \xrightarrow{(\alpha, d)} (R, S, F) \quad (1)$$

Con esta definición, se puede establecer una hipótesis de investigación como: la implementación de un procedimiento $\alpha = \{SE\}$ que permite mejorar d para que el estado de SS cambie de $I = \{\emptyset\} \rightarrow F = \{\text{"Satisfecho"}\}$. La propuesta de investigación consiste en probar diferentes métodos α y diferentes medidas de desempeño d , como los dados en (2) y (3).

$$\alpha = \{\text{Técnicas tradicionales, Sistema Basado en Reglas, Sistema Experto}\} \quad (2)$$

$$d = \{\text{Tiempo de respuesta, Costo del servicio,}\} \quad (3)$$

Tabla 2. Medidas de desempeño en PT.

Medida	Autores
Minimizar costos	[38], [12], [9], [11], [10]
Minimizar retrasos en los trabajos	[38], [12], [9], [11]
Minimizar tiempo de terminación	[38], [12], [9], [11]
Maximizar capacidad	[38], [10]
Maximizar satisfacción cliente	[38], [12], [9], [11]
Maximizar moral del operario	[38], [10]
Maximizar Throughput	[38], [12], [9], [11]
Minimizar el Trabajo en proceso	[38], [12], [9], [11]

Fuente: elaboración propia.

Tabla 3. Técnicas de solución de problemas en PT

Técnica	Ejemplos
Programación Matemática	Programación Lineal, Entera, Dinámica.
Heurísticas	Reglas de atención: Tiempo de procesamiento mínimo, Tiempo de holgura de procesamiento mínimo, Regla de Johnson, etc.
Metaheurísticas	Búsqueda Local y en vecindarios, Recocido simulado, Algoritmos Genéticos, Búsqueda de enjambres
Simulación	Simulación de Eventos discretos, Simulación por agentes
Inteligencia Artificial	Sistemas Basados en Reglas, Sistemas Expertos, Lógica Difusa, Redes neuronales, Colonia de hormigas

Fuente: elaboración propia.

CONSTRUCCIÓN DEL MODELO METODOLÓGICO

A continuación se expone la construcción de un modelo metodológico para la PTSS basado en IS a partir del análisis de procesos o actividades concernientes al proceso de investigación. La IS se puede emplear como soporte metodológico y descriptivo. El modelamiento se realizará mediante artefactos de software en un ambiente centrado en procesos buscando demostrar su funcionalidad en la representación [45].

En la figura 2 se presenta la construcción de este modelo metodológico de cuatro fases. En la primera fase se debe definir el problema de investigación

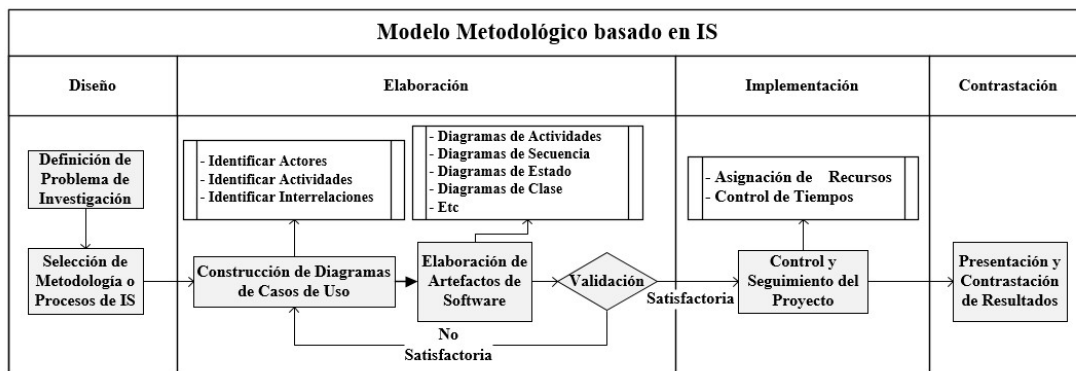


Figura 2. Construcción del modelo metodológico.

y seleccionar la metodología o proceso de desarrollo de software que más se ajuste al problema. Dentro de estas se pueden seleccionar, *Cascada*, *RUP*, *spiral*, *big bang*, *ASD*, entre otras [46]–[48], [7].

Por otro lado, en la fase de Elaboración se hará uso de una herramienta que permite definir un ambiente centrado en procesos denominado *Coloso*. Esta plataforma se compone por varios lenguajes como *archimate* en su capa de arquitectura, UML en la capa de diseño y Java en la capa de implementación [49]. En esta fase se debe representar las interacciones entre los elementos y actores del proceso de investigación, siendo un proceso iterativo sobre la metodología o el proceso seleccionado en la fase inicial de diseño.

Una vez validada la arquitectura, se tiene la fase de implementación, en la cual se realiza un control y seguimiento del proyecto mediante el manejo de recursos y tiempo. Finalmente, se tiene la fase de contrastación donde se deben presentar y contrastar los resultados. Estas dos etapas, no se presentarán en el desarrollo de este documento debido a que corresponden a momentos posteriores a la consecución de resultados del proyecto de investigación específico.

DESARROLLO Y RESULTADOS

En esta sección se mostrará el resultado de esta propuesta para las fases de diseño y elaboración de la figura 2, en el desarrollo de la propuesta de investigación como una arquitectura de IS. La figura 3 (a) muestra la estructura para el diseño del prototipo de sistema experto basada como un metaproceto en la herramienta de desarrollo Coloso, [49].

Como estrategia de modelado para el manejo del ciclo del software se utiliza el paradigma de los procesos ágiles que ofrecen flexibilidad y colaboración, además de un desarrollo de software iterativo e incremental [50] en contraste con los procesos de cascada [46] que son lineales y se basan en metodologías de planificación e implementación.

Para el proceso de gestión se abordará la estrategia mediante una metodología de desarrollo de software adaptativo (ASD, Adaptive Software Development) en las figuras 3 y 4. La metodología ADS fue planteada por Jim Highsmith y Sam Bayer a comienzos de 1990 [51]. Se basa en el principio de adaptación continua a circunstancias cambiantes dentro de un paradigma de especular-colaborar-aprender [52], [53]. Se seleccionó UML como plataforma estándar en los flujos de trabajo para las fases de iniciación del proyecto, plan adaptativo del ciclo, ingeniería de componentes y revisión de calidad [54]. La fase de QA Final y Entrega del proceso no se presenta, ya que corresponde a una fase de implementación para el control seguimiento del proyecto.

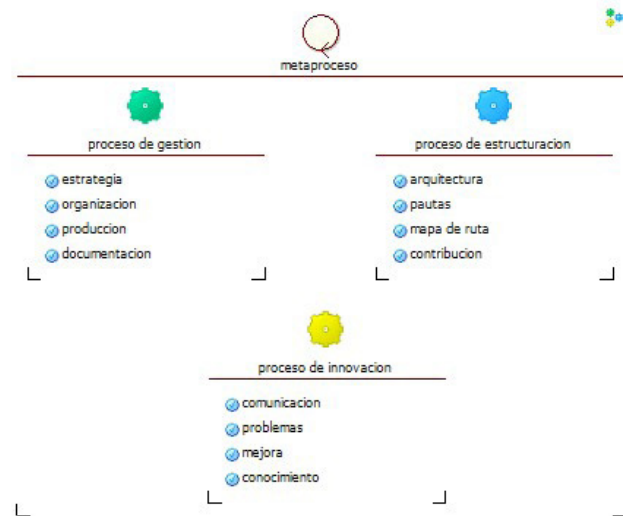


Figura 3. Modelo de software metaproceto.

Fuente: [49].

Iniciación del proyecto

Para la fase de iniciación del proyecto se plantea el desarrollo de la investigación titulado como “Tesis”, en el cual se tienen tres actores (tesista, director y jurado) que tiene influencia sobre las actividades de formulación, desarrollo y contrastación como lo muestra la figura 5.

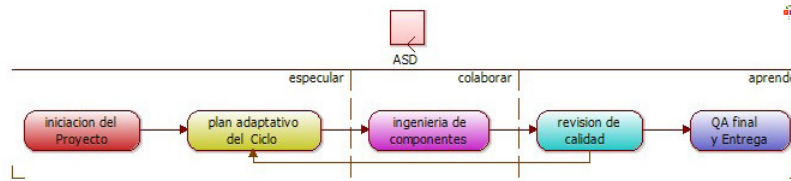


Figura 4. Modelo ASD para estrategia, vista de gestión.

Fuente: [49].

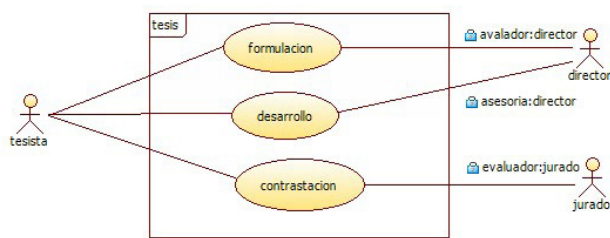
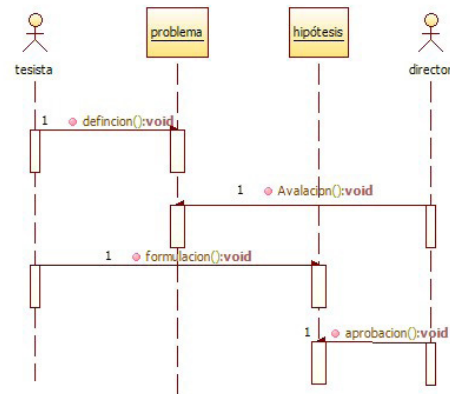


Figura 5. Diagrama de casos de uso: problema, fase de iniciación del proyecto.

Fuente: [49].



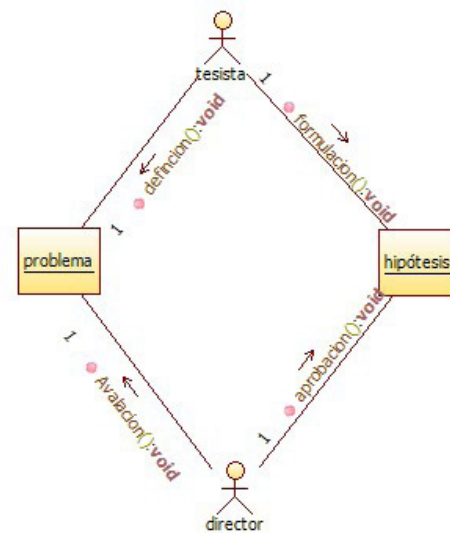
(a)

Plan adaptativo del ciclo

En la figura 6 (a) se desarrolla la actividad de formulación, en la cual interviene el tesisista en el planteamiento del problema y la hipótesis de investigación que se buscara contrastar. Mientras que el director tiene el rol de aprobación de estas. La figura 6 (b) presenta la misma estructura vista como un diagrama de comunicación.

Ingeniería de componentes y de revisión de calidad

En la figura 7 se muestra el diagrama de la actividad de desarrollo, la cual detalla en particular las fases de construcción del sistema experto para el objeto de estudio. En este diagrama se muestra la interacción entre el Director en su rol de revisor para posteriormente darle el aval; y el tesisista para el desarrollo de todas las actividades enmarcadas en el prototipo y la revisión de calidad.



(b)

Figura 6. Formulación, Fase de Plan adaptativo del ciclo. (a) Diagrama de workflow, (b) Diagrama de Comunicaciones.

Fuente: [49]

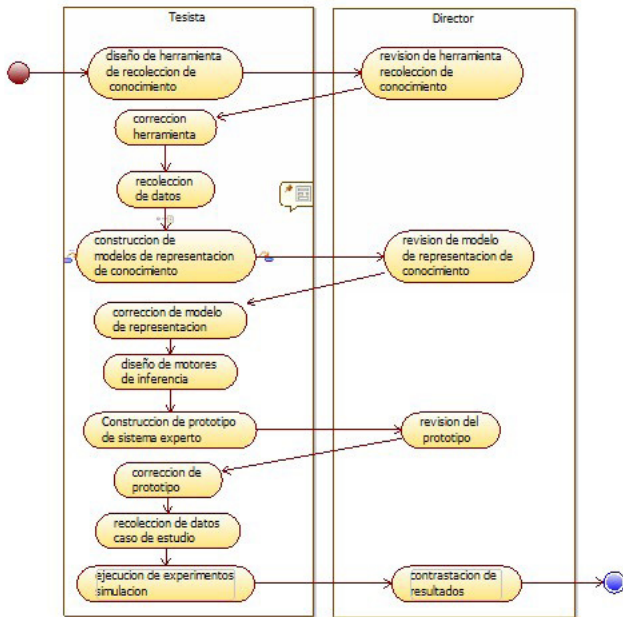


Figura 7. Diagrama de actividades: Desarrollo, fase de ingeniería de componentes.
Fuente: [49].

En esta fase de ingeniería de componentes se plantea la actividad en donde se contrastan (figura 8) los resultados obtenidos con la realidad y se discuten con el actor jurado para su comprobación. Esta actividad tiene un insumo que se realiza mediante la comprobación de la hipótesis de investigación basada en un diagrama de estados (figura 9) y una clase (figura 10).

La hipótesis x e y debe plantearse de acuerdo a las estructuras establecidas en la sección 3.2, de acuerdo a los métodos a y las medidas de desempeño d , en este caso se planteó una estructura general que se puede modificar con algún patrón de desarrollo de software, como el denominado “fabrica” que permite generar múltiples escenarios [55]–[57].



Figura 8. Diagrama de secuencia: Contratación, Fase de revisión de calidad.
Fuente: [49].

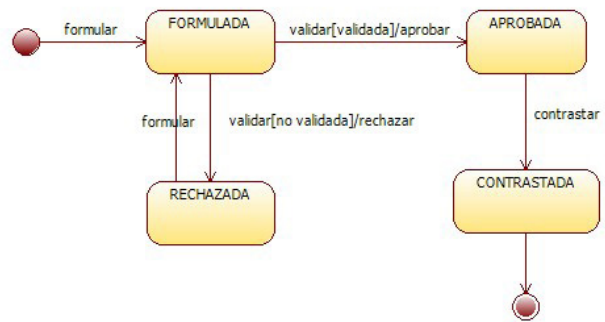
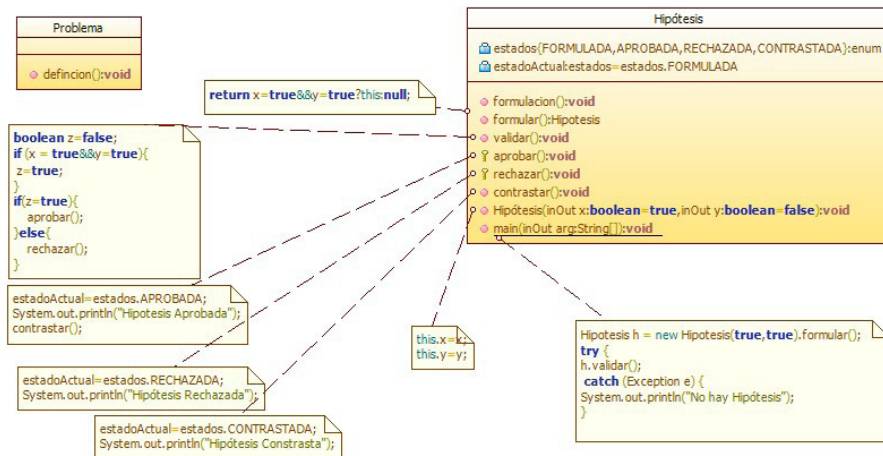


Figura 9. Diagrama de estados: Hipótesis, Fase de ingeniería de componentes.
Fuente: [49].



Fuente: [49]. **Figura 10.** Clase: Hipótesis, fase de ingeniería de componentes.
Fuente: [49].

CONCLUSIONES

Con este trabajo se muestra como la IS permite representar la metodología de investigación mediante el uso de diagramas de lenguajes formales y objetos, que representan los procesos y como los actores influyen en ellos dentro de una fase de estrategia de desarrollo. Cada una de las diferentes interacciones entre los actores y elementos de la investigación se logró establecer, en las tres actividades definidas: formulación, desarrollo y contrastación.

De esta manera, se puede establecer que la IS es un enfoque que permite abstraer y representar, mediante un lenguaje formal, la metodología de un proceso de investigación y poder emplear como un instrumento para la planeación y el control del proyecto de investigación mediante la posible incorporación del consumo de recursos y control de cumplimiento de actividades y resultados. Esta puede ser una fase posterior para trabajos futuros con esta propuesta de modelado de un proceso de investigación. Se utilizó la estrategia de modelado para el manejo del ciclo del software dentro del paradigma de los procesos ágiles que ofrecen flexibilidad y colaboración, además de un desarrollo de software iterativo e incremental [50], utilizando la metodología ASD [51] para el desarrollo de los módulos específicos de programación de tareas, así como trabajo futuro se puede emplear un enfoque de modelos de dominio específico como lo plantean [58].

AGRADECIMIENTOS

Este trabajo es parcialmente soportado por el Centro de Investigaciones y Desarrollo Científico de la Universidad Distrital Francisco José de Caldas (Colombia) bajo el proyecto No. 2-602-468-14. Los autores desean agradecer a los revisores anónimos por sus comentarios y sugerencias para la mejora de este artículo.

REFERENCIAS

- [1] M. T. y Tamayo, *Metodología formal de la investigación científica*. México: Editorial Limusa, 1980.
- [2] M. Bunge, *La investigación científica: su estrategia y su filosofía*. Buenos Aires: Siglo XXI, 2000.
- [3] E. A. R. Moguel, *Metodología de la Investigación*. Tabasco: Univ. J. Autónoma de Tabasco, 2005.
- [4] R. H. Sampieri, C. F. Collado, and P. B. Lucio, *Metodología de la investigación*. México: McGraw-Hill/Interamericana, 2006.
- [5] S. Bolaños and S. Lopez, "LMPS como propuesta alterna a BPMN para el modelado de proceso de software". *Tecnura: Tecnología y Cultura Afirmando el Conocimiento*, no. 34, pp. 157–170, 2012.
- [6] C. Ghezzi, D. Mandrioli, and M. Jazayeri, *Fundamentals of software engineering*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1991.
- [7] I. Sommerville, *Software Engineering*, 9th ed. USA: Pearson Education, 2011.
- [8] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz, *Handbook on Scheduling: From Theory to Applications*. New York: Springer, 2007.
- [9] J. H. Heizer and B. Render, *Principles of operations management*. Boston: Pearson Education, 2010.
- [10] G. Méndez-Giraldo, *Programación de tareas-Scheduling*. Bogotá, D. C.: Universidad Distrital Francisco José de Caldas, 2011.
- [11] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. New York: Springer, 2012.
- [12] R. G. Schroeder, S. M. Goldstein, and M. J. Rungtusanatham, *Operations management: contemporary concepts and cases*. Boston: McGraw-Hill Irwin, 2008.
- [13] S. O. Duffuaa and K. S. Al-Sultan, "Mathematical programming approaches for the management

- of maintenance planning and scheduling". *Journal of Quality in Maintenance Engineering*, vol. 3, no. 3, pp. 163–176, Sep. 1997.
- [14] M. L. Pinedo, "Manufacturing Models". in *Planning and Scheduling in Manufacturing and Services*, New York: Springer, 2009, pp. 19–35.
- [15] S. Anily, C. A. Glass, and R. Hassin, "The scheduling of maintenance service". *Discrete Applied Mathematics*, vol. 82, no. 1–3, pp. 27–42, Mar. 1998.
- [16] H. Date and T. Matsuo, "Effects of at-home nursing service scheduling in multiagent systems". *New Challenges in Applied Intelligence Technologies*, vol. 134, N. T. Nguyen and R. Katarzyniak, Eds. Springer Berlin Heidelberg, 2008, pp. 245–254.
- [17] M. L. Pinedo, "Interval Scheduling, Reservations, and Timetabling". *Planning and Scheduling in Manufacturing and Services*, Springer New York, 2009, pp. 207–230.
- [18] M. L. Pinedo, "Planning and Scheduling in Health Care". *Planning and Scheduling in Manufacturing and Services*, Springer New York, 2009, pp. 291–316.
- [19] R. M'Hallah and A. Alkhabbaz, "Scheduling of nurses: A case study of a Kuwaiti health care unit". *Operations Research for Health Care*, vol. 2, no. 1–2, pp. 1–19, Mar. 2013.
- [20] K.-C. Huang, T.-C. Huang, M.-J. Tsai, and H.-Y. Chang, "Moldable Job Scheduling for HPC as a Service". *Future Information Technology*, J. J. (Jong H. Park, I. Stojmenovic, M. Choi, and F. Xhafa, Eds. Springer Berlin Heidelberg, 2014, pp. 43–48.
- [21] S. L. Vargo and R. F. Lusch, "Service-dominant logic: Continuing the evolution". *Journal of the Academy of Marketing Science*, vol. 36, no. 1, pp. 1–10, Mar. 2008.
- [22] M. Böttcher and K.-P. Fähnrich, "Service Systems Modeling: Concepts, Formalized Meta-Model and Technical Concretion". *The Science of Service Systems*, H. Demirkan, J. C. Spohrer, and V. Krishna, Eds. Springer US, 2011, pp. 131–149.
- [23] H. Demirkan, J. C. Spohrer, and V. Krishna, "Introduction of the Science of Service Systems". *The Science of Service Systems*, H. Demirkan, J. C. Spohrer, and V. Krishna, Eds. Boston, MA: Springer US, 2011, pp. 1–11.
- [24] S. I. Gass and M. C. Fu, Eds., "Service Systems". *Encyclopedia of Operations Research and Management Science*, Springer US, 2013, pp. 1392–1392.
- [25] A. Polyvyanyy and M. Weske, "Flexible Service Systems". *The Science of Service Systems*, H. Demirkan, J. C. Spohrer, and V. Krishna, Eds. Springer US, 2011, pp. 73–90.
- [26] J. C. Spohrer, H. Demirkan, and V. Krishna, "Service and Science". *The Science of Service Systems*, pp. 325–358, Jan. 2011.
- [27] S. Bhatnagar, H. Prasad, and L. Prashanth, "Service Systems". in *Stochastic Recursive Algorithms for Optimization*, Springer London, 2013, pp. 225–241.
- [28] M. L. Pinedo, "Service Models". *Planning and Scheduling in Manufacturing and Services*, Springer New York, 2009, pp. 37–49.
- [29] D. Barrera, N. Velasco, and C. A. Amaya, "A network-based approach to the multi-activity combined timetabling and crew scheduling problem: Workforce scheduling for public health policy implementation". *Computers and Industrial Engineering*, vol. 63, no. 4, pp. 802–812, Dec. 2012.
- [30] J. Zhibin and H. Qixiang, "A Service-based Method to Generate Shuttle Bus Timetable in Accordance with Rail Transit Timetable". *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 1890–1897, Nov. 2013.
- [31] M. L. Pinedo, "Scheduling and Timetabling in Sports and Entertainment". *Planning and Scheduling in Manufacturing and Services*, Springer New York, 2009, pp. 231–254.
- [32] M. L. Pinedo, *Planning and Scheduling in Manufacturing and Services*, 2nd ed. Springer, 2009.
- [33] A. A. Kovacs, S. N. Parragh, K. F. Doerner, and R. F. Hartl, "Adaptive large neighborhood

- search for service technician routing and scheduling problems". *Journal of Scheduling*, vol. 15, no. 5, pp. 579–600, Oct. 2012.
- [34] P. Do Van, A. Barros, C. Bérenguer, K. Bouvard, and F. Brissaud, "Dynamic grouping maintenance with time limited opportunities". *Reliability Engineering and System Safety*, vol. 120, pp. 51–59, 2013.
- [35] A. Goel and F. Meisel, "Workforce routing and scheduling for electricity network maintenance with downtime minimization". *European Journal of Operational Research*, vol. 231, no. 1, pp. 210–228, Nov. 2013.
- [36] E. Remy, F. Corset, S. Despréaux, L. Doyen, and O. Gaudoin, "An example of integrated approach to technical and economic optimization of maintenance". *Reliability Engineering and System Safety*, vol. 116, pp. 8–19, Aug. 2013.
- [37] H. Atabakhsh, "A survey of constraint based scheduling systems using an artificial intelligence approach". *Artificial Intelligence in Engineering*, vol. 6, no. 2, pp. 58–73, Apr. 1991.
- [38] G. Méndez-Giraldo, *Sistemas Cooperativos Asistidos para la Programación de la Producción en la Industria Manufacturera Colombiana*, 1st ed. Bogotá, Colombia: Universidad Distrital Francisco José de Caldas, Centro de Investigaciones y Desarrollo Científico, 2001.
- [39] M. Berglund and J. Karlton, "Human, technological and organizational aspects influencing the production scheduling process". *International Journal of Production Economics*, vol. 110, no. 1–2, pp. 160–174, Oct. 2007.
- [40] A. Madureira, I. Pereira, and N. Sousa, "Collective intelligence on dynamic manufacturing scheduling optimization". *Proceedings 2010 IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications, BIC-TA 2010*, 2010, pp. 1693–1697.
- [41] I. Pereira and A. Madureira, "Self-Optimization module for Scheduling using Case-based Reasoning". *Applied Soft Computing*, vol. 13, no. 3, pp. 1419–1432, Mar. 2013.
- [42] A. Madureira, I. Pereira, P. Pereira, and A. Abraham, "Negotiation mechanism for self-organized scheduling system with collective intelligence". *Neurocomputing*, vol. 132, pp. 97–110, May 2014.
- [43] D. Ing, "Service Systems in Changing Paradigms; An Inquiry Through the Systems Science". *The Science of Service Systems*, H. Demirkan, J. C. Spohrer, and V. Krishna, Eds. Springer US, 2011, pp. 267–296.
- [44] D. McFarlane, "An Engineering Perspective on Service Science". *The Science of Service Systems*, H. Demirkan, J. C. Spohrer, and V. Krishna, Eds. Springer US, 2011, pp. 257–266.
- [45] S. J. Bolaños Castro, V. H. Medina García, and J. S. Carrión, "Marco conceptual para el metaproceto de desarrollo de software". *Ninth LACCEI Latin American and Caribbean Conference (LACCEI'2011)*, 2011.
- [46] W. W. Royce, "Managing the development of large software systems". *proceedings of IEEE WESCON*, 1970, vol. 26.
- [47] B. W. Boehm, "A Spiral Model of Software Development and Enhancement". *Computer*, vol. 21, no. 5, pp. 61–72, May 1988.
- [48] R. Pressman, *Ingeniería de Software*, 7th ed. MCGRAW-HILL, 2010.
- [49] B. C. Sandro Javier, G. C. Rubén, S. M. Oscar, P. E. Jordán, M. G. Victor Hugo, S. J. Bolaños, R. G. Crespo, O. Sanjuan Martinez, J. P. Espada, and V. H. Medina García, "COLOSO A Development Environment Centered Process and Intent". 2012, pp. 1–6.
- [50] A. Patel, A. Seyfi, M. Taghavi, C. Wills, L. Na, R. Latih, and S. Misra, "A comparative study of agile, component-based, aspect-oriented and mashup software development methods". *Technicki vjesnik / Technical Gazette*, vol. 19, no. 1, pp. 175–189, 2012.
- [51] James A. Highsmith, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House Publishing Company, 2000.

- [52] M. Alnoukari, Z. Alzoabi, and S. Hanna, "Applying adaptive software development (ASD) agile modeling on predictive data mining applications: ASD-DM methodology". *2008 International Symposium on Information Technology*, 2008, vol. 2, pp. 1–6.
- [53] S. Singh and I. Chana, "Introducing Agility in Cloud Based Software Development through ASD". *International Journal of u- and e- Service, Science and Technology*, vol. 6, no. 5, pp. 191–202, Oct. 2013.
- [54] P. Dan and P. Neil, *UML 2.0 in a Nutshell*. 2005.
- [55] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, 1994.
- [56] S. J. Bolaños Castro, R. González Crespo, and V. H. Medina García, "Antipatterns: a compendium of bad practices in software development processes". *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 1, no. 4, pp. 41–46, 2011.
- [57] S. J. Bolaños Castro, R. Gonzalez Crespo, and V. H. Medina Garcia, "Patterns of Software Development Process". *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 1, no. 4, p. 33, 2011.
- [58] C. E. Montenegro Marín, J. M. Cueva Lovelle, O. Sanjuán Martínez, and P. A. Gaona Garcia, "Desarrollo de un lenguaje de dominio específico para sistemas de gestión de aprendizaje y su herramienta de implementación 'KiwiDSM' mediante ingeniería dirigida por modelos". *Ingeniería*, vol. 15, no. 2. pp. 67–81, 2010.

