

Una Propuesta PDDL para la Planificación de la Reparación como Proceso de Negocio

Irene Barba, Carmelo Del Valle

Universidad de Sevilla

{irenebr,carmelo}@us.es

Resumen

Hoy en día, uno de los aspectos más críticos en el proceso de desarrollo de software es la eficiencia. Para mejorarla, se pueden aplicar técnicas de planning y scheduling a la planificación del desarrollo de software, considerando razonamiento temporal y sobre recursos. Por otro lado, los sistemas de gestión de procesos de negocio permiten gestionar de manera adecuada el desarrollo de software, ya que éste se puede modelar como un proceso de negocio. Este trabajo presenta una propuesta para gestionar adecuadamente las etapas de modelado y ejecución de procesos de negocio que requieren decisiones de planning (selección y ordenación parcial de las actividades que se van a ejecutar) y scheduling (asignación de recursos y ordenación total). La principal contribución de este trabajo, es que tanto las decisiones de planning como las de scheduling se toman en tiempo de ejecución, dando lugar a una gestión más eficiente y flexible de los procesos de negocio. A modo de ejemplo, el problema de la planificación de la reparación (problema complejo y representativo que incluye planning y scheduling) se gestiona con la propuesta actual. Para resolver este problema, se propone una especificación PDDL 2.1.

1. Introducción

Actualmente, la mayoría de organizaciones necesitan adaptarse a las nuevas condiciones comerciales, al mismo tiempo que responder a las presiones de un mercado competitivo, por lo que existe un creciente interés en la gestión

eficiente de procesos de negocio (Business Process, BP). Esto puede aplicarse también a las organizaciones de desarrollo software. De una forma similar, la gestión de BP (BP Management, BPM) hace uso de métodos, técnicas y software para el diseño, la ejecución, el control y el análisis de los procesos, teniendo en cuenta el equipo de trabajo, las organizaciones, los documentos, y otras fuentes de información [1]. Por otro lado, los sistemas BPM (BPMS) incluyen las herramientas para la gestión de los BP. Los BPMS son capaces de gestionar el proceso de desarrollo de software, ya que dicho proceso puede modelarse como un BP. En la mayoría de los casos, los BPMS ignoran el tratamiento de recursos en la fase de diseño, considerándolos sólo en tiempo de ejecución. Esto puede no resultar adecuado cuando diferentes actividades pueden requerir recursos limitados en periodos solapados de tiempo.

Por otro lado, los problemas de scheduling [4] vienen caracterizados por un conjunto de tareas relacionadas por restricciones temporales, donde la ejecución de cada tarea requiere el uso de recursos (compartidos y limitados), de forma que dichas tareas pueden competir por el uso de dichos recursos. En este tipo de problemas, el objetivo es encontrar un plan factible de forma que se satisfagan tanto las restricciones temporales como las de recurso, normalmente optimizando una o varias funciones objetivo. En el campo de la planificación en la Inteligencia Artificial (AI planning) [8], las tareas que se deben ejecutar no están establecidas a priori. La especificación de los problemas de planning, en general, incluye el estado inicial, el objetivo a alcanzar, y

un conjunto de acciones que se pueden aplicar a un estado para alcanzar otro. En este tipo de problemas, es necesario seleccionar un conjunto adecuado de acciones que deben ejecutarse en un orden correcto para alcanzar un objetivo, en general optimizando algunas métricas y permitiendo paralelismo en la ejecución de tareas.

Actualmente, existe un gran interés en la integración de planning y scheduling (P&S) debido a la gran cantidad de problemas reales que engloban ambas problemáticas [3], como el mantenimiento y la planificación de la reparación [14]. Los grafos And/Or [9] se pueden utilizar como representación base para muchos de los problemas que involucran P&S, incluyendo el problema de la planificación de la reparación, que se estudia en el presente trabajo. Para resolver estos problemas, se han aplicado distintas técnicas de diversa naturaleza, desde programación matemática hasta una gran variedad de métodos de inteligencia artificial [12]. Por otro lado, muchos problemas pueden perseguir la optimización de múltiples funciones objetivo que pueden entrar en conflicto, y que deben ser consideradas al mismo tiempo. En los problemas de optimización multi-objetivo [6], normalmente no existe una única solución, sino que se pueden encontrar un conjunto de soluciones Pareto optimal. Se han desarrollado algunos planificadores para resolver problemas multi-objetivo, tales como MO-GTR [13]. En este trabajo se consideran dos objetivos típicos en planning y scheduling, la minimización de la duración y el coste totales del plan.

Por otro lado, el lenguaje Planning Domain Description Language (PDDL) es un estándar para la representación de dominios de planificación. PDDL soporta varios aspectos, tales como STRIPS, efectos condicionales, cuantificación universal sobre universos dinámicos, especificación de restricciones, etc. PDDL 2.1 [7], una extensión de PDDL, incluye nuevas características: permite manejar valores numéricos que pueden ser utilizados en las precondiciones y los efectos de las acciones; las acciones pueden tener una duración asociada; y es capaz de tratar con funciones objetivos relativas

al plan. Estas características son muy útiles para resolver de forma adecuada el problema de la planificación de la reparación, como puede verse más adelante.

El presente trabajo presenta una propuesta para el modelado y la ejecución de procesos de negocio que requieren decisiones tanto de planning (selección y ordenación parcial de las actividades que deben ser ejecutadas) como de scheduling (asignación de recursos y ordenación total). La principal contribución es que ambos tipos de decisiones P&S se toman en tiempo de ejecución, incrementando la eficiencia y flexibilidad del proceso de negocio. A modo de ejemplo, el problema de la planificación de la reparación (problema complejo y representativo que incluye planning y scheduling) se gestiona con la propuesta actual. Para este problema se considera la minimización de la duración y el coste del plan, cuando éste se ejecuta en un entorno con múltiples recursos compartidos y limitados, considerando: duraciones y costes de tareas, y estimación de las duraciones y los costes necesarios para llevar a cabo operaciones auxiliares, tales como el transporte de subsistemas entre distintos recursos, o los cambios de configuración de los recursos. Se considera que las tareas pueden ejecutarse en más de un modo de ejecución, cada una utilizando diferente recurso o configuración, y posiblemente con una duración y un coste asociado distinto. Además, este trabajo presenta una propuesta PDDL 2.1 que puede ser utilizada como entrada de un planificador genérico para resolver el problema de la planificación de la reparación. También se incluyen los resultados obtenidos por un planificador genérico en la resolución de un conjunto de ejemplos. En un trabajo previo [2], se muestra una especificación basada en satisfacción de restricciones para resolver el problema de la planificación de la reparación.

El resto del trabajo se organiza de la siguiente manera: el apartado 2 detalla el problema de la reparación considerado, el apartado 3 muestra la especificación PDDL para dicho problema, el apartado 4 presenta la transformación de grafos And/Or a modelos de BP, el apartado 5 muestra algunos resultados experimenta-

les y, por último, el apartado 6 resume algunas conclusiones y trabajos futuros.

2. El Problema de la Planificación de la Reparación

En este trabajo se consideran sistemas compuestos por un conjunto de componentes. Cuando se diagnostica un componente defectuoso, se debe llevar a cabo un plan de sustitución/reparación para solucionar el problema. Este plan de reparación está formado por tres pasos, que pueden solaparse:

1. Proceso de desconexión: conjunto de tareas que deben aislar el componente defectuoso.
2. Acción de reparación: tarea para reparar el componente defectuoso.
3. Proceso de conexión: conjunto de tareas que deben volver a conectar el sistema.

En el plan de reparación propuesto, pueden aparecer diferentes subsistemas (uno o más componentes que están conectados). Además, se consideran dos tipos de tareas:

- Tareas de conexión/desconexión: se ejecutan en una máquina determinada con una configuración concreta para obtener un subsistema (en el caso de tareas de conexión), o dos subsistemas (en el caso de tareas de desconexión).
- Tareas auxiliares: se deben al uso de recursos compartidos. Se consideran dos tipos de operaciones: operaciones de cambio de configuración, que cambian la configuración de un recurso cuando dos tareas sucesivas con distinta configuración utilizan dicho recurso; y operaciones de transporte, que trasladan los subsistemas entre recursos cuando el recurso donde el subsistema se obtiene es distinto del recurso donde se requiere.

Se considera una duración y un coste asociado a la ejecución de cada tarea.

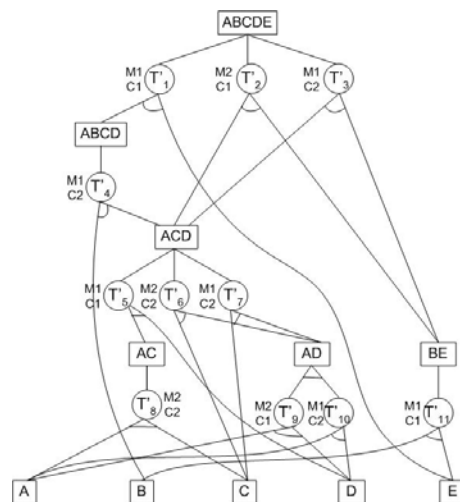


Figura 1: El grafo And/Or de un sistema formado por 5 componentes.

2.1. El Grafo And/Or

Existen varias representaciones para el problema de la planificación de la reparación. Una de ellas es mediante grafos And/Or [9], que permiten representar el conjunto de todos los planes de conexión y desconexión factibles de manera natural. La figura 1 muestra el grafo And/Or para un sistema compuesto por 5 componentes. Dichos grafos están formados por 2 tipos de nodos: nodos Or y nodos And. El primero de ellos se corresponde con subconjuntos de componentes del sistema, de forma que la raíz representa el sistema completo mientras los nodos hoja representan los componentes individuales. Los nodos And se corresponden con las tareas que conectan los subsistemas: una arista descendente descompone un subsistema en 2, mientras una ascendente se puede ver como la tarea reversible que une 2 subsistemas en uno compuesto.

Para el mismo nodo Or, pueden existir varios nodos And (tareas) debajo, representando diferentes alternativas para conectar/desconectar los subsistemas correspondientes. En los grafos And/Or, cada plan de conexión/desconexión está asociado a un árbol, que es un camino que comienza en el nodo

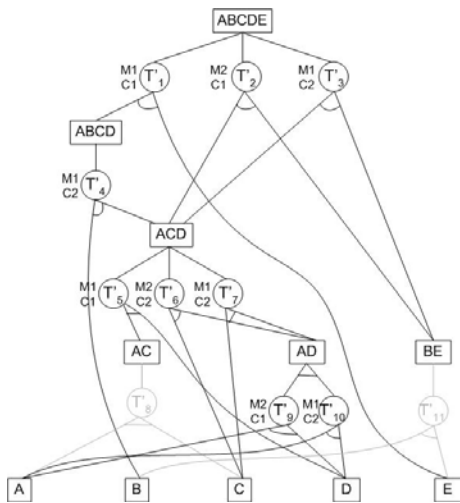


Figura 2: El grafo And/Or simplificado cuando el componente defectuoso es D

raíz y termina en los nodos hoja. Una ventaja importante que presenta esta representación es que se muestran claramente las tareas que pueden ejecutarse en paralelo.

2.2. Tareas Multi-modo

Como se ha comentado anteriormente, se considera que las tareas pueden ejecutarse en más de un modo de ejecución, cada una utilizando diferentes recursos o configuración, y, posible-mente, con un coste y una duración distintos. Teniendo en cuenta esto, pueden existir distintas opciones tanto de conexión como de desconexión.

En este trabajo, se considera que cada tarea con un modo de ejecución determinado se corresponde con un nodo And en el grafo (por ejemplo T'_2 y T'_3 en la Fig. 1)

2.3. Algunas Consideraciones

El modelo de planificación desarrollado supone que: (A1) todas las tareas son reversibles (A2) los subsistemas que no incluyen los componentes defectuosos no se desconectan. Con respecto a (A2), para el problema de la Fig. 1, si D es el componente defectuoso, entonces los

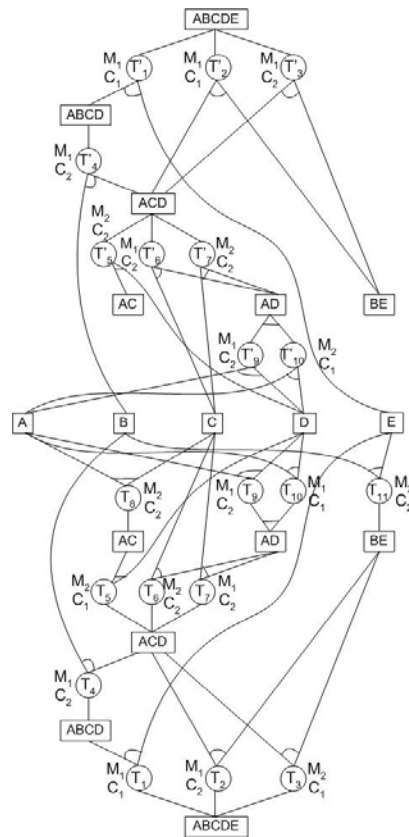


Figura 3: El grafo And/Or de reparación simplificado cuando el componente defectuoso es D

subsistemas AC y BE no se desconectan, por lo tanto las tareas T'_8 y T'_{11} no se seleccionan para el proceso de desconexión (Fig. 2).

Fig. 3 presenta el grafo And/Or de reparación extendido y simplificado para el problema de la Fig. 1 cuando el componente defectuoso es D . En este grafo, la parte superior representa el proceso de desconexión, la parte intermedia los componentes individuales, y la parte inferior el proceso de conexión.

2.4. Optimización Multi-objetivo

En el trabajo actual, se consideran dos funciones objetivo en la búsqueda de un plan factible, la minimización del tiempo y el coste

Cuadro 1: Predicados para el problema de la planificación de la reparación.

Predicado	Descripción
(constr-conex ?s - subsistema)	El subsistema se ha obtenido en el proceso de conexión y no ha sido utilizado todavía para otra operación.
(constr-desconex ?s - subsistema)	El subsistema se ha obtenido en el proceso de desconexión y no ha sido utilizado todavía para otra operación.
(en ?s - subsistema ?r - recurso)	El subsistema <i>s</i> está en el recurso <i>r</i> .
(conf ?r - recurso ?c - config)	El recurso <i>r</i> tiene configuración <i>c</i> .
(tarea-conex ?s1 ?s2 ?s - subsistema ?r - recurso ?c - config)	Existe una tarea de conexión que se ejecuta en el recurso <i>r</i> con configuración <i>c</i> y conecta <i>s1</i> y <i>s2</i> para obtener <i>s</i> .
(tarea-desconex ?s ?s1 ?s2 - subsistema ?r - recurso ?c - config)	Existe una tarea de desconexión que se ejecuta en <i>r</i> con configuración <i>c</i> y desconecta <i>s</i> para obtener <i>s1</i> y <i>s2</i> .
(defec ?s - subsistema)	Es el componente defectuoso.
(hoja ?s - subsistema)	Es una hoja en el grafo And/Or.
(libre ?r - recurso)	El recurso está libre.

totales del plan de reparación.

Para resolver los problemas de optimización multi-objetivo [6], se puede utilizar el método de la suma ponderada. Dicho método consiste en definir una nueva función objetivo que puede ser optimizada por resolutores mono-objetivo, $\sum w_i f_i$, donde $w_i \geq 0$ para cada función objetivo f_i considerada. Es recomendable normalizar las funciones objetivo con una escala adecuada.

Para el modelo propuesto, las funciones objetivos planteadas combinan la optimización de la duración y el coste total del plan de reparación, considerando diferentes combinaciones de peso para cada una (más detalles en el apartado 5). Este método es fácil de utilizar y, si todos los pesos son positivos, el mínimo de la función objetivo es siempre Pareto optimal.

3. Especificación PDDL

La especificación PDDL incluye dos elementos: el dominio para predicados y acciones, y el problema, que contiene los objetos, el estado inicial y el objetivo a alcanzar.

3.1. Descripción del Dominio

La definición del dominio en una especificación PDDL contiene diferentes puntos::

Predicados: Propiedades de los objetos que pueden ser ciertas o falsas (Tabla 1). La razón por la cual se han definido **constr-conex** y **constr-desconex** como predicados distintos es para facilitar el trabajo de búsqueda del planificador.

Funciones (Fluents): Permite manejar valores numéricos. Se pueden utilizar en las precondiciones o en los efectos de las acciones, y se le da valor en el fichero problema (Tabla 2).

Acciones/Operadores (Con duración): Permiten cambiar el estado del mundo (Figuras 4 y 5). La acción **conexion** actúa sobre los subsistemas **s1** y **s2** para obtener **s**, utilizando el recurso **r** con configuración **conf**, y duración dada por la función **dur-conex**. El coste de dicha acción se añade al fluente **cost-acumul** del recurso **r**. De forma similar, se define la acción de **desconexion**. La acción de reparación **repara** el subsistema **s** en el recurso **r**, con duración dada por la función **reparar**. El coste de esta acción se añade al fluente **cost-acumul** de **r**. En el proceso

Cuadro 2: Funciones para el problema de la planificación de la reparación.

Función	Descripción
(camb-conf ?r - recurso ?c1 ?c2 - config)	Tiempo requerido para cambiar el recurso r de c1 a c2.
(cost-conf ?r - recurso ?c1 ?c2 - config)	Coste requerido para cambiar el recurso r de c1 a c2.
(mov ?s - subsistema ?r1 ?r2 - recurso)	Tiempo requerido para cambiar el subsistema s del recurso r1 a r2.
(cost-mov ?s - subsistema ?r1 ?r2 - recurso)	Coste requerido para cambiar el subsistema s del recurso r1 a r2.
(dur-conex ?s1 ?s2 ?s - subsistema ?r - recurso ?conf - config)	Tiempo requerido para conectar los subsistemas s1 y s2 para obtener s.
(cost-conex ?s1 ?s2 ?s - subsistema ?r - recurso ?conf - config)	Coste requerido para conectar los subsistemas s1 y s2 para obtener s.
(dur-desconex ?s ?s1 ?s2 - subsistema ?r - recurso ?conf - config)	Tiempo requerido para desconectar el subsistema s para obtener s1 y s2.
(cost-desconex ?s ?s1 ?s2 - subsistema ?r - recurso ?conf - config)	Coste requerido para desconectar el subsistema s para obtener s1 y s2.
(reparar ?s - subsistema)	Tiempo requerido para reparar el subsistema s.
(cost-rep ?s - subsistema)	Coste requerido para reparar el subsistema s.
(cost-acumul ?r - recurso)	Este fluent es el coste acumulado para cada recurso en cada instante. Se usa para definir la función objetivo relativa al coste.

de desconexión, cuando un subsistema no contiene el componente defectuoso, no sigue descomponiéndose, por lo que desaparece del proceso de desconexión para unirse al de conexión (*desconexion-a-conexión*). La acción *mover* transporta el subsistema *s* desde el recurso *r1* a *r2*, con duración *mov*. El coste de esta acción se añade al fluent *cost-acumul* de *r2*. Finalmente, la acción *cambia-config* cambia la configuración de *r* de *c1* a *c2*. El coste de dicha acción se añade a *cost-acumul* de *r*.

3.2. Descripción del Problema

La especificación PDDL del problema contiene los siguientes puntos:

Objetos: Se han definido los objetos de la Fig. 6 para el problema de la Fig. 3. Inicialmente, el sistema puede estar en cualquier sitio, lo que se representa mediante el recurso *REC_INIC*, y los recursos pueden presentar cualquier configuración, lo que se representa mediante *CO*.

Estado inicial: Para el problema de la Fig. 3, el estado inicial se muestra en la Fig. 6.

Objetivo: predicados que deben ser ciertos tras la ejecución del plan. En el problema de la reparación, el objetivo consiste en obtener el sistema completo tras la reparación del componente defectuoso (Fig. 6).

Función objetivo: Medidas de calidad del plan (metrics). Para el problema de la reparación, se han seleccionado algunas funciones objetivo para ser minimizadas, una de ellas aparece en la Fig. 6.

4. Modelado y Ejecución del Problema de la Planificación de la Reparación como un BP

Como ejemplo de aplicación de nuestra propuesta, el problema de la planificación de la reparación, representado mediante un grafo And/Or, se va a modelar a través de un lenguaje de modelado/ejecución para BP, concretamente BPMN [15].

```
(:durative-action co\ne\-\xión
:parameters (?s1 ?s2 ?s - subsistema ?r - recurso
?c - config)
:duration
(= ?duration (dur-conex ?s1 ?s2 ?s ?r ?c))
:condition
(and (at start (tarea-conex ?s1 ?s2 ?s ?r ?c))
(at start (libre ?r))
(at start (conf ?r ?c))
(at start (en ?s1 ?r))
(at start (en ?s2 ?r))
(at start (constr-conex ?s1))
(at start (constr-conex ?s2)))
:effect
(and (at start (not (libre ?r)))
(at start (not (en ?s1 ?r)))
(at start (not (en ?s2 ?r)))
(at start (not (is-built-conec ?s1)))
(at start (not (is-built-conec ?s2)))
(at end (libre ?r))
(at end (en ?s ?r))
(at end (constr-conex ?s))
(at end (increase (cost-acumul ?r)
(cost-conec ?s1 ?s2 ?s ?r ?c)))
(:durative-action REPARAR
:parameters (?comp - subsistema ?r - recurso)
:duration (= ?duration (reparar ?comp))
:condition
(and (at start (defec ?comp))
(at start (constr-desconex?comp))
(at start (en ?comp ?r)))
:effect
(and (at start (not (is-built-disconec ?comp))
(at end (constr-conex ?comp))
(at start (not (at ?comp ?r)))
(at end (en ?comp ?r))
(at start (not (defec ?comp)))
(at end (increase (cost-acumul ?r)
(cost-rep ?comp))))))
```

Figura 4: Especificación PDDL para las acciones conexión y reparar.

Tanto los grafos And/Or como los modelos para BP deben contemplar todas las alternativas de ejecución posibles, y durante la fase de ejecución del BP, métodos AI P&S serán utilizados para seleccionar y ordenar las actividades adecuadas para obtener un plan de ejecución óptimo.

El modelo BP propuesto para el problema de la planificación de la reparación Fig. 8.a) contiene tres pools: *Cliente*, *Módulo de Reparación* y *Planificador*. El *Cliente* es un pool abstracto que actúa como intermediario entre el usuario y el BP. El usuario debe especificar los componentes defectuosos, lo que da lugar al comienzo de la ejecución del BP. Al final de dicha ejecución, se envía al cliente un mensaje que contiene las actividades que

```
(:action DESCONEXION-A-CONEXION
:parameters (?s - subsistema)
:precondition (and (hoja ?s)
(constr-desconex?s))
:effect (and (not (constr-desconex?s))
(constr-conex ?s)))
(:durative-action MOVER
:parameters (?s - subsistema ?r1 ?r2 - recurso)
:duration (= ?duration (mov ?s ?r1 ?r2))
:condition (at start (en ?s ?r1))
:effect (and (at start (not (en ?s ?r1)))
(at end (en ?s ?r2))
(at end (increase (accum-cost ?r2)
(cost-mov ?s ?r1 ?r2))))
(:durative-action CAMBIAR-CONFIG
:parameters (?r - recurso ?c1 ?c2 - config)
:duration (= ?duration (camb-conf ?r ?c1 ?c2))
:condition (and (at start (conf ?r ?c1))
(at start (libre ?r)))
:effect (and (at start (not (libre ?r)))
(at end (libre ?r))
(at start (not (conf ?r ?c1)))
(at end (conf ?r ?c2))
(at end (increase (cost-acumul ?r)
(cost-conf ?r ?c1 ?c2))))
```

Figura 5: Especificación PDDL para las acciones desconexión-a-conexion, mover y cambia-config.

han sido seleccionadas junto con su información temporal. El *Módulo de Reparación* es un pool abstracto que contiene los servicios web. Dichos servicios basan su implementación en técnicas AI P&S, y son los responsables de la planificación óptima de las actividades. Cada vez que la ejecución del plan de reparación (*Módulo de Reparación*) alcanza una puerta OR, el pool *Planificador* le indica el camino a seguir (planning). Además, dicho pool establece el orden de ejecución para las tareas que utilizan el mismo recurso, en el caso de que éstas entren en conflicto (scheduling). Durante la ejecución del plan de reparación, el *Módulo de Reparación* y el *Planificador* se intercambian información acerca del BP.

A continuación, se detalla la transformación desde un grafo And/Or a un modelo BP. A modo de ejemplo, en la figura 8 aparece el BPMN correspondiente al grafo de la figura 7. Dicho BPMN está formado por 2 diagramas: la Fig. 8.a muestra las actividades correspondientes a las tareas de conexión/desconexión sin expandir; mientras en la Fig. 8.b aparece la ejecución de dichas actividades de forma detallada. En el BP de la Fig. 8, en primer lugar, el *Cliente* envía un mensaje al *Módulo de*

```
(define (problem grafo)
(:domain reparacion)
(:objects M1 M2 - recurso
  C0 C1 - config
  ABCDE ABCD ACD AC AD BE A B C D E - subsistema
  INIT_LOC - almacen)
(:init
(en ABCDE REC\_INIC)(constr-desconex ABCDE)
(defec D)(libre M1)(conf M1 C0)(hoja A) ...
(tarea-desconex ABCDE ABCD E M1 C1)
(= (dur-desconex ABCDE ABCD E M1 C1) 10)
(= (cost-desconex ABCDE ABCD E M1 C1) 124) ...
(= (cost-conf M1 C0 C1) 0)(= (camb-conf M1 C3 C1) 4)
(= (cost-conf M1 C1 C3) 36)(= (mov AD INIT\_LOC M1) 0)
(= (cost-mov AD INIT\_LOC M1) 0)(= (mov AD M1 M2) 1)
(= (cost-mov AD M1 M2) 15)...
(:goal (constr-conex ABCDE))
(:metric minimize (+(*(total-time) 10)
  +(cost-acumul M1)(cost-acumul M2))))
```

Figura 6: Especificación PDDL del problema de Fig. 3.

Reparación indicándole el componente defectuoso. Tras esto, una herramienta basada en la especificación PDDL 2.1 explicada anteriormente, se encarga de generar un plan inicial optimizando las funciones objetivo correspondientes. Este plan inicial, en la mayoría de los casos, se mejorará de forma iterativa a medida que avanza la ejecución del BP, como se explica posteriormente. Tras la generación del plan inicial, dos ramas se ejecutan en paralelo en el *Módulo de Reparación*:

- Proceso de optimización (Fig. 8.a): Puede ser que el plan inicial no sea óptimo. Además, los parámetros reales pueden diferir, en mayor o menor medida, de los estimados. Teniendo en cuenta ambos factores, se propone una actividad de actualización, que está continuamente intentando mejorar la solución actual, considerando los valores reales de los parámetros. Para realizar esto, se invoca un servicio web basado en técnicas de AI (*Actualiza plan*), de forma que el plan se actualiza cada vez que se encuentra una solución mejor.
- Plan de reparación (Fig. 8.a): las actividades BP se ejecutan considerando la planificación establecida por el *Planificador*. Cada relación del grafo And/Or, en la que están involucrados varios com-

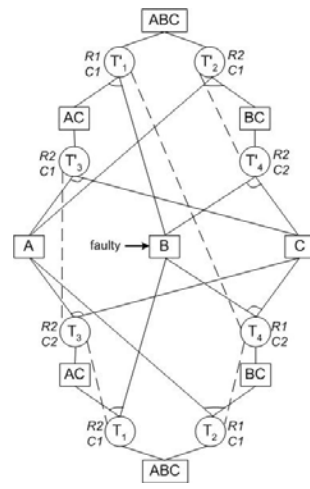
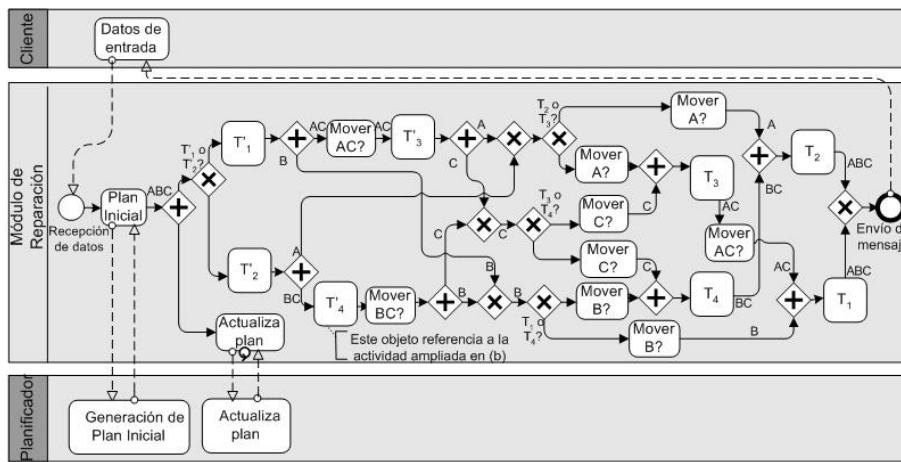


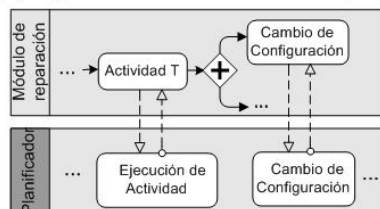
Figura 7: Un grafo And/Or extendido para un sistema formado por 3 componentes.

ponentes, da lugar a otra relación en el modelo BPMN. Concretamente, un subsistema para el que existen varias tareas de desconexión alternativas, se transforma en una relación del tipo Exclusive Data-Based Gateway, con tantas salidas como opciones de desconexión; una tarea de desconexión que da lugar a dos subsistemas se transforma en una relación del tipo Parallel Gateway con dos salidas, una por subsistema; un subsistema que en el proceso de conexión puede ser el resultado de varias tareas alternativas de conexión, se transforma en una relación del tipo Exclusive Data-Based Gateway con tantas entradas como tareas alternativas; y, por último, una tareas de conexión que conecta dos subsistemas se transforma en una relación del tipo Parallel Gateway con dos entradas, una por subsistema.

Por otro lado, el recurso en el cual se obtiene un subsistema tras una actividad de conexión/desconexión puede ser diferente del recurso donde se requiere, por lo que es necesario considerar la actividad *Mover Subsistema?* entre dos actividades sucesivas que actúan sobre el mismo subsistema. En la Fig. 8.b, la *Activi-*



(a) Diagrama BPMN con actividades sin expandir



(b) Diagrama BPMN con actividades expandidas

Figura 8: BPMN para el problema de la Fig. 7

dad T es la ejecución real de la tarea de conexión/desconexión. Un servicio web debe gestionarla, ya que debe establecer su tiempo de comienzo, su duración, etc. Teniendo en cuenta la información de ejecución de las tareas, se almacenan los valores reales, y se utilizan para la actualización del plan. La actividad *Cambio de Configuración*, en el caso de que sea necesaria, se debe realizar en paralelo con el tratamiento de los subsistemas que se han sido obtenidos tras la ejecución de la actividad (Fig. 8.b). La ejecución de dicha actividad se controla mediante el *Planificador*, ya que su ejecución provoca el bloqueo de recursos, lo que tienen una gran influencia en la ejecución del plan. Además, el hecho de que esta actividad se ejecute o no depende de varios factores que deben ser analizados por el *Planificador*.

5. Resultados Experimentales

Para realizar los resultados experimentales, se ha utilizado un planificador genérico, SG-Plan [5]. La tabla 3 muestra el número de nodos en un conjunto de grafos And/Or correspondientes a un conjunto de sistemas hipotéticos formados por 30 y 40 componentes. Suponiendo que cada componente individual debe ser reparado, en dicha tabla se incluyen el número de nodos Or, And (tareas de conexión), And' (tareas de desconexión) y de planes de reparación posibles ($\#RP$) para los grafos simplificados. Para obtener resultados acerca del funcionamiento del modelo propuesto en un entorno de tareas multi-modo, cada grafo se ha extendido incluyendo un 10% de tareas multi-modo (M30-1, M30-2, M30-3, M40-1, M40-2 y M40-3). Cada fila representa a un conjunto de 80 instancias, con diferentes combinaciones para algunos parámetros, tales

Cuadro 3: Número de nodos And, Or (media)

Prob	Grafo And/Or		Grafo And/Or simplificado			
	#Or	#And	#Or	#And	#And'	#RP
30-1	348	630	223	327	240	1213
30-2	404	828	303	520	365	9200
30-3	415	863	310	546	384	12846
40-1	649	1518	433	833	575	23005
40-2	770	2143	621	1489	984	248408
40-3	756	2060	598	1400	925	197551

Cuadro 4: Resultados Comparativos (calidad de las soluciones)

Prob	f_{Time}		f_{MO20}		f_{MO10}		f_{Cost}	
	T	C	T	C	T	C	T	C
30-1	575,78	6775,61	575,78	6775,61	575,78	6775,61	575,78	6775,61
30-2	563,68	6821,89	569,09	6892,68	563,68	6821,89	563,68	6821,89
30-3	658,40	7889,76	648,11	7750,80	658,40	7889,76	658,40	7889,76
40-1	526,47	6302,39	532,97	6358,04	526,47	6302,39	526,47	6302,39
40-2	587,13	6910,68	588,31	6892,90	587,13	6910,68	587,13	6910,68
40-3	633,62	7821,95	596,66	7372,09	633,62	7821,95	633,82	7821,95
M30-1	582,34	6644,25	582,34	6644,25	582,34	6644,25	582,34	6644,25
M30-2	574,78	6931,07	574,67	6931,07	574,67	6931,07	574,67	6931,07
M30-3	674,84	7927,21	674,84	7927,21	674,84	7927,21	674,84	7927,21
M40-1	536,69	6615,35	536,69	6615,35	536,69	6615,35	536,69	6615,35
M40-2	622,72	7195,27	622,72	7195,27	622,72	7195,27	622,72	7195,27
M40-3	667,28	8279,40	667,28	8279,40	667,28	8279,40	667,28	8279,40

como duración de las tareas, recursos, configuraciones, y componente defectuoso. El coste de cada tarea es una función que depende del recurso requerido, la configuración establecida para el mismo y la duración de dicha tarea. Estos experimentos se han llevado a cabo en un 2,66GHz Intel Core 2 Duo con 4GB RAM.

Se han seleccionado 4 funciones objetivos para ser minimizadas: el coste, la duración, y dos funciones objetivo combinadas (aplicando el método de la suma ponderada, $w_t t_{OR}(P) + w_c cost_{total}$, para $w_t = 10$ y $w_c = 1$ (f_{MO10}), y para $w_t = 20$ y $w_c = 1$ (f_{MO20})). Se han elegido los valores 10 y 20 porque el coste de un plan de reparación es, de media, 15 veces la duración total del plan.

Algunos resultados experimentales se muestran en las tablas 4 (calidad de las soluciones obtenidas) y 5 (tiempos medios de ejecución).

6. Conclusiones y Trabajo Futuro

Los sistemas de gestión de procesos de negocio son capaces de gestionar el desarrollo de software, ya que dicho desarrollo puede modelarse como un proceso de negocio. Este trabajo presenta una propuesta para modelar y ejecutar procesos de negocio que involucran decisiones de planning (selección y ordenación parcial de las actividades que deben ejecutarse) y scheduling (asignación de recursos y ordenación total) en tiempo de ejecución. A modo

Cuadro 5: Tiempo de ejecución (media)

Prob	f_{Time}	f_{MO20}	f_{MO10}	f_{Cost}
30-1	0,15	0,31	0,81	0,26
30-2	0,43	0,82	0,83	0,75
30-3	0,47	0,86	0,84	0,74
40-1	0,85	1,84	1,90	1,67
40-2	0,65	1,57	1,72	1,51
40-3	1,02	2,21	2,22	2,18
M30-1	0,88	0,31	0,35	0,73
M30-2	4,88	0,93	0,95	4,10
M30-3	4,70	0,87	0,95	3,73
M40-1	11,96	1,84	1,94	11,04
M40-2	9,53	1,70	1,71	9,62
M40-3	11,81	2,40	2,45	14,63

de ejemplo, el problema de la planificación de la reparación (problema complejo y representativo que incluye planning y scheduling) se gestiona con la propuesta actual. Para resolver este problema, se propone una especificación PDDL 2.1.

Como trabajo futuro, se pretenda analizar otros tipos de procesos de negocio que involucren planning y scheduling. Además, se considera interesante el estudio de diferentes planificadores para conocer las contribuciones de cada uno de ellos al problema de la planificación de la reparación.

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía (P08-TIC-04095), por el Ministerio de Ciencia e Innovación (TIN2009-13714) y por the European Regional Development Fund (ERDF/FEDER).

Referencias

[1] van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M., *Business Process Management: A Survey*, Int. Conf. BPM 2003, Proceedings, 1–12, 2003.

[2] Barba, I., Del Valle, C., Borrego, D., *A multiobjective constraint optimization model for multimode repair plans*, 6th International Conference on Informatics, Control, Automation and Robotics 1, 355–358, 2009.

[3] Boddy, M.S., Cesta, A., Smith, S.F., *ICAPS-04 Workshop on Integrating Planning into Scheduling*, AAAI Press, 2004.

[4] Brucker, P., Knust, S., *Complex Scheduling (GOR-Publications)*, Springer-Verlag New York, Inc., 2006.

[5] Chen, Y., Hsu, C., Wah, B., *Temporal planning using subgoal partitioning and resolution in SGPlan*, Journal of Artificial Intelligence Research 26, 323–369, 2006.

[6] Coello, C.A., *Evolutionary multi-objective optimization: A historical view of the field*, IEEE Computational Intelligence Magazine 1(1), 28–36, 2006.

[7] Fox, M., Long, D., *PDDL 2.1. An extension to PDDL for expressing temporal planning domains*, Journal of Artificial Intelligence Research 20, 61–124, 2003.

[8] Ghallab, M., Nau, D., Traverso, P., *Automated Planning: Theory and Practice*, Morgan Kaufmann, 2004.

[9] Homem de Mello, L.S., Sanderson, A.C., *And/or graph representation of assembly plans*, IEEE Transactions on Robotics and Automation 6(2), 188–189, 1990.

[10] ILOG, <http://www.ilog.fr/>, France.

[11] Kolisch, R., Drexel, A., *Local for multi-mode resource-constrained project*, IIE Transactions (Institute of Industrial Engineers) 29(11), 987–999, 1999.

[12] Lambert, A.J.D., *Disassembly sequencing: a survey*, Int. Jour. Production Research 41(16), 3721–3759, 2003.

[13] Refanidis, I., Vlahavas, I., *Multiobjective heuristic state-space planning*, Artificial Intelligence 145 (1-2), 1–32, 2003.

- [14] Smith, D., Frank, J., Jónsson, A., *Bridging the gap between planning and scheduling*, Knowledge Engineering Review 15(1), 47–83, 2000.
- [15] White, S.A. et al, *Business Process Modeling Notation (BPMN), Working draft, Version 1.0*, BPML.org, 2004.