

UN MODELO DE INTEGRACIÓN DE PLANIFICACIÓN Y SCHEDULING¹

A. Garrido E. Marzal L. Sebastiá F. Barber

Dpto. Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

Camino de Vera s/n, 46071 Valencia, Spain

e-mail: {agarridot, emarzal, lstarin, fbarber}@dsic.upv.es

Palabras clave: planificación, scheduling, restricciones.

Keywords: planning, scheduling, constraints.

RESUMEN. *En diversas aplicaciones de planificación aparecen restricciones respecto a temporalidad de las acciones y de sus efectos, estados del proceso, etc. Esto introduce una problemática temporal en el proceso. Por otra parte, en la mayoría de problemas reales, la ejecución de un plan conlleva la utilización de recursos, generalmente limitados y compartidos. Esto da lugar a un problema adicional de scheduling de acciones y recursos en el tiempo. A pesar de los resultados obtenidos de forma aislada en planificación y scheduling, ha habido pocos intentos de integración. Generalmente, después de la planificación, el scheduler realiza la asignación de recursos y validación de las restricciones del problema. En este artículo proponemos un modelo para llevar a cabo el proceso anterior de forma integrada: obtención y validación del plan durante su generación. El comportamiento integrado de los procesos es coordinado por un módulo de control encargado de gestionar la interacción entre ellos, y toma las decisiones globales sobre su actuación. El objetivo es obtener un plan válido, correcto, ejecutable, y eventualmente óptimo.*

ABSTRACT. *Temporal constraints over actions and their effects or process states may appear in planning problems. In many real problems, the execution of a plan involves the use of resources, generally limited and shared. For this reason, an additional stage of scheduling of actions and resources in time becomes more necessary. In spite of the obtained results in a separate way in planning and scheduling, there have been few integration intents. Usually, the scheduler carries out the resource allocation and validation of the constraints of the problem after the planning process. We present a model to execute this process in an integrated way: obtaining and validating the plan during its generation. The integrated behaviour of the processes is coordinated by a control module in order to manage the interaction between them, giving more priority selection approaches and making the global decisions on its performance. Therefore, the aim is to obtain a valid, executable, and optimal plan.*

1.- INTRODUCCIÓN.

En el mundo real, existen muchos problemas (industriales, problemas de transporte, planificación de procesos productivos, etc.) que pueden ser tratados como problemas de planificación y scheduling (Bertolotti 93, Gervasio 92). Por una parte, se trata de obtener un plan, como una secuencia de acciones que alcance un objetivo, y por otra, delimitar la ejecutabilidad de las acciones en el tiempo, asignándoles los recursos necesarios y teniendo en cuenta las restricciones generales del problema. Los métodos clásicos para solucionar estos problemas se basan principalmente en técnicas de Investigación Operativa (Baptiste 95b) y de Inteligencia Artificial. En

¹ Este trabajo ha sido propuesto por el Grupo de Planificación y Scheduling de la Universidad Politécnica de Valencia (<http://www.dsic.upv.es/users/ia/gps>) y soportado por el proyecto CICYT/TAP98-0345.

las técnicas de Investigación Operativa la secuenciación parcial de las acciones, o posibles planes alternativos, ya se conoce y el problema se basa principalmente en obtener la secuenciación óptima de acciones y/o asignación óptima de recursos. Por otro lado, los métodos de Inteligencia Artificial se pueden utilizar para determinar planes correctos -por un planificador (Yang 97)- y para realizar la asignación de recursos -por un scheduler (Dorn 93). Ambos métodos tienen gran utilidad en multitud de problemas reales. A pesar de ello, el incremento de la complejidad de los problemas actuales (utilización de un gran número de recursos, restricciones temporales sobre la aplicabilidad de las acciones y sus efectos sobre los estados del problema, etc.) obliga a utilizar nuevas aproximaciones, más flexibles y más potentes.

Tradicionalmente, los métodos para solucionar problemas de planificación y scheduling se dividen en dos etapas diferentes (Srivastava 99). En la primera (etapa de planificación) se obtiene un plan como una secuencia ordenada de acciones, y durante la segunda (etapa de scheduling) se asignan los recursos, intentando optimizarlos, y garantizando el cumplimiento de las restricciones existentes (Beck 95). Así, el planificador construye un plan que resuelve el problema. Posteriormente, el scheduler asignará los recursos usando criterios de optimización y garantizando, adicionalmente, su ejecutabilidad. Desde este punto de vista, los dos procesos son independientes y sin ninguna relación entre ellos durante su respectiva ejecución.

La idea principal de este artículo es la integración de las etapas de planificación y scheduling: si los dos procesos colaboran durante la resolución del problema, se prevé la obtención de una solución mejor y más eficiente. Por ello, proponemos una arquitectura independiente del dominio como resultado de nuestra línea de trabajo actual. En esta arquitectura, el scheduler trabaja de forma simultánea con el planificador.

En la sección 2 caracterizamos el problema a resolver y presentamos los elementos del lenguaje de especificación. En la sección 3, explicamos la arquitectura para la integración de planificación y scheduling, detallando su estructura y funcionamiento, y comentando brevemente ambos procesos. En la sección 4 finalizamos con las conclusiones.

2.- PROBLEMA DE PLANIFICACIÓN Y SCHEDULING.

En este apartado presentaremos las restricciones que el sistema debe manejar así como los elementos del lenguaje que permiten especificar el dominio del problema.

2.1. Caracterización del problema. Restricciones.

Un sistema de planificación-scheduling debe manejar todo tipo de restricciones, tanto temporales como sobre recursos y sus respectivas utilidades. Por ejemplo, un sistema de este tipo debería ser capaz de gestionar eficientemente las siguientes restricciones:

- **Restricciones sobre recursos limitados.** Aunque cada recurso dispone de sus propias características, existen otras que son comunes a la mayoría de ellos como cantidad, tiempo de servicio, disponibilidad de recursos, función de coste, etc. Las restricciones sobre disponibilidad deben ser, en todo momento, garantizadas por el sistema, mientras que las funciones de evaluación o coste deben ayudar a la hora de tomar decisiones para conocer la calidad del plan generado. Debido a la existencia de recursos compartidos en el tiempo, el sistema debe ser capaz de asegurar la utilización disjunta de estos recursos entre distintas acciones.
- **Restricciones sobre el estado de los objetos y su persistencia.** El valor de los slots o atributos de los objetos identifica en todo momento el estado del objeto. Puede haber restricciones sobre dicho estado si se exige que se mantenga durante un plazo mínimo o máximo de tiempo (persistencia), o si es necesario que se alcance antes de un determinado instante.
- **Otras restricciones temporales.** En este punto se debe tener en cuenta todas las restricciones temporales sobre acciones, recursos, etc., como por ejemplo los intervalos en los que se debería ejecutar una determinada acción y utilizar los recursos asociados a ella. Adicional-

mente, puede ser necesario la consecución de un estado meta antes de un instante dado o el cumplimiento de plazos máximos para algunas acciones específicas.

2.2. Elementos del lenguaje de especificación.

A fin de modelar los problemas de planificación y scheduling, se define un lenguaje de especificación, mediante el cual el usuario define la jerarquía de objetos, acciones, recursos disponibles, restricciones temporales, situación inicial y objetivos del problema. Las restricciones del problema pueden ser implícitas (no-utilización simultánea de recursos) o explícitas (precedencias de acciones u órdenes, duraciones, persistencia temporal de los efectos, etc.) Mediante este lenguaje, cuya gramática se define en el Anexo I, el usuario podrá especificar los siguientes conceptos:

Jerarquía de Objetos. Las aproximaciones clásicas de planificación utilizan un lenguaje declarativo -predicados de primer orden- (Penberthy 92). Por el contrario, el lenguaje propuesto utiliza una estructura basada en *frames* donde se definen clases e instancias. Los literales mantendrán siempre la misma estructura: (*<nombre-clase> <objeto> <nombre-slot> <valor>*). Los objetos pueden disponer de slots temporales y no temporales. Los slots temporales mantienen la secuencia histórica de sus valores tomados en el tiempo, así como el instante de tiempo en el que su valor fue modificado. Sin embargo, los slots no temporales sólo mantienen el valor actual.

Acciones Primitivas. Definen las precondiciones, efectos, tests, utilización de recursos, duración de las acciones y persistencia de las mismas. La utilización de un recurso se puede restringir temporalmente dentro del intervalo de la ejecución de la propia acción. Estas acciones primitivas son consideradas atómicas, es decir, no pueden dividirse en otras acciones.

Macro-Acciones. Se trata de un tipo especial de acciones con un mayor nivel de abstracción. Estas macro-acciones agrupan un conjunto de acciones primitivas e incluso otras macro-acciones. Además, definen un contexto donde se pueden añadir relaciones de orden parcial y restricciones temporales entre dichas acciones.

Jerarquía de Recursos. Define una jerarquía de clases e instancias de recursos que son compartidos por varias acciones. Para cada recurso se puede establecer el número de elementos disponibles, tiempo de servicio por defecto, restricciones temporales en cuanto a su disponibilidad y funciones de coste. Se pueden plantear dos tipos de recursos: concretos y abstractos. Los concretos se identifican y se caracterizan uno a uno, mientras que los abstractos se especifican globalmente. Por ejemplo, una flota de cinco vehículos indistintos se puede considerar como un recurso abstracto, mientras que un vehículo_i concreto y con características específicas representaría un recurso concreto.

Definición del Objetivo. El objetivo se define como un conjunto de tuplas (*<nombre-clase> <objeto> <nombre-slot> <valor>*) que representan el estado meta que se desea alcanzar.

3.- ARQUITECTURA DE LA INTEGRACIÓN DE PLANIFICACIÓN Y SCHEDULING.

El objetivo de un sistema de planificación es obtener un plan correcto cuya ejecución consiga la meta. Sin embargo, el plan sugerido puede no ser ejecutable por el uso de restricciones de recursos u otras restricciones temporales existentes en los datos del problema. En esta situación, el plan es inconsistente para el scheduler y se debe obtener un plan alternativo. Por otra parte, aunque un plan sea ejecutable puede no ser eficiente u óptimo en cuanto a su coste, número de recursos utilizados, etc. La principal idea de la integración de ambos procesos es garantizar la ejecutabilidad del plan y satisfacer óptimamente las restricciones del problema, simultáneamente con la construcción del plan. Si el sistema está plenamente integrado, se obtiene la ventaja de poder combinar el proceso de planificación con el de scheduling. Los beneficios obtenidos con

esta integración parecen claros: el planificador y el scheduler cooperarán para obtener el resultado como un plan ejecutable y óptimo.

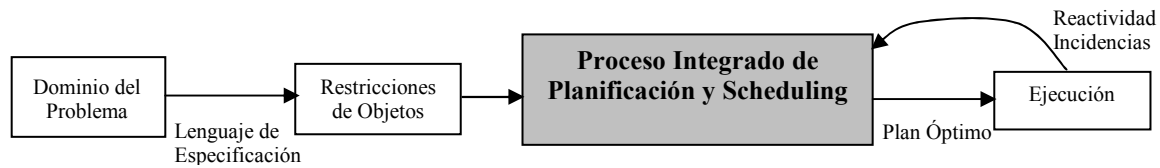


Fig. 1. Visión global del sistema.

En el modelo propuesto (Fig. 1), existe una integración de los procesos de planificación y scheduling. La resolución del problema comienza con la definición del dominio del problema realizado por el usuario mediante el lenguaje de especificación introducido en el apartado 2. El usuario define los objetos, restricciones sobre los objetos, acciones, recursos y objetivos del problema que serán utilizados en ambos procesos. Una vez obtenido el plan óptimo mediante el proceso de planificación-scheduling, descrito en el apartado 3.1, éste pasa a ejecución. Durante la ejecución del plan pueden aparecer nuevas restricciones del problema, incidencias, etc., debiéndose producir una reactividad en el proceso readaptando el plan obtenido.

3.1. Estructura y comportamiento del proceso integrado.

La estructura interna del proceso integrado de planificación y scheduling puede verse en la Fig. 2. Una cuestión inicial es la representación de la información. Por una parte, el planificador debe acceder a los datos referentes a acciones y a su ordenación, situación inicial, metas, etc. Por otra parte, el scheduler debe mantener todo lo relacionado con la asignación y utilización de recursos, restricciones temporales (tanto de recursos como de acciones), e incluso el orden entre acciones. De esta forma, existe una información común a ambos procesos. Por un lado, puede optarse por un esquema distribuido de la información con mayor flexibilidad y autonomía. Sin embargo, a fin de evitar la réplica de la información común a ambos procesos y permitir un mayor nivel de integración se ha optado por una estructura de información común y compartida. Estos datos globales (Fig. 2) corresponden a los objetos del problema, acciones, metas, recursos, restricciones temporales y de utilización de recursos, etc. Todos estos objetos se almacenan en el espacio global al que accede tanto el planificador como el scheduler. Además de esta información común, los dos procesos dispondrán de una memoria local en la que se mantendrán los datos privados a cada proceso, como asignaciones locales o estructuras internas del mismo, y que no sean necesarios a nivel global.

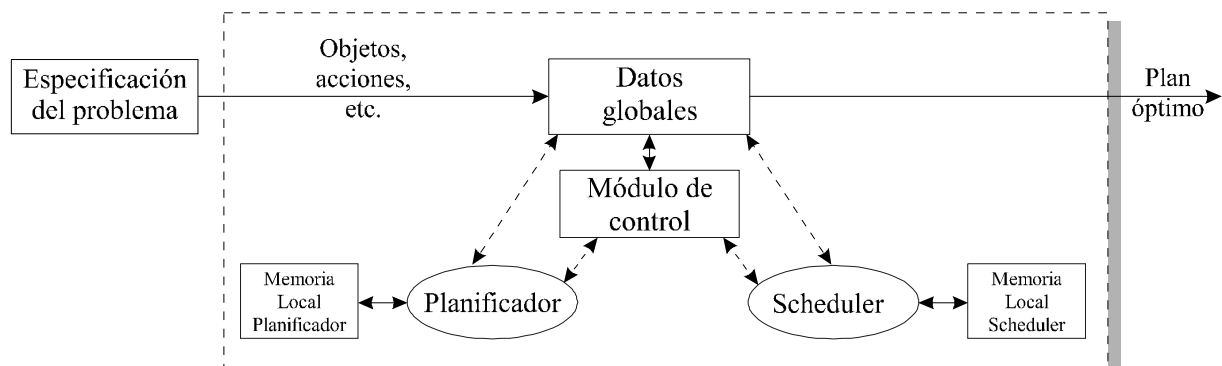


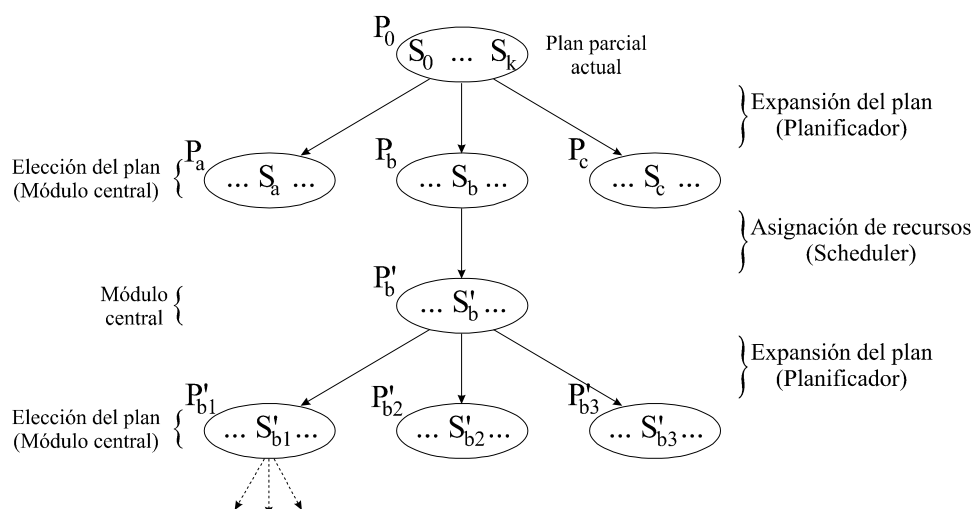
Fig. 2. Estructura del proceso integrado de planificación y scheduling.

Una vez creados todos los objetos del dominio, empieza el proceso de planificación y scheduling propiamente dicho. El módulo de control, encargado de gestionar la interactividad

entre planificador y scheduler, solicita al planificador planes parciales que resuelvan alguno de los objetivos del problema. Estos planes parciales son devueltos a la memoria global y el módulo de control solicita interactivamente al scheduler una validación de acuerdo a las restricciones existentes y, si es necesario, una asignación de recursos a las acciones. Estos recursos pueden venir definidos inequívocamente en la propia acción, o bien asignados por el planificador en su propio proceso. En caso contrario, el scheduler efectúa la asignación de recursos a las acciones de acuerdo a los requerimientos de las mismas especificados en su propia descripción, a la disponibilidad de recursos, a las restricciones previamente existentes y a los propios criterios de optimización residentes en el scheduler. Cada asignación de recursos es, por tanto, devuelta por el scheduler a la memoria global.

Estudiemos con un poco más de detalle este proceso (Fig. 3). Partiendo de un plan parcial actual P_0 , se van expandiendo sucesivamente los planes parciales que conduzcan a la situación final u objetivo. Los pasos que se siguen son los enumerados a continuación:

1. El módulo central pasa el control al planificador que devuelve varios planes alternativos (P_a , P_b y P_c), que introducen nuevos pasos (S_a , S_b y S_c) por los que se puede continuar.
2. El módulo central selecciona el plan (P_b) por el que desea continuar de acuerdo a unos criterios de optimización. Para poder tomar esta decisión debe solicitar al sistema de planificación y al sistema de scheduling sus respectivos criterios de optimización. Una vez recogida esta información la ponderará y obtendrá el plan más prometedor, que será el siguiente a ser expandido. La justificación de que sea el módulo central el que elija el mejor plan viene del hecho de que un plan puede ser óptimo de acuerdo a algún criterio del planificador y no serlo desde el punto de vista del scheduler, y viceversa. Por este motivo el módulo central recogerá toda la información y obtendrá un plan óptimo desde un punto de vista global.
3. La información sobre el plan seleccionado (P_b) es recogida por el scheduler que se encarga de realizar la asignación de recursos si fuera necesario y la comprobación de restricciones temporales. Con ello se obtiene el nuevo plan resultante (P_b'). Puede ocurrir que un orden parcial entre las acciones se restrinja por imposición del scheduler, debido por ejemplo, a una restricción en cuanto a la utilización de recursos. Por lo tanto, la secuencia de acciones impuesta por el scheduler será siempre igual o más restrictiva que la impuesta por el planificador.
4. El módulo central invoca al planificador para que genere los planes sucesores de P_b' , solicitando la resolución de alguna precondition del mismo plan volviendo al punto 1. Este com-



portamiento se mantendrá sucesivamente hasta obtener una solución que resuelva todos los objetivos del problema y que cumpla todas las restricciones del mismo.

Fig. 3. Generación de planes y asignación de recursos gestionado por el módulo central.

Como puede observarse en la figura anterior, los dos procesos de planificación y scheduling comparten la información para obtener una integración completa. El módulo central se encarga de activar ambos procesos y parametrizar su activación en base a criterios globales. Adicionalmente, la expansión de planes es gestionada por el módulo central cuyo criterio es el resultado de una ponderación de valoraciones del planificador y del scheduler. Por otra parte, la poda de planes puede producirse tanto por inconsistencias en el planificador, debido a amenazas irresolubles, como en el scheduler debido al incumplimiento de restricciones temporales o uso de recursos. Esta eliminación de estos planes parciales es una de las principales ventajas de la integración de los procesos de planificación y scheduling.

3.2. Proceso de Planificación.

El objetivo en planificación (Yang 97) es construir un plan como una secuencia de acciones para conseguir el objetivo del problema. El planificador que se propone está basado en un planificador POCL (*Partial Order Causal Link*). Durante muchos años, esta aproximación ha sido considerada como válida para la resolución de problemas de planificación (Weld 99). No obstante, en los últimos años, la planificación POCL parece haber sido abandonada en favor de nuevas aproximaciones como GraphPlan (Blum 97). Sin embargo, y de acuerdo al objetivo global planteado respecto a la integración de los procesos de planificación y scheduling, consideramos más adecuado utilizar una aproximación de planificación POCL debido a que está ampliamente estudiada y permite una integración más intuitiva por la forma de generar los planes.

Para añadir restricciones de alto nivel entre acciones, usamos el concepto previamente introducido de macro-acciones. En un primer nivel de planificación, se obtiene un plan abstracto, compuesto por una secuencia parcialmente ordenada de macro-acciones, que se refinará en sucesivos niveles de planificación, de forma similar a la planificación de Redes Jerárquicas de Tareas o HTN (Currie 91).

3.3. Proceso de Scheduling.

El objetivo del proceso de scheduling es asegurar la corrección del plan en función de los recursos disponibles y de las restricciones temporales impuestas por el problema. Adicionalmente, dispone de criterios para garantizar la optimalidad del plan generado, de acuerdo a funciones de evaluación de costes y otros criterios. En el proceso de scheduling, a menudo, se utiliza un problema de satisfacción de restricciones o CSP que pueda evaluar todos los posibles valores de asignación a las variables del problema (Fox 90, Freuder 92, Kumar 92, Sadeh 96), buscando aquella solución que cumpla con todas las restricciones iniciales del problema. Sin embargo, en la arquitectura integrada que proponemos, las restricciones no son conocidas a priori, ya que van apareciendo a medida que el planificador avanza en la generación del plan. Por lo tanto el scheduler no puede comportarse como un típico CSP, requiriendo en cambio asegurar la consistencia, propagando los efectos de las nuevas restricciones, introducidas interactivamente en cada nuevo plan parcial que vaya generando el planificador. Igualmente, debe evitar hacer asignaciones innecesarias en las etapas iniciales. Por lo tanto, y gracias a esta característica, el scheduler contará con un mayor grado de flexibilidad al mantener, únicamente, las ventanas temporales en las que las acciones pueden ejecutarse. En lugar de fijar el instante en el que ejecutar una acción o utilizar un recurso, mantiene una restricción consistente sobre el intervalo en el que dicha operación debería llevarse a cabo.

El sistema de scheduling para resolver este problema se basa en un módulo gestor de restricciones temporales disyuntivas y métricas (*Temporal Constraint Network Manager –TCNM-*) que trata un *Temporal Constraint Satisfaction Problem –TCSP-* (Barber 98, Dechter 91). El gestor necesita validar las restricciones garantizando la consistencia y minimalidad. Los algoritmos del TCNM se basan en un modelo temporal métrico-disyuntivo entre puntos de tiempo e intervalos (Baptiste 95a, Barber 93).

3.4. Tratamiento de las inconsistencias.

Las inconsistencias pueden producirse tanto en el planificador como en el scheduler. Las inconsistencias del planificador son las producidas por una amenaza irresoluble. Así, el planificador comunica la inviabilidad de dicho plan al módulo central que lo eliminará. Las inconsistencias provocadas por el scheduler se originan por una asignación de recursos no posible o por una violación de alguna restricción temporal. En este caso se produciría un backtracking en la etapa de scheduling controlado por el módulo central, desechándose el plan actual. Sin embargo, puede no ser necesario modificar la secuencia de acciones que constituyen el plan actual. Podría ser suficiente una reasignación de recursos a las acciones del plan que garantice que la secuencia de acciones existente, junto con la nueva asignación de recursos, sea correcta y consistente. En la Fig. 3, este tipo de backtracking no implica ningún *salto* de rama, sino únicamente una reasignación de recursos dentro del mismo plan. No obstante, la modificación de recursos puede obligar a una readaptación en el orden de las acciones para así cumplir las restricciones entre recursos.

3.5. Criterios de optimización.

Para hacer más eficiente el proceso de resolución del problema se pueden aplicar varios criterios de decisión. Por una parte, se puede considerar el criterio para hacer más eficiente el proceso de búsqueda de la solución (obtención de un plan finalmente ejecutable). Por otra parte existen distintos criterios para mejorar la calidad de la solución. Para hacer más eficiente el propio procedimiento de planificación y de scheduling se pueden utilizar las mismas técnicas empleadas en los enfoques tradicionales de planificación y de scheduling que separaban ambos procesos (Srivastava 99). Por ejemplo, el planificador puede establecer como mejor solución aquel plan con menor número de acciones, mientras que el scheduler puede dar prioridad a aquella asignación de recursos que consuma el mínimo número de ellos. Para poder incrementar el grado de optimalidad de la solución mientras se genera, deberían aplicarse nuevas técnicas integradas.

Como se ha comentado, es el módulo central el que, una vez obtenida la información sobre los distintos criterios de optimalidad, toma las decisiones acerca de qué plan es la *mejor* solución. Para tomar estas decisiones, se ponderan los criterios de optimalidad del planificador y del scheduler. Esta función de ponderación puede ser dinámica, es decir, variar la importancia de un criterio u otro de acuerdo al estado de resolución del problema.

4.- CONCLUSIONES.

En este artículo se ha expuesto un modelo para el desarrollo de una arquitectura integrada de planificación y scheduling, cuyo objetivo es la construcción de una herramienta para resolver problemas reales. Mediante un lenguaje de especificación de problemas y con los beneficios que se prevén de la integración de los procesos de planificación y scheduling, se pueden resolver problemas que incluyan restricciones temporales y utilización de recursos limitados. Mediante los distintos criterios de optimización comentados se puede mejorar la eficiencia en el proceso de resolución y obtener unas soluciones que garanticen un cierto grado de calidad u optimalidad.

Actualmente se dispone de una primera versión de la arquitectura propuesta, implementada en CLOS (*Common Lisp Object System*). El objetivo de esta versión es validar la arquitectura propuesta y la interacción entre los procesos. Los resultados obtenidos muestran la viabilidad de esta propuesta, requiriendo sin embargo profundizar en los criterios de optimización de cada proceso y la combinación de los mismos en el módulo central. Adicionalmente, se trata la incorporación de restricciones sobre el estado de los objetos y sobre la persistencia de los efectos de las acciones, así como el tratamiento de recursos renovables y parcialmente utilizables. Como futuras líneas de investigación, nos proponemos la inclusión de nuevas técnicas de planificación, como GraphPlan, la incorporación de restricciones temporales más expresivas en el

scheduler (Jonsson 98) y la mejora de la reactividad en la ejecución del plan debido a las incidencias producidas en su ejecución.

REFERENCIAS.

- Baptiste, P. y C. Le Pape (1995a): A Theoretical and Experimental Comparison of Constraint Propagation Techniques for Disjunctive Scheduling. In Morgan Kaufmann: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp.600-606, 1995.
- Baptiste, P., C. Le Pape y Nuijten, W. (1995b): Incorporating Efficient Operations Research Algorithms in Constraint-Based Scheduling. *First International Joint Workshop on Artificial Intelligence and Operations Research*. Timberline Lodge, Oregon, 1995.
- Barber, F. (1993): A Metric Time-Point and Duration-Based Temporal Model. *SIGART-ACM Bulletin* (4), 3, pp.30-49, 1993.
- Barber, F. (1998): Reasoning on complex disjunctive temporal constraints. An extended version. *Technical Report*, DSIC-II/17/1998, Dpto. de Sistemas Informáticos y Computación, UPV, España, 1998.
- Beck, H. y Tate, A. (1995): Open Planning, Scheduling and Constraint Management Architectures. In the *British Telecommunication's Technical Journal*, Special Issue on Resource Management, 1995.
- Bertolotti, E., Giannone, G. (1993): Jobs sequencing in a weaving factory. *IJCAI-93 Workshop on Knowledge-Based Production Planning, Scheduling, and Control*, 1993.
- Blum, A.L. y Furst, M.L (1997): Fast Planning through Planning Graph Analysis. *Artificial Intelligence* (90), pp.281-300, 1997.
- Currie, K. y Tate, A. (1991): O-Plan: the Open Planning Architecture. *Artificial Intelligence* (52), pp.49-86, 1991.
- Dechter, R., Meiry, J. y Pearl, J. (1991): Temporal Constraints Networks. *Artificial Intelligence* (49), pp.61-95, 1991.
- Dorn, J., y Froeschl, K. (1993): *Scheduling of Production Processes*. Ed. Ellis Horwood, 1993.
- Fox, M. S. y Sadeh, N. (1990): Why is Scheduling difficult? A CSP Perspective. *9th European Conference on Artificial Intelligence*, 1990.
- Freuder, E. C. y Wallace, R. J. (1992): Partial constraint satisfaction. *Artificial Intelligence* (58), pp.21-70, Elsevier, 1992.
- Gervasio, M. T. y DeJong, G. F. (1992): A Completable Approach to Integrating Planning and Scheduling. In Morgan, Kaufmann: *Artificial Intelligence Planning Systems: Proceedings of the First International Conference (AIPS 92)*, pp.275-276, 1992.
- Jonsson, P. y Bäckström, C. (1998): A unifying approach to temporal constraint reasoning. *Artificial Intelligence* (102), pp.143-155, 1998.
- Kumar, V. (1992): Algorithms for Constraint Satisfaction Problems: A Survey. *AI Magazine* 13(1), pp.32-44, 1992.
- Penberthy, S. y Weld, D.S. (1992): UCPOP: A sound, complete, partial-order planner for ADL. In Morgan, Kaufmann: *Proc. of the 1992 International Conference on Principles of Knowledge Representation and Reasoning*, pp.103-114, Los Altos, CA, 1992.
- Sadeh, N. M. y Fox, M. S. (1996): Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem. *Artificial Intelligence* (86), pp.1-41, 1996.
- Srivastava, B. y Kambhampati, S. (1999): Efficient Planning Trough Separate Resource Scheduling. *AAAI Spring Symp on Search Strategy under Uncertain and Incomplete Information*, Mar 1999.
- Weld, D.S. (1999): Recent Advances in AI Planning. Technical Report UW-CSE-98-10-01. October 1998. To appear in *AI Magazine*, 1999.
- Yang, Q. (1997): *Intelligent Planning. A Decomposition and Abstraction Based Approach*. Ed. Springer-Verlag, Berlin, Heidelberg, 1997.

ANEXO I. GRAMÁTICA DEL LENGUAJE DE ESPECIFICACIÓN.

;Jerarquía de objetos

```

<class-definition> ::= (class <class-name> [(is-a <superclass-depend>)]
                        [<class-role>] {<slot-definition>}*)
<superclass-depend> ::= RESOURCE | {<class-name>}+
<class-role>        ::= (role concrete | abstract)
<slot-definition>  ::= (slot <slot-name>
                        [(type <data-type>)]           [(occurs temporal | non-temporal)]
                        [(range <slot-value> <slot-value>)] [(values {<slot-value>}+)]
                        [(default <slot-value>)]         [(allocation class | instance)])
<class-name>       ::= symbol

```

