

GENERADOR DE MOTORES DE SCHEDULING EN DOMINIOS INDUSTRIALES

Daniel Díaz, Francisco Ibañez, Raymundo Forradellas

LISI- Instituto de Informática – Dpto. de Informática

FCEfyN - Universidad Nacional de San Juan

CUIM – Av. Ignacio de la Roza 590 (O), Rivadavia – J5402DCS San Juan

{ddiaz, fibanez, kike}@iinfo.unsj.edu.ar

CONTEXTO

Este trabajo está encuadrado en una línea de I+D+i que este grupo de trabajo desarrolla en el LISI (Laboratorio Integrado de Sistemas Inteligentes) desde 1996, orientado a resolver problemas complejos de Planning, Scheduling y Asignación de Recursos, en el ámbito de los sistemas industriales. El trabajo actual guarda relación con esta línea, incorporando nuevas tecnologías y funcionalidades provenientes de la ingeniería de software.

RESUMEN

En este trabajo se describe una propuesta para desarrollar sistemas de scheduling en el dominio de sistemas industriales, utilizando como base un generador de motores de scheduling, para una instancia específica de un dominio determinado. El constructor de motores de scheduling utiliza una combinación de métodos provenientes de la ingeniería del software y de la inteligencia artificial. Más específicamente, se presenta una integración entre SPL - *Software Product Line*, y la programación con restricciones.

Palabras clave: Scheduling, Inteligencia Artificial, Reutilización de software, Software Product Line.

1 INTRODUCCION

Existen varias definiciones para scheduling: “Scheduling es el problema de asignar recursos limitados a tareas en el tiempo con el objeto de optimizar uno o más objetivos” (Barker, 1974). “Scheduling trata con la asignación temporal de trabajos a recursos limitados donde se debe considerar un conjunto de restricciones” (Sauer, 2000b). Los problemas de scheduling aparecen en una gran cantidad de dominios, en este artículo tratamos con los problemas de scheduling en la industria, los cuales se conocen como *production scheduling* o programación de la producción.

Considerando la definición del problema de scheduling, la solución constituye la asignación temporal de máquinas a trabajos (referidos también como actividades o tareas). En la industria, un sistema de scheduling es el corazón del sistema de planificación y control de la producción (Vollmann et al., 1997). Normalmente el sistema de scheduling interactúa con otros sistemas de fabricación. Por ejemplo, un sistema de scheduling necesita conocer la capacidad de los recursos y disponibilidad de la materia prima. Este tipo de información la brinda el MRP - *Material requirement planning* (Orlicky, 1975). El sistema de scheduling genera como salida un plan de producción, el cual es puesto en práctica por la planta.

El sistema encargado de gestionar la operación de planta es el MES - *Manufacturing Execution System* (Boucher and Yalcin, 2006). Este sistema interactúa con el sistema de scheduling a fin de adecuar el scheduling predictivo al scheduling real (re-scheduling). Los sistemas contenidos en el ERP - *Enterprise Resource Planning* (Shanks and Seddon, 2000) también pueden interactuar con sistema de scheduling. A modo de ejemplos, ventas puede observar a futuro la carga de máquinas y así ajustar su política de ventas, los mantenimientos preventivos se pueden planificar en función de las simulaciones de carga de máquinas obtenidas desde el sistema de scheduling.

Según (Yen and Pinedo, 1994), un sistema de scheduling se compone diferentes módulos: (1) módulos de base datos y base de conocimiento, (2) módulos del motor de scheduling y (3) módulo de interface de usuario.

1.1 EVOLUCIÓN DE LOS SISTEMAS DE SCHEDULING

La investigación en el diseño y desarrollo de sistemas de scheduling se puede clasificar en 4

generaciones. Inicialmente, esta clase de problemas fue abordada por el área de la ingeniería industrial, y posteriormente por el dominio de la investigación operativa. En los comienzos del siglo pasado, Henry L. Gantt (Gantt, 1903), fue uno de los primeros en proponer un sistema de scheduling: *daily balance* “el equilibrio diario” que consiste en un método de programar y de registrar el trabajo. En (MacNiece, 1951), se discuten dos técnicas de scheduling *Loading*, y *Planning Board*. En 1960 se desarrolló *Planalog Control Board*, una herramienta que constaba de una pizarra y entre otras cosas podía forzar restricciones de precedencia. Los primeros sistemas de scheduling computarizados nacieron a la par del MRP - *Material Requirement Planning*, (Wight, 1984). Estos fueron desarrollados con las primeras tecnologías de hw y sw disponibles. En aquel entonces el desarrollo requería un gran esfuerzo y sus costos y tiempos eran elevados. En la segunda generación surgen los ISS - *Intelligent Scheduling Systems*, tales como ISIS (Fox and Smith, 1984) y OPIS de (Smith, 1994). Esta fue una etapa donde el problema de scheduling se comenzó a abordar con técnicas de inteligencia artificial tales como algoritmos genéticos, *simulated annealing*, redes neuronales y *constraint satisfaction techniques*. En esta generación aparecen los sistemas de scheduling que se construyen utilizando técnicas de KBS - *Knowledge Based Systems* (Sauer, 2000a). La tercera generación se inició con el surgimiento del paradigma orientado, dando origen a sistemas de scheduling desarrollados con esta tecnología (Yen and Pinedo, 1994). Esto permitió el desarrollo de *frameworks* especializados en scheduling, los cuales encapsulan técnicas de inteligencia artificial (Le Pape, 1994).

Una cuarta generación de sistemas de scheduling está surgiendo actualmente. En esta etapa, el constante avance tecnológico hace que la arquitectura de sistemas de la empresa crezca día a día y esto requiere que los sistemas sean integrables, flexibles, de rápida implementación y bajo mantenimiento. La flexibilidad tiene un alto impacto en los costos de mantenimiento, dado que modificar un

sistema inflexible suele ser costoso, tanto en tiempo como en dinero. Últimamente la reutilización sistemática está dando lugar a esta cuarta generación de sistemas de scheduling, así lo demuestran los últimos trabajos de investigación sobre sistemas de scheduling en el área de KBS (Rajpathak et al., 2003). Esta propuesta adhiere a esta tendencia, usando técnicas de inteligencia artificial embebidas en *frameworks* orientados a objetos para resolver problemas de scheduling, y por otro lado, utilizando los últimos avances en la ingeniería de software, como son el desarrollo de familias de productos de software y SPL, para modelar, diseñar, implementar sistemas de scheduling.

1.2 PROBLEMA DE INVESTIGACIÓN

El objetivo principal del tema de investigación es demostrar que dado un dominio industrial, es posible desarrollar un proceso constructor de motores de scheduling, considerando una instancia X del dominio industrial.

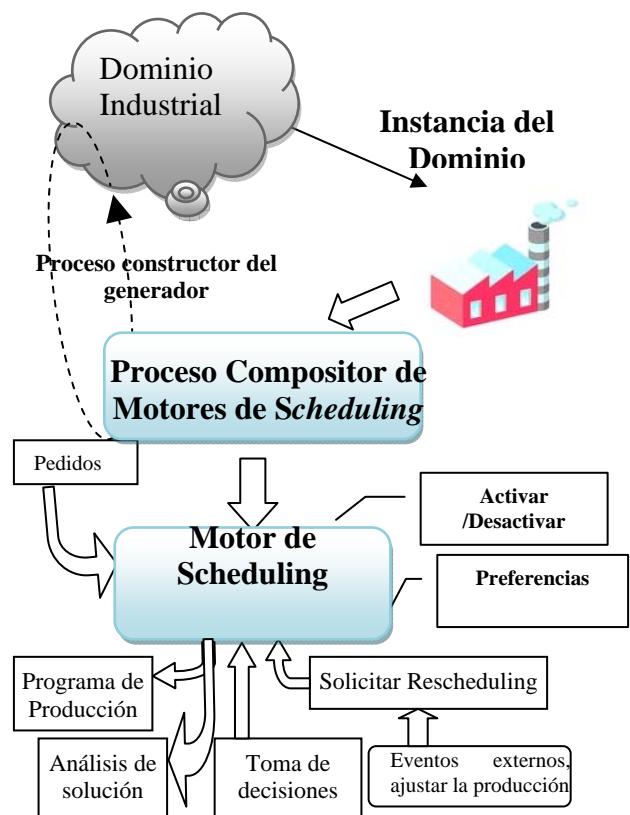


Figura 1 Problema de Investigación

La instancia queda descrita por un modelo conceptual, que describe a las máquinas, procesos, hojas de rutas, y demás objetos de la

instancia. Este modelo conceptual constituye la entrada al proceso compositor de motores de solución, el cual produce como salida, un motor de solución que debe responder a las distintas instancias del problema que tiene dicha compañía.

Una instancia del problema queda determinada por las distintas necesidades que ocurren en el día a día de la empresa. Por ejemplo, máquinas no disponibles, nuevos pedidos, urgencias, etc. Cada instancia es en realidad un problema de scheduling.

Los componentes básicos del problema son “el proceso compositor de motores de solución” y el “motor de solución”, los cuales se muestran en la Figura 1.

1.3 PROPUESTA DE SOLUCIÓN

En esta propuesta, se muestra que es posible construir un generador de motores de scheduling para dar solución al problema planteado, mediante programación por restricciones y SPL.

La programación por restricciones es una de las técnicas que permite solucionar problemas de scheduling (Barták., 2003) (Baptiste et al., 2001).

La ingeniería de SPL es un paradigma para desarrollar aplicaciones de software (sistemas y productos de software intensivos) usando una plataforma de componentes reusables y de customización en masa (Pohl et al., 2005).

Nuestra propuesta tiene como eje principal la reutilización, tanto de la construcción con reuso y para reuso. La ingeniería SPL utiliza el reuso como camino para construir sistemas, utilizando los conceptos de la ingeniería de dominios, pero acotando el dominio bajo estudio a las características del conjunto concreto de productos (existentes, planeados, o futuros) a desarrollar.

La ingeniería de línea de productos de software tiene dos procesos (Weiss and Lai, 1999, Boeckle et al., 2004, van der Linden, 2002).

La Ingeniería de Dominio: Es el proceso responsable de establecer la plataforma reusable y por lo tanto definir las similitudes y las variabilidades de la línea de producto.

La Ingeniería de Aplicación: Es el proceso responsable de derivar aplicaciones de la línea

de producto, a partir de la plataforma establecida en la ingeniería de dominio.

Regresando a nuestro problema, el Proceso Compositor de Motores de Scheduling es una línea de productos. Podemos ver entonces que este proceso se compone de dos subprocesos, ingeniería de dominio e ingeniería de aplicación. En el primero se establece una plataforma reusable que contiene todos los artefactos de software para construir un motor de scheduling a medida para una instalación industrial del dominio bajo estudio. Mientras que el segundo proceso, la ingeniería de aplicación, construye el motor de scheduling.

Nuestro proceso de construcción de motores de scheduling también adhiere al enfoque generativo de desarrollo de software (Czarnecki, 2005), permitiendo la generación automáticamente de una aplicación de la línea de producto, mediante la definición y uso de un DSL – *Domain Specific Language* o Lenguaje de Propósito Específico. (Greenfield and Short, 2004, Mernik et al., 2003, Fowler, 2005). En nuestro caso el modelo conceptual de la instancia del dominio se formaliza usando un DSL. Así, el proceso generador toma esta especificación y genera un motor de scheduling para la instancia del dominio. Una de las ventajas de usar DSL es que el motor puede ser construido por usuarios expertos del dominio, sin que sean expertos en tecnologías informáticas. Cuando el motor no se adecua a la realidad de la planta, se puede modificar usando el conocimiento del dominio, sin necesidad de programación. Sin embargo, para cada instancia (empresa del dominio) se tienen diferentes necesidades que no pueden ser capturadas al momento de definir la plataforma, y por ese motivo, la ingeniería de dominio resulta un proceso que está permanentemente modificando o agregando conocimiento a la plataforma reusable. Por último, otra particularidad de este enfoque es que el conocimiento obtenido en cada implementación se puede proveer a las demás instancias del dominio.

2 LINEAS DE I + D

Las líneas de I+D son las siguientes:

- Algoritmos de scheduling para soportar preferencias parametrizables
- Calidad en Scheduling
- Software product lines y software product families
- Enterprise reference models y Scheduling References models
- Variabilidad y similitudes en constraint programming
- Implementaciones en la industria del envase flexible

3 RESULTADOS OBTENIDOS/ESPERADOS

Como resultados obtenidos se pueden mencionar los siguientes: (1) Formulación del problema, (2) Estudio de estado del arte sobre problemas de scheduling, técnicas de solución de problemas y sistemas de scheduling, integración del sistema de scheduling, *software product lines* y *software product families*. (3) Vinculación con empresas que tienen la problemática abordada, tareas de campo. (4) Obtención de una beca para hacer una pasantía de 6 meses en el Institut National Polytechnique de Lorraine (INPL) ubicado en Nancy Francia con fecha de comienzo en septiembre de 2009.

Como tareas actuales, se esta terminado el reporte de estado del arte en la temática abordada (examen de calificación programa de doctorado, capitulo de tesis).

Diseñando e Implementando un prototipo de generador de motores de scheduling para el dominio de la industria del envase flexible.

Como tareas Futuras. Generalizar el método para generador de motores de scheduling.

Publicar resultados, redactar y defender la tesis.

4 FORMACION DE RECURSOS HUMANOS

La Formación de Recursos Humanos está sustentada en el trabajo de tesis de doctorado de Daniel Diaz Araya becario de CONICET con beca Tipo II.

5 BIBLIOGRAFIA

BAPTISTE, P., PAPE, C. L. & NUIJTEN, W. (2001) *Constraint-Based Scheduling - Applying Constraint Programming to Scheduling Problems*, Springer.

BARKER, K. R. (1974) *Elements of sequencing and scheduling*, New York, John Wiley and Sons.

BARTÁK., R. (2003) Constraint-based Scheduling: An Introduction for Newcomers. *Intelligent Manufacturing Systems* 69-74.

BOECKLE, G., KNAUBER, P., POHL, K. & SCHMID, K. (2004) *Software-Produktlinien: Methoden, Einführung und Praxis (in German)*, Heidelberg, dpunkt.

BOUCHER, T. & YALCIN, A. (2006) *Design of Industrial Information Systems*, Academic Press.

CZARNECKI, K. (2005) Overview of Generative Software Development. *Unconventional Programming Paradigms*.

FOWLER, M. (2005) Language Workbenches: The Killer-App for Domain Specific Languages?

FOX, M. S. & SMITH, S. F. (1984) ISIS - a knowledge-based system for factory scheduling. *Expert Systems*, 1, 25-49.

GANTT, H. L. (1903) A graphical daily balance in manufacture. *Transactions of the American Society of Mechanical Engineers*, 24, 1322-1336.

GREENFIELD, J. & SHORT, K. (2004) *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*, Indianapolis, Wiley.

LE PAPE, C. (1994) Implementation of Resource Constraints in ILOG SCHEDULE: A Library for the Development of Constraint-Based Scheduling Systems. *Intelligent Systems Engineering* 3, 3, 55-66.

MACNIECE, E. H. (1951) *Production Forecasting, Planning, and Control*, New York, John Wiley & Sons.

MERNIK, M., HEERING, J. & SLOANE, A. M. (2003) When and how to develop domain-specific language. *Technical Report SEN-E0309* Amsterdam, CWI Available from <http://www.cwi.nl/ftp/CWIreports/SEN/SEN-E0309.pdf>.

ORLICKY, J. (1975) *MRP, The New Way of Life in Production and Inventory Management*, McGraw-Hill Book Company.

POHL, K., BÖCKLE, G. & LINDEN, F. J. V. D. (2005) *Software Product Line Engineering: Foundations, Principles and Techniques*, Springer.

RAJPATHAK, D., MOTTA, E., ZDRAHAL, Z. & ROY, R. I. (2003) A Generic Library of Problem Solving Methods for Scheduling Applications. *Second International Conference on Knowledge Capture (KCAP'2003)*. Florida, USA,.

SAUER, J. (2000a) Knowledge-based Systems in Scheduling. IN LEONDES, T. L. (Ed.) *Knowledge-Based Systems Techniques and Applications*. San Diego, USA, Academic Press.

SAUER, J. (2000b) Knowledge-Based Systems Techniques and Applications in Scheduling. IN

- LEONDES, C. T. (Ed.) *Knowledge-Based Systems Techniques and Applications* Academic Press.
- SHANKS, G. & SEDDON, P. (2000) Enterprise resource planning (ERP) systems. *Journal of Information Technology*, 15, 243-244.
- SMITH, F. S. (1994) *OPIS: A Methodology and architecture for reactive scheduling*, Morgan Kaufmann.
- VAN DER LINDEN, F. (2002) Software Product Families in Europe: The ESAPS and CAFÉ Projects. *IEEE Software*, 19, 41-49.
- VOLLMANN, T. E., BERRY, W. L. & WHYBARK, D. C. (1997) *Manufacturing Planning and Control Systems*, New York, Irwin/McGraw-Hill.
- WEISS, D. M. & LAI, C. T. R. (1999) *Software Product-Line Engineering - A Family-Based Software Development Process*, Massachusetts, Addison-Wesley.
- WIGHT, O. W. (1984) *Production and Inventory Management in the Computer Age*, Wight (Oliver) Publications Inc., U.S.
- YEN, B. P.-C. & PINEDO, M. (1994) On the design and development of scheduling systems. *Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*.