

Hacia un modelo híbrido de simulación de la producción de software en un entorno multiproyecto

Javier Navascués Fernández-Victorio¹, Isabel Ramos Román² y Miguel Toro Bonilla²

¹ Departamento de Organización Industrial y Gestión de Empresas, Universidad de Sevilla;

² Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla

¹jnavascues@us.es

Abstract. La simulación del ciclo de vida de los proyectos software o de partes de éste es un activo campo de investigación en la ingeniería del software. El presente informe analiza la literatura sobre modelos de simulación que puedan ser útiles para estudiar las organizaciones de desarrollo de software. La finalidad del trabajo es construir un modelo de simulación en el que se puedan implementar diversas políticas de asignación de recursos basadas en las técnicas de gestión de proyectos propias de la investigación operativa en un contexto de mejora de procesos y teniendo en cuenta el carácter multiproyecto.

La simulación de proyectos software

La simulación de proyectos software es un campo de investigación que registra un importante nivel de actividad. En la tesis de Mercedes Ruiz Carreira (18) se recoge una relación de trabajos de simulación del proceso software entre los años 1991 y 2000. En el origen de todos ellos está el trabajo de Abdel-Hamid y Madnick (2) que supuso un exhaustivo intento de modelar el proceso de inspección formal con un grado de detalle muy elevado. Reúne las actividades relacionadas con la gestión de recursos humanos, la producción, la planificación y el control en un modelo de Dinámica de Sistemas que se ha convertido en una referencia en este campo de investigación. A partir de ahí en la misma fuente se recogen referencias a una serie de trabajos que han ido enfocando diferentes perspectivas y problemas dentro del marco general antes citado.

Con posterioridad a la publicación de este trabajo, la producción no ha cesado, enriqueciéndose el campo de la producción de modelos y acometiéndose esfuerzos de modelado híbrido dentro del campo de la simulación e integrándolo con métodos y herramientas de otros campos, contemplándose:

- El empleo simultáneo de varios paradigmas de modelado (continuo y discreto, fundamentalmente)
- La integración entre la simulación y la aplicación de métodos y modelos de optimización del campo de la Investigación Operativa y de la Inteligencia Artificial

- La integración de la simulación dentro de herramientas de ayuda a la decisión en tiempo de ejecución básándose en la metodología de agentes
- La integración de la simulación en modelos generales de representación y explotación del conocimiento

Desde el punto de vista académico y profesional la referencia fundamental es la serie de talleres PROSIM (*Workshop of Software Process Simulation and Modeling*) iniciada en 1998 y fusionada a partir de 2006 con el taller internacional SPW (*Software Process Workshop*), que se celebra en el marco de las conferencias ICSE (*International Conference on Software Engineering*) organizadas por la SCM y el IEEE (7,15,16,20).

El entorno multiproyecto

El entorno multiproyecto, con sus elevadas dependencias, está naturalmente inclinado a multiplicar los efectos cruzados de los diferentes bucles de realimentación que gobierna la evolución del estado de la organización de desarrollo. Es significativo, por tanto, que el esfuerzo por la simulación de este tipo de entorno sea relativamente reducido en comparación con el caso de un único proyecto. El propio Abdel Hamid (1) llama la atención sobre lo inapropiados que pueden ser los métodos y modelos de estimación cuando no tienen en cuenta las implicaciones a escala de toda la organización.

En el artículo citado (1) se presenta un trabajo en el que se modela específicamente la transferencia de personal entre varios proyectos que se están desarrollando en paralelo. La transferencia obedece a una serie de reglas relacionadas con la presión del plazo. Se realizan dos experimentos: uno de aceleración de los plazos y otro en el que se emula la regla MINSLACK para asignar recursos de la organización a los proyectos que compiten por ellos.

En (17) se presenta un modelo de dinámica de sistemas en el que se modela un ciclo de vida incremental en términos que conlleva relaciones de concurrencia entre diferentes paquetes de trabajo que, a otra escala, equivalen a la competencia por recursos de varios proyectos simultáneos, con la complicación adicional que dicha interrelación se establece no sólo por esa competencia por los recursos sino también por la información que fluye de un paquete de trabajo a otro, acoplándolos por tanto también a través de ese mecanismo.

Los autores proponen un modelo sencillo basado en una estructura simplificada del modelo de Abdel Hamid semejante a la recogida en (19), que relaciona recursos, tiempo y esfuerzo con el paquete de trabajo. Esta sencilla estructura se denomina el *triángulo del proceso*. Este triángulo puede definirse, modularmente, a diferentes niveles jerárquicos: paquetes de trabajo, fases, entregas, proyectos y organización en su conjunto.

Para acoplarlos diferentes módulos se emplea tanto la asignación de recursos a módulos como el acoplamiento de los trabajos. El primero es una regla de reparto de recursos entre paquetes de trabajo (o módulos de nivel superior) en base a prioridades y presiones de plazo. El acoplamiento de los trabajos implica tratar el trabajo de *corrección de errores* y las *transiciones*. La corrección de errores puede aparecer

dentro de un mismo paquete de trabajo o en otro subsiguiente. Trasladando la corrección de errores a una entrega posterior se *externalizan* estas tareas. Las transiciones se producen cuando una fase madura lo suficiente como para iniciarse la siguiente. Las transiciones pueden producirse de diversas formas: mediante *hitos* o por *unidades* de trabajo que al concluirse se convierten en *inputs* de las siguientes. Esto permite modelar la superposición de fases. El objetivo último del modelo es analizar las estrategias de desarrollo incremental, tanto a nivel previo (planificación) como su implementación y su modificación en tiempo de ejecución.

La asignación de los recursos a los proyectos

En la literatura revisada aparecen básicamente tres enfoques:

1. Generar una solución a partir de un modelo de investigación operativa y simular luego las consecuencias de su aplicación, generalmente por medio del método de Montecarlo.
2. Generar el espacio de estados a través del proceso de simulación y luego aplicar un heurístico de la investigación operativa para optimizar el estado del sistema.
3. Modelar el decisor explícitamente como un agente dotado de las reglas heurísticas; en este caso se trata de una herramienta de apoyo a la dirección del proyecto para que evalúe las consecuencias de las decisiones adoptadas de forma dinámica

Los heurísticos aplicados son de diferentes tipos:

- reglas simples de prioridad
- programación dinámica (métodos aproximados)
- algoritmos de búsqueda \emph{greedy} (miopes)
- algoritmos genéticos
- *sequeaky wheel optimization*; un algoritmo de búsqueda en el espacio de las estrategias de prioridad

Antonio *et al.* (4), presentan una aproximación al problema basada en la teoría de colas y la simulación estocástica para el apoyo a la planificación, gestión y control de la asignación de personal a un proyecto de mantenimiento multifase. Se trata de la adaptación al efecto 2000, Y2K, de un gran sistema de información de una entidad financiera desarrollado en COBOL/JCL que requiere la constitución de diferentes equipos trabajando en varios centros geográficamente distribuidos. Se estudia una configuración centralizada desde el punto de vista del cliente así como otras configuraciones más desagregadas. Los métodos de teorías de colas y la simulación estocástica se emplean para evaluar las políticas de asignación de personal a los equipos y la posibilidad de alcanzar los hitos previstos para la ejecución del proyecto.

En otro segundo artículo (3) se emplean los *algoritmos genéticos* para generar soluciones para el problema de teoría de colas teniendo en cuenta la necesidad de corregir errores y de reprogramar y abandonar algunas de las tareas inicialmente previstas. La simulación de Montecarlo de las colas junto con los algoritmos genéticos permite generar secuencias y asignar personal a los equipos. El impacto posible del trabajo de corrección de errores, el abandono de tareas y los errores e

incertidumbres en las estimaciones iniciales se caracteriza con funciones de distribución que se utilizan como entradas para la simulación.

Padberg, en varios artículos(9, 10, 11, 12, 13, 14) presenta su trabajo en el que se modelan como problemas *markovianos* de decisión en los que se tienen en cuenta, además de la variación estocástica de la duración de las actividades, la posibilidad de vueltas atrás y retrasos. Para ello se generan políticas de secuenciación con un algoritmo aproximado de *programación dinámica*.

Con un conjunto de problemas similares pero que se diferencian en la necesidad mayor o menor de recursos especializados y el grado de integración entre los componentes del producto, se simula la aplicación de la política óptima generada por el algoritmo con las políticas clásicas de listas. Se comprueba que cuanto mayor es la necesidad de recursos especializados y cuanto menor es el grado de integración entre los componentes, más destaca la solución propuesta sobre las reglas de listas.

Browning *et al*, (21) presentan un modelo se basa en la *design structure matrix*. La varianza en la duración y el coste en los proyectos de desarrollo de producto son atribuibles según su criterio en gran medida a las iteraciones, algo que es común también en el proceso software. Dichas iteraciones pueden o no ocurrir dependiendo del comportamiento de una serie de variables que el modelo simula como variables estocásticas cuya probabilidad de ocurrencia depende de la existencia de determinados paquetes de información que activan un proceso de reelaboración. Un proceso de reelaboración puede, a su vez, desencadenar otros sucesivos afectando así al proyecto en su totalidad. Para hacer frente a este problema utilizan el método de Montecarlo para simular políticas de secuenciación robustas generadas por un algoritmo genético.

Özdamar (8) presenta una propuesta de aplicación de reglas de prioridad en proyectos definidos con el formalismo *fuzzy*. Presenta un heurístico para la programación de las actividades de un modelo de desarrollo en cascada que se modela utilizando la lógica borrosa. la duración de las actividades depende del modo de ejecución seleccionado representado por una función de pertenencia trapezoidal. También se modela la disponibilidad de los recursos diferenciando entre dos categorías, una de personal más cualificado que comparte varias tareas y - por tanto - assignable de forma continua, y otra categoría de personal que no realiza multitareas y que se presenta en unidades discretas.

El objetivo del modelo es minimizar el tiempo de realización. Para ello se emplea un algoritmo de generación de secuencias en serie y se ensayan diferentes reglas de prioridad entre las más sencillas empleadas en los problemas de secuenciación. El modelo permite la resecuenciación dinámica para escoger las variaciones en las estimaciones a lo largo de la ejecución del proyecto.

En (5) Joslin presenta un modelo basado en agentes que simula de forma dinámica la reasignación de personal a las tareas en un proyecto con diferentes funcionalidades que deben entregarse en un plazo fijo.

Se trata de un sistema de ayuda a la decisión (DSS) basado en un heurístico denominado *Squeaky Wheel* que explora en el espacio de las estrategias en lugar de en el espacio de las soluciones. El simulador puede implementar diferentes reglas de asignación si bien la versión que se presenta en el artículo sólo presenta los primeros resultados para reglas muy sencillas.

La mecánica básica es la de reasignar el personal entre tareas conforme se aproximan los plazos y se detecta el riesgo de no cumplirlos. El modelo contempla el efecto aprendizaje cuando se produce un cambio de tarea y también el efecto de la sobrecarga de comunicación por lo que incorpora no linealidades el tipo de las que se presentan en los modelos clásicos de dinámica de sistemas. Los autores anuncian la futura implementación de reglas más sofisticadas, como las basadas en algoritmos genéticos.

Propuesta de un modelo híbrido multiproyecto

En el contexto descrito hasta aquí se encuentra a faltar una propuesta capaz de integrar los siguientes aspectos:

- El carácter multiproyecto de la producción de software
- El impacto de las actividades de medición y mejora de procesos en el comportamiento de los proyectos y del conjunto
- La implementación de estrategias de asignación de recursos sofisticadas

La propuesta es desarrollar un modelo dinámico de simulación jerárquico capaz de integrar estas facetas. El modelo de simulación será de tipo híbrido por la necesidad de capturar con la granularidad necesaria procesos y sucesos de naturalezas muy diferentes. Se pretende con ello poder modelar simultáneamente:

- El carácter discreto de:
 - El ciclo de vida de un producto software caracterizado por la sucesión de fases definidas y caracterizadas cada una de ellas por sus correspondientes entregables e hitos
 - Las decisiones de asignación (y reasignación) de recursos y programación (y reprogramación) de actividades
 - El proceso de registro para la obtención de métricas necesarias para la adopción de decisiones
 - Los eventos externamente determinados o derivados de las decisiones del punto anterior que pueden afectar al ciclo de vida o a cualquiera de sus fases
- El carácter continuo de:
 - El proceso de trabajo con el correspondiente consumo de esfuerzo y la eventual generación de errores
 - El entorno general de la empresa (movimiento laboral, ciclos de capacitación del personal, desarrollo de infraestructuras de soporte a los procesos de producción, etc.)
 - La mejora de las capacidades de los procesos

Si bien el problema específico que se desea modelar en esta investigación no tiene una correspondencia clara en la literatura, sí que hay en las referencias revisadas elementos suficientes con los que construir un modelo híbrido que permita representarlo. Los componentes serían:

- **Los modelos reducidos como "átomos".** El modelo presentado en (19) puede servir para constituir el núcleo básico de la simulación de la evolución de las tareas. Su integración en un marco más amplio obligará a redefinir los bucles de

realimentación que se cierran en el propio átomo y cuales traspasan las frontreas de estos.

- **Los triángulos de procesos.** La estructura de integración jerárquica de los \emph{triángulos de procesos} descritos por Powell permite a la vez modelar la jerarquía tareas - fases - proyectos - cartera y representar, a través de los tres primeros niveles un modelo de desarrollo incremental, que puede reducirse al modelo en cascada con facilidad.
- **La mejora del proceso como tarea del proceso.** La incorporación de las actividades de ingeniería y soporte como tareas específicas del proyecto como se propone en (23) permite modelar explícitamente el *trade-off* entre costes y beneficios de la mejora de procesos.
- **La orientación a agentes.** El modelo presentado por Joslin, orientado a la búsqueda en el espacio de las estrategias de priorización de las actividades permite simular las reglas heurísticas de asignación de recursos.

El formalismo DEVS, *discrete event system specification*, propuesto por Ziegler (22) posibilita representar con naturalidad un sistema híbrido al gestionar conjuntamente una simulación continua con un manejo muy flexible de la lógica de eventos y los flujos de información asociados al inicio y finalización de tareas y a la monitorización de las mismas. Este formalismo presenta además la ventaja de que el modelo básico, basado en un sistema muy simple denominado el DEVS *atómico* y en un acoplamiento elemental de dos átomos, es escalable y muy fácilmente modelable con la tecnología orientada a objetos. Esto permite generar una clase de modelos fácilmente parametrizables y reutilizables en el que se pueden diferenciar, y a la vez mantener coherentes, los procesos continuos del entorno de la organización y del propio desarrollo de las tareas elementales incluyendo los efectos de realimentación, y los procesos discretos tanto de flujos materiales (entregables, módulos, ...) como de medida, control y decisión.

Este trabajo se ha realizado dentro del proyecto “Calidad Predecible y gestionada mediante simulación y técnicas de prueba en etapas tempranas” (TIN 2007-67843-C06-03)

Bibliografía

- (1) AbdelHamid TK. A Multiproject perspective of single-project dynamics. *J.Syst.Software* 1993 sep.;22(3):151-165.
- (2) AbdelHamid TK, Madnick SE. *Software Project Dynamics An Integrated Approach*. Englewood Cliffs NJ: Prentice Hall; 1991.
- (3) Antoniol G, Cimitile A, Di Lucca GA, Di Penta M. Assessing staffing needs for a software maintenance project through queuing simulation. *IEEE Trans Software Eng* 2004;30(1):43-58.
- (4) Antoniol G et al, A robust search-based approach to project management in the presence of abandonment, rework, error and uncertainty. *Proceedings - 10th International Symposium on Software Metrics, METRICS 2004; 14 September 2004 through 16 September 2004; ; 2004.*
- (5) Joslin, D., Poole, W., Agent-based simulation for software project planning. *Winter Simulation Conference; 2005.*

- (6) Meier C, Yassine AA, Browning TR. Design Process Sequencing With Competent Genetic Algorithms. *Transactions of the ASME* 2007;129:566.
- (7) Münch J, Pfahl D. Special issue on ProSim 2005, The 6th international workshop on software process simulation and modeling, St. Louis, Missouri, USA may 2005. *Software Process: Improvement and Practice* 2006;11(4):339-343.
- (8) Özdamar L, Alanya E. Uncertainty Modelling in Software Development Projects (With Case Study). *Annals of Operations Research* 2001;102:157-178.
- (9) Padberg F. On the potential of process simulation in software project schedule optimization. *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International* 2005;2:127-130 Vol. 1.
- (10) Padberg F. Computing optimal scheduling policies for software projects. *Software Engineering Conference, 2004. 11th Asia-Pacific* 2004:300-308.
- (11) Padberg F. Linking software process modeling with Markov decision theory. *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International* 2004;2:152-155 vol.2.
- (12) Padberg F. A software process scheduling simulator. *Software Engineering, 2003. Proceedings. 25th International Conference on* 2003:816-817.
- (13) Padberg F. Using process simulation to compare scheduling strategies for software projects. *Software Engineering Conference, 2002. Ninth Asia-Pacific* 2002:581-590.
- (14) Padberg F. A study on optimal scheduling for software projects. *Software Process: Improvement and Practice* 2006;11(1):77-91.
- (15) Pfahl D, Raffo DM, Rus I, Wernick P editors. *ProSim 05. The 6th International Workshop on Software Process Simulation and Modeling. Proceedings : May 14-15, 2005, Adams Mark Hotel, St. Louis, Missouri, USA. : Fraunhofer IRB Verlag, Stuttgart;* 2005.
- (16) Pfahl D, Rus I. Special issue on ProSim 2004, The 5th International Workshop on Software Process Simulation and Modeling, Edinburgh, Scotland, May 2004. *Software Process: Improvement and Practice* 2005;10(3):251-253.
- (17) Powell A, Mander K, Brown D. Strategies for lifecycle concurrency and iteration – A system dynamics approach. *Journal of Systems and Software*, 1999 4/15;46(2-3):151-161.
- (18) Ruiz Carreira M, Ramos I, Toro M. Modelado y simulación del proceso de desarrollo de software: Una técnica para la mejora de procesos. Técnicas cuantitativas para la gestión en la ingeniería del software Oleiros, La Coruña: Netbiblo SL; 2007. p. 69-90.
- (19) Ruiz M, Ramos I, Toro M. A simplified model of software project dynamics. *Journal of Systems and Software* 2001 12/15;59(3):299-309.
- (20) Wernick P, Scacchi W. Special Issue on ProSim 2003, The 4th International Workshop on Software Process Simulation and Modeling, Portland, OR May 2003. *Software Process: Improvement and Practice* 2004;9(2):51-53.
- (22) Zeigler BP, Kim TG, Praehofer H. Theory and practice of modeling and simulation. New York: Academic Press; 2000.