

INGENIERÍA DE SOFTWARE ORIENTADA A AGENTES

METODOLOGÍAS Y HERRAMIENTAS

Juan Carlos Canto Rodríguez
José Ángel Pérez Rejón (1)

Ingeniería de software orientada a agentes

El surgimiento de nuevos paradigmas de desarrollo como en el caso de los Sistemas Multiagentes (SMA) ha traído consigo la introducción de nuevos conceptos tales como autonomía, reactividad, proactividad y cooperación, proporcionando una poderosa herramienta de abstracción para la solución de problemas complejos distribuidos.

La falta de soporte por parte de la Ingeniería de Software (IS) tradicional, ha propiciado el surgimiento de una variedad de Metodologías de Ingeniería de Software Basadas en Agentes (MISOA). Esta diversidad puede parecer una ventaja, pero ¿se dispone del tiempo suficiente para poder analizar cada metodología? ¿Se está realmente seguro que se eligió la metodología adecuada para la realización del sistema?.

El principal esfuerzo de la Ingeniería de Software Orientada a Agentes (ISOA) está encaminado hacia el desarrollo de SMA, es por estos motivos que el paradigma orientado a agentes ha tomado impulso y se ha invertido trabajo en el diseño y especificación de MISOA, que intentan cubrir las necesidades de especificación de los SMA.

Figura 1

Aunque actualmente la IS nos provee de métodos para poder realizar de manera eficiente el desarrollo de sistemas de software, no toman en cuenta

motivación de los componentes del sistema, es debido a estas razones por las que surge el concepto de Ingeniería de Software Orientada a Agentes (ISOA)[15, 25].

Metodologías

Como resultado del trabajo invertido en el desarrollo de MISOA han surgido una cantidad considerable de éstas, dentro de las cuales podemos mencionar MaSE[9], Prometheus[21], Tropos[3], MESSAGE[4], GAIA[24], CommonKADS[8], BDI[17], Vowel Engineering[10], MASCommonKADS[13], ZEUS[19], INGENIAS[11], MESMA[7], AUMI[16], ROADMAP[23], AOR[2], DESIRE[18], MASSIVE[20], SODA[5], AML[6], PASSI[1] y ADELFE[14] por mencionar algunas.

Muchos trabajos de investigación han surgido para reforzar este paradigma y como parte de este trabajo los investigadores, desarrolladores o simpatizantes del paradigma de agentes han enfocado su esfuerzo en evaluar algunas MISOA conocidas.

La gran cantidad de información obtenida es con la finalidad de tener una visión clara de las características y el objetivo de cada una de las MISOA, el resultado de estas evaluaciones es presentada en reportes, en tablas, o simplemente en artículos de difusión científica lo que para el usuario

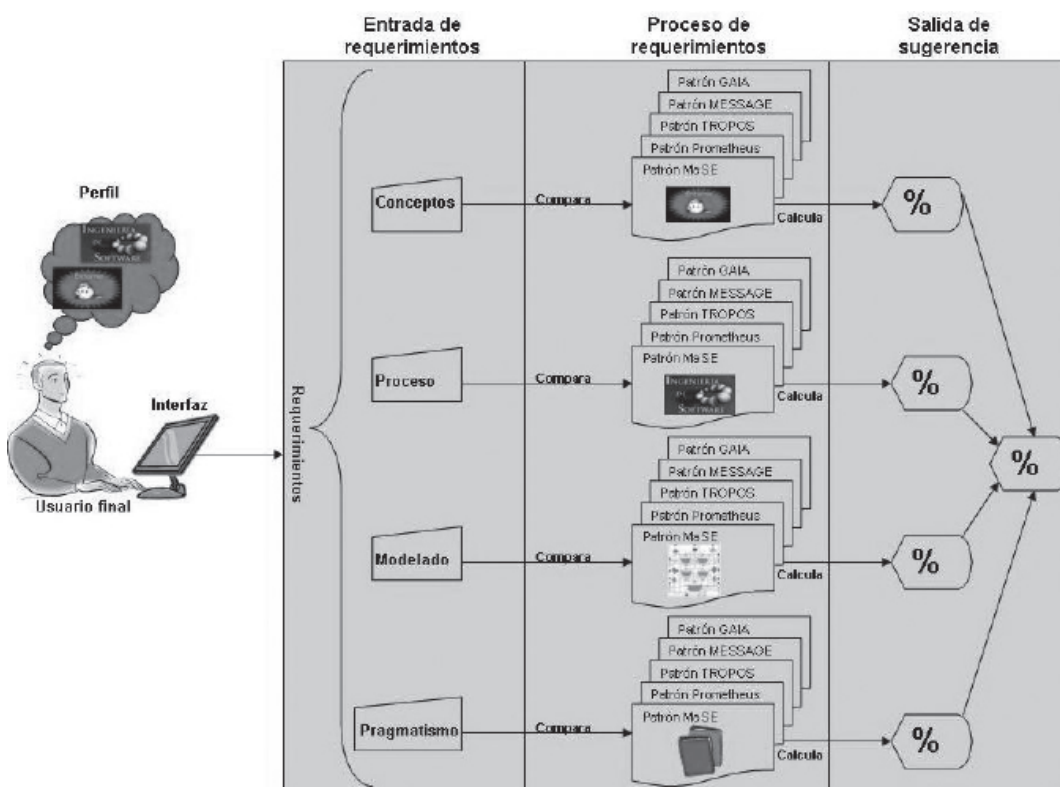
novato o inexperto en el paradigma de agentes sería un tanto complicado extraer la información que necesita, así como obtener claridad del porqué usar alguna MISOA para el desarrollo de su sistema.

Herramientas: Suggest

De la clasificación de las características en trabajos de evaluación mencionados anteriormente podrían obtenerse patrones de caracterización y así poder desarrollar alguna herramienta que nos proporcione apoyo para decidir sobre el uso de una MISOA.

Básicamente lo que se propone es la utilización de la documentación obtenida como parte del esfuerzo realizado por parte de los investigadores en el área de agentes y tratar de desarrollar herramientas que ayuden a la comprensión de estos términos pero sobre todo a seleccionar adecuadamente la metodología para el desarrollo de los SMA.

El planteamiento para el desarrollo de una herramienta la cual se denomina Suggest se basa en la arquitectura general de un sistema (véase Figura 1), la cual se compone de entradas, proceso, salida y almace-



las necesidades de especificación de los SMA en actividades relacionadas tales como: la planificación de tareas, el intercambio de información con lenguajes de comunicación orientado a agentes, movilidad del código o

namiento.

Docentes de tiempo completo en la Dependencia Área Ciencias de la Información de la Universidad Autónoma del Carmen (1)

Dicha herramienta Suggest es el resultado del proceso de unificación de los criterios analizados en el trabajo de investigación de un grupo de investigadores australianos [12] y se presenta la interfaz de dicha herramienta de manera general (véase Figura 2) ya que en documentos o artículos posteriores se pretende mostrar el proceso detallado de unificación de criterios y de la propia herramienta.

La herramienta es una herramienta diseñada en el lenguaje de programación java, usando el entorno de desarrollo Netbeans, la cual presenta una interfaz sencilla clasificando los criterios en cuatro ventanas independientes.

Los tipos de sugerencias que se pueden seleccionar son: por criterio, es decir, una sugerencia dependiendo de la ventana seleccionada en ese momento; o integral, es decir, una sugerencia tomando en cuenta todos los criterios.

Como puede observarse en la interfaz se compone de cuatro ventanas, clasificadas en Conceptos, Proceso, Modelado y Pragmático, el usuario final puede elegir el criterio que más desee y seleccionar en cada una de las características listadas en este criterio.

Aunque existe una diversidad en cuanto a Metodologías de Ingeniería de Software Orientada a Agentes debe considerarse que al menos en nuestro país, esta es un área que apenas comienza a explotarse, por lo que aun no se dispone de información o de herramientas que nos ayuden a acelerar el proceso de aprendizaje en cuanto a tener una visión rápida de la metodología que más se apege a las necesidades específicas de cada proyecto de software.

Figura 2

Los resultados obtenidos en las pruebas de la herramienta fueron satisfactorios, ya que la mayoría de las afirmaciones se comprobó con un porcentaje aceptable, concluyendo que el objetivo de la herramienta se cumple en cuanto a la función primordial que consiste en proporcionar una sugerencia lo más exacta posible, esto quiere decir que se está quitando el sesgo en cuanto a las opiniones de los autores de las metodologías pero podría generarse un sesgo por la labor realizada por este trabajo de investigación al unificar valores, otra característica de la herramienta es la flexibilidad de su diseño porque se puede actualizar conforme cambien o se especialicen las metodolo-

Criterios

Conceptos Proceso Modelado Pragmático

1.- Una metodología que modele agentes que actúen sin supervisión, consideras que es:

☐ Importante ☐ Deseable ☐ No requerido

2.- Una metodología que modele las actitudes mentales de los agentes, consideras que es:

☐ Importante ☐ Deseable ☐ No requerido

3.- Una metodología que modele agentes con la habilidad de alcanzar objetivos en cierto plazo, consideras que es:

☐ Importante ☐ Deseable ☐ No requerido

4.- Una metodología que modele agentes que sean rectivos en su entorno, consideras que es:

☐ Importante ☐ Deseable ☐ No requerido

5.- Una metodología que modele agentes multi objetivos, consideras que es:

☐ Importante ☐ Deseable ☐ No requerido

6.- Una metodología que modele el entorno de los agentes con acciones y percepciones, consideras que es:

☐ Importante ☐ Deseable ☐ No requerido

7.- Una metodología que modele la habilidad de los agentes para trabajar en grupo, consideras que es:

☐ Importante ☐ Deseable ☐ No requerido

8.- Una metodología que modele la habilidad de una conversación válida entre los agentes, consideras que es:

☐ Importante ☐ Deseable ☐ No requerido

9.- Una metodología la cual use conceptos claros y entendibles, consideras que es:

☐ Importante ☐ Deseable ☐ No requerido

Tipo de sugerencia

☐ Integral ☐ Por criterio

Resultados de la Sugerencia

Información adicional

Son los conceptos de agentes involucrados en la metodología, básicamente la ontología, por lo que no todas las metodologías incluyen estos conceptos.

gías incluidas, la actualización se puede realizar cambiando únicamente los patrones de características.

Con esta herramienta se pretende obtener algunos beneficios no solo en el área de Ingeniería de Software si no también contribuir en la selección adecuada de una MISOA, pero sobre todo en proporcionar al desarrollador una herramienta la cual le oriente a elegir alguna MISOA para el desarrollo de su sistema de manera oportuna.



Referencias

- [1] C. Bernon, M. P. Gleizes, S. Peyruqueou, and G. Picard. Adelfe, a methodology for Adaptive Multi-Agent Systems Engineering. In *ESAW - Engineering Societies in the Agents World III: Third International Workshop*, volume 2577, pages 70-81, Madrid, Spain, September 16-17 2002. Springer Berlin / Heidelberg.
- [2] F. M. T. Brazier, B. Dunin-Keplicz, N. R. Jennings, and J. Treur. Modelling Distributed Industrial Processes in a Multi-Agent Framework. In S. Kirn and G. M. P. O'Hare, editors, *Cooperative Knowledge Processing*, pages 212-229. Springer-Verlag: Heidelberg, Germany, 1996.
- [3] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, 8 (3):203-236, 2004.
- [4] G. Caire, W. Coulier, F. J. Garijo, J. Gómez, J. Pavón, F. Leal, P. Chainho, P. E. Kearney, J. Stark, R. Evans, and P. Massonet. Agent Oriented Analysis Using Message/UML. In P. Ciancarini and M. Wooldridge, editors, *AOSE - Agent Oriented Software Engineering*, pages 119-135, Montreal, Canada, 2001. Springer-Verlag.
- [5] R. Cervenka and I. Trencansky. Modeling Social Aspects of Multiagent Systems: The AML Approach. In J. Muller and F. Zambonelli, editors, *The Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 05). Workshop 7: Agent-Oriented Software Engineering (AOSE)*, pages 85-96, Universiteit Utrecht, The Netherlands, 2005.
- [6] M. Cossentino, P. Burrafato, S. Lombardo, and L. Sabatucci. Introducing Pattern Reuse in the Design of Multi-Agent Systems. In *Agent Technologies, Infrastructures, Tools, and Applications for E-Services: NODE 2002 Agent-Related Workshops*, volume 2592 of *Lecture Notes in Computer Science*, pages 107-120, Erfurt, Germany, October 8-9 2002. Springer Berlin / Heidelberg.
- [7] P. Cuesta, A. Gómez, J. C. Gonzales, and F. J. Rodriguez. The MESMA approach for AOSE. In *Proceedings of Fourth Iberoamerican Workshop on Multi-Agent Systems (Iberagents '2002)*, a workshop of *IBERAMIA '2002*, the VIII Iberoamerican Conference on Artificial Intelligence, November 2002.
- [8] R. de Hoog, R. Martil, B. Wielinga, R. Taylor, C. Bright, and W. Van de Velde. The Common KADS model set, 1993. The Common KADS model set. ESPRIT Project P5248 KADS-II/M1/DM1.1b/UvA/018/5.0, University of Amsterdam and others.
- [9] S. A. Deloach, W. M. F., and C. H. Sparkman. Multiagent System Engineering. *International Journal of Software Engineering and Knowledge Engineering*, 11(3):231-258, 2001.
- [10] Y. Demazeau. From interactions to collective behaviour in agent-based systems, 1995.
- [11] J. Gómez and J. Pavón. Agent Oriented Software Engineering with INGENIAS. In *Proceedings of the Third International Central and Eastern European Conference on Multi-Agent Systems*, volume 2691 of *LNCS*, pages 394-403. Springer Verlag, 2003.
- [12] K. Hoa. Evaluating and Comparing Agent-Oriented Software Engineering Methodologies. Tesis de maestría, School of Computer Science and Information Technology, RMIT University, Australia, 2003.
- [13] C. A. Iglesias, M. Garijo, J. G. Centeno, and J. R. Velasco. Analysis and Design of Multiagent Systems Using MAS-common KADS. In *ATAL '97: Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*, pages 313-327, London, UK, 1998. Springer-Verlag.
- [14] N. R. Jennings. Agent-Oriented Software Engineering. In F. J. Garijo and M. Boman, editors, *Proceedings of the 9th European Workshop on MAAMAW - Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering*, volume 1647, pages 1-7. Springer-Verlag: Heidelberg, Germany, 2 1999.
- [15] N. R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 177(2):277-296, 2000.
- [16] T. Juan, A. Pearce, and L. Sterling. ROADMAP: Extending the Gaia Methodology for Complex Open Systems. *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*, ACM Press, pages 3-10, Bologna, Italy, July 2002.
- [17] D. Kinny, M. George, and A. Rao. A Methodology and Modelling Technique for Systems of BDI Agents. In W. V. de Velde and J. W. Perram, editors, *Seventh European Workshop on Modelling Autonomous Agents in a Multi Agent World*, volume 1038, pages 56-71, Eindhoven, The Netherlands, 1996. Springer-Verlag: Berlin, Germany.
- [18] J. Lind. The Massive Development Method for Multiagent Systems. In J. Bradshaw and G. Arnold, editors, *Proceedings of the 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 2000)*, pages 339-354, Manchester, UK, 2000. The Practical Application Company Ltd.
- [19] H. S. Nwana, D. T. Ndumu, L. C. Lee, and J. C. Collis. Zeus: A collaborative agents toolkit. In *Proceedings of the Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, pages 377-392, 1998.
- [20] A. Omicini. SODA: Societies and Infrastructures in the Analysis and Design of Agent-Based Systems. In *AOSE - Agent-Oriented Software Engineering: First International Workshop*, volume 1957, pages 185-193, Limerick, Ireland, June 10 2000.
- [21] L. Padgham and M. Winiko. Prometheus: A pragmatic methodology for engineering intelligent agents. In J. Debenham, B. Henderson-Sellers, N. Jennings, and J. Odell, editors, *Proceedings of the workshop on Agent-oriented methodologies at OOPSLA(Object-Oriented Programming, Systems, Languages and Applications)*, pages 105-117, Seattle, USA, November 4 2002.
- [22] G. Wagner. A Uml prole for external AOR models. In *AOSE - Agent-Oriented Software Engineering III: Third International Workshop*, pages 138-149, Bologna, Italy, July 5 2002. Springer Berlin / Heidelberg.
- [23] M. Wooldridge, N. R. Jennings, and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285-312, 2000.
- [24] M. J. Wooldridge and N. R. Jennings. Software Engineering with Agents: Pitfalls and Pratfalls. *IEEE, Internet Computing*, 3(3):20-27.