

Un enfoque dirigido por modelos para la creación de sistemas robóticos

Claudia Pons, Carlos Neil, Roxana Giandini, Gabriela Pérez and Marcelo de Vincenzi

CIC, Universidad Abierta Interamericana (UAI) y Facultad de Informática de la UNLP

Buenos Aires, Argentina

Resumen

La mayoría de los sistemas robóticos tienden a ser complejos de mantener y reusar porque los frameworks existentes están basados principalmente en enfoques sobre el código. Esto significa que el proceso de desarrollo de software está acotado a la implementación de sistemas usando lenguajes de programación específicos. Y, a medida que el sistema evoluciona, crece en tamaño y complejidad. Incluso cuando estos enfoques abordan las necesidades de mercados centrados en la robótica, las metodologías y las herramientas utilizadas actualmente no pueden hacer frente a las necesidades de un proceso de desarrollo complejo. El objetivo general de nuestro trabajo es la definición de un marco metodológico, respaldado por un conjunto de herramientas, para hacer frente a los requerimientos del proceso de desarrollo de software para robots. Más aún, un desafío mayor es realizar el paso desde el desarrollo basado en el código al desarrollo dirigido por modelos en lo que a sistemas de software para robots se refiere.

Palabras clave: sistemas de software para robots, proceso de desarrollo de software, ingeniería de software, desarrollo dirigido por modelos

1. CONTEXTO

Los sistemas robóticos son esencialmente sistemas distribuidos de tiempo real. Tienen necesidades especiales, acordes a su naturaleza de sistema tiempo real y a las características de su entorno. Además, este tipo especial de sistemas necesita de una mejor calidad que un sistemas de propósito general, y tiene que estar preparado para hacer frente al entorno físico con el que interactúa, el cual suele ser muchas veces incierto y cambiante. Por ello, características tales como confiabilidad y seguridad son fuertemente requeridas en este dominio.

Por otra parte, los sistemas robóticos están hechos de diferentes componentes y sensores, lo que resulta en una arquitectura muy compleja y altamente variable. A menudo, los caminos de comunicación y control dentro del propio sistema están estrechamente relacionados con la configuración actual del propio robot. En consecuencia, estos robots solo pueden ser armados, configurados y programados por expertos.

Los enfoques tradicionales utilizados en el proceso de desarrollo de este tipo de sistemas están basados principalmente en codificar las aplicaciones sin ningún tipo de técnica de modelado. Y, aunque estas

aplicaciones se utilizan en diferentes sistemas robóticos, podemos identificar varios problemas. Entre ellos, vale la pena mencionar que no hay documentación clara sobre las decisiones de diseño que se tomaron durante la fase de codificación, por lo que se dificultan tanto la evolución como el mantenimiento de estos sistemas. Al utilizar lenguajes de programación específicos perdemos la posibilidad de generalizar conceptos que pueden ser extraídos, reutilizados y aplicados en otros sistemas, lo que nos permitiría evitarnos recodificar todo desde cero cuando se lo necesite.

En consecuencia, observamos que los enfoques de desarrollo tradicionales están llegando a un límite. Las metodologías y las herramientas utilizadas actualmente no alcanzan a cubrir las necesidades de estos sistemas tan complejos. En este contexto, está ampliamente aceptado que deben establecerse nuevos enfoques para poder abarcar todas las necesidades de los procesos de desarrollo en sistemas robóticos de hoy en día. El Desarrollo Basado en Componentes (CBD) (Szyperski, 2002), la Arquitectura Orientada a Servicios (SOA) (Bell 2008 y 2010), así como la Ingeniería de software Dirigida por Modelos (MDE) (Stahl, 2006) (Pons y otros, 2010) y el Modelado Específico de Dominio (DSM) (Steven y Juha-Pekka, 2008), son algunas de las prometedoras tecnologías en el dominio de los sistemas robóticos.

En nuestro proyecto, investigamos sobre el uso actual de estas modernas técnicas de ingeniería de software para mejorar el desarrollo de sistemas de software para robots y su nivel real de automatización. Considerando que los sistemas existentes ya fueron codificados, el desafío mayor es

realizar el paso desde el desarrollo basado en el código hacia el desarrollo dirigido por modelos para sistemas robóticos, para extraer los conceptos generales y específicos de las aplicaciones existentes basadas en diferentes lenguajes de programación. Nuestro objetivo es la definición de un marco metodológico (compuesto por modelos y código), respaldado por un conjunto de herramientas, para hacer frente a los requerimientos del proceso de desarrollo de software para robots teniendo en cuenta los enfoques previamente existentes.

2. OBJETIVOS, HIPÓTESIS Y RESULTADOS ESPERADOS

En este contexto, es importante trabajar en pos de la aplicación de principios de ingeniería para hacer frente a la complejidad de los sistemas robóticos, ya que no podemos esperar un crecimiento significativo utilizando sistemas artesanales de una sola pieza. Por otro lado, las interfaces y el comportamiento de los sistemas robóticos deben ser definidos en un nivel de abstracción mayor para poder ser reutilizados en diferentes plataformas. Separando el conocimiento sobre robótica de las tecnologías de implementación es esencial para fomentar la reutilización y el mantenimiento. Así, aplicando tecnologías de ingeniería de software existentes (como MDE, SOA y CBD) para la creación de sistemas de software para robots, se podrá ahorrar una gran cantidad de tiempo y esfuerzo al crecer la reusabilidad de estos sistemas.

Dentro de este contexto, los objetivos principales de este proyecto son:

- Resumir la evidencia existente en cuanto a la aplicación de tecnologías de ingeniería de software en el campo del desarrollo de

- sistemas robóticos, tales como SOA, MDE y CBD.
- Identificar baches en la investigación actual para sugerir posibles nuevas áreas de investigación
 - Proveer un contexto para desarrollar nuevas actividades de investigación en forma apropiada
 - Trabajar sobre aplicación de técnicas modernas al desarrollo de sistemas robóticos, lo que significaría un avance en el campo
 - Definir una metodología abierta para el proceso de desarrollo de sistemas robóticos.
 - Crear herramientas para el proceso de desarrollo de sistemas robóticos. Ejemplos de estas herramientas son: un lenguaje específico de dominio acompañado por un editor gráfico, facilidades para la generación de código, integración con servicios web, editores para la definición de componentes, etc.
 - Proveer herramientas técnicas y metodológicas con gran capacidad de adaptación para hacer frente al ritmo de desarrollo en el campo de la robótica y a las diferencias entre las diferentes plataformas.
 - Realizar una serie de experimentos para evaluar la efectividad y la factibilidad de la propuesta en la construcción de sistemas robóticos complejos.

3. LINEAS DE INVESTIGACIÓN Y RELEVANCIA DEL PROBLEMA

Como la complejidad del software para robots es alta, en la mayoría de los casos la reutilización está limitada al nivel de librerías. En el nivel más bajo, fueron creadas una gran cantidad de librerías para sistemas robóticos que realizan tareas como cálculos

matemáticos para la cinemática, la dinámica y la visión artificial, tal como (Bruyninckx, 2001). En lugar de armar los sistemas con bloques que aseguren ciertos servicios, el proceso de integración para otro sistema está casi siempre limitado a la reimplementación de toda la lógica para unir las diferentes librerías. Casi siempre la integración global es llevada a cabo por un cierto middleware específico. Obviamente, esto no solo es caro y desperdicia muchos recursos de expertos en robótica altamente capacitados, sino que también no se realiza ningún proceso de crecimiento para mejorar la robustez en general. Hemos experimentado este problema en carne propia. Estuvimos programando robots para educación durante más de 10 años (GIRA, 2013) (CAETI, 2013) y hemos observado en los últimos años la aparición de robots orientados a usuarios “no expertos” que dio lugar al desarrollo de una gran cantidad de proyectos educativos utilizando robots. Estos proyectos aplican robots en diferentes niveles educacionales, desde el jardín de infantes hasta los altos estudios, especialmente en el área de física y tecnología. En este contexto, uno de los problemas que encontramos es que el hardware de los robots experimenta un cambio constante. Además, su uso no es el mismo en diferentes regiones y en los diferentes niveles educativos. Por lo tanto, las interfaces técnicas de estos robots deben ocultar estas diferencias, para que los profesores y maestros no tengan que cambiar su material una y otra vez. Un ejemplo de estas interfaces es “PhysicalEtoys” (GIRA, 2013), que propone una plataforma educativa estándar para programar robots, independientemente de si

están basadas en Arduino, Lego u otras tecnologías.

Desde esta perspectiva, está ampliamente aceptado que deben establecerse nuevos enfoques para cubrir las necesidades de los procesos de desarrollo en robótica de hoy en día. El Desarrollo Basado en Componentes (CBD) (Szyperski, 2002), la Arquitectura Orientada a Servicios (SOA) (Bell 2008 y 2010), así como la Ingeniería de software Dirigida por Modelos (MDE) (Stahl, 2006) (Pons y otros, 2010) y el Modelado Específico de Dominio (DSM) (Steven y Juha-Pekka, 2008), son algunas de las prometedoras tecnologías en el dominio de los sistemas robóticos.

4. TRABAJOS RELACIONADOS

Hemos identificado una tendencia creciente con respecto a la aplicación tanto de desarrollo basado en componentes como de arquitecturas orientadas a servicios y desarrollo dirigido por modelos, aunque estas técnicas han sido aplicadas de forma aislada. Algunos trabajos (Basu y otros, 2011; Biggs, 2010; Brooks y otros, 2005; Jawawi y otros, 2008; Min Yang Jung y otros, 2010) han tomado ventaja del desarrollo basado en componentes para desarrollar sistemas robóticos, mientras que otras propuestas (Amoretti y otros, 2007; Cesetti y otros, 2010; Ebenhofer y otros, 2013; Yang y otros, 2013) han aplicado arquitecturas orientadas a servicios para desarrollar este tipo de sistemas. En cambio, solo se han encontrado propuestas preliminares para aplicar desarrollo dirigido por modelos en robótica (Arney y otros, 2010; Baer y otros, 2007; Baumgartl y otros, 2013; Brugali y Scyurra, 2009; Brugali y Shakhimardanov, 2010; Dhouib y otros, 2012; HyunSeung Son y otros, 2008; Iborra y otros, 2009; Jorges y

otros, 2007; Jung y otros, 2005; Poppa y otros, 2012; Sanchez y otros, 2010; Schlegel, 2012; Thomas y otros, 2013; Wei y otros, 2009), mientras que solo un trabajo combina las tres técnicas mencionadas (Tsai y otros, 2008). Al momento, no hay una propuesta que aproveche todas las ventajas de combinar CBD, SOA y MDE al desarrollo de software para robots, como fue relevado en (Pons y otros, 2012).

5. FORMACIÓN DE RECURSOS HUMANOS

El equipo está integrado por cinco investigadores senior, provenientes de 2 universidades (UNLP y UAI), quienes combinan sus conocimientos en robótica e ingeniería de software. El resto del equipo está integrado por estudiantes de pregrado y postgrado. En el contexto de este proyecto se están desarrollando 4 tesis de licenciatura en Informática, 5 tesis de Maestría en TI y 3 tesis doctorales (1 financiada por CONICET y 1 por el programa DoctorAR). Se trabaja además en colaboración con la Universidad de Viena a través de un convenio bi-lateral financiado por el Mincyt.

6. REFERENCIAS

- Amoretti, M.; Zanichelli, F.; Conte, G.; A Service-Oriented Approach for Building Autonomic Peer-to-Peer Robot Systems Enabling Technologies: Infrastructure for Collaborative Enterprises, 2007. WETICE 2007. 16th IEEE International Workshops on, Page(s): 137 - 142 (2007)
- Arney, D.; Fischmeister, S.; Lee, I.; Takashima, Y.; Yim, M.; Model-Based Programming of Modular Robots. 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), Page(s): 66 - 74 (2010)
- Baer, P. A.; Reichle, R.; Zapf, M.; Weise, T.; Geihs, K.; A Generative Approach to the Development of Autonomous Robot Software. EASE '07. Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems. (2007)

- Barner, S., Geisinger, M., Buckl, C., Knoll, A.: EasyLab: Model-based development of software for mechatronic systems. In: IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications. Beijing, China (2008)
- Basu, A.; Bensalem, B.; Bozga, M.; Combaz, J.; Jaber, M.; Nguyen, T.; Sifakis, J.; Rigorous Component-Based System Design Using the BIP Framework Software, IEEE Volume: 28 , Issue: 3 Page(s): 41 – 48. (2011)
- Baumgartl, J., Buchmann, T., Henrich, D., Westfechtel, B.: Towards easy robot programming: Using DSLs, code generators and software product lines ICSoft 2013 - Proceedings of the 8th International Joint Conference on Software Technologies 2013, Reykjavik; Iceland; Pages 548-554.. ISBN: 978-989856568-6; (July 2013)
- Bell, M., "Introduction to Service-Oriented Modeling". Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley & Sons. pp. 3. ISBN 978-0-470-14111-3. (2008).
- Bell, M., SOA Modeling Patterns for Service-Oriented Discovery and Analysis. Wiley & Sons. pp. 390. ISBN 978-0470481974. (2010)
- Biggs, G.; Flexible, adaptable utility components for component-based robot software. Robotics and Automation (ICRA), 2010 IEEE International Conference on, Page(s): 4615 – 4620. (2010)
- Brooks, A., Kaupp, T., Makarenko, A., Oreback, A., Williams, S.: Towards component-based robotics. In: Proc. of 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'05), pp. 163–168. (2005)
- Brugali, D.; Scandurra, P.; Component-based robotic engineering (Part I) [Tutorial] Robotics & Automation Magazine, IEEE Volume: 16 , Issue: 4, Page(s): 84 - 96 (2009)
- Brugali, D.; Shakhimardanov, A.; Component-Based Robotic Engineering (Part II) Robotics & Automation Magazine, IEEE Volume: 17 , Issue: 1 , Page(s): 100 – 112. (2010)
- Bruyninckx, H., Open robot control software: The OROCOS project. In: Proceedings of 2001 IEEE International Conference on Robotics and Automation (ICRA'01), vol. 3, pp. 2523–2528. Korea (2001).
- CAETI (Centro de Altos Estudios en Tecnología Informática). Roboticprojects. <http://www.caeti.uai.edu.ar>. Access November 1st. 2013.
- Cesetti, A.; Scotti, C. P.; Di Buo, G.; Longhi, S.; A Service Oriented Architecture supporting an autonomous mobile robot for industrial applications Control & Automation (MED), 18th Mediterranean Conference on, Page(s): 604 – 609. (2010)
- Dhouib, S., Kchir, S., Stinckwich, S., Ziadi, T., and Ziane, M. RobotMLL, "A domain-specific language to design, simulate and deploy robotic applications". In Noda, I., Ando, N., Brugali, D., and Kuffner, J., editors, Simulation, Modeling, and Programming for Autonomous Robots, vol. 7628 of Lecture Notes in Computer Science, pages 149–160. Springer (2012).
- Ebenhofer, G., Bauer, H., Plasch, M., Zambal, S., Akkaladevi, S.C., Pichler, A.: A system integration approach for service-oriented robotics. 2013 IEEE 18th International Conference on Emerging Technologies and Factory Automation, ETFA 2013; Cagliari; Italy; ISSN: 19460740; (September 2013).
- Gerkey, B.P., Vaughan, R.T., Howard, A., Most valuable player: a robot device server for distributed control. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1226–1231. Wailea, Hawaii (2001)
- GIRA, Grupo de Investigación en Robótica Autónoma del CAETI: PhysicalEtoys <http://tecnodacta.com.ar/gira/> (last access May 1st, 2013).
- Hyun Seung Son, Woo Yeol Kim; Kim, R., Semi-automatic Software Development Based on MDD for Heterogeneous Multi-joint Robots. In Future Generation Communication and Networking Symposia, FGCNS '08. : 2008 , Page(s): 93 – 98 (2008)
- Iborra, A.; Caceres, D.; Ortiz, F.; Franco, J.; Palma, P.; Alvarez, B.; Design of Service Robots. Experiences Using Software Engineering. IEEE Robotics & Automation Magazine 1070-9932/09/ IEEE Page(s): 24 – 33. MARCH 2009
- Jawawi, D.N.A.; Deris, S.; Mamat, R.; Early-Life Cycle Reuse Approach for Component- Based Software of Autonomous Mobile Robot System. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. Ninth ACIS International Conference on, Page(s): 263 – 268. (2008)
- Jorges, S.; Kubczak, C.; Pageau, F.; Margaria, T.; Model Driven Design of Reliable Robot Control Programs Using the jABC. Engineering of Autonomic and

- Autonomous Systems, 2007.EASE '07. Fourth IEEE International Workshop on, Page(s): 137-148 (2007)
- Jung, E.; Kapoor, C.; Batory, D.; Automatic code generation for actuator interfacing from a declarative specification Intelligent Robots and Systems, 2005. (IROS 2005).IEEE/RSJ International Conference on. Page(s): 2839 - 2844 (2005)
- Microsoft, "Microsoft robotics developer studio," 2009,<http://msdn.microsoft.com/en-us/robotics/default.aspx>, visited on March 11th 2009.
- Min Yang Jung; Deguet, A.; Kazanzides, P.; A component-based architecture for flexible integration of robotic systems Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Page(s): 6107 - 6112 (2010)
- Nesnas, I., Wright, A., Bajracharya, M., Simmons, R., Estlin, T.: CLARAty and challenges of developing interoperable robotic software. In: Procs of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), vol. 3, pp. 2428–2435 (2003)
- OMG Robotics Domain Task Force (Robotics DTF) <http://robotics.omg.org/> (access Nov 1st, 2013).
- Pons, C., Giandini, R., Pérez, G., "Desarrollo de Software Dirigido por Modelos. Teorías, Metodologías y Herramientas", Ed: McGraw-Hill Education. ISBN: 978-950-34-0630-4 (2010)
- Pons, C., Giandini, R., Arévalo, G., A systematic review of applying modern software engineering techniques to developing robotic systems. *Revista Ingeniería e Investigación* Vol. 32 No. 1 (2012)
- Poppa, F., Zimmer, U.: RobotUI - A software architecture for modular robotics user interface frameworks, 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems, IROS 2012; Vilamoura, Algarve; Portugal; Pages 2571-2576; (October 2012).
- Sanchez, P; Alonso, D; Rosique, F; Alvarez, B; Pastor, J; Introducing Safety Requirements Traceability Support in Model-Driven Development of Robotic Applications. *Computers, IEEE Transactions on*, Issue: 99 (2010)
- Schlegel, C., Steck, A., Lotz, A., Robotic Software Systems: From Code-Driven to Model-Driven Software Development. Chapter 23 in *Robotic Systems - Applications, Control and Programming*, book edited by Ashish Dutta, ISBN 978-953-307-941-7. (2012)
- SmartSoft.[Online]. Available: <http://smart-robotics.sourceforge.net/>(2013, Jul.)
- Stahl, M Voelter. Model Driven Software Development. John Wiley. (2006)
- Steven, K., Juha-Pekka, T., Domain-Specific Modeling. John Wiley & Sons, Inc. 2008. (2008)
- Zyperski, C., Component Software: Beyond Object-Oriented Programming. 2nd ed. Addison-Wesley Professional, Boston ISBN 0-201-74572-0 (2002).
- Thomas, U., Hirzinger, G., Rumpe, B., Schulze, C., Wortmann, A: A new skill based robot programming language using UML/P Statecharts. *Proceedings - IEEE International Conference on Robotics and Automation, ICRA 2013; Germany; Pages 461-466; (May 2013).*
- Tsai, W.T., Qian Huang, Xin Sun. A Collaborative Service-Oriented Simulation Framework with Microsoft Robotic Studio® Simulation Symposium, ANSS 2008. 41st Annual, Page(s): 263 – 270 (2008)
- Wei Hongxing; Duan Xinming; Li Shiyi; Tong Guofeng; Wang T.; A component based design framework for robot software architecture. *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, Page(s): 3429 - 3434 (2009).
- Yang, T.-H., Lee, W.-P.: A service-oriented framework for the development of home robots. *International Journal of Advanced Robotic Systems; Volume 10, Article number 122; ISSN: 17298806; (February 2013).*