

formas de interacción con el entorno (pueden ser de ultrasonido, infrarrojos, etc.) y también tienen protocolos de datos y comunicación disímiles, por lo que, las técnicas de reutilización tienen que adaptarse a ello para eliminar la variabilidad software de este tipo de objetos. Aunque, una solución podría ser especificar estándares de interfaces software para este tipo de artefactos tecnológicos (Zug et al., 2010).

También, se pudo notar que numerosos investigadores que están realizando reutilización de software no se apropian de los resultados ya obtenidos por otros, debido a que la mayor parte de las investigaciones se desarrollan mediante tecnologías ad-hoc, por lo tanto se está realizando un retrabajo en el área que podría ser aprovechado en otros campos de igual importancia. Posiblemente, esto se deba a que las tecnologías desarrolladas no cumplen con las expectativas de los programadores y se debe mejorar la calidad de las herramientas finales.

Entre los principales desafíos del área, se encuentra la forma de elegir, integrar y alinear de manera adecuada los componentes reutilizables que se desarrollen, y de esta forma atender a las funcionalidades requeridas teniendo en cuenta la particularidad del dominio. Entre estas particularidades que se deben sortear se encuentran la interacción física con el ambiente, la heterogeneidad de los elementos hardware, además, de la alta confiabilidad que requieren este tipo de sistemas. Por lo tanto se debe desarrollar estrategias de reutilización que cumplan con estos requerimientos del dominio (Doukas and Thramboulidis, 2011), (Brugali, 2015).

En el documento de Sun et al. en (Sun et al., 2012) se plantean dos desafíos que tiene la reutilización de software en los robots industriales, el primero tiene que ver con la complejidad de adaptar el software a los cambios que tienen constantemente los sistemas robóticos, pues, estos trabajan en distintos productos, acciones y tareas lo que hace que se cambien frecuentemente configuraciones e interacciones para abordar una nuevas tareas. La consecución de este desafío implicaría la creación de sistemas robóticos más versátiles y una mejora en la velocidad de desarrollo software en este tipo de dispositivos. Mientras que, un segundo desafío plantea desarrollar software para múltiples plataformas (multiplataforma), es decir, realizar una correcta reutilización para evitar la reprogramación y poder utilizar piezas software para distintos tipos de fabricante, esto evitaría que la obsolescencia de este tipo de sistemas no sea tan acelerada. Finalmente, un desafío importante en el área considerando las mejoras que produce la combinación de enfoques de reutilización de software en otros dominios, podría ir dirigido a desarrollar una solución que permita combinar enfoques de reutilización de manera intuitiva, lo cual permitiría adoptar las mejores ventajas de cada uno de los enfoques utilizados.

5.10 Otros datos extraídos

Se extrajo información de los artículos encontrados que no necesariamente tienen que estar relacionados con las preguntas de investigación planteadas, pero que son útiles para el mapeo sistemático. Entre estas cuestiones se encuentra el lenguaje de programación que se está utilizando para el desarrollo de las herramientas finales y el nivel de automatización de estas.

En la Figura 9 se observa que C++ y XML son los lenguajes predominantes (31% y 30% de uso respectivamente) para la

construcción de aplicaciones software para robots industriales, el uso de C++ tiene que ver con la utilización de ROS pues sus módulos están construidos en ese lenguaje de programación para una mejor interacción entre estos. En esta misma línea, XML también ocupa un lugar predominante y aunque no es un lenguaje de programación, se utiliza comúnmente, para soluciones con el enfoque MDE. Por otra parte, JAVA fue utilizado un 8.5% de las veces, sobre todo cuando se aplicó el enfoque OOR, aunque hoy en día ha ido perdiendo fuerza, debido a que su auge se enmarca en los años 90, cuando no existían alternativas a su portabilidad. También, C# se utilizó en algunos casos aislados, pero sin nada notable. Asimismo, algunos lenguajes han sido utilizados solo en una investigación, como .NET, Esterel, Puma o Delphi. Finalmente, algunas herramientas se construyeron en base a su propio lenguaje específico del dominio, que está relacionado con las investigaciones que utiliza el enfoque MDE.

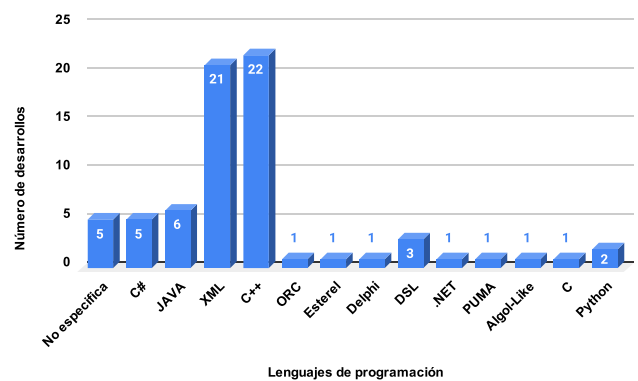


Figura 9. Distribución de lenguajes de programación para herramientas desarrolladas

También, se buscó encontrar el nivel de automatización que ofrecían las herramientas desarrolladas (se considera automatización, al tiempo que se ahorra el programador en realizar o adaptar el software gracias a la herramienta desarrollada). El nivel más alto de automatización es para los generadores de código y los softwares de programación gráfica, porque, el desarrollador no debe realizar programación manual o pocas modificaciones al código final, este apartado corresponde al 20.5% de las herramientas desarrolladas. Asimismo, existen herramientas como los entornos de trabajo o lógicas de intercambio que tienen un nivel de automatización medio, pues, el desarrollador de aplicaciones robóticas debe realizar un trabajo de acondicionamiento y generación del código sobre el cual trabajar, pero con la ventaja, que cuenta con pautas predefinidas que agilizan el desarrollo del software, este nivel corresponde al 49.4% de las herramientas analizadas. Finalmente, el nivel más bajo es para las arquitecturas, metodologías e interfaces de programación de aplicaciones (API), debido a que el desarrollador debe generar el código desde cero, este nivel ocupa el 30.1% de las herramientas examinadas. Por lo tanto se puede decir que la cuota más alta de automatización de herramientas corresponde al nivel medio, que puede estar relacionado con el hecho de que el área de estudio se encuentra en un momento de maduración tecnológica, y que seguramente en pocos años se incrementará la creación de herramientas finales, como generadores de código o entornos de programación gráfica.

Por último, en el siguiente enlace se encuentra la información recolectada para observar y analizar de forma detallada: <https://cutt.ly/IyFvA2Y>.

6. Conclusiones

La reutilización de software se ha convertido en una parte importante de la construcción de los sistemas robóticos industriales (Estévez et al., 2016), por tanto, se están haciendo grandes esfuerzos para agilizar/mejorar el desarrollo de software de este tipo de dispositivos. Los resultados permiten establecer que se trata de un campo de investigación en auge, pues, existe una tendencia positiva en la frecuencia de publicaciones anuales. De acuerdo con el estudio, los enfoques de reutilización de software más utilizados en el dominio son el MDE con un 30.1%, seguido por, el CBSE con el 24.7%. Asimismo, cabe mencionar que los productos de investigación que más se proponen en esta área son los marcos de trabajo con un 42.5%, mientras, que un 17.8% se centra en soluciones arquitectónicas. Por lo tanto, se puede decir que se están desarrollando (marcos o arquitecturas) elementos de base que normalmente soportan enfoques y otros productos, lo que proporciona bases sólidas para continuar con las investigaciones en esta área.

La investigación permite determinar que la computación en la nube y la comunicación multimodal son soluciones emergentes, y jugarán un papel importante en el futuro para mejorar el proceso de desarrollo software para robots industriales. La tendencia a inclinarse por estos enfoques es clara. Entre las principales ventajas se encuentra que el primero intenta eliminar limitaciones de hardware y ayudar al trabajo colaborativo entre dispositivos robóticos (Hu et al., 2017), mientras que, el segundo permite la programación y reprogramación de estos sistemas de manera rápida y por personal no experto (Maurtua et al., 2017).

Entre los principales desafíos de investigación en el área se encuentra la combinación de diferentes enfoques de reutilización de software, para obtener las ventajas de cada uno y optimizar el proceso de desarrollo (Weyns, 2018). Esto está relacionado con la abstracción e integración de las piezas reutilizables de software desarrolladas, para responder a los nuevos requisitos de los sistemas robóticos.

Los enfoques de reutilización más utilizados están relacionados con realizar el proceso de abstracción del sistema o sus funciones para lograr la reutilización de software. Asimismo, el proceso de composición también ha sido clave para mejorar y adaptar la reutilización en este dominio, un ejemplo claro es el enfoque CBSE que se está convirtiendo en una forma natural de representar los sistemas robóticos y sus componentes, mientras que, el enfoque SOA proporciona un mecanismo de comunicación para los componentes en un nivel de abstracción más cercano a la implementación, logrando cierta independencia del hardware y los lenguajes de programación. También cabe mencionar, que el uso de la descomposición y la abstracción no son procesos que deban realizarse por separado, y que, de hecho, puedan mejorarse mutuamente como se ha demostrado en apartados anteriores. El MDE es justamente el enfoque que busca una separación de preocupaciones a través de modelos en diferentes niveles de abstracción.

Se puede dar una propuesta novedosa de reutilización combinando enfoques, permitiendo que los servicios empaqueten componentes, y simultáneamente, los componentes, servicios y sus interacciones puedan ser abstraídos como modelos para que el desarrollador trabaje con elementos abstractos, y genere soluciones concretas sin tener que lidiar con detalles técnicos específicos a las tecnologías subyacentes.

Los resultados de este mapeo sistemático permiten establecer los múltiples acercamientos entre las técnicas de reutilización de software y el dominio de los SR industriales. El sistema operativo robótico puede considerarse como una alternativa a tener en cuenta como referente para que los desarrolladores integren diferentes enfoques de ingeniería de software en los sistemas robóticos industriales (Estefo et al., 2019). La mejor forma de adoptar y establecer estas buenas prácticas de programación podría ser incluir a ROS en la oferta académica de las instituciones educativas, de manera que los programadores se acondicionan a la herramienta desde experiencias tempranas. También, es importante indicar que se debe establecer una curva de aprendizaje menos pronunciada para ROS o para cualquier otro desarrollo, que busque ser punta de lanza para implementar completamente estos enfoques (Bruyninckx et al., 2013), para que de esta forma los desarrolladores puedan adoptar estas herramientas de forma natural y abstraer aspectos técnicos y específicos de tecnologías de bajo nivel.

Referencias

- Adi, W., Sekiyama, K., 2015. A component-based framework for molecular robotic development as smart drug system, in: 2015 International Symposium on Micro-NanoMechatronics and Human Science (MHS). Presented at the 2015 International Symposium on Micro-NanoMechatronics and Human Science (MHS), IEEE, Nagoya, Japan, pp. 1–5. <https://doi.org/10.1109/MHS.2015.7438319>
- Ahmad, A., Babar, M.A., 2016. Software architectures for robotic systems: A systematic mapping study. *Journal of Systems and Software* 122, 16–39. <https://doi.org/10.1016/j.jss.2016.08.039>
- Arne, N., Nico, H., Dennis, W., Sebastian, W., 2016. A survey on domain-specific modeling and languages in robotics. *Journal of Software Engineering for Robotics* 7, 75–99.
- Backhaus, J., Reinhart, G., 2017. Digital description of products, processes and resources for task-oriented programming of assembly systems. *J Intell Manuf* 28, 1787–1800. <https://doi.org/10.1007/s10845-015-1063-3>
- Bandi, A., Williams, B.J., Allen, E.B., 2013. Empirical evidence of code decay: A systematic mapping study, in: 2013 20th Working Conference on Reverse Engineering (WCRE). Presented at the 2013 20th Working Conference on Reverse Engineering (WCRE), IEEE, Koblenz, Germany, pp. 341–350. <https://doi.org/10.1109/WCRE.2013.6671309>
- Beck, J.E., Reagin, J.M., Sweeny, T.E., Anderson, R.L., Garner, T.D., 2000. Applying a component-based software architecture to robotic workcell applications. *IEEE Trans. Robot. Automat.* 16, 207–217. <https://doi.org/10.1109/70.850639>
- Bhavsar, P., Patel, S.H., Sobh, T.M., 2019. Hybrid Robot-as-a-Service (RaaS) Platform (Using MQTT and CoAP). Presented at the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, Atlanta, GA, USA, pp. 974–979. <https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00171>
- Brugali, D., 2015. Model-Driven Software Engineering in Robotics: Models Are Designed to Use the Relevant Things, Thereby Reducing the Complexity and Cost in the Field of Robotics. *IEEE Robot. Automat. Mag.* 22, 155–166. <https://doi.org/10.1109/MRA.2015.2452201>

- Brugali, D. (Ed.), 2007. Software Engineering for Experimental Robotics, Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-68951-5>
- Brugali, D., Hochgeschwender, N., 2018. Software product line engineering for robotic perception systems. *International Journal of Semantic Computing* 12, 89–107.
- Brugali, D., Scandurra, P., 2009. Component-based robotic engineering (Part I) [Tutorial]. *IEEE Robot. Automat. Mag.* 16, 84–96. <https://doi.org/10.1109/MRA.2009.934837>
- Bruyninckx, H., Klotzbücher, M., Hochgeschwender, N., Kraetzschmar, G., Gherardi, L., Brugali, D., 2013. The BRICS component model: a model-based development paradigm for complex robotics software systems, in: *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*. Presented at the the 28th Annual ACM Symposium, ACM Press, Coimbra, Portugal, p. 1758. <https://doi.org/10.1145/2480362.2480693>
- Bubeck, A., Maidel, B., Lopez, F.G., 2014. Model Driven Engineering for the Implementation of User Roles in Industrial Service Robot Applications. *Procedia Technology* 15, 605–612. <https://doi.org/10.1016/j.procty.2014.09.021>
- Chen, Y., Du, Z., García-Acosta, M., 2010. Robot as a Service in Cloud Computing, in: *2010 Fifth IEEE International Symposium on Service Oriented System Engineering*. Presented at the 2010 Fifth International Symposium on Service Oriented System Engineering (SOSE), IEEE, Nanjing, China, pp. 151–158. <https://doi.org/10.1109/SOSE.2010.44>
- Ciccozzi, F., Di Ruscio, D., Malavolta, I., Pelliccione, P., Tumova, J., 2017. Engineering the software of robotic systems, in: *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. Presented at the 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), IEEE, Buenos Aires, pp. 507–508. <https://doi.org/10.1109/ICSE-C.2017.167>
- Degueule, T., Combemale, B., Blouin, A., Barais, O., Jézéquel, J.-M., 2017. Safe model polymorphism for flexible modeling. *Computer Languages, Systems & Structures* 49, 176–195. <https://doi.org/10.1016/j.cl.2016.09.001>
- Doukas, G., Thramboulidis, K., 2011. A Real-Time-Linux-Based Framework for Model-Driven Engineering in Control and Automation. *IEEE Trans. Ind. Electron.* 58, 914–924. <https://doi.org/10.1109/TIE.2009.2029584>
- Estefo, P., Simmonds, J., Robbes, R., Fabry, J., 2019. The Robot Operating System: Package reuse and community dynamics. *Journal of Systems and Software* 151, 226–242. <https://doi.org/10.1016/j.jss.2019.02.024>
- Estévez, E., García, A.S., García, J.G., Ortega, J.G., 2018. ART2ool: a model-driven framework to generate target code for robot handling tasks. *Int J Adv Manuf Technol* 97, 1195–1207. <https://doi.org/10.1007/s00170-018-1976-z>
- Estévez, E., Sánchez García, A., Gámez García, J., Gómez Ortega, J., 2017. Aproximación Basada en UML para el Diseño y Codificación Automática de Plataformas Robóticas Manipuladoras. *Revista Iberoamericana de Automática e Informática Industrial RIAI* 14, 82–93. <https://doi.org/10.1016/j.riai.2016.11.001>
- Estévez, E., Sánchez-García, A., Gámez-García, J., Gómez-Ortega, J., Satorres-Martínez, S., 2016. A novel model-driven approach to support development cycle of robotic systems. *Int J Adv Manuf Technol* 82, 737–751. <https://doi.org/10.1007/s00170-015-7396-4>
- Gherardi, L., Brugali, D., 2014. Modeling and reusing robotic software architectures: The HyperFlex toolchain, in: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Presented at the 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Hong Kong, China, pp. 6414–6420. <https://doi.org/10.1109/ICRA.2014.6907806>
- Gorecki, S., Duq, Y., Ribault, J., Zacharewicz, G., Perry, N., 2019. Distributed Simulation For A Modeling And Simulation Tool: Papyrus, in: *2019 Spring Simulation Conference (SpringSim)*. Presented at the 2019 Spring Simulation Conference (SpringSim), IEEE, Tucson, AZ, USA, pp. 1–12. <https://doi.org/10.23919/SpringSim.2019.8732868>
- Guerin, K.R., Riedel, S.D., Bohren, J., Hager, G.D., 2014. Adjutant: A framework for flexible human-machine collaborative systems, in: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Presented at the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), IEEE, Chicago, IL, USA, pp. 1392–1399. <https://doi.org/10.1109/IROS.2014.6942739>
- Haage, M., Piperagkas, G., Papadopoulos, C., Mariolis, I., Malec, J., Bekiroglu, Y., Hedelind, M., Tzovaras, D., 2017. Teaching Assembly by Demonstration Using Advanced Human Robot Interaction and a Knowledge Integration Framework. *Procedia Manufacturing* 11, 164–173. <https://doi.org/10.1016/j.promfg.2017.07.221>
- Hu, B., Wang, H., Zhang, P., Ding, B., Che, H., 2017. Cloudroid: A Cloud Framework for Transparent and QoS-Aware Robotic Computation Outsourcing, in: *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. Presented at the 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), IEEE, Honolulu, CA, USA, pp. 114–121. <https://doi.org/10.1109/CLOUD.2017.23>
- Jawawi, D., Deris, S., Mamat, R., 2007. Software Reuse for Mobile Robot Applications Through Analysis Patterns. *The International Arab Journal of Information Technology* 4, 9.
- Kitchenham, B., 2004. Procedures for Performing Systematic Reviews. Keele, UK, Keele Univ. 33, 1–26.
- Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S., 2009. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology* 51, 7–15. <https://doi.org/10.1016/j.infsof.2008.09.009>
- Lotz, A., Hamann, A., Lange, R., Heinzemann, C., Staschulat, J., Kesel, V., Stampfer, D., Lutz, M., Schlegel, C., 2016. Combining robotics component-based model-driven development with a model-based performance analysis, in: *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAP)*. Presented at the 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAP), IEEE, San Francisco, CA, USA, pp. 170–176. <https://doi.org/10.1109/SIMPAP.2016.7862392>
- Lunghi, G., Marin, R., Di Castro, M., Masi, A., Sanz, P.J., 2019. Multimodal Human-Robot Interface for Accessible Remote Robotic Interventions in Hazardous Environments. *IEEE Access* 7, 127290–127319. <https://doi.org/10.1109/ACCESS.2019.2939493>
- Maurtua, I., Fernández, I., Tellaache, A., Kildal, J., Susperregi, L., Ibarguren, A., Sierra, B., 2017. Natural multimodal communication for human-robot collaboration. *International Journal of Advanced Robotic Systems* 14, 172988141771604. <https://doi.org/10.1177/1729881417716043>
- Möckel, R., Dahl, L., Christopher, S.M., 2020. Interdisciplinary Teaching with the Versatile Low-Cost Modular Robotic Platform EDMO, in: Moro, M., Alimisis, D., Iocchi, L. (Eds.), *Educational Robotics in the Context of the Maker Movement*. Springer International Publishing, Cham, pp. 135–146.
- Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic Mapping Studies in Software Engineering. Presented at the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE). <https://doi.org/10.14236/ewic/EASE2008.8>
- Pons, C., Giandini, R., Arévalo, G., 2012. A systematic review of applying modern software engineering techniques to developing robotic systems. *Ingeniería e Investigación* 32, 58–63.
- Pons, C., Pérez, G., Giandini, R., Neil, C., de Vincenzi, M., 2017. Ingeniería de Software Dirigida por Modelos Aplicada a Sistemas Robóticos Usando los Estándares de la OMG. Presented at the XIX Workshop de Investigadores en Ciencias de la Computación, Buenos Aires, pp. 555–559.
- Rastogi, N., Dutta, P., Krishna, V., Gotewal, K.K., 2017. Implementation of an OROCOS based Real-Time Equipment Controller for Remote Maintenance of Tokamaks, in: *Proceedings of the Advances in Robotics on - AIR '17*. Presented at the the Advances in Robotics, ACM Press, New Delhi, India, pp. 1–6. <https://doi.org/10.1145/3132446.3134900>
- Ronchieri, E., Canaparo, M., 2019. Metrics for Software Reliability: a Systematic Mapping Study. *JID* 22, 5–25. <https://doi.org/10.3233/jid-2018-0008>
- ROS.org | Powering the world's robots [WWW Document], 2020. . ROS.org | Powering the world's robots. URL <http://www.ros.org/> (accessed 5.18.20).
- Rudorfer, M., Guhl, J., Hoffmann, P., Kruger, J., 2018. Halo Pick'n'Place. Presented at the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, pp. 1219–1222. <https://doi.org/10.1109/ETFA.2018.8502527>
- Salman, S.M., Struhar, V., Papadopoulos, A.V., Behnam, M., Nolte, T., 2019. Fogification of Industrial Robotic Systems: Research Challenges, in: *Proceedings of the Workshop on Fog Computing and the IoT, IoT-Fog '19*. Association for Computing Machinery, New York, NY, USA, pp. 41–45. <https://doi.org/10.1145/3313150.3313225>
- Smith, R., Smith, G., Wardani, A., 2005. Software reuse in robotics: Enabling portability in the face of diversity, in: *IEEE Conference on Robotics, Automation and Mechatronics, 2004*. Presented at the 2004 IEEE Conference on Robotics, Automation and Mechatronics, IEEE, Singapore, pp. 933–938. <https://doi.org/10.1109/RAMECH.2004.1438043>
- Souza, F.C., Santos, A., Andrade, S., Durelli, R., Durelli, V., Oliveira, R., 2018. Automating Search Strings for Secondary Studies, in: Latifi, S. (Ed.), *Information Technology - New Generations, Advances in Intelligent Systems and Computing*. Springer International Publishing, Cham, pp. 839–848. https://doi.org/10.1007/978-3-319-54978-1_104

- Stenmark, M., Haage, M., Topp, E.A., 2017. Simplified Programming of Reusable Skills on a Safe Industrial Robot: Prototype and Evaluation, in: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. Presented at the HRI '17: ACM/IEEE International Conference on Human-Robot Interaction, ACM, Vienna Austria, pp. 463–472. <https://doi.org/10.1145/2909824.3020227>
- Sun, Y., Gray, J., Bulheller, K., von Baillou, N., 2012. A Model-Driven Approach to Support Engineering Changes in Industrial Robotics Software, in: France, R.B., Kazmeier, J., Breu, R., Atkinson, C. (Eds.), Model Driven Engineering Languages and Systems, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 368–382. https://doi.org/10.1007/978-3-642-33666-9_24
- Tibermacine, C., Sadou, S., Ton That, M.T., Dony, C., 2016. Software architecture constraint reuse-by-composition. Future Generation Computer Systems 61, 37–53. <https://doi.org/10.1016/j.future.2016.02.006>
- Trapani, S., Indri, M., 2017. Task modeling for task-oriented robot programming, in: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). Presented at the 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Limassol, pp. 1–8. <https://doi.org/10.1109/ETFA.2017.8247650>
- Vrochidou, E., Manios, M., Papakostas, G.A., Aitsidis, C.N., Panagiotopoulos, F., 2018. Open-Source Robotics: Investigation on Existing Platforms and Their Application in Education, in: 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). Presented at the 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), IEEE, Split, pp. 1–6. <https://doi.org/10.23919/SOFTCOM.2018.8555860>
- Vyatkin, V., 2011. IEC 61499 as enabler of distributed and intelligent automation: State-of-the-art review. IEEE transactions on Industrial Informatics 7, 768–781. <https://doi.org/10.1109/TII.2011.2166785>
- Wan, J., Tang, S., Yan, H., Li, D., Wang, S., Vasilakos, A.V., 2016. Cloud Robotics: Current Status and Open Issues. IEEE Access 1–1. <https://doi.org/10.1109/ACCESS.2016.2574979>
- Wei, H., Duan, X., Li, S., Tong, G., Wang, T., 2009. A component-based design framework for robot software architecture, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. Presented at the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), IEEE, St. Louis, MO, USA, pp. 3429–3434. <https://doi.org/10.1109/IROS.2009.5354161>
- Wei, H., Shao, Zhenzhou, Huang, Z., Chen, R., Guan, Y., Tan, J., Shao, Zili, 2016. RT-ROS: A real-time ROS architecture on multi-core processors. Future Generation Computer Systems 56, 171–178. <https://doi.org/10.1016/j.future.2015.05.008>
- Wenger, M., Eisenmenger, W., Neuschwandtner, G., Schneider, B., Zoitl, A., 2016. A model-based engineering tool for ROS component compositioning, configuration and generation of deployment information, in: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA). Presented at the 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Berlin, Germany, pp. 1–8. <https://doi.org/10.1109/ETFA.2016.7733559>
- Weyns, D., 2018. Engineering Self-Adaptive Software Systems – An Organized Tour, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W). Presented at the 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), IEEE, Trento, pp. 1–2. <https://doi.org/10.1109/FAS-W.2018.00012>
- Wigand, D.L., Nordmann, A., Goerlich, M., Wrede, S., 2017. Modularization of Domain-Specific Languages for Extensible Component-Based Robotic Systems, in: 2017 First IEEE International Conference on Robotic Computing (IRC). Presented at the 2017 First IEEE International Conference on Robotic Computing (IRC), IEEE, Taichung, Taiwan, pp. 164–171. <https://doi.org/10.1109/IRC.2017.34>
- Yassin, N.I.R., Omran, S., El Houbay, E.M.F., Allam, H., 2018. Machine learning techniques for breast cancer computer aided diagnosis using different image modalities: A systematic review. Computer Methods and Programs in Biomedicine 156, 25–45. <https://doi.org/10.1016/j.cmpb.2017.12.012>
- Yoong, L.H., Bhatti, Z.E., Roop, P.S., 2012. Combining iec 61499 model-based design with component-based architecture for robotics, in: International Conference on Simulation, Modeling, and Programming for Autonomous Robots. Springer, pp. 349–360. https://doi.org/10.1007/978-3-642-34327-8_32
- Zug, S., Schulze, M., Dietrich, A., Kaiser, J., 2010. Programming abstractions and middleware for building control systems as networks of smart sensors and actuators, in: 2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010). Presented at the Factory Automation (ETFA 2010), IEEE, Bilbao, pp. 1–8. <https://doi.org/10.1109/ETFA.2010.5641341>