

Reutilización de software en la robótica industrial: un mapeo sistemático

Solis, A., Hurtado, J.

Grupo IDIS, Universidad del Cauca, Calle 5 No. 4 - 70, Popayán, Colombia.

To cite this article: Solis, A., Hurtado, J. 2020. Software reuse in industrial robotics: a systematic mapping. Revista Iberoamericana de Automática e Informática Industrial 17, 354-367. <https://doi.org/10.4995/riai.2020.13335>

Resumen

Existe una tendencia a utilizar los enfoques de reutilización de software en el dominio de los sistemas robóticos industriales, con el fin de acelerar su desarrollo. Aunque algunos estudios muestran los beneficios de desarrollar usando diferentes enfoques de reutilización, estas prácticas no se han incorporado masivamente en la industria, principalmente, debido al desarrollo de software propietario por parte de los fabricantes y a la diversidad del hardware subyacente. Sin embargo, estos estudios han sido de gran valor para avanzar en su adopción. A través de un mapeo sistemático de la literatura, se muestra la adopción de los diferentes enfoques de reutilización, dentro de los cuales se analizan los más utilizados como la ingeniería dirigida por modelos MDE (Model-Driven Engineering), el desarrollo basado en componentes CBSE (Component-based Software Engineering) y la arquitectura basada en servicios (SOA). Por otro lado, se analizan los marcos de trabajo por ser las soluciones más utilizados y en términos de herramientas, se enfatiza en ROS (Robot Operating System) como una plataforma de referencia para el desarrollo rápido de aplicaciones. El principal reto identificado en esta área de estudio es definir estrategias combinadas y prácticas de los enfoques de reutilización MDE, CBSE y SOA, con el fin de aprovechar las diferentes ventajas de reutilización que cada uno ofrece.

Palabras clave: Reutilización, robótica industrial, manipulador robótico, programación de robots.

Software reuse in industrial robotics: a systematic mapping

Abstract

There is a tendency to use software reuse approaches in the domain of industrial robotic systems, to accelerate their development. Although some studies show the benefits of developing using different reuse approaches, these practices have not been massively incorporated in the industry, mainly due to the development of proprietary software by manufacturers and the diversity of the underlying hardware. However, these studies have been of great value in advancing their adoption. Through a systematic mapping of the literature, the adoption of different reuse approaches is shown, within which the most widely used are analyzed, such as Model-Driven Engineering (MDE), Component-based Software Engineering (CBSE) and Service-Oriented Architecture (SOA). On the other hand, the frameworks are analyzed because they are the most used solutions and in terms of tools, ROS (Robot Operating System) is emphasized as a reference platform for the rapid development of applications. The main challenge identified in this area of study is to define combined and practical strategies of the MDE, CBSE, and SOA reuse approaches, to take advantage of the different reuse advantages that each one offers.

Keywords: Reusability, industrial robotics, robotic manipulator, robot programming.

1. Introducción

El desarrollo de los Sistemas Robóticos (SR) industriales, siempre ha tenido un alto grado de complejidad debido a la interdisciplinariedad requerida para su construcción (Möckel

et al., 2020), que incluye aspectos como el modelado de los actuadores, manipulación de protocolos de comunicación, calibración de sensores y el desarrollo de software, entre otros. Sin embargo, no existe un consenso general sobre qué técnicas o metodologías son ideales a la hora de desarrollar software

para robots, aunque han surgido alternativas prometedoras tales como, Ingeniería dirigida por modelos (MDE, siglas del inglés Model-driven Engineering), Ingeniería de software basada en componentes (CBSE, siglas del inglés Component-based Software Engineering) o Arquitectura orientada a servicios (SOA, siglas del inglés Service-Oriented Architecture) pero no existe una aceptación general sobre estas (Pons et al., 2012).

El software para los robots industriales juega un papel fundamental, debido a que permite la integración de los elementos hardware que participan en el desarrollo de los robots (Ciccozzi et al., 2017), además proporciona inteligencia a los dispositivos electrónicos (Jawawi et al., 2007). Actualmente, los sistemas robóticos desempeñan un papel cada vez más importante en la industria y en la vida cotidiana, lo cual trae como consecuencia exigencias cada vez más difíciles de cumplir. A medida que los SR son más complejos, la necesidad de utilizar estrategias de reutilización de software se vuelven evidentes, para así evitar el retrabajo en el desarrollo de activos de software que ya existen y poder enfocarse en nuevos desafíos (Smith et al., 2005).

Entre los principales factores que han afectado el nivel de adopción de técnicas o metodologías de reutilización de software en la robótica industrial se encuentra la resistencia de los desarrolladores a adoptar estos enfoques, a pesar de haber demostrado en múltiples campos los buenos resultados que pueden tener (Pons et al., 2017). Otra importante causa, es que la mayor parte de cuota de mercado para la creación de software robótico son empresas, que realizan inversiones a largo plazo y que ante todo exigen confiabilidad (Vrochidou et al., 2018), por lo tanto a los desarrolladores les resulta difícil adoptar nuevas técnicas de ingeniería de software que no estén completamente probadas y listas para producción. Finalmente, la heterogeneidad de componentes hardware y software que existen en el mercado obstaculizan la aplicación de estas técnicas (Estévez et al., 2017), pues dificultan algunos procesos como la abstracción de modelos, detección de patrones, identificación del nivel de granularidad, entre otros, los cuales son pasos fundamentales para realizar una correcta reutilización del software (Brugali, 2007).

Existen pocas revisiones sistemáticas que permitan realizar una retrospectiva de las técnicas de reutilización de software y los sistemas robóticos, entre ellas se encuentra la revisión sistemática de literatura de Pons et al. en (Pons et al., 2012), la cual plantea un análisis detallado de las técnicas o metodologías que mejoran el desarrollo de software en el dominio de la robótica. En este estudio se comprueba que existe una creciente tendencia a utilizar los enfoques MDE, CBSE y SOA en el dominio robótico, pero indica que son casos aislados. Por tanto, concluye que existen aún lagunas en el conocimiento de los desarrolladores con respecto a las técnicas de reutilización. También es importante mencionar, que se utilizó como base para la presente investigación el artículo de Kitchenham en (Kitchenham, 2004), que establece las principales bases para portar las revisiones sistemáticas al área de ciencias de la computación.

La principal diferencia entre este mapeo sistemático y otros estudios con características parecidas es que este está enfocado

en el dominio de los robots industriales, mientras que otros se enfocan en el dominio general de los robots. Este es el caso de la investigación realizada por Ahmad y Babar en (Ahmad and Babar, 2016) donde realiza un análisis de las principales arquitecturas utilizadas en el dominio general de los sistemas robóticos. En el documento concluye que los enfoques CBSE y SOA son prometedores para mejorar el desarrollo de software en estos dispositivos.

Finalmente, en la revisión bibliográfica realizada se encontró que las revisiones o mapeos sistemáticos no realizan una discriminación por tipo de robot, lo cual genera un vacío en el conocimiento para la replicación de este tipo de metodologías, debido a que cada tipo de sistema robótico es un mundo aparte, con características específicas.

A continuación, se desarrolla un mapeo sistemático de literatura de los enfoques de reutilización de software en el dominio de los sistemas robóticos industriales.

2. Planificación del mapeo sistemático de literatura

Un mapeo sistemático de literatura es un proceso mediante el cual se puede recopilar información sobre un tema en específico de manera ordenada, metódica y replicable para proporcionar una visión general al lector (Kitchenham et al., 2009). El presente mapeo aplicó el protocolo guía de mapeos/revisiones sistemáticas propuesto por Petersen et al. (Petersen et al., 2008). Esta investigación se desarrolla en cinco etapas (Figura 1): (1) definir las preguntas de la investigación, (2) realizar la búsqueda de estudios primarios, (3) revisar los documentos basándose en criterios de inclusión/exclusión, (4) clasificar los documentos y (5) extraer y mapear los datos recopilados. Cada etapa produce un resultado que se utiliza como insumo para las siguientes etapas.

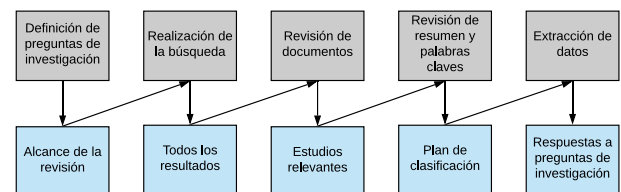


Figura 1. Proceso de realización de mapeo sistemático propuesto por Petersen et al. (Petersen et al., 2008).

A continuación, se presenta el proceso de elaboración del mapeo sistemático.

2.1. Definición de las preguntas de investigación

Las preguntas de investigación planteadas buscan establecer en qué contexto se encuentran las técnicas de reutilización de software en el dominio de los sistemas robóticos industriales, para encontrar carencias existentes en el área y poder investigar cómo se está realizando el proceso de abstracción e implementación de los enfoques de reutilización. En la Tabla 1, se presentan cada una de las preguntas y su motivación. A través de estas se pudo seleccionar, analizar y categorizar la información encontrada del área de estudio.

Tabla 1. Preguntas de investigación utilizados para el mapeo sistemático.

Preguntas de investigación	Motivante	Respuestas
¿Qué estudios existen que utilicen enfoques o técnicas de reutilización de software aplicadas al dominio de los SR industriales?	Determinar el número de publicaciones actuales y las tendencias de la reutilización de software en el dominio de los SR industriales.	<ul style="list-style-type: none"> • Component-based • Object-oriented • Domain-specific • Model-driven • Service-oriented
¿Cuál es la frecuencia de publicación de investigaciones que hagan reutilización de software para la construcción de los SR industriales?	Comprobar con qué frecuencia se están realizando estudios que impliquen la reutilización de software en el dominio de la robótica industrial, además de observar si la línea de tendencia es positiva y, por lo tanto, si es un tema actual.	Año de creación de la herramienta software
¿Qué tipo de herramientas se están desarrollando para implementar las técnicas de reutilización de software en los SR industriales?	Establecer qué tipo de herramientas (framework, DSL, aplicaciones, etcétera) se están desarrollando para integrar la reutilización de software en el dominio de los SR industriales.	<ul style="list-style-type: none"> • API • Arquitectura • Marco de trabajo • Generador de código • Método o modelo • Middleware • Otra herramienta
¿Con qué frecuencia se relacionan las herramientas y los enfoques de reutilización de software en el dominio de los robots industriales en los estudios analizados?	Determinar cómo se están distribuyendo las herramientas creadas junto con los enfoques de reutilización de software para esta forma encontrar donde existen vacíos en el conocimiento o mayor evidencia empírica para sustentar hipótesis o desarrollar teorías.	Se realizará el cruce de información con las respuestas 1 y 3
¿En qué contexto se está haciendo los principales aportes al dominio de la robótica con técnicas de reutilización de software?	Encontrar en qué contextos se está desarrollando la reutilización de software en el dominio de la robótica industrial, para de esta forma observar donde se realizan las mayores contribuciones y en donde existen vacíos en el área conocimiento.	<ul style="list-style-type: none"> • Académico • Industrial • Ambos
¿Cómo se está implementado las estrategias de reutilización de software en SR industriales?	Determinar cómo se están implementando las estrategias de reutilización de software en el dominio de los robots industriales.	Se desarrollará al momento de responder la pregunta de investigación
¿Se han utilizado los enfoques de forma combinada o de manera aislada?	Cómo se están desarrollando las metodologías de forma separada o en combinación con otros métodos o metodologías.	Se desarrollará al momento de responder la pregunta de investigación
¿Las técnicas de reutilización de software en el dominio de la robótica industrial se están aplicando de manera ad-hoc o siguiendo un estándar o metodología ya probada?	Encontrar si se están realizando los esfuerzos de incluir la reutilización en el dominio robótico industrial de manera aislada o, al contrario, si los desarrolladores se están apropiando de tecnologías ya probadas como marcos de trabajo (framework), arquitecturas o API.	<ul style="list-style-type: none"> • Sistema operativo robótico • Otras tecnologías • Tecnología Ad-Hoc
¿Cuáles son los principales problemas y retos en el área de reutilización de software en el dominio de los sistemas robóticos industriales?	Determinar hacia qué dirección se dirige el área de reutilización de software en el dominio robótico industrial.	Se desarrollará al momento de responder la pregunta de investigación

2.2. Estrategia de búsqueda

Se utilizó una cadena de búsqueda (Tabla 2) para obtener información de forma sistemática en las bases de datos bibliográficas. Esta se construyó usando los criterios PICOC que son una excelente forma de estructurar conceptos de manera ordenada, metodológica y abarcando todos los temas que se desean encontrar (Souza et al., 2018). Las bases bibliográficas utilizadas fueron: ACM Digital Library, EBSCO Information Services, IEEE Digital Library y ScienceDirect, que son aptas para realizar búsquedas sistemáticas en el

contexto de la ingeniería de software (Ronchieri and Canaparo, 2019). Además, se añadió Scopus pues se considera la mayor base de datos bibliográfica de artículos científicos y de investigación. Es importante aclarar que la cadena de búsqueda se aplicó al título, el resumen y a las palabras clave de todas las bases de datos mencionadas, excepto por ScienceDirect en donde la cadena de búsqueda tuvo que reducirse a ocho términos, porque este sitio web acepta este máximo de operadores lógicos. Asimismo, se realizó una búsqueda manual en Google Académico y se seleccionaron los artículos

de mayor relevancia, los cuales relacionan la reutilización de software con los sistemas robóticos industriales (el buscador tiene en cuenta pertinencia, número de citas, autor y la fecha de publicación para medir la relevancia de los artículos), posteriormente estas investigaciones se incluyeron entre los estudios a analizar. La búsqueda en las bases de datos bibliográficas comprende todos los estudios hasta abril de 2020 sin ventana de tiempo restrictiva.

Tabla 2. Cadena de búsqueda aplicada a cada una de las bases de datos

Cadena de búsqueda
("robot arm" OR "articulated robot" OR "articulated robots" OR "handling robot" OR "handling robots" OR "industrial robot" OR "industrial robots" OR "robotic arm" OR "robotic arms" OR "robotic manipulators" OR "robot manipulator" OR "robots arm" OR "robots manipulators") AND ("programming" OR "software development" OR "software engineering" OR "software reuse" OR "software reusable") AND ("Component-based" OR "CBD" OR "CBSE" OR "Domain-specific" OR "DSL" OR "framework" OR "Model-driven" OR "MDA" OR "MDD" OR "MDE" OR "Service-oriented" OR "SOA" OR "SOA" OR "code generation" OR "generative programming")

2.3. Metodología de selección de estudios primarios

Para llevar a cabo un mapeo sistemático de manera metodológica y ordenada se decidió seguir las directrices reportadas por Petersen et al. en (Petersen et al., 2008), cuyo objetivo es proporcionar un mapeo sistemático de literatura completo definiendo de manera técnica el alcance de este. En la siguiente figura se observa que la investigación se hace de manera iterativa, con constante retroalimentación dependiendo de los resultados.

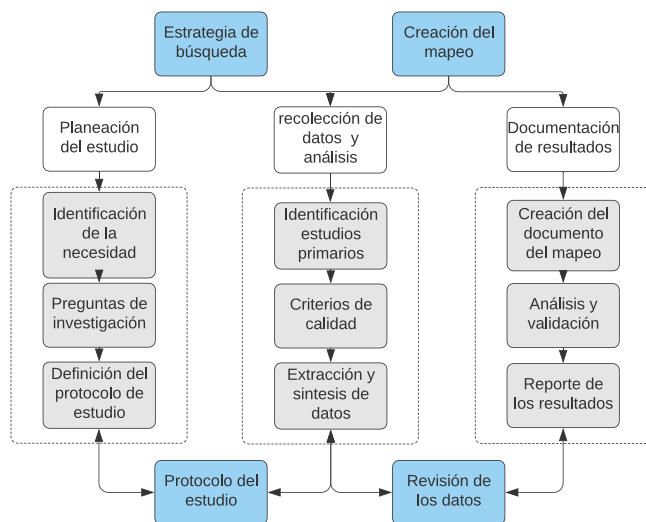


Figura 2. Metodología utilizado en el mapeo sistemático de literatura sugerido por Petersen et al. en (Petersen et al., 2008).

2.4. Criterios de selección de estudios primarios

Siguiendo las directrices reportadas por Petersen et al. en (Petersen et al., 2008), se utilizaron los criterios de inclusión y exclusión (Tabla 3) para determinar cuáles son los estudios

más relevantes para la investigación y de esta forma responder las preguntas planteadas (Yassin et al., 2018). Un estudio fue considerado relevante si cumple con todos los criterios de inclusión, en cambio, si un estudio cumple con al menos un criterio de exclusión no será tenido en cuenta.

Tabla 3. Criterios de inclusión y exclusión de la investigación

Criterios de inclusión
Estudios presentados en inglés que tengan en cuenta la reutilización de software en los SR industriales.
La investigación presenta una solución concreta al área de reutilización de software en robótica industrial.
Artículos completos publicados en revistas, conferencias o congresos con revisión por pares.
Criterios de exclusión
El estudio primario no presenta una solución concreta al área de la robótica industrial.
Estudios no accesibles en texto completo.
Estudios que presentan material no revisado por pares.
Estudios secundarios o terciarios que reportan y analizan resultados de otras investigaciones.
Publicaciones duplicadas de los mismos autores

3. Estrategia de extracción de datos

La estrategia de extracción de datos se basó en recopilar información asociada a cada pregunta de investigación, utilizando una serie de posibles respuestas y de esta forma asegurar la aplicación de los mismos criterios para todo el trabajo (columna denominada "respuestas" en la Tabla 1).

Posteriormente se buscó encontrar el nivel de automatización de las herramientas desarrolladas en las investigaciones analizadas, donde se quiere determinar hasta qué punto las nuevas propuestas de reutilización están ayudando a automatizar el desarrollo de aplicaciones para estos dispositivos. El nivel alto corresponde a los generadores de código, que permiten crear el software listo para personalizar a la medida del SR industrial, el nivel medio para los marcos de trabajo (framework) o lógicas de intercambio (middleware) que tienen que adaptar el código para cada plataforma. Finalmente, el nivel bajo para las técnicas de modelización o arquitecturas que se utilizan como una guía para realizar el desarrollo de aplicaciones software.

Por otra parte, se cuestionó acerca de los lenguajes de programación en que están desarrolladas las herramientas finales, para de esta forma identificar cuáles son los más utilizados y si existe algún patrón de uso.

4. Etapa de ejecución

En esta etapa de la investigación se ejecutaron los pasos establecidos en las etapas anteriores, por lo tanto se ha dividido en tres fases y se realizó con la ayuda de la herramienta en línea Parsifal (<https://parsif.al/>).

En la fase 1, se aplicó la cadena de búsqueda a cada una de las cinco bases de datos bibliográficas, se obtuvieron un total de 1102 resultados. Posteriormente, se emplearon los criterios de inclusión y exclusión sobre el resumen, palabras claves y el título a cada uno de los artículos, así como la eliminación de

investigaciones duplicadas, dando como resultado un total de 152 resultados, denominados “candidatos a estudios primarios”. Posteriormente en la fase 2 se realizó el mismo proceso de aplicar criterios de selección, pero en este caso sobre el artículo completo, teniendo como resultado final un total de 73 artículos, denominados “artículos primarios”. Finalmente, en la fase 3 se realizó la extracción de datos, la caracterización de los estudios y las respuestas a las preguntas de investigación propuestas. Por otra parte, con respecto a los estudios duplicados se seleccionaron según el mayor índice de citación que tenían y aquellos de acceso completo a la publicación. En el caso de los estudios secundarios, no se tuvieron en cuenta para realizar la extracción de información, pero sí, para comparar los resultados de estos con los de la presente investigación.

5. Resultados y análisis de resultados

Una vez realizada la búsqueda en cada una de las bases de datos (1102 artículos resultantes), se ejecutó la fase 1 (análisis del resumen, título y palabras claves) donde se obtuvieron 152 investigaciones resultantes, finalmente, después de hacer la lectura completa de cada uno de los artículos se obtuvieron 73 artículos primarios.

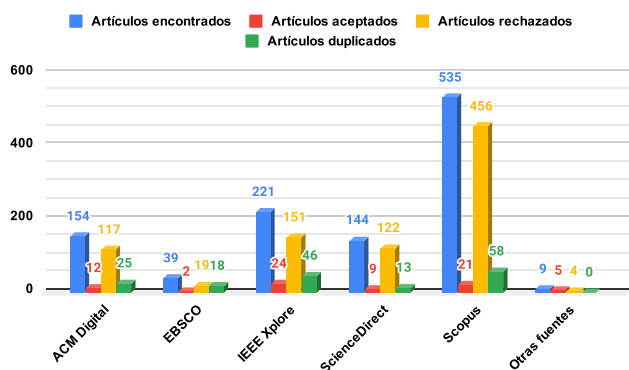


Figura 3. Aportes a la investigación de cada una de las bases de datos y motores de búsqueda académicos antes de ser evaluados los artículos.

En la Figura 3 se detalla cuál fue la contribución de cada una de las bases de datos, los artículos primarios, duplicados y rechazados. Además, permite inferir que la mitad de los resultados sin analizar corresponden a la base de datos Scopus, mientras que, la base de datos científica “IEEE Xplore Digital Library” es la fuente que más artículos primarios aportó (artículos aceptados).

A continuación, se les da solución a las preguntas de investigación planteadas anteriormente.

5.1. P1: ¿Qué estudios existen que utilicen enfoques o técnicas de reutilización de software aplicadas al dominio de los sistemas robóticos industriales?

En la actualidad, han surgido alternativas probadas que permiten incrementar la productividad en el desarrollo a nivel de software de los sistemas robóticos industriales, entre las técnicas más preponderantes se encuentran: ingeniería dirigida por modelos, ingeniería de software basada en componentes, arquitectura orientada a servicios y la arquitectura de software para un dominio específico (DSSA, siglas en inglés Domain-

specific Software Architecture). Sin embargo, también existen otros enfoques menos tradicionales, como la Robótica Orientada a Objetos (OOR, siglas del inglés Object-Oriented Robotics) o la Programación Orientada a Tareas (TOP, siglas en inglés Task-Oriented Programming).

En la Figura 4 se observa que la técnica más utilizada de reutilización de software es el MDE con un 30.1%, este enfoque comprende la creación de modelos abstractos de software y los transforma sistemáticamente en implementaciones concretas. Un modelo captura características importantes de la realidad y descuida los detalles irrelevantes (Brugali, 2015). Una herramienta común para realizar el modelado en este dominio es el Eclipse Modeling Framework (EMF, (Degueule et al., 2017)), por su facilidad para generar código, construir herramientas y otras aplicaciones a partir de modelos. Usualmente las investigaciones que utilizan el MDE son realizadas en cuatro pasos aproximadamente. En primer lugar, efectúa un análisis del dominio en el que se realiza la abstracción de este, luego, se plasma en un meta-modelo, seguidamente, se implementa el perfil de modelado en un lenguaje generalmente UML (Bandi et al., 2013), finalmente se lleva todo a un entorno de modelado, que puede ser Papyrus (Gorecki et al., 2019) u otra herramienta. El objetivo del MDE es mejorar el proceso de generación de código a partir de un resumen de modelos que describen el dominio robótico.

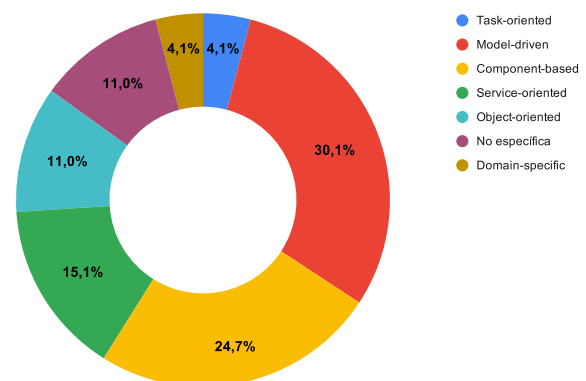


Figura 4. Distribución de metodologías de reutilización de software en el dominio de la robótica industrial.

Posteriormente, se encuentra el enfoque CBSE con un 24.7% de los estudios, su orientación cambia el énfasis de construcción de la programación tradicional a la composición de sistemas modulares, a través de una mezcla de componentes estándar y a medida por lo tanto si una pieza del software está dañada no se tiene que crear desde el principio el programa, y solo es necesario reemplazar dicho componente mejorando de esta forma la mantenibilidad del software de los SR industriales (Adi and Sekiyama, 2015). Este enfoque busca separar los módulos principales del dominio robótico en componentes, considerándose estos como una pieza de software que efectúa una funcionalidad importante (por ejemplo, sensores, actuadores, planificación de rutas, etc.). Los componentes son elementos reutilizables, mantenibles y pueden ser ensamblados para construir distintas configuraciones de aplicaciones (Brugali and Scandurra, 2009). La comunicación entre componentes se realiza mediante protocolos de solicitud-respuesta, y las conexiones

de comunicación sobre canales establecidos entre un proveedor y un solicitante (Ahmad and Babar, 2016). El enfoque CBSE resulta muy atractivo para los sistemas robóticos distribuidos y permite encapsular las funcionalidades de forma coherente y mantenible, dando solución a problemas específicos que se presentan en este tipo de formas de trabajo (Doukas and Thramboulidis, 2011), (Brugali, 2015).

El tercer enfoque más utilizados es el SOA con un 15.1%, este considera un servicio como una representación lógica de una actividad del dominio que puede comunicarse a través de una interfaz estándar de intercambio de mensajes. Este enfoque se caracteriza por el bajo acoplamiento de los servicios hacia las tecnologías que los implementan mejorando de esta forma el proceso de reutilización, pues sus funcionalidades son independientes del hardware que los contiene. El uso de SOA en los sistemas robóticos industriales está directamente relacionado con la utilización de servicios web, que permiten ampliar sus posibilidades y eliminar limitaciones de este tipo de dispositivos (Bhavsar et al., 2019).

Los estudios que no especificaron el tipo de enfoque están relacionados con el desarrollo de interfaces gráficas (comunicación multimodal), que permiten mejorar la programación de los robots industriales sin utilizar ningún tipo de modelamiento o descomposición del sistema. La razón de incluirlos en el estudio está relacionada con la aceleración en el proceso de construcción de aplicaciones software para estos dispositivos, debido a que utilizan la comunicación multimodal a través de un proceso de demostración para enseñar al robot lo que debe hacer, es decir capturan y transforman las demostraciones en tareas que el robot posteriormente ejecuta, sin necesidad de programación adicional (Haage et al., 2017).

5.2. P2: ¿Cuál es la frecuencia de publicación de investigaciones que hagan reutilización de software para la construcción de sistemas robóticos industriales?

En la Figura 5 se puede apreciar la distribución anual de publicaciones realizadas en relación con la reutilización de software y el dominio de los sistemas robóticos industriales. Es destacable indicar que los primeros 20 años correspondientes al periodo 1985 a 2005 tuvo un flujo de publicación relativamente bajo, con un total de 14 investigaciones realizadas correspondientes al 19.2% de los estudios totales, en cambio, el periodo comprendido del 2007 al 2020 se realizaron 59 estudios lo que equivale al 80.8% restante. Particularmente, en los últimos tres años (2020-2017) se han publicado el 27.4% de las investigaciones, mientras, que el 72.6% sobrante ocurrió en los 31 años anteriores. También, hay que detallar que el año 2019, tuvo la mayor cantidad de publicaciones sobre este tema. Esto sugiere que es un área de investigación actual, con problemas relevantes y con acogida por la comunidad de investigadores.

El interés de la comunidad científica se acrecentó a partir del nacimiento del Sistema Operativo Robótico (ROS) en el año 2007, convirtiendo a este marco de trabajo en punta de lanza para el desarrollo de sistemas robóticos a nivel de hardware y software (Wei et al., 2016), aumentando significativamente, a partir de ese año, el número de publicaciones en esta área del conocimiento. Específicamente, en el periodo comprendido entre 2017-2020, ROS estuvo

presente en el 60% de las investigaciones, mientras que, los desarrollos a la medida (Ad-hoc) se realizaron en un 30%, dejando el 10% restante, al CERN Robotic Framework (Lunghi et al., 2019) y el software Lego Mindstorms, lo que permite intuir el aumento en el uso de ROS como herramienta que apoya la reutilización de software.

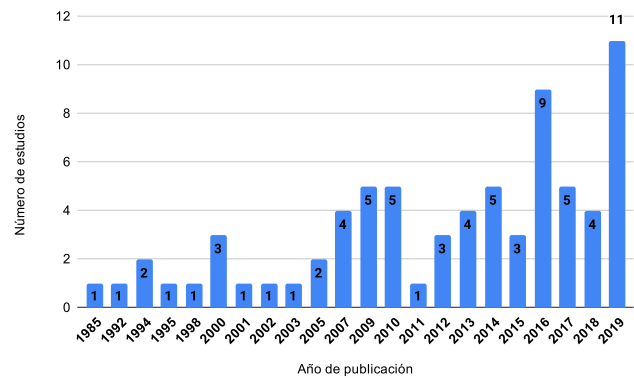


Figura 5. Distribución de investigaciones de reutilización de software en el dominio de los SR industriales.

5.3. P3: ¿Qué tipo de herramientas se están desarrollando para implementar las técnicas de reutilización de software en los SR industriales?

Las herramientas desarrolladas para apoyar los enfoques de reutilización van desde arquitecturas software, pasando por marcos de trabajo y llegando finalmente, a los generadores de código automático.

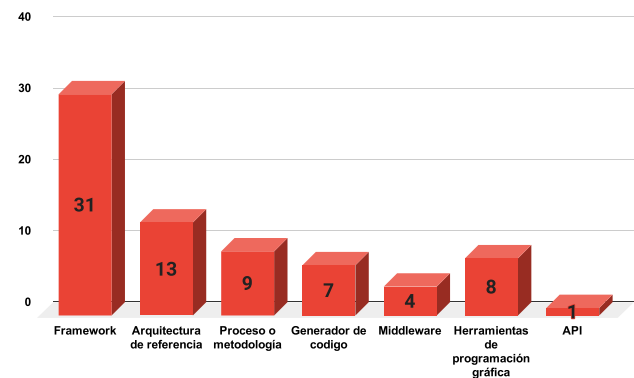


Figura 6. Tipos de herramientas desarrolladas en el dominio de los sistemas robóticos industriales.

En la Figura 6 se puede observar que las herramientas más desarrolladas son los marcos o entornos de trabajo con un 42.5%, seguido de las arquitecturas de desarrollo con un 17.8%, posteriormente los procesos o metodologías con un 12.3% de los aportes totales a esta área. Una herramienta importante como los generadores de código se presentan en un 9.6% de los desarrollos totales. Esta distribución de herramientas sugiere que el área está realizando los pasos iniciales para establecer un conjunto estandarizado de conceptos, prácticas y criterios como base para integrar la reutilización de software en los SR industriales, pero en la que otras estrategias de reutilización pueden ser pensadas como la

generación automática de programas o herramientas de programación gráfica.

5.4 P4: ¿Con qué frecuencia se relacionan las herramientas y los enfoques de reutilización de software en el dominio de los robots industriales en los estudios analizados?

En la Figura 7 se puede apreciar que los enfoques que tienen la mayor distribución de herramientas son los enfoques MDE y CBSE, además la mayor cantidad de contribuciones se concentran en estos dos de forma continua. El enfoque SOA también tiene una gran distribución de herramientas de diferentes tipos, aunque las contribuciones se hacen con menos frecuencia que los enfoques mencionados.

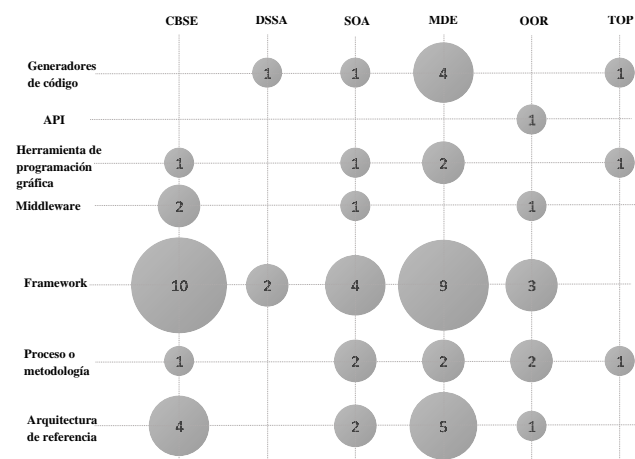


Figura 7. Gráfico de burbujas donde se puede observar de manera gráfica la distribución de herramientas según el enfoque de reutilización de software.

La robótica orientada a objetos, la arquitectura de software específica del dominio y la programación orientada a tareas son enfoques ortogonales a los otros enfoques, los cuales orientan conceptual o técnicamente la organización de la reutilización. Asimismo, los marcos de trabajo (framework) son las herramientas que son más transversales a todos los enfoques y donde la mayoría de las contribuciones se realizan de forma continua. Existe una mayor cantidad de trabajos que se concentran en la parte baja (Figura 7), que corresponde a las herramientas de bajo nivel, tales como las arquitecturas de software o los entornos de trabajo, surgiendo oportunidades de investigación en la parte superior, como los generadores de código. Existe una brecha en el enfoque CBSE con respecto a los generadores de código, donde se puede observar que no hay estudios de este tipo y podría ayudar a mejorar la reutilización de software en el dominio, considerando aspectos de calidad como escalabilidad, portabilidad y soporte del ciclo de vida de las aplicaciones (Estévez et al., 2018).

Algunas investigaciones que desarrollan herramientas de bajo nivel (marcos de trabajo, lógicas de intercambio o arquitecturas) y que se apoyan en instrumentos como ROS u OROCOS, tratan de asegurar que sus desarrollos puedan trabajar con el mayor número de plataformas que ayudan a la reutilización. Lo contrario ocurre con aquellas investigaciones que generan sus desarrollos de forma independiente, debido a que buscan darle un mejor soporte al ciclo de desarrollo de software. Este es el caso de la investigación de Estévez et al. (Estévez et al., 2016), que busca no solo generar código de

forma automática, sino que también apoya el adición de nuevas tareas que se requieran.

5.5 P5: ¿En qué contexto se está haciendo los principales aportes al dominio de la robótica con técnicas de reutilización de software?

Se están realizando continuas contribuciones al desarrollo de aplicaciones robóticas, principalmente desde dos lugares, la academia (investigación) y la industria (inversión) e incluso en casos concretos estos dos estamentos trabajan de manera conjunta. La mayoría de los aportes provienen de la academia, en forma de nuevas metodologías, técnicas o aportaciones novedosas. Generalmente, estas nuevas contribuciones se prueban mediante un estudio de caso y más adelante, si esta es válida, las empresas adoptan estas aportaciones y las llevan al contexto de la industria. Por otro lado, es difícil encontrar empresas que inviertan directamente en investigación, aunque, hay casos específicos en los que las fábricas desarrollan una línea de productos software de aplicaciones robóticas para brazos industriales (Brugali and Hochgeschwender, 2018). La reutilización de software en el dominio de los robots industriales a nivel de empresa (industria) ha sido dominado por los enfoques MDE y CBSE, siendo los enfoques más utilizados y con mayor evidencia empírica de investigación.

5.6 P6: ¿Cómo se está implementando las estrategias de reutilización de software en SR industriales?

A continuación se describe cómo se están aplicando los principales enfoques en el dominio de los SR industriales.

5.6.1 Ingeniería dirigida por modelos (MDE)

Dentro de la reutilización de software en el dominio de la robótica industrial, el enfoque MDE ha surgido como una alternativa probada y funcional. Este enfoque consiste en encapsular la complejidad en modelos, mejorar las interfaces y definir las arquitecturas para un dominio específico. Los modelos se les considera piezas de software que son representaciones abstractas de un sistema real o un fenómeno, considerando las propiedades que son relevantes y descartando los detalles que no lo son (Brugali, 2015). Normalmente, el enfoque MDE se considera una arquitectura con múltiples meta-niveles definidos por modelos (Mi), que permiten formar la pila de reflexión MDE. En el nivel más bajo se encuentra el modelo de objetos (M0) que es el código fuente ejecutándose sobre el nivel. Después, está el nivel de modelo de implementación específico a una aplicación en el dominio (M1). Posteriormente, la forma de representación para describir el dominio específico, se definen en el nivel conocido como el meta-modelo (M2). Finalmente, en el nivel más alto de abstracción, existe el modelo que representa las abstracciones y restricciones para modelar cualquier tipo de conocimiento de un dominio, se le denomina meta-meta-modelo (M3) (Arne et al., 2016). En la robótica industrial, este estándar ha sido adoptado de modo que el dominio se representa en tres o cuatro niveles dependiendo de la herramienta a desarrollar. Por ejemplo, en el documento de Bruyninckx en (Bruyninckx et al., 2013) el nivel más alto de la pila (M3) está representado por un meta-meta-modelo Ecore proporcionado por la herramienta de modelización EMF. El nivel (M2) está representado por un modelo abstracto (meta-

modelo) independiente de la plataforma. En la capa (M1) se representa el modelo de aplicación de los robots industriales. Por último, en este caso (M0) es una instancia del mundo real, representado por los marcos de trabajo OROCOS o ROS, pues, la herramienta desarrollada busca estructurar y formalizar el desarrollo de aplicaciones robóticas, mas no llevarlo a la implementación como tal.

En el documento de Bubeck et al. en (Bubeck et al., 2014) buscan separar mediante roles de usuario el desarrollo de aplicaciones robóticas industriales, utilizando la capa (M3) para el meta-meta-modelo Ecore, seguidamente, la capa (M2) que proporciona la cadena de herramientas para describir los meta-modelos, en este caso, un lenguaje específico del dominio para la herramienta, posteriormente, la capa (M1) brinda los mecanismos para describir el modelo. Por último, la capa (M0) es la implementación de la herramienta, permitiendo generar código para acoplarse con los nodos de ROS.

Cabe señalar que de la totalidad de investigaciones que utilizan el enfoque MDE, solo el 18.9% sigue las capas de abstracción establecidas por la OMG, estas se denominan M0 (instancias), M1 (modelo del sistema), M2 (modelo del modelo) y M3 (modelo de M2). Estos niveles permiten la adopción de la ingeniería dirigida por modelos de una manera más formal. Por otra parte, el 81.8% restante de los estudios no siguen esta propuesta, pero si adoptan el enfoque MDE de forma particular, en este caso, se realizan modelos donde cada uno representa de manera abstracta los conocimientos y actividades del dominio robótico industrial.

5.6.2 Ingeniería de Software Basada en Componentes (CBSE)

El enfoque CBSE está siendo ampliamente utilizada para hacer reutilización de software en los SR industriales, funciona mediante solicitudes entre proveedor y cliente. En general, los clientes solicitan funcionalidades que son proporcionadas por un proveedor, el cual suministra una colección de componentes reutilizables y listos para usar. Estos se encuentran almacenados en un depósito (Ahmad and Babar, 2016). Este enfoque utiliza los principios de modularidad, extensibilidad y reutilización, que actualmente son requeridos en el campo de la robótica para optimizar el desarrollo de este tipo de sistemas (Wigand et al., 2017).

Específicamente, Wei et al. en (Wei et al., 2009) realiza la descomposición del robot industrial en dos partes: la primera el control del sistema robótico y la segunda el sistema controlado. El primero corresponde a la aplicación software del robot, mientras que la segunda a los componentes del hardware, por lo tanto el documento propone una lógica de intercambio que permita interactuar estas dos partes. Para ello, el sistema se descompone en cuatro componentes generales (alojados en un repositorio de componentes). El primero denominado “componente atómico”, este es el único que interactúa con el hardware del dispositivo. Posteriormente, se encuentra el “Componente compuesto” que es la unión de varios componentes atómicos para proporcionar mayor granularidad al sistema, un ejemplo de esto es el chasis de un robot que comprende varios sensores y motores, el componente permite la interacción con estos dispositivos más allá de la variabilidad del número de elementos. Seguidamente, se encuentra el “componente algorítmico”, el cual encapsula

todos los algoritmos que necesita el robot, por ejemplo, la planificación de caminos, los filtros Kalman o los modelos cinemáticos. Finalmente, se encuentra el “componente de aplicación” que es el lugar donde los componentes interactúan para hacer una aplicación concreta del sistema robótico. Cada uno de los cuatro componentes principales puede interactuar con los subcomponentes (por ejemplo, el componente algoritmo puede interactuar con un subcomponente que contiene las cinemáticas inversas del robot), y estos deben obedecer.

En esta misma línea, en el documento de Beck et al. en (Beck et al., 2000), la reutilización permite que los recursos y operaciones de fabricación común, se realice a través de las familias de productos. A diferencia de la investigación anterior, el proceso de reutilización por componentes se hace de la siguiente forma: El código probado y funcional del sistema robótico se sustenta en componentes básicos de la arquitectura. Posteriormente, el repositorio de componentes permite proporcionar estos elementos con características específicas, para mejorar las prestaciones del sistema robótico. Finalmente, la arquitectura proporciona un entorno de reutilización para la definición de componentes básicos, que permite el modelado de recursos del sistema, pasos y recetas de piezas. En resumen, el sistema está compuesto por una serie de componentes generales que proporciona funcionalidades básicas del robot (algoritmos, trayectorias, etc.), pero permite adaptar otros componentes dependiendo de la aplicación (diferentes sensores o actuadores) que se requiera.

5.6.3 Arquitectura Orientada a Servicios (SOA)

La implementación de SOA en el dominio de los SR industriales, sigue el mismo modelo de su contraparte en el desarrollo de software, donde existen un conjunto de servicios que se comunican entre sí, utilizando protocolos de comunicación estándar (Bhavsar et al., 2019). En esta arquitectura se consideran tres elementos fundamentales: corredor, publicador y suscriptor junto con sus protocolos de comunicación. El proceso de interacción entre los elementos es el siguiente, el publicador proporciona una representación del servicio al corredor, luego, el suscriptor accede al corredor y recibe información de los servicios disponibles y por último, el suscriptor se conecta con el publicador para solicitar los servicios requeridos (Ahmad and Babar, 2016).

En el documento de Rudorfer et al. en (Rudorfer et al., 2018) integra el enfoque SOA al dominio robótico, intentando mejorar el proceso de programación de estos dispositivos, mediante una interfaz de arrastre intuitivo. La investigación propone encapsular funcionalidades como servicios y que se comuniquen a través de una interfaz común permitiendo la interoperabilidad, la reutilización y la escalabilidad. Dependiendo de la tarea a realizar por el robot, las funcionalidades (servicios) pueden ser intercambiadas, permitiendo la adaptación de los sistemas a la medida de los requerimientos.

También, en el documento de Chen et al. en (Chen et al., 2010) combina la arquitectura orientada a servicios y la computación en la nube, acuñando el término “robot as service (RaaS)” que basa su funcionamiento en tres directrices principales: la unidad RaaS es un proveedor de servicios en el que un desarrollador o cliente puede desplegar nuevos

servicios o eliminarlos. La segunda directriz está relacionada con que la unidad RaaS contiene aplicaciones, el cliente puede acceder a nuevas funcionalidades que se encuentren disponibles. Finalmente, la unidad RaaS es el publicador de servicios donde un cliente puede buscar los servicios y aplicaciones disponibles en la unidad directorio. Es decir, tiene las mismas características que el enfoque original, con la diferencia de que el editor está en la nube y es accesible desde un protocolo web a distancia. Una variación de este enfoque se encuentra propuesta en el documento de Salman et al, en (Salman et al., 2019), el autor plantea utilizar la computación en la niebla en los SR industriales, la cual consiste en reemplazar las limitaciones de cómputo de los controladores de los robots, y trasladarlo a la capa niebla. Aunque, en este documento se plantea una arquitectura, pero no se implementa con un caso específico de estudio.

Por último, existe una tendencia creciente a integrar la nube con los sistemas robóticos permitiendo que accedan a recursos de almacenamiento y cómputo a distancia, además de mejorar las actividades y procesos colaborativos entre estos dispositivos (Wan et al., 2016). Esto convierte a SOA en una alternativa para tener en cuenta para realizar reutilización de software en el dominio.

5.6.4 Otros enfoques de reutilización utilizados

Otro enfoque importante encontrado en la revisión de la literatura fue la robótica orientada a objetos, aunque, actualmente no es el más elegido por los desarrolladores, fue uno de los primeros acercamientos de la ingeniería de software en el dominio robótico. La OOR utiliza tres conceptos principales: el componente central o núcleo, componentes genéricos y componentes específicos. Estos tienen propiedades de interconexión, herencias y usos. Por jerarquía, los componentes genéricos deben contener componentes específicos, mientras que, los componentes específicos deben heredar características de los componentes genéricos. Asimismo, el componente principal encapsula el control de bajo nivel, que son instancias de los controladores hardware y software que permiten la interacción entre estos. Permitiendo de esta forma reutilizar esta lógica para robots con características similares. Por lo tanto los componentes genéricos utilizan el núcleo para proporcionar la funcionalidad general del sistema robótico, mientras que, los componentes específicos heredan las funcionalidades de los componentes genéricos (Tibermacine et al., 2016).

Un enfoque que se ha utilizado en menor medida el OOR, que consiste en abstraer las características comunes de las labores que realiza el robot y descartar los elementos no comunes (sensores, actuadores y sistemas operativos), de modo que el desarrollador pueda centrarse en la tarea que debe realizar el dispositivo, y no preocuparse por la variabilidad del hardware. Generalmente, este enfoque se realiza a través de comunicación multimodal y se utiliza cuando se requiere un trabajo colaborativo entre sistemas robóticos (Trapani and Indri, 2017), (Backhaus and Reinhart, 2017).

Por último, también ha surgido una alternativa para mejorar el desarrollo de aplicaciones software para robots, la cual utiliza comunicación multimodal. Esta consiste en demostrarle al robot mediante acciones (gestos o habla) las tareas que debe realizar. En este caso el desarrollador ahorra tiempo porque no

debe programar trayectorias o cinemáticas, debido a que el robot imita las acciones aprendidas, sumado a esto permite a personas no expertas poder interactuar y reprogramar de manera sencilla el dispositivo, evitando tiempos demasiado largos de reprogramación o apagado (Guerin et al., 2014), (Stenmark et al., 2017). Además se puede intuir que existe una tendencia creciente a desarrollar herramientas de programación gráfica, más allá, del proceso de reutilización que pueda existir.

5.7 P7: ¿Se han utilizado los enfoques de forma combinada o de manera aislada?

En la búsqueda se encontró que existen pocos estudios que combinen técnicas de reutilización de software en el ámbito de la robótica industrial, en el artículo de Doukas y Thramboulidis en (Doukas and Thramboulidis, 2011) se halló que se hace uso del MDE y CBSE de manera conjunta. La investigación propone dividir el sistema robótico en modelos y acoplarlos con la norma IEC 61499 (Vyatkin, 2011), que es un estándar para procesos industriales y medición, ésta defiende que la mejor manera de construir procesos eficientes es a través de componentes. De esta forma la investigación plantea combinar estos dos paradigmas para mejorar la construcción de sistemas robóticos, permitiendo definir un modelo de aplicación (MDE) del robot usando conceptos abstraídos en componentes (CBSE). También, otra investigación que combina dos enfoques de reutilización de software en el dominio, es la realizada por Wenger et al. en (Wenger et al., 2016) donde describe la creación de una aplicación denominada ReApp que es un marco de trabajo basado en ROS, capaz de proporcionar y ampliar habilidades a los sistemas robóticos más allá del lenguaje en el que esté programado. La herramienta utiliza el MDE para abstraer las funciones del sistema robótico, y los desarrolladores pueden agregar nuevas habilidades a este, en forma de componentes que se comunican naturalmente con otros y con el sistema robótico. Igualmente, utiliza la norma IEC 61499, que permite a ReApp seguir sus directrices para operar de manera correcta y eficiente, especialmente comportamiento en tiempo real del sistema robótico industrial.

Por otra parte, los 71 estudios restantes no presentaron explícitamente la utilización de enfoques de reutilización de manera conjunta. Ahora, las investigaciones (Lotz et al., 2016) y (Yoong et al., 2012) han combinado los enfoques MDE y CBSE en el dominio robótico general, que son casos exitosos, y pueden ser tomados como ejemplo para implementarse en el dominio robótico industrial. La mayor ventaja observada en este tipo de investigaciones es que se aprovecha lo mejor de cada uno de los enfoques (combinación MDE y CBSE). Permitiendo proporcionar una forma intuitiva de crear componentes siguiendo el modelo de ejecución sincrónico, y a su vez permitir la comunicación entre esos componentes de forma natural y sin intermediarios.

5.8 P8: ¿Las técnicas de reutilización de software en el dominio de la robótica industrial se están aplicando de manera ad-hoc o siguiendo un estándar o metodología ya probada?

Se han desarrollado herramientas que permiten realizar el proceso de reutilización en el dominio de forma más sencilla, alternativas como el sistema operativo robótico, OROCOS e

incluso CORBA buscan mejorar y agilizar el proceso de desarrollo de software. En la Figura 8 se observa la distribución de las investigaciones encontradas, separadas por las tecnologías en que se basaron para hacer la reutilización. En este caso el 49.3% del total de las investigaciones no están basadas en ningún estándar particular o en algún proyecto anterior (tecnologías Ad-hoc), en este tipo de trabajos se implementan las técnicas de reutilización de software a medida. Por otra parte, el 30.1% utiliza a ROS como base para sus investigaciones, que generalmente tienen que ver con la creación de nuevas librerías para el sistema operativo o nuevas funcionalidades. También, se encuentra OROCOS con un 4.1% de los estudios, este busca desarrollar un marco modular para el control de robots y máquinas de uso general, de forma accesible para todos. En otras palabras, establece una infraestructura para el desarrollo de sistemas en tiempo real en lenguaje C++ (Rastogi et al., 2017). También se encontró que existen alternativas menos populares como CORBA y RTLinux.

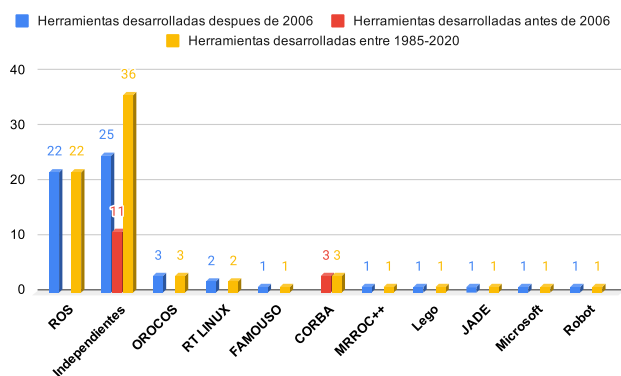


Figura 8. Distribución de tecnologías que apoyan la reutilización de software en los SR industriales.

Por otra parte, en el año 2007 existió un aumento notable en el flujo de trabajos e investigaciones que incluían los enfoques de ingeniería de software en los sistemas robóticos, paralelamente, en ese año surgió ROS, que se ha convertido gradualmente en un punto de referencia para introducir metodologías y/o herramientas de reutilización de software en el dominio de la robótica (Wei et al., 2016). Si se realiza un filtro de las investigaciones hechas desde el año 2007 (creación de ROS) hasta la fecha, se puede encontrar que los estudios que utilizan ROS aumentan a un 37.3%, mientras, que las investigaciones con tecnologías Ad-Hoc decrecen a un 42.4%. Además, el uso de OROCOS crece en este periodo de tiempo, estableciéndose como una opción para tener en cuenta. Es importante mencionar que CORBA desaparece de la gráfica, pues, su utilización se dio en mayor medida entre los años 2000 y 2005, por tanto, es una herramienta que actualmente no satisface las necesidades de los desarrolladores de aplicaciones robóticas.

Con los datos anteriores se deduce que existe una tendencia creciente a utilizar ROS, y este podría establecerse como un referente para introducir técnicas de reutilización de software en el dominio de la robótica industrial, debido a que ofrece funcionalidades valiosas como el proveer una infraestructura de comunicación mediante la modalidad publicador y suscriptor, haciendo uso de nomenclaturas y registros que son

proporcionados por un maestro. También, tiene opciones específicas para cada tipo de robot, es decir, más allá de los componentes centrales, brinda características secundarias como librerías de diseño de interfaces, herramientas de diagnóstico, mapeo, localización y navegación. Incluso, permite trabajar con otros softwares como Gazebo, OpenCV, Qt y MoveIt, que actualmente son activos importantes que facilitan el desarrollo para sistemas robóticos (“ROS.org | Powering the world’s robots,” 2020). ROS utiliza una licencia BSD, lo que ha permitido desarrollar numerosas bifurcaciones que permiten enfocarse en dispositivos específicos, este el caso de MicroROS orientado a llevar las funcionalidades del sistema operativo robótico a los microcontroladores o el caso de ROS-Industrial centrado en brazos robóticos. Este aumento en la oferta de herramientas que apoyan la reutilización de software ha permitido a los desarrolladores, ampliar sus investigaciones y no limitarlos a un nicho específico. Además, se espera que en el futuro aumente la adopción de este marco de trabajo por parte de los programadores, puesto que, el desarrollo de nuevas tecnologías que funcionan de forma independiente disminuye, seguramente, porque los desarrolladores han encontrado en el sistema operativo robótico u otras posibilidades un soporte adecuado para realizar sus investigaciones.

5.9 P9: ¿Cuáles son los principales problemas y retos en el área de reutilización de software en el dominio de los sistemas robóticos industriales?

La implementación de los enfoques de reutilización de software en la robótica industrial se está realizando mediante el desarrollo de piezas de software de grano fino, que permiten adaptar los sistemas robóticos a las necesidades del cliente, por lo tanto se toman pequeñas funcionalidades de los sistemas robóticos y se personalizan según los requerimientos. Este tipo de abstracción tiene ventajas permitiendo el desarrollo de una gran variedad de sistemas construidos de múltiples piezas de software reutilizables. Aunque esto ha generado una problemática creciente en el dominio, que tiene que ver con la falta de apoyo a la reutilización, pues, no se genera soluciones efectivas a problemas reiterados de diseño arquitectónico (Brugali, 2015), que pueden ser resueltos realizando procesos de abstracción de grano grueso, abarcando así problemas concurrentes en el diseño de los SR industriales. El documento de Gherardi y Brugali en (Gherardi and Brugali, 2014) intenta resolver este problema desarrollando una arquitectura de referencia, que permite un mayor nivel de abstracción del dominio robótico, pero no ha tenido el impacto esperado y los desarrolladores no han adoptado esta solución como propia.

Otro problema recurrente en este dominio está relacionado con la heterogeneidad de los componentes físicos de los robots, cualquier cambio en la configuración del sistema puede implicar una cadena de cambios en otros componentes, que pueden requerir herramientas específicas o tiempos de reprogramación muy grandes, lo que genera incertidumbre para los desarrolladores, pues, estos dudan de realizar el proceso de reutilización si posteriormente deben invertir grandes cantidades de tiempo en acondicionar el sistema a nuevos requerimientos o tareas. Asimismo, surge otra problemática específicamente entre los sensores y actuadores, pues, estos cumplen diferentes funcionalidades, distintas

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.

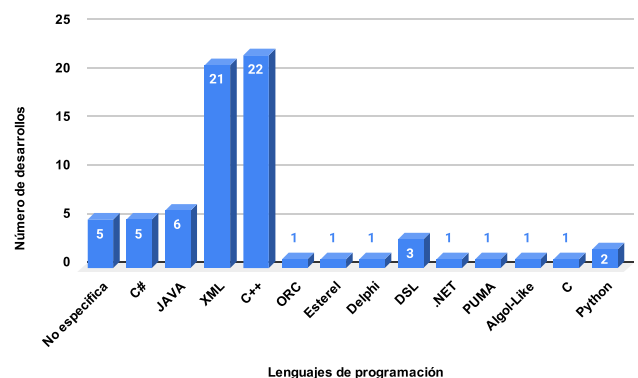


Figura 9. Distribución de lenguajes de programación para herramientas desarrolladas

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., Ibarra, 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>