

Open Services for Lifecycle Collaboration: Um Estudo de Mapeamento Sistemático

Bruno Marcelo S. Ferreira¹, Rafael S. Torres¹, Fábio P. Basso¹,
Elder Rodrigues¹, Maicon Bernardino¹, Rafael Z. Frantz²

¹Universidade Federal do Pampa (UNIPAMPA), PPGES
Alegrete - RS

²Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUÍ)
Ijuí - RS

brunoferreira.aluno@unipampa.edu.br

Abstract. *The software industry invests in modern tools throughout the software development lifecycle. However, there are challenges to achieve an end-to-end integrated environment such as data integration and artifact traceability. To mitigate these challenges, many approaches have been proposed for integration. In this context, Open Services for Lifecycle Collaboration (OSLC) is an open standard for tool interoperability, which allows data federation throughout Software Engineering (SE) application lifecycles. This study presents a systematic mapping study about OSLC, analyzing 59 primary studies and addressing integration issues.*

Resumo. *A indústria de software investe em ferramentas modernas ao longo de todo o ciclo de desenvolvimento de software. No entanto, existem desafios para alcançar um ambiente integrado de ponta a ponta, como por exemplo estabelecer a rastreabilidade dos artefatos. Para mitigar esses desafios, muitas abordagens foram propostas para integração de ferramentas de software. Nesse contexto, o Open Services for Lifecycle Collaboration (OSLC) é um padrão aberto para interoperabilidade de ferramentas, que permite a federação de dados ao longo do ciclo de vida de aplicações de Engenharia de Software (ES). Este artigo apresenta um estudo de mapeamento sistemático sobre OSLC, analisando 59 estudos primários e abordando questões de integração.*

1. Introdução

Ferramentas de software são usadas para apoiar times a concluir atividades ao longo do ciclo de vida de desenvolvimento de software. No entanto, essas ferramentas geralmente são projetadas sem o suporte necessário para serem integradas a outras ferramentas de software. Além disso, possuem configurações, funcionalidades, fabricantes e manipulam diversos tipos de artefatos. Esse cenário adiciona desafios para a obtenção de um ambiente integrado de ponta a ponta, como por exemplo estabelecer e manter relacionamentos entre artefatos através de toda cadeia de ferramentas.

Ambientes integrados de forma totalmente automatizada são um fenômeno raro na indústria devido ao número de características envolvidas no processo de interoperabilidade de ferramentas [Wicks and Dewar 2007]. Apesar disso, muitas abordagens foram

propostas para *assets* fornecidos como serviços [Biehl et al. 2014]. Essas abordagens visam a minimizar os problemas relacionados ao processo de integração de ferramentas adotados nos contextos de Engenharia de Software (ES) por meio de uma arquitetura típica de serviços [Zhang and Møller-Pedersen 2014].

Nesse contexto, uma emergente solução industrial foi proposta: Open Services for Lifecycle Collaboration (OSLC) [OSLC 2020]. O OSLC define um modelo de representação comum para artefatos produzidos ao longo do projeto, bem como métodos que permitem que outras aplicações manipulem. Essas especificações são definidas na especificação principal do OSLC, que consiste em regras e padrões para o uso de tecnologias como o RESTful e do protocolo HTTP. Além da especificação principal, existem domínios OSLC que a estendem e definem regras para seus respectivos domínios (Gerenciamento de Requisitos, Gerenciamento de Qualidade, Gerenciamento de Configuração e Mudanças). Dessa maneira, o OSLC fornece interoperabilidade em diferentes domínios de desenvolvimento de software sem obter conhecimento sobre como as ferramentas operam internamente.

O OSLC é baseado nos princípios de Dados Ligados e usa tecnologias como protocolo HTTP e RESTful. Além disso, cada artefato é representado por arquivos RDF/XML estruturados com base em domínios OSLC. Os relacionamentos entre os artefatos são obtidos através de uma estrutura composta por três elementos: sujeito, predicado e objeto. Por exemplo, em um ambiente formado pelas ferramentas de gerenciamento de requisitos (Ferramenta A) e gerenciamento de testes (Ferramenta B), um requisito na Ferramenta A pode ser validado por um caso de teste na Ferramenta B por meio do OSLC.

Em uma cadeia de ferramentas OSLC, uma ferramenta pode ser classificada como um provedor ou consumidor. As ferramentas provedoras são projetadas para armazenar e prover dados, permitindo as ferramentas consumidoras fácil acesso para navegar, criar e recuperar dados [Aichernig et al. 2014]. Existem três abordagens para prover suporte OSLC em ferramentas de software: abordagem nativa, *plugin* e adaptador. A abordagem mais recomendada é o desenvolvimento de adaptadores, pois não é preciso possuir conhecimento das tecnologias que envolvem a implementação da ferramenta.

Uma vez que a comunidade OSLC está ativa desde 2008, uma pergunta em aberto é "qual é o estado da arte e a prática do OSLC para integração de ferramentas e ambientes de Engenharia de Software?". A seguir, é apresentado um Estudo de Mapeamento Sistemático (SMS) que analisa 59 estudos primários que relatam o uso do OSLC.

O restante do artigo está organizado da seguinte forma: na seção 2 o protocolo do estudo de mapeamento é exposto e a Seção 3 discute os resultados do SMS. A Seção 4 discute as ameaças à validade do SMS. Por fim, a Seção 5 conclui este trabalho.

2. Estudo de Mapeamento Sistemático

Nesta seção, é descrito o processo aplicado em nosso estudo de mapeamento sistemático. O protocolo seguido foi proposto por Petersen *et al.* [Petersen et al. 2015], que define um processo para execução de mapeamentos sistemáticos na área de ES. O protocolo completo deste trabalho está disponível no Google Drive ¹.

¹https://drive.google.com/drive/folders/14Fus_yehj1RxRXP53_V1ATxgzsdfo7Bd?usp=sharing

2.1. Questões de Pesquisa

O Estudo de Mapeamento Sistemático possui as seguintes questões de pesquisa (Q):

Q1. *Em quais fases do ciclo de vida do software o trabalho utiliza o padrão OSLC como solução?* Os estudos de integração de aplicações voltados para contextos de ES são normalmente associados a determinados ciclos de vida da aplicação. Assim, esta questão de pesquisa busca caracterizar os estudos que contribuem para determinadas disciplinas de ES, fornecendo um mapa de cobertura que permite inferir se os processos de desenvolvimento de software são totalmente automatizados.

Q2. *Quais são as facetas da contribuição do estudo?* Nosso objetivo é caracterizar as contribuições da área de acordo com sua faceta: seja com ferramentas, modelagem, processos e metodologias.

Q3. *Quais os tipos de pesquisa mais frequentes sobre OSLC?* Por fim, para compreender a maturidade dos estudos do ponto de vista empírico, também buscamos classificar os estudos de acordo com o tipo de avaliação adotada.

2.2. Strings de Busca

A busca pelos trabalhos aconteceram em 5 bibliotecas digitais amplamente utilizada pela comunidade científica: ACM Digital Library², IEEEExplore³, Springer Link⁴, Science Direct⁵ e Scopus⁶. As *strings* de busca foram elaboradas com o objetivo de obter uma ampla cobertura de estudos sobre o padrão OSLC, como mostra a Tabela 1.

Tabela 1. Strings de Busca

Base de Dados	String de busca
ACM DL	(“Open Services for Lifecycle Collaboration” OSLC) AND (Integration Interoperability “Linked Data” Lifecycle Traceability)
IEEE Xplore	(“Open Services for Lifecycle Collaboration” OR OSLC) AND (Integration OR Interoperability OR “Linked Data” OR Lifecycle OR Traceability)
Springer Link	(“Open Services for Lifecycle Collaboration” OR OSLC) AND (Integration OR Interoperability OR “Linked Data” OR Lifecycle OR Traceability)
Science Direct	(“Open Services for Lifecycle Collaboration” OR OSLC) AND (Integration OR Interoperability OR “Linked Data” OR Lifecycle OR Traceability)
Scopus	TITLE-ABS-KEY ((“Open Services for Lifecycle Collaboration” OR OSLC) AND (Integration OR Interoperability OR “Linked Data” OR Lifecycle OR Traceability))

3. Análise dos Resultados

Esta seção apresenta a análise dos resultados obtidos através da execução do nosso protocolo. Os resultados são baseados nos artigos selecionados durante o processo de triagem, conforme ilustrado na Figura 1. Na primeira etapa, as strings de busca foram utilizadas

²<https://dl.acm.org/>

³<https://ieeexplore.ieee.org/>

⁴<https://link.springer.com/>

⁵<https://www.sciencedirect.com/>

⁶<https://www.scopus.com/>

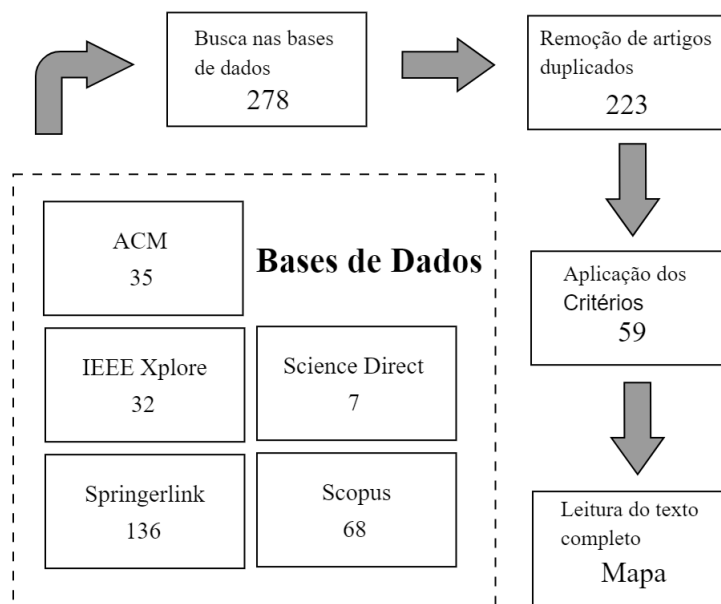


Figura 1. Processo de Triagem dos Estudos

nas bases de dados e foram retornados 278 artigos. Em seguida, os estudos duplicados entre as bases de dados foram desconsiderados, restando 223 artigos. Assim, após aplicar os critérios de inclusão e exclusão com base no título, resumo, palavras-chave e introdução desses artigos, foram considerados 59 estudos de acordo com o tema de pesquisa.

3.1. Q1. Em quais fases do ciclo de vida do software o padrão OSLC é utilizado?

Para responder a Q1, foram mapeadas as fases do ciclo de vida do software em que a solução do estudo foi empregada. Para isso usamos as disciplinas do Rational Unified Process (RUP). De acordo com os resultados do SMS, a maioria das soluções propostas são para os domínios de Gerenciamento de Requisitos, Análise & Projeto, Gerenciamento de Configuração e Mudanças e Testes, como mostra a Tabela 2. Todas as ferramentas no conjunto de ferramentas foram consideradas parte da solução OSLC. Portanto, os estudos podem ser categorizados em mais de uma disciplina.

Tabela 2. Estudos Distribuídos pelas Disciplinas do RUP

Disciplina	Quantidade de artigos
Modelagem de Negócios	1
Requisitos	21
Análise & Projeto	29
Implementação	6
Teste	25
Implantação	0
Gerenciamento de Configuração e Mudanças	14
Gerenciamento de Projetos	6
Ambiente	2
N/A	11

As abordagens OSLC se empenham em resolver problemas comuns entre os artigos. No domínio de requisitos, o OSLC é proposto para estabelecer a rastreabilidade entre os requisitos e o restante do projeto. Além disso, as ferramentas de requisitos de software são frequentemente usadas nos exemplos de conjuntos de ferramentas como um ponto de partida para o fluxo de trabalho.

A partir de conceitos de Dados Ligados, artigos como [VanZandt 2015], [Buffoni et al. 2017] apresentam abordagens para rastrear o histórico das mudanças por meio das propriedades do OSLC, incluindo as partes interessadas que o modificaram e tarefas nas quais os artefatos estão associados. Nesse sentido, abordagens de domínio Gerenciamento de Configuração e Mudanças como [Lednicki et al. 2016], [Pikus et al. 2019], [Adjepon-Yamoah et al. 2015], [Regan et al. 2015], [Aoyama et al. 2014], [Aoyama et al. 2013] são propostas para controle de versão e automação de tarefas por meio dos serviços OSLC.

Os resultados do SMS reforçam a necessidade de integrar diferentes domínios de desenvolvimento de software desde suas fases iniciais. Portanto, podemos citar ferramentas de Engenharia Orientada a Modelos (MDE), que são citadas na fase de Análise & Projeto como soluções para minimizar esses esforços de integração. Nesse contexto, o Eclipse Lyo [El-khoury 2016] propõe um gerador de código-fonte aberto para interfaces de ferramentas em conformidade com OSLC no ambiente eclipse, que foi citado por artigos que foram usados no desenvolvimento de adaptadores OSLC. Ainda, há trabalhos que propõem abordagens para apoiar a integração ou transformação de modelos heterogêneos como BPMN, UML, SysML, SoaML e EMF [d. Martino et al. 2016], [Lu et al. 2018], [Arnould 2018], [Mustafa and Labiche 2017], [Zhang and Møller-Pedersen 2013], [Biehl et al. 2012], [Biehl et al. 2012].

Outra observação é um número considerável de soluções aplicadas à disciplina de Teste devido aos contextos em que o OSLC é normalmente aplicado. Por se tratarem de sistemas críticos de segurança, os conjuntos de ferramentas são compostos por ferramentas de V&V.

3.2. Q2. Quais são as facetas da contribuição do estudo?

Identificar quais facetas de contribuição são mais exploradas em uma área de pesquisa é importante para encontrar lacunas de pesquisa ou avaliar tendências na área. Em relação à faceta de contribuição dos estudos, estabelecemos quatro categorias: modelo, ferramenta, método e processo.

Os resultados mostram que a maioria dos artigos contribui para a faceta de Ferramentas (30). Esta categoria está relacionada à implementação de adaptadores de ferramenta OSLC, serviços da web OSLC ou ferramentas para visualizar artefatos de dados. Esses trabalhos propõem exemplos de conjuntos de ferramentas que usam adaptadores OSLC como uma solução de integração. Em relação à categoria Modelo (20), trabalhos como: [Zhang and Møller-Pedersen 2014], [Gürdür et al. 2018], [Alvarez-Rodríguez et al. 2018], [Gallina et al. 2016], [Mustafa and Labiche 2017], cobrem a proposta para estender as especificações de domínios OSLC. Outro tópico representativo na categoria de modelo são abordagens para gerar especificações OSLC em ambientes baseados em MDE. Nos exemplos da categoria Processo (3) encontrados, o OSLC é usado para automatizar atividades que envolvem a rastreabilidade de artefa-

tos, conforme mostrado em trabalhos de [Baumgart and Ellen 2014], [Regan et al. 2015] e [Regan et al. 2014].

Em outros trabalhos, são propostos Métodos (20) ou novas abordagens para o uso de OSLC. Assim, foi possível identificar que as contribuições dos estudos estão voltadas para a resolução de problemas em um domínio específico, como uma atividade ou um tipo de indústria. É um indicativo da lacuna de pesquisa de abordagens para soluções globais sobre OSLC, em processos especialmente planejados para ES.

3.3. Q3. Quais os tipos de pesquisa mais frequentes sobre OSLC?

A maturidade de uma pesquisa pode ser determinada pela forma como o estudo afeta a prática na área. Por esse motivo, adotamos as categorias propostas pelo estudo de Wieringa *et al.* [Wieringa et al. 2005], mapeando os tipos de pesquisa empregados por cada pesquisa. Assim, são considerados os seguintes tipos de estudos: *Evaluation Research*, *Validation Research*, *Solution Proposal*, *Experience Report*, *Opinion Paper* e *Philosophical Paper*.

A maioria dos artigos é classificada como *Solution Proposal* (36), reforçando a ideia de que o OSLC possui implementações como prova de conceito e protótipos em ambientes reais. Alguns trabalhos sugerem a realização de estudos de caso, embora sejam apresentados dados insuficientes, levando a um pequeno número de estudos empíricos, incluindo *Validation Research* (3) e *Evaluation Research* (5). Os artigos classificados como *Philosophical Papers* (12) discutem como usar o OSLC para solucionar problemas, mas não apresentam sua implementação. Existem também diretrizes, opiniões e relatórios baseados em experiências profissionais na indústria, resultando em artigos de *Experience Report* (5) e *Opinion Paper* (1), afirmando que o OSLC é utilizado em ambientes reais.

4. Ameaças à Validade

Os trabalhos empíricos devem ser capazes de conter informações suficientes para que seja possível replicá-lo apenas com o acesso ao protocolo. Entretanto, durante sua execução existem alguns fatores que podem interferir nos resultados, sendo tratados como ameaças à validade. A seguir seguem algumas ameaças deste SMS:

Validade interna: Esse tipo de ameaça refere-se à validade da análise realizada, validando se as conclusões derivadas dos dados são válidas internamente. Nesse sentido, perguntas de pesquisa bem definidas possibilitaram concluir resultados de acordo com o tópico de pesquisa. Além disso, uma preocupação constante do estudo foi garantir que o processo de seleção de trabalhos ocorresse conforme os critérios de inclusão e exclusão definidos. Para esse fim, foi cuidadosamente investido um grande esforço para filtrar os estudos primários, com o auxílio de ferramentas para organização e filtros dos trabalhos utilizados pelos pesquisadores.

Validade externa: Diz respeito sobre o quão generalizadas são as descobertas do estudo de mapeamento. Nesse sentido, a representatividade estatística pode ser uma ameaça. Por considerar um número expressivo de artigos, essa ameaça é tratada pelo refinamento da *string* de busca por mais de um pesquisador, permitindo a inclusão de 59 estudos primários, os quais caracterizam um bom poder estatístico e a adaptação do texto para cada uma das bases selecionadas.

Validade de construção: O trabalho foi executado com base em um protocolo bem definido e amplamente utilizado na área de Engenharia de Software. Para garantir a extração dos dados, o esquema de classificação foi revisado e discutido por um segundo pesquisador. Além disso, o trabalho conta com critérios de qualidade para identificar os trabalhos mais relevantes para o tema de pesquisa. Entretanto, as categorias escolhidas para o esquema de classificação podem ser consideradas genéricas demais. Para mitigar esse viés, as categorias foram escolhidas a partir da base de conhecimento do RUP.

Validade da conclusão: Uma ameaça a conclusão pode ocorrer na fase de classificação, no qual pode ser influenciada pelo viés do *know-how* dos autores. Para superar esse viés, foi utilizado um formulário de extração de dados que contém todos os dados necessários para responder as questões de pesquisa.

5. Considerações Finais

As especificações OSLC estão se tornando cada vez mais populares na indústria de software. Ao longo dos anos, desafios e práticas foram relatados na literatura para conectar artefatos entre domínios, aplicações e organizações. Nesse sentido, este trabalho empírico ajuda a compreender as áreas potenciais de emprego do OSLC com base na análise de artigos selecionados. Por exemplo, nossos resultados mostram as motivações e contribuições do OSLC para apoiar a integração de ferramentas com base nas disciplinas da Engenharia de Software, contexto industrial, facetas da contribuição e tipo de avaliação de tais trabalhos.

Os trabalhos encontrados enfocam a geração de código-fonte para conectar ferramentas. Os autores procuram maneiras de representar os artefatos bem como explorar ontologias e a extensão dos domínios OSLC. Além disso, é consenso da indústria integrar ferramentas por meio de adaptadores, visto que as ferramentas geralmente não possuem suporte nativo ao OSLC. Nesse contexto, as abordagens de Desenvolvimento Dirigido por Modelos (MDD) e MDE são de interesse da indústria para gerar essas interfaces e ao longo dos anos, vem se consolidando como a melhor alternativa para o desenvolvimento de adaptadores OSLC. Apesar disso, a geração de código não é 100% automática e ainda não atingiu o nível de integração de todo o ambiente. Este cenário pode estar relacionado com o elevado desafio técnico necessário para integrar tais aplicações.

O primeiro trabalho que encontramos foi publicado em 2010 e relata a pesquisa e o potencial para uma nova especificação para integração de ferramentas. Desde então, pesquisadores exploraram o OSLC e propuseram novas abordagens para melhorar a integração de ferramentas em ambientes ALM [Tüzün et al. 2019].

As principais vantagens do OSLC estão relacionadas aos Dados Ligados e envolve não apenas adaptadores de ferramentas para integração ponto a ponto, mas também abordagens propostas para promover a substituição de ferramentas na cadeia de ferramentas, incluindo modificação de especificações de domínio OSLC e soluções para atividades automatizadas para integração de ferramentas. Todos esses elementos podem ser realizados por meio de abordagens como o MDE, que através de uma representação comum, pode fornecer todo o dispositivo que permite a automação das atividades e integração de todo o ambiente de desenvolvimento de software.

O OSLC é motivado por problemas de integração de contextos industriais de alta complexidade. Nesse sentido, as soluções de integração propostas pelos autores geral-

mente são projetadas para domínios específicos de aplicações de software de engenharia de sistemas. Por exemplo, a maioria dos estudos é dedicada a sistemas embarcados e a outros contextos de aplicações da indústria automobilística, naval, entre outros. Embora o OSLC proponha a integração de ponta a ponta de todo o ciclo do software, os estudos mostram que as cadeias de ferramentas existentes são, em sua maior amostra, adotadas nos estágios iniciais do projeto de software e também nas que envolvem atividades de verificação e validação.

Por fim, nós seguimos um protocolo replicável que pode ser implementado ao longo dos anos para identificar novas contribuições na área. Esse fator é especialmente importante para a comunidade acadêmica, pois fornece uma visão geral do estado da arte e da prática, sobre as áreas exploradas, as deficiências para adoção do OSLC e também as áreas inexploradas. Além disso, este trabalho pode servir para tomada de decisão no desenvolvimento de soluções de integração na indústria.

6. Agradecimentos

Este estudo foi parcialmente financiado através da Pró-Reitoria de Pesquisa (PROPESQ), com bolsa do programa AGP (Apoio a Grupos de Pesquisa), e pela FAPERGS, por meio do projeto ARD N. 19/2551-0001268-3.

Referências

- Adjepon-Yamoah, D., Romanovsky, A., and Iliasov, A. (2015). A reactive architecture for cloud-based system engineering. In *Proceedings of the 2015 International Conference on Software and System Process, ICSSP 2015*, pages 77–81, New York, NY, USA. ACM.
- Aichernig, B. K., Hörmaier, K., Lorber, F., Nickovic, D., Schlick, R., Simoneau, D., and Tiran, S. (2014). Integration of requirements engineering and test-case generation via osc. In *2014 14th International Conference on Quality Software*, pages 117–126.
- Alvarez-Rodríguez, J., Mendieta, R., Vara, J., Fraga, A., and Llorens, J. (2018). Enabling system artefact exchange and selection through a linked data layer. *Journal of Universal Computer Science*, 24(11):1536–1560. cited By 1.
- Aoyama, M., Yabuta, K., Kamimura, T., Inomata, S., Chiba, T., Niwa, T., and Sakata, K. (2013). Promis: A management platform for software supply networks based on the linked data and osc. In *2013 IEEE 37th Annual Computer Software and Applications Conference*, pages 214–219.
- Aoyama, M., Yabuta, K., Kamimura, T., Inomata, S., Chiba, T., Niwa, T., and Sakata, K. (2014). A resource-oriented services platform for managing software supply chains and its experience. In *2014 IEEE International Conference on Web Services*, pages 598–605.
- Arnould, V. (2018). Using model-driven approach for engineering the system engineering system. In *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, pages 608–614.
- Baumgart, A. and Ellen, C. (2014). A recipe for tool interoperability. In *2014 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, pages 300–308.

- Biehl, M., El-Khoury, J., Loiret, F., and Törngren, M. (2014). On the modeling and generation of service-oriented tool chains. *Software & Systems Modeling*, 13(2):461–480.
- Biehl, M., El-Khoury, J., and Törngren, M. (2012). High-level specification and code generation for service-oriented tool adapters. In *2012 12th International Conference on Computational Science and Its Applications*, pages 35–42.
- Biehl, M., Gu, W., and Loiret, F. (2012). Model-based service discovery and orchestration for oslc services in tool chains. In Brambilla, M., Tokuda, T., and Tolksdorf, R., editors, *Web Engineering*, pages 283–290, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Buffoni, L., Pop, A., and Mengist, A. (2017). Traceability and impact analysis in requirement verification. In *Proceedings of the 8th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, EOOLT '17, pages 95–98, New York, NY, USA. ACM.
- d. Martino, B., Esposito, A., Nacchia, S., and Maisto, S. A. (2016). Towards a uniform semantic representation of business processes, uml artefacts and software assets. In *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)*, pages 543–548.
- El-khoury, J. (2016). Lyo code generator: A model-based code generator for the development of oslc-compliant tool interfaces. *SoftwareX*, 5:190 – 194.
- Gallina, B., Padira, K., and Nyberg, M. (2016). Towards an iso 26262-compliant oslc-based tool chain enabling continuous self-assessment. In *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)*, pages 199–204.
- Gürdür, D., Feljan], A. V., El-khoury, J., Mohalik], S. K., Badrinath, R., Mujumdar], A. P., and Fersman, E. (2018). Knowledge representation of cyber-physical systems for monitoring purpose. *Procedia CIRP*, 72:468 – 473. 51st CIRP Conference on Manufacturing Systems.
- Lednicki, L., Sapienza, G., Johansson, M. E., Seceleanu, T., and Hallmans, D. (2016). Integrating version control in a standardized service-oriented tool chain. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 323–328.
- Lu, J., Wang, J., Chen, D., Wang, J., and Törngren, M. (2018). A service-oriented tool-chain for model-based systems engineering of aero-engines. *IEEE Access*, 6:50443–50458.
- Mustafa, N. and Labiche, Y. (2017). Employing linked data in building a trace links taxonomy. pages 186–198. cited By 2.
- OSLC (2020). Open services for lifecycle collaboration primer web page. Accessed at February 2020.
- Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1 – 18.

- Pikus, Y., Weissenberg, N., Holtkamp, B., and Otto, B. (2019). Semi-automatic ontology-driven development documentation: Generating documents from rdf data and dita templates. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC '19*, pages 2293–2302, New York, NY, USA. ACM.
- Regan, G., Biro, M., Flood, D., and McCaffery, F. (2015). Assessing traceability - practical experiences and lessons learned. *Journal of Software: Evolution and Process*, 27(8):591–601. cited By 6.
- Regan, G., Biro, M., Mc Caffery, F., Mc Daid, K., and Flood, D. (2014). A traceability process assessment model for the medical device domain. In Barafort, B., O'Connor, R. V., Poth, A., and Messnarz, R., editors, *Systems, Software and Services Process Improvement*, pages 206–216, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Tüzün, E., Tekinerdogan, B., Macit, Y., and İnce, K. (2019). Adopting integrated application lifecycle management within a large-scale software company: An action research approach. *Journal of Systems and Software*, 149:63 – 82.
- VanZandt, L. (2015). Enabling rational decision making with provenance-annotated oslc relationships. In *2015 IEEE International Symposium on Systems Engineering (ISSE)*, pages 346–352.
- Wicks, M. N. and Dewar, R. G. (2007). Controversy corner: A new research agenda for tool integration. *J. Syst. Softw.*, 80(9):1569–1585.
- Wieringa, R., Maiden, N., Mead, N., and Rolland, C. (2005). Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.*, 11(1):102–107.
- Zhang, W. and Møller-Pedersen, B. (2013). Establishing tool chains above the service cloud with integration models. In *2013 IEEE 20th International Conference on Web Services*, pages 372–379.
- Zhang, W. and Møller-Pedersen, B. (2014). Modeling of tool integration resources with oslc support. In *2014 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, pages 99–110.