



DOI: 10.5335/rbca.v11i3.8469 Vol. 11, № 3, pp. 39–46 Homepage: seer.upf.br/index.php/rbca/index

ORIGINAL PAPER

Supporting collaborative modelling in UML class diagrams

Raul Antonio Cortiana Neto^{(1),1}, Kleinner Farias^{(1),1}, Vinicius Bischoff^{(1),1}, Lucian José Gonçales^{(1),1}

¹Applied Computing Graduate Program (PPGCA), University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil.

raulac@edu.unisinos.br; kleinnerfarias@unisinos.br; viniciusbischof@edu.unisinos.br; lucianj@edu.unisinos.br

Received: 2018-08-06. Revised: 2019-05-22. Accepted: 2019-09-10.

Abstract

Background. Modeling architectural aspects of the system is an essential activity in software development. In this context, developers work in parallel, and collaborate to define application software models, such as class diagrams. **Problem.** Although many software modeling tools have been proposed, there is a lack of distributed collaboration features. **Solution.** This study proposes C-SAMT, a web tool for collaborative modeling of UML class diagrams. Developers can benefit from using C-SAMT when performing modeling tasks, such as creating domain models in parallel, and collaboratively. **Evaluation.** We recruited 20 industry professionals to perform a qualitative evaluation of the tool through a questionnaire. **Results.** Majority of Industry professionals (85%, 17/20) reported that totally agree that the communication channel of the tool worked properly, they also (80%, 16/20) totally agreed the models generated collaboratively with C-SAMT had small numbers of conflicts, and they also (90%, 18/20) perceived improved productivity gains using the proposed tool. **Conclusion.** C-SAMT provided a collaborative environment, which were positively evaluated by developers, who reported that C-SAMT had a good communication channel between team members.

Keywords: Class Diagrams; Software Artifacts; Collaboration; UML.

Resumo

Background. A modelagem de aspectos arquiteturais do sistema é uma atividade essencial no desenvolvimento de software. Nesse contexto, os desenvolvedores trabalham em paralelo colaborando para definir modelos de software, por exemplo, diagramas de classes. **Problema.** Embora muitas ferramentas de modelagem de software tenham sido propostas, há uma falta de recursos de colaboração distribuída. **Solução.** Este estudo propõe o C-SAMT, uma ferramenta da Web para modelagem colaborativa de diagramas de classes UML. Os desenvolvedores podem se beneficiar do uso do C-SAMT ao executar tarefas de modelagem, como a criação de modelos de domínio em paralelo e de forma colaborativa. **Avaliação.** Foram recrutados 20 profissionais da indústria para realizar uma avaliação qualitativa da ferramenta através de um questionário. **Resultados.** A maioria dos profissionais (85%, 17/20) concordaram totalmente que o canal de comunicação da ferramenta funcionou adequadamente, eles também (80 %, 16/20) concordaram totalmente que os modelos gerados colaborativamente com a C-SAMT resultaram em um pequeno número de conflitos, e eles também (90%, 18/20) perceberam ganhos de produtividade aprimorados usando a ferramenta proposta. **Conclusão.** A C-SAMT forneceu um ambiente colaborativo que foi avaliado positivamente pelos desenvolvedores, os quais relataram que a C-SAMT tinha um bom canal de comunicação entre os membros da equipe.

Palavras-Chave: Diagrama de Classe; Artefatos de Software; Colaboração; UML.

1 Introduction

The process of globalization and constant technological innovation of software-development companies is motivating the demand for collaborative and distributed development (Nunes and Falbo, 2006). The companies have applied efforts to reduce manual activities and improving collaborative activities (Conboy, 2009, Rodríguez et al., 2017) aiming to raise production, and reduce business costs. Thus the use of collaborative systems in software-development processes is an alternative for software companies, due to its well-known benefits de Lange et al. (2016), such as, collaboration between project participants regardless their location, and time zone.

In addition, according to Mistrik et al. (2010) and Lucas et al. (2017) collaborative work improves the decision making as well as the dissemination of new ideas. Nowadays, companies have interest on adopting techniques for the collaborative software modeling due to their interface in which enables developers, and collaborators visualize the occurring changes on realtime in the software project. These tools also provides some benefits such as, activities are synchronized between those involved in the project, productivity gains on development cycle, and consequently, the reduction of project costs (Nunes and Falbo, 2006, Conboy, 2009, Rodríguez et al., 2017). This is because these tools notify changes and conflicts between versions in real time to development teams (Mistrik et al., 2010).

Modelling tools lack some essential functionality to support collaboration on enterprise environments e.g., changing artefacts simultaneously by distributed software development teams. For example, Google docs is a tool that supports this functionality in the context of online documents. When a realtime synchronization of the artefact does not occur, the documents become outdated; otherwise, parallel changes will be usually contradictory, increasing the effort of teams for correction of model inconsistencies (Farias et al., 2014).

According to Xavier et al. (2019) Collaborative software modeling has become a trend due to the significant growth in software industry and academic research. Collaborative Software Modelling for local teams become more feasible because the team can interact informally with each other discussing about the models in development and to solve possible conflicts that may happen. Productivity may decrease with globally distributed teams due to a lack of modeling collaborative tools. In addition, according to Xavier et al. (2019) modeling collaborative tools still dependent of audio chat and other collaborative functionalities like files share.

A review of the proposed tools was executed with regards to some criteria a collaborative tool should implement: (C1) supporting real-time changes to software artefacts simultaneously; (C2) communication between users; (C3) identification of each users in the same section; (C4) location of places the users are acting; (C5) stores the user who did the last update on the content; (C6) Collaboration of nonconnected users; (C7) allow multi-platform use; (C8) support to mobile devices; (C9) integration with other development tools; and (C10) allow the publication of updates notifications. It was found that wellknown approaches adopted to support the development teams on collaborative software modelling do not meet specific criteria essentials to developers. This were also reported on existing literature (Rodríguez et al., 2017, Mistrík et al., 2010, Nicolaescu et al., 2018, de Lange et al., 2016). Furthermore, it also was found a lack of studies exploring this issue (Rodríguez et al., 2017, Nicolaescu et al., 2018), as well as empirical studies about the perception of software developers during collaborative software modelling activities (Farias et al., 2014).

For this, this paper proposes C–SAMT, a web–based tool that supports collaborative modelling of UML class diagrams. Developers can use C–SAMT on modelling tasks, such as creating class diagrams. A qualitative evaluation was performed applying a questionnaire on 20 developers from industry. The main results suggest that C–SAMT communication channel worked properly between developers, it also can increase productivity, as well as produce a number of reduced conflicts on output class diagrams.

This paper is organized as follows. Section 2 presents the theoretical basis, discussing the main concepts used throughout the work. Section 3 describes the tool design and implementation aspects of the proposed tool. Section 5 presents a comparative analysis of software modelling tools. Section 6 presents the main related works. Finally, Section 7 describes the final considerations and future work.

2 Background

This section presents essential concepts to understand this research. Libraries used on the development of the proposed tool are described.

2.1 JointJS and TogetherJS

JointJS is a modern and extremely effective API for creating deferent types of diagrams and graphs. The JavaScript, HTML 5, and JointJS is a set of technologies that enables the renderization of diagrams with static or even fully interactive structures. JointJS is a modern API, supporting the mobile browsers (Client.IO, 2019).

The TogetherJS is an open Source and free JavaScript library. It provides the proper features to turn collaboration efficient on web browsers. Specifically, this API enable users help each other collaboratively in real-time. TogetherJS has a simple, easy, and fast for users (Mozilla, 2019). The next section describes the design and implementation aspects of the proposed collaboration tool prototype.

3 C-SAMT: Collaborative Modeling Tool

This section presents the C-SAMT. Section 3.1 presents the system requirements. Section 3.2 describes the proposed architecture. Section 3.3 discusses the implementation aspects of the proposed tool.

3.1 System Requirements

This session presents the definition of the technical and functional requirements of the proposed tool. The requirements point what the proposed system should do, also represent constraints in the development that were followed during the development of the tool. The requirements were specified to attend the found gaps on related works (Section 6). Table 1 describes the specified requirements of the proposed tool.

3.2 Architecture

This section presents the component diagram of components of the C-SAMT, and describes each component. Fig. 1 shows the components HTML5, JointJS and TogetherJS:

- **HTML**: this component is responsible for managing the tool. The HTML part is responsible for calling the JointJS, and TogetherJS JavaScript libraries;
- TogetherJS: this component is responsible for assigning collaboration, and also manage the main collaborative functionalities:
 - Real-Time Collaboration: component responsible for managing the real-time collaboration capabilities. This component enables users change, edit, and view at same time;
 - Joint Navigation: turns the navigation possible with more than two users within the same domain;
 - Text chat: manages the functionality of text chat;
 - Audio Chat: manages the functionality of audio chat. This functionality is based on RTC Web technology for audio chat between users;
 - User View: enables the identification of each user from a profile. In addition, enables users to upload an image, and change other information of their profile, such as name and color.
 - User Focus: manages the actions of each user's cursor.
- JointJS: component responsible for assigning the diagrams and graphs;
 - **Class Diagram**: allows the editing of the class diagrams through the JointJS API.

3.3 Implementation Aspects

The TogetherJS API assigns collaboration to web pages. When the web-page is created by HTML, it is sufficient that the API is only "called" within the code so that all



Figure 1: C-SAMT Component Diagram.

its features work perfectly. When a dynamic content is presented through JavaScript, TogetherJS requires a function be implemented to other users view the content changes. Fig. 2 presents this function.

grap	h.on('change:position', function(cell) {
	<pre>var _seletor = '[model-id=' + cell.attributes.id + ']',</pre>
	<pre>_posicao = 'translate(' + cell.attributes.position.x + ',' + cell.attributes.position.y + ')';</pre>
	<pre>if (TogetherJS.running) {</pre>
	<pre>TogetherJS.send({type: "changePosition", seletor: _seletor, posicao: _posicao });</pre>
	}
}) ;	
Toge	therJS.hub.on("changePosition", function (msg) {
	<pre>\$(msg.seletor).attr('transform', msg.posicao);</pre>
});	



The tool being discussed in this article fits in this context because the class diagram is created based on the JointJS API. TogetherJS had to be implemented to allow users to view the changes of the class diagram in real-time. In relation to the JointJS API, it allows the creation of diagrams and graphs. The construction of the class diagram of the tool was performed according to the code shown in Fig. 3, and Fig. 4 presents the diagram generated from this code.

Moreover, the proposed tool has as the main objective to support the edition of UML class diagrams collaboratively. The tool also has other features, such as communication via text and audio chat, user location within the page through the cursor view, and identification of each user of the same section. Each of these features will be detailed as follows. Fig. 4 presents the initially generated class diagram in the proposed tool. As previously mentioned users belonging to the same section can change this diagram at same time and on real-time.

• User Identification: allows the user changing the characteristics of his/her profile, such as: rename his/her profile, update the user identification photo,

Table 1: List of tool requirements

Requirement							
ID	Description	-					
Req-01	Real-time changes	Users must be able to make changes to the same diagram and view changes in real time.					
Req-02	Users Cursors with focus	Users must be able to view the cursors and clicks of users in the same section.					
Req-03	Chat communication	A chat must be embedded in the system for the communication of users.					
Req-04	Communication via audio	The tool must be composed of a communication chat via audio.					
Req-05	Viewing users	Users must be able to view who are "logged in" in the same section.					
Req-06	Navigation together	Other users will be directed to the new location every time the main user navigate to a new position on the page.					
Req-07	Class Diagram	The system will allow all "logged in" users in the same section edit the class diagram on real-time.					
Req-08	Start collaboration	The tool has a button on the menu that starts the collaboration.					
Req-09	Fast response time when adding a user to the section	The system cannot take more than 5 seconds to add the user when inviting a user to the section.					
Req-10	Fast response time to start the collaboration menu	The system cannot take more than 5 seconds to start the collaboration.					
Req-11	Fast response Time to load page assets	The class diagram should appear in less than 7 seconds when starting the system page.					

var uml = joint.shapes.uml;

```
var classes = {
mammal: new uml.Interface({
    id: 'interface',
    position: { x:300 , v: 50 },
    size: { width: 240, height: 100 },
    name: 'Mammal',
    attributes: ['dob: Date'],
    methods: ['+ setDateOfBirth(dob: Date):
    Void', '+ getAgeAsDays(): Numeric']
Đ,
person: new uml.Abstract({
    id: 'abstract',
    position: { x:300 , y: 300 },
    size: { width: 240, height: 100 }
    name: 'Person',
    attributes: ['firstName: String',
    'lastName: String'],
    methods: ['+ setName(first: String,
             last: String): Void', '+ getName(): String']
1).
bloodgroup: new uml.Class({
    id: 'bloodgroup',
    position: { x:20 , y: 190 },
    size: { width: 220, height: 100 },
    name: 'BloodGroup',
    attributes: ['bloodGroup: String'],
    methods: ['+ isCompatible(bG: String): Boolean']
H),
address: new uml.Class({
    id: 'address',
    position: { x:630 , y: 190 },
    size: { width: 160, height: 100 },
    name: 'Address',
    attributes: ['houseNumber: Integer','streetName: String'
    methods: []
11.
```

Figure 3: Example of using TogetherJS.hub.on (Mozilla, 2019).

and change the color of their profile. Their profile can be viewed on the same section by other users;

- **Invite a User**: generates a link to invite another user to the collaborative section. When the other user receives this link, he will receive a notification asking if he wants to enter the section;
- Audio Chat: allows connected users to communicate

with each other using audio;

• **Text Chat**: this functionality gives users the option to chat via text.



Figure 4: Class diagram available in C-SAMT tool.

4 C-SAMT Features

This section presents the C–SAMT main functionalities and describes each one of them.

Profile Setup: Fig. 5.(a) presents a functionality that allows users to have name and profile color updated and also the possibility to upload a picture in each avatar.

Profile Checking: Fig. 5.(b) shows a functionality where enables users to check their profile information. For this, they must click on profile icon, and then they can configure their avatar picture, profile color, and also the user name.

Chat feature: this feature enables users to send and receive text messages. Fig. 5.(c) shows users sending text messages through the chat window. Users are also identified by each Profile characteristics.

Invite collaborators: Fig. 5.(d) presents the Invite Collaborator feature. A link will be generated automatically when clicked on second item in collaborative menu. The new collaborator only needs

to access the environment through this link.

Collaboration environment: Fig. 5.(e) shows three users using the proposed tool, C–SAMT. This figure captures the exact moment they locate the cursors of each collaborator. The small circles generated on the screen shows the click of a user.

Fig. 6 presents a class diagram being changed by two users in parallel. The different cursors can identify both users.

Audio Chat: Users can send audio messages through the collaborative menu provided. This feature enables collaborators sending the instructions by recorded audio. They can also communicate instructions through a direct call. This feature was implemented using the TogetherJS API.

5 Evaluation

The evaluation was performed through a questionnaire answered by 20 professionals. This questionnaire aimed at evaluating the functionality of the proposed collaborative modelling tool. This questionnaire was elaborated based on a five point Likert scale (Jamieson, 2004). Specifically, these options are: (1) Strongly Disagree, (2) Partially Disagree; (3) I do not know; (4) Partly Agree; and (5) Totally Agree.

Team Communication using the C-SAMT tool. Fig. 7 presents the obtained results from the questionnaire regarding the communication channel of the tool. All the participants agreed that the communication channel of the tool worked properly. Specifically, the collected results show that 85% (17/20) totally agree, and 15% (3/20) partially agree. These results are interesting because according to Mistrík et al. (2010), a good communication channel between team members can provides some benefits, such as, improved understanding of the solution, and of the elaborated diagram, as well as agility in decision making, and the dissemination of new ideas.

Conflicts between model versions using the C-SAMT tool. Fig. 8 shows the results related to the number of conflicts on output models produced on the tool. Developers reported that models generated collaboratively with the C-SAMT had few conflicts. Specifically, 80% (16/20) totally agree, and 5% (1/20) partially agree that few conflicts were present on generated software models. Finally, only 15% (3/20) of participants were neutral. We attribute this result due to the social interaction of developers during the development of software models. Multiple developers were working and checking emerging inconsistencies at same time. These benefits are analogous to obtained using pair programming (Begel and Nagappan, 2008, Williams and Kessler, 2002).

Productivity. Fig. 9 presents the results related to the perception of the participants regarding productivity gains using the proposed tool. Developers had a positive experience with C-SAMT. They reported that the collaborative development with this tool were productive. Results show that 90% (18/20) of industry professionals totally agree, and 5% (1/20) partially

agree that they perceived had gained productivity on modeling the class diagrams using the C-SAMT tool. In addition, this result reinforces the previous results collected in this evaluation: they reported that the tool enables their communication, and generated output models with a reduced number of conflicts. Consequently, their perception on productivity gains makes sense. In addition, Mistrík et al. (2010) highlights that "collaborative tools provide an improved coordination of activities". This also impacts on productivity gains.

6 Related Works

This section presents a comparative analysis of the related works that focus on supporting collaboration. The tools that are part of this study, for the most part, are used within the software industry.

The Web-Based, Collaborative, Computer-Aided Sequential Control Design Tool (Yen et al., 2003): this tool has as main objective to design a sequential control system for devices and electrical circuits. The software is designed to enable several users work collaboratively on the Internet browser. In this tool only one user can make changes, and others can only view for keeping up with the changes.

Collaborative Project Management Software (Romano et al., 2002): This article aims to describe a prototype of a collaborative tool for project management, called C-PMS (Collaborative Project Management Software). The main qualities of C-PMS tool are the efficiency, and effectiveness in the activities carried out by developers.

Users experiences in collaborative writing using Collaboratus, an Internet-based collaborative work (Lowry et al., 2002): The Collaboratus is a tool for work-groups over the Internet. It also presents its collaborative functionality that supports work-groups.

IBM Rational Software Architect (IBM, 2018): IBM RSA is a robust tool for modelling, and designing software artefacts. IBM developed this tool which is highly recognized in industry.

Borland Together (Borland, 2018): It is a set of software modelling tools that enable the implementation, design, and analysis of software architectures. This tool also provides collaborative functionality to users.

COMA – The Tool for a Collaborative Modelling (Rittgen, 2008): prototype tools for UML modelling that enables collaborative support for group modelling. COMMA tool provides models negotiation, and provides information synthesis through an existent UML modeling tool.

ColD SPA - The Tool for Collaborative Process Model Development (Lee et al., 2000): ColD SPA provides a collaborative modelling support through a web browser. This tool does not enables collaborators to change models in real-time and simultaneously.

This works compares the previously mentioned tools according the following criteria (C):

• C1: supports simultaneous and real-time changes



Figure 5: Invite Collaborators feature.

Table 2: Comparativ	e table	between	the	collaborative	tools
---------------------	---------	---------	-----	---------------	-------

Tools	Criteria									
10018		C2	C3	C4	C5	C6	C7	C8	C9	C10
Web-based and Collab. Soft. Tool (Yen et al., 2003)	0	۲	۲	0	۲	۲	۲	0	0	0
C-PMS (Romano et al., 2002)	0	۲	۲	0	۲	۲	0	0	0	•
Colaborattus (Lowry et al., 2002)	0	٠	۲	0	0	٠	0	0	٠	•
IBM RSA (IBM, 2018)	0	۲	٠	0	۲	۲	0	0	۲	•
Borland Together (Borland, 2018)	0	۲	٠	0	۲	۲	0	0	۲	0
COMMA (Rittgen, 2008)	0	۲	0	0	۲	۲	0	0	0	0
ColD (Lee et al., 2000)	0	۲	۲	0	۲	۲	0	0	۲	0
Tenned, Annlas o Decement employ										

Legend: ●Apply ⊖Does not apply

to the same software artefact;

- C2: communication between users;
- C3: identification of each user;
- **C4**: locate the users action;
- C5: stores the user who did the last update on the content;
- C6: collaboration of non-connected users;
- C7: allow multi-platform use;
- C8: support to mobile devices support;
- **C9**: integration with other tools;
- C10: allow the publication of updates notifications.

Table 2 presents the comparison between produced tools, in relation to the criteria (C) previously mentioned. The general considerations regarding these

results are described bellow:

- Proposed tools do not enable developers changing and evolving artefacts on real-time (C1), they also not enable to locate where another users are acting (C4), they do not support mobile devices (C8), and only one of them (Web-based and Collab. Soft. Tool (Yen et al., 2003)) is a multi-plataform (C7);
- Existing tools strongly supports the communication between users (C2), and enables the collaboration with non-connected developers (C6). In other words, their updates are committed as soon they get logged to the application;
- In exception of COMMA (Rittgen, 2008), majority of them identifies each user on the collaborative



Figure 6: UML Diagram designed with simultaneous collaboration.

Can the tool's communication kernels work?

		(1) Strongly Disagree	0	0%
85%		(2) Partially Disagree	0	0%
		(3) I do not know	0	0%
	15%	(4) Partially agree	3	15%
		(5) I otally agree.	17	85%

Figure 7: The communication channel of the proposed tool.

Does the tool reduce the likelihood of conflicts between versions ?



Figure 8: Amount of conflicts produced in the output models.

environment (C3). In addition, majority of them records the users that make the last modification on the diagram (C5), in exception of the Collaboratus Tool (Lowry et al., 2002);

• Some of them (Collaboratus, IBM RSA, Borland Togheter, and ColD) supports the integration with another development tools (C9), and other tools (C- Do the collaborative features provide greater productivity?



Figure 9: Developers' perception of the productivity with C-SAMT.

PMS, Colaborattus, and IBM RSA) notifies another developer about the recent changes (C10).

Finally, the IBM RSA were the tool that meet the majority of the comparison criteria (C2, C3, C5, C6, C9, and C10), while COMMA were the tool that had less attended the comparison criteria (C2, C5, and C6). The proposed tool was built to attend all the comparisons criteria.

7 Conclusions

Developers demands for collaborative tools for evolving software models in parallel on current development environments. In order to seek gaps and overlapping points on existing tools a comparative analysis were conducted in this work. This analysis compared seven related tools according important criteria, such as support for simultaneous and real-time chances, communication and tool integration. The main result of this analysis was the lack of support for developers change artefacts on real-time.

Therefore, there are limited support for collaboration in UML diagrams despite many modelling tools have been proposed. For this, this study proposed C-SAMT, a web-based tool to support collaborative modelling of UML Class diagrams. In order to evaluate this tool, a qualitative evaluation was performed with 20 developers. They used C-SAMT in their daily work tasks, and evaluated qualitatively the tool through a questionnaire. The usage of the tool enabled the communication between team members, few number of conflicts present on the output model, and team perceived improved productivity.

Future works will focus on testing the effectiveness of the proposed tool. Specifically, experiments to test the precision, and recall in relation to state-of-the art tools.

References

Begel, A. and Nagappan, N. (2008). Pair programming: What's in it for me?, Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '08, ACM, New York, NY, USA, pp. 120–128. http://doi.acm.org/10. 1145/1414004.1414026.

- Borland (2018). Borland together. Available at http:// www.borland.com/Products/RequirementsManagement/ Together.
- Client.IO (2019). Jointjs javascript diagraming library. Avaiilable at http://www.jointjs.com.
- Conboy, K. (2009). Agility from first principles: reconstructing the concept of agility in information systems development, *Information Systems Research* 20(3): 329-354. http://dx.doi.org/10.1287/isre. 1090.0236.
- de Lange, P., Nicolaescu, P., Derntl, M., Jarke, M. and Klamma, R. (2016). Community application editor: Collaborative near real-time modeling and composition of microservice-based web applications., *Modellierung (Workshops)*, pp. 123–128.
- Farias, K., Garcia, A. and Lucena, C. (2014). Effects of stability on model composition effort: an exploratory study, *Software & Systems Modeling* **13**(4): 1473–1494. https://doi.org/10.1007/s10270-012-0308-2.
- IBM (2018). Ibm rational software architect. Available at http://www.ibm.com/developerworks/downloads/r/ architect/index.html.
- Jamieson, S. (2004). Likert scales: how to (ab) use them, *Medical education* 38(12): 1217–1218. https:// doi.org/10.1111/j.1365-2929.2004.02012.x.
- Lee, J. D., Hickey, A. M., Zhang, D., Santanen, E. and Zhou, L. (2000). Cold spa: a tool for collaborative process model development, Annual Hawaii International Conference on System Sciences, 2000, IEEE, pp. 10–15. https://doi.org/10.1109/ HICSS.2000.926588.
- Lowry, P. B., Albrecht, C. C., Lee, J. D. and Nunamaker, J. F. (2002). Users' experiences in collaborative writing using collaboratus, an internetbased collaborative work, System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on, IEEE, pp. 243–252. https://doi.org/ 10.1109/HICSS.2002.993879.
- Lucas, E. M., Oliveira, T. C., Farias, K. and Alencar, P. S. (2017). Collabrdl: A language to coordinate collaborative reuse, *Journal of Systems and Software* pp. 505–527. https://doi.org/10.1016/j.jss.2017. 01.031.
- Mistrík, I., Grundy, J., van der Hoek, A. and Whitehead, J. (2010). Collaborative Software Engineering: Challenges and Prospects, Springer Berlin Heidelberg. https:// doi.org/10.1007/978-3-642-10294-3_19.
- Mozilla (2019). Togetherjs collaboration made easy. Available at http://www.togetherjs.com.
- Nicolaescu, P., Rosenstengel, M., Derntl, M., Klamma, R. and Jarke, M. (2018). Near real-time collaborative modeling for view-based web information systems engineering, *Information Systems* **74**(P1): 23–39. https://doi.org/10.1016/j.is.2017.07.008.

- Nunes, V. B. and Falbo, R. d. A. (2006). Uma ferramenta de gerência de configuração integrada a um ambiente de desenvolvimento de software, V Simpósio Brasileiro de Qualidade de Software, Vila Velha, Brazil.
- Rittgen, P. (2008). Coma: A tool for collaborative modeling, *CAiSE Forum*, Vol. 344, pp. 61–64.
- Rodríguez, P., Haghighatkhah, A., Lwakatare, L. E., Teppola, S., Suomalainen, T., Eskeli, J., Karvonen, T., Kuvaja, P., Verner, J. M. and Oivo, M. (2017). Continuous deployment of software intensive products and services: A systematic mapping study, *Journal of Systems and Software* 123: 263–291. https: //doi.org/10.1016/j.jss.2015.12.015.
- Romano, N. C., Chen, F. and Nunamaker, J. F. (2002). Collaborative project management software, System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on, IEEE, pp. 233–242. https://doi.org/10.1109/HICSS.2002.993878.
- Williams, L. and Kessler, R. (2002). *Pair programming illuminated*, Addison–Wesley Longman Publishing Co., Inc.
- Xavier, M. S. d. L., Farias, K., Barbosa, J., Gonçales, L. and Bishoff, V. (2019). Umlcollab: A hybrid approach for collaborative modeling of uml models, *Proceedings of the XV Brazilian Symposium on Information Systems*, SBSI'19, ACM, New York, NY, USA, pp. 30:1–30:8. http://doi.acm.org/10.1145/3330204.3330239.
- Yen, C., Li, W.-J. and Lin, J.-C. (2003). A web-based, collaborative, computer-aided sequential control design tool, *IEEE control systems* 23(2): 14–19. https: //doi.org/10.1109/MCS.2003.1188768.