2017 IEEE 25th International Requirements Engineering Conference

Reusable Goal Models

Mustafa Berk Duran
Department of Electrical and Computer Engineering, McGill University
McConnell Engineering Building, 3480 University Street, Montréal, Québec H3A0E9, Canada
berk.duran@mail.mcgill.ca

***Abstract*—Goal models help elicit, specify, analyze, and validate requirements as they capture hierarchical representations of system requirements, possible solutions, stakeholder objectives, and their relationships. In the early requirements phase, goal models aid requirements engineers in understanding the goals of stakeholders and exploring solution alternatives based on their impact on these goals. Despite these strengths of goal modeling, reuse of goal models has received limited attention in the goal modeling community for various reasons. We argue that the requirements that need to be fulfilled at least for the reuse of goal models are as follows: (i) ensure that trade-off reasoning via goal model evaluation is possible through the reuse hierarchy, (ii) provide the means to delay decisions to a later point in the reuse hierarchy when more complete information is available, while still allowing the reusable artifact to be analyzed, (iii) take into account constraints imposed by other modeling notations when evaluating reusable goal models, (iv) allow context dependent information to be modeled so that the goal model can be used and analyzed in various application contexts, and (v) last but not the least, provide a well-defined interface for reuse. This research abstract argues that existing goal modeling approaches do not fully address these five required capabilities for reusability, and reports on both completed and planned research work.**

***Index Terms*—Goal model; reuse; context; requirements reuse; model-driven requirements engineering.**

I. Motivation

Goal models capture hierarchical representations of stakeholder objectives, system requirements, and possible solutions in addition to the relationships between them. In the early requirements phase, goal modeling helps requirements engineers understand stakeholder goals and explore alternative solutions and their potential impacts on these goals.

Goal modeling is important for abstracting away the unnecessary information (e.g., implementation details) to help stakeholders understand the requirements and communicate their needs and expectations. Various goal modeling languages exist in literature. Key examples include KAOS [1], the NFR framework [2], i* [3], Tropos [4], and the Goal-oriented Requirement Language (GRL) [5]. Even though differences exist among these languages in terms of the concepts they include, there are also a lot of similarities. In a broad sense, the concept of a *goal* exists in all languages. While the NFR framework only uses *softgoals* (i.e., goals for which there is no objective measure of satisfaction), i* and GRL further classify intentional elements into *goals, softgoals*, *tasks*, and *resources*. AND and OR *decompositions* among goals is possible in all languages. All languages except KAOS cover *contribution links*, although the means to express contributions differ.

Software reuse is powerful due to the potential benefits it provides such as increased quality, productivity, and reliability with faster time-to-market and lower cost. Reuse is defined as the process of creating software systems from existing software artifacts instead of creating them from scratch [6] and as the use of existing artifacts in a new context [7].

In the context of reuse, goal models may be used either as *standalone artifacts* to help better understand stakeholder objectives and compare potential solutions in the early requirements phase or as *connected artifacts* to describe and compare reusable artifacts.

- *Standalone artifacts* are used to describe a problem situation and can be reused in different contexts by themselves while maintaining the provided analysis capabilities to better understand the needs of different stakeholders and considered candidate solutions.
- *Connected artifacts*, in collaboration with other models, are used to describe reusable artifacts to express the impacts of the reusable artifact on high-level goals and qualities. In this situation, goal models can answer why a reusable artifact should be chosen over another candidate artifact as they allow alternative candidate solutions to be evaluated with trade-off analysis, i.e., goal models can guide a modeler through design choices among alternative reusable artifacts and solutions.

A comprehensive goal modeling language should support reuse of both standalone and connected goal models.

II. Statement of the Problem

Even though goal modeling can benefit from reuse to increase quality, productivity, and reliability, to reduce time-to-market, and to lower cost, goal model reuse has received limited attention in the goal modeling community. Hence, the main research question to answer is:

How can goal models be made reusable?

Answering this question involves determining the goal model concepts and algorithms needed to reuse a goal model and benefit from the strengths of goal modeling. As one of the initial and stepping stone contributions of this research, the requirements that need to be fulfilled at least for the reuse of goal models are identified and introduced in the remainder of this section. The requirements were identified based on experience with work on requirements reuse [8], Reusable Aspect Models (RAM) [9], and Concern-Orientation [10]. The requirements are needed for these application contexts, but we

2332-6441/17 $31.00 © 2017 IEEE
DOI 10.1109/RE.2017.34

cannot guarantee that other requirements may exist for other contexts. The requirements are used both in the assessment of the current state of goal modeling literature and as guidelines for a comprehensive solution for reusable goal models.

R.1. *A reusable goal modeling approach shall ensure that trade-off reasoning via goal model evaluation is possible through the reuse hierarchy.*

Reusable artifacts form a reuse hierarchy where lower-level reusable artifacts are composed together to build larger, higher-level artifacts. Analysis and validation that is available for a single model at a single level must be ensured to be available through the entire reuse hierarchy. For example, in the context of GRL, analysis in the form of trade-off reasoning is performed by a propagation-based evaluation mechanism [11] (i.e., the evaluation mechanism takes the satisfaction value of a child element and propagates it up to the parent according to the type and value of the link between them). To support the trade-off reasoning through the reuse hierarchy, e.g., the way contribution values are assigned and satisfaction values are interpreted by different modelers working on different levels in the reuse hierarchy needs to be investigated. At the same time, the complexity of the evaluation must also be managed throughout a growing reuse hierarchy.

R.2. *A reusable goal modeling approach shall provide the means to delay decisions to a later point in the reuse hierarchy when more complete information is available, while still allowing the reusable artifact to be analyzed.*

The exact requirements for a reusable artifact are typically unknown as the context in which the artifact is going to be used is unknown in advance [12]. In general, two reuse scenarios are applicable to goal models. In the first scenario, the modeler has all the information about the system, the context in which the reusable artifact is going to operate, and its complete requirements. If that is the case, the modeler can pick and reuse the most appropriate reusable artifact. In the second scenario, the modeler lacks some of the information, however, it is still important to allow them to proceed with the limited information at hand. This requires the modeling approach to support the partial reuse of a reusable artifact and leaving some decisions open to be made at the later stages of development. Essentially, a goal model describes a set of choices that are evaluated with respect to objectives of stakeholders. The first reuse scenario allows decisions to be made for each choice, whereas, the second reuse scenario leaves some choices to be made when more information is available higher up in the reuse hierarchy.

R.3. *A reusable goal modeling approach shall take into account constraints imposed by other modeling notations when evaluating reusable goal models.*

Goal models cannot always be considered in isolation as additional constraints on various goal model elements may be expressed in other modeling notations that are used in collaboration with goal models. Typically, external constraints exist for tasks, i.e., the candidate solutions. E.g., a workflow model may express causal relationships between tasks at runtime. Similarly, a feature model may describe which tasks can be selected together to build a functionality or a system [13]. If these constraints are ignored during evaluation, goal model analysis may result in an inaccurate evaluation. Trying to model all of the aforementioned information in the goal model results in an overly complex goal modeling notation that is difficult to analyze and understand. Consequently, different requirements models are used in collaboration with goal models to benefit the modeler as each notation has their own strengths. It is also important to note that such additional constraints make the evaluation of goal models in reuse hierarchies (see R.1) much more challenging as the analysis requires backtracking or other forms of non-linear exploration of the goal model rather than employing the usual straightforward propagation algorithms [13].

R.4. *A reusable goal modeling approach shall allow context dependent information to be modeled so that the goal model can be used and analyzed in various application contexts.*

Designing a generic software artifact has its challenges. Although it requires considering a wide range of scenarios in which the artifact could be reused, it often results in the use of placeholders instead of concrete model elements to represent unknown application-specific information to keep the reusable artifact generic. Hence, it is important to allow elements of a goal model to adjust to different circumstances. For example, depending on the application context, a candidate solution may contribute to a high-level goal either positively or negatively or with varying degrees of impact.

R.5. *A reusable goal modeling approach shall provide a well-defined interface for reuse.*

In order to increase encapsulation and modularity, and strengthen the reuse hierarchy, many successful reusable artifacts have clearly defined interfaces [14][15] and goal models are no exception, i.e., a reuse interface must also be defined for goal models. This interface would allow hiding lower-level goal model elements behind the interface of the current artifact from the next higher-level artifact that wants to reuse this current artifact. A clearly defined reuse interface allows modelers to differentiate model elements of the reused lower-level artifact from those of the reusing artifact both at the model and the metamodel levels.

To summarize, an approach that can answer the main research question must address these five requirements. While we cannot claim that these requirements are sufficient for all reuse contexts, they are nevertheless necessary to make goal models reusable from our experience of applying goal models in a reuse context. The hypotheses for this research are built upon these requirements and explained in the next section. A reusable goal modeling approach should support the reuse of both standalone and connected goal models. For the reuse of a standalone goal model, the modeler realizes that the current stakeholders, their objectives, and potential solutions are similar to previous ones, and hence the standalone goal model can be reused during the early requirements phase. In this case, all requirements apply except for R.3, i.e., external constraints do not need to be taken into account as the goal model is used in isolation. For the reuse of a connected goal model, the modeler realizes that some properties of reusable artifacts apply to the current problem situation. These reusable artifacts do not necessarily have to be requirements-level artifacts but can also describe system design or implementation. The modeler then uses the connected goal models to decide which reusable artifact to reuse. In this case, all five requirements apply, because the goal model is used in connection with other models which may impose additional constraints on the goal model.

III. Hypotheses and Methodology

Given the main research question and the five requirements identified in Section II, the hypotheses for this research are stated as follows:

H.1) Current goal modeling technologies do not fully satisfy the five requirements R.1 to R.5: analysis and validation capability through the reuse hierarchy, enabling the delaying of decisions, handling constraints imposed by other modeling notations, means to model and analyze context dependent information, and a defined reuse interface.

H.2) It is feasible to determine the required goal model concepts and tool-supported algorithms for goal model reuse in GRL that satisfy the five requirements R.1 to R.5: providing trade-off reasoning via goal model evaluation through reuse hierarchies in the presence of delayed decision-making while taking constraints imposed by other models into account, allowing context dependent information to be modeled and analyzed, and providing an interface for reuse.

For the assessment of the first hypothesis (H.1), a comparative analysis of the proposed approach with existing goal model reuse techniques is performed with respect to their ability to satisfy the five requirements. For this analysis, a two-part systematic literature review investigating *goal model reuse* and *context and goal models* is conducted.

To check whether the second hypothesis (H.2) holds, goal model concepts required for reuse are formulated for GRL and evidences of feasibility and scalability are gathered via the implementation of tool support and necessary algorithms that support the five requirements. Proof-of-concept implementations are provided for the TouchCORE [16] and jUCMNav [17] tools. Furthermore, with the tool support provided through the proof-of-concept implementations, an example case study is modeled using the proposed approach to provide further evidence of feasibility and scalability. GRL is chosen based on our experience with it, but the results of this work can still be applied to other languages given the similarity of major goal modeling languages.

IV. State of the Art

For the investigation of the existing body of knowledge on goal model reuse and an assessment of the existing solutions, a two-part systematic literature review is conducted. The systematic literature review aims to answer (i) to what extent do goal modeling approaches satisfy the five requirements R.1 to R.5, (ii) which goal modeling concepts are used or introduced to represent context in a goal modeling notation, (iii) what other requirements models, if any, are used in collaboration with reusable goal models, (iv) for which reusable goal modeling approaches does tool support exist, and (v) which research themes need to be investigated further to realize contextual and reusable goal models and fulfill all identified requirements (R.1 to R.5).

The two-part systematic literature review covers both reuse and context in goal models with each step of the review documented and all results tracked. Out of the initial publication list of more than 2,400 unique hits, retrieved from seven major academic search engines, a final list of 68 publications and 11 comparisons/assessments of goal modeling approaches are analyzed in more detail. Due to space restrictions, only a few example publications are used for the explanation of the results of the systematic literature review in this section.

Although several goal modeling approaches partially provide support for analysis or validation (R.1), (e.g., GRL Catalogues [18], i* Modules [19], AoGRL [20], Contextual Goal Models [21], URN [22], and i*-Context [23]), the absence of a reuse hierarchy (reuse is done via copy and paste) results in one big monolithic goal model which is complex to evaluate, making it even more difficult in the presence of additional constraints from other modeling notations.

A complete mechanism that explicitly specifies the means to delay decisions in the reuse hierarchy (R.2) is not provided by any of the existing approaches. This requirement is partially satisfied by most of the approaches since they use qualitative reasoning to cope with uncertainty (e.g., GRL Catalogues [18] and i* Modules [19]) or allow use of data obtained at runtime for re-evaluation of goal satisfactions or system reconfiguration (e.g., Contextual Goal Models [21] and Extended Tropos Goal Model [24]). However, they do not address reuse hierarchies and require knowledge of the whole system.

Existing techniques provide very little support to take additional constraints imposed on goal models by other modeling formalisms into account. Even those techniques that model additional information with different formalisms connected to goal models (e.g., Aspects with Tropos [25] allows modelers to express workflow models with Use Case Map (UCM) notation [5] that are connected with goal models) do not leverage the constraints imposed by those complementary modeling languages in the evaluation. This points to a need for a general framework for handling constraints on goal models that can handle various modeling formalisms.

The majority of existing approaches allow context dependent information to be modeled (R.4) using a wide variety of modeling constructs to represent context. E.g., Contextual

Goal Models [21] uses or-decomposition, means-end, actor dependency, root goals, and-decomposed goals, and contribution to softgoals as variation points in their goal models. Context-enriched Goal Models [26] identifies a set of contextual tags and is concerned with the visibility of model elements, whereas URN [22] uses key performance indicators (KPI) and beliefs for context. However, it is not clear which of the modeling constructs are more fitting to express context in reuse hierarchies. For this purpose, it is necessary to assess each of these modeling constructs in terms of how it could support reuse hierarchies, a reuse interface, and external constraints imposed by other models. Based on the results of this assessment, it may be necessary to add new concepts to existing goal modeling approaches and develop new evaluation mechanisms to support contextual and reusable goal models.

Even though some of the existing approaches provide the necessary module boundaries (e.g., pointcuts in Aspect KAOS [27], patterns in NFR [28], and templates in GoPF approach [29]) to provide an interface for reuse (R.5), most of the existing goal model reuse approaches do not fully satisfy R.5 as modularization and hence the reuse hierarchy is lost upon composing the models. At that point, it is not possible for the user of the reusable artifact to distinguish model elements at different levels. On the other hand, goal templates used in the GoPF approach [29] and aspect markers used in AoGRL [20] maintain clear model boundaries after they are reused and hence these approaches satisfy R.5. Based on the literature review, the sparse support for reuse boundaries to satisfy R.5 indicates that the goal modeling community is yet to converge on a generally accepted definition of what constitutes the reuse interface of a goal model.

Various approaches in the literature make use of different types of supplementary models in addition to goal models. Behavioral models in the form of workflow models (i.e., UCMs and process models), sequence diagrams (e.g., Patterns in NFR [28]), and state diagrams make up the majority whereas several structural modeling notations such as role models, organization models (e.g., Task Knowledge Patterns [30]), use case models, Entity Relation Diagrams (ERD), and context models (e.g., Adaptation Goal Model [31]) are also used. Furthermore, feature models are also used in collaboration with goal models to model variability.

To summarize, results of the systematic literature review indicate that *even though various goal modeling approaches satisfy individual requirements, there exists a need for a holistic goal modeling environment that supports all five requirements* stated in Section II. Such an approach would be a novel contribution and an advance in the landscape of goal modeling research.

V. Technical Challenges

One of the major strengths of goal modeling is the capability to perform trade-off analysis. E.g., in GRL, a *strategy* depicts a candidate solution via assigning initial satisfaction values to a set of elements in the model. A satisfaction value expresses the degree with which a softgoal or goal is satisfied, a task is performed, or a resource is available. Unless a satisfaction value of a node is assigned by the modeler with a strategy, the satisfaction value of a node is calculated with the help of propagation-based evaluation [2][11], using the incoming links (type of the link and the contribution value if it is a contribution link) and the satisfaction values of the children nodes. Therefore, modelers can compare several strategies with each other using the resultant satisfaction values of high-level goal model elements. This facilitates the decision making process among alternative solutions or prioritization of the requirements and/or stakeholder goals.

Contribution values and how they are assigned are an integral part of goal model evaluation. Three types of contribution values are known for main goal modeling approaches such as the NFR Framework, GRL, and i*: qualitative, quantitative, and real-life values. We argue that a reusable goal model must use either real-life values (if they are measurable/available) or *relative values* for contribution links [32]. The designer of the reusable artifact assigns relative contribution values with the sole intention of differentiating competing solutions locally within the reusable artifact. This approach eliminates the need for a global agreement among modelers on what contribution values mean as is needed for non-relative contribution values. Global agreements are unlikely to exist for vague goals such as user convenience or security. Relative contribution values lead to relative satisfaction values, which need to be normalized for composability and reusability reasons. Without normalization, goal model elements with relative satisfaction values cannot be composed together (i.e., one would compare apples and oranges). Normalization ensures that each satisfaction value is in a given range (i.e., $[0, 100]$), and hence composable. For this purpose, it is necessary to determine an element's actual minimum and maximum satisfaction values, to be normalized to 0 and 100, respectively. The need to normalize these actual minimum and maximum satisfaction values in the presence of additional constraints from other modeling notations requires backtracking or other forms of non-linear exploration of the goal model, which introduces computational complexity that needs to be handled by the normalization algorithm.

A crucial technical challenge of this research is to manage the computational complexity introduced by the proposed capabilities to address the five requirements from Section II. While initial work has been done to enable the evaluation of goal models with relative contribution values in a reuse hierarchy (R.1) taking external constraints into account (R.3) [13][32][33][34], delayed decisions (R.2) and contextual information (R.4) still need to be investigated.

VI. Contributions

The first contribution **(C.1)** of this doctoral research is the identification of the five requirements that need to be fulfilled at least for the reuse of goal models. To check whether the stated hypotheses of this research hold and to address the stated technical challenges, the results of the research activities, described as part of the research methodology in Section III, are to date:

C.2) The two-part systematic literature review to assess the first hypothesis is focused on *goal model reuse* and *context and goal models* in the existing body of work. Findings of this literature review have shown that none of the existing goal model reuse approaches completely addresses the five requirements and that there is a need for a comprehensive goal modeling approach that supports all five requirements.

C.3) For the evaluation of reusable goal models in the absence of real-life measurements, a new approach to assign and interpret contribution values was defined. For this purpose, the concepts of relative contribution values and normalization with scaling factors and offsets were introduced [32]. The GRL metamodel was extended to cover these new concepts [33], which are prerequisites to satisfy R.1.

C.4) The evaluation mechanism was updated to handle external constraints imposed by feature models [32] and workflow models [13]. The GRL metamodel was extended to support handling of workflow constraints by identifying run-time and design-time goals. The updated evaluation mechanism allows for more accurate trade-off analysis in reusable goal models as required by R.3.

C.5) The GRL metamodel was extended to include constructs for reuse links and feature impact elements to provide an interface for reuse (R.5) and allow the goal model evaluation to span the entire reuse hierarchy [33][34] as required by R.1. With the proposed changes to goal model evaluation, our approach deals with the problem of large monolithic models resulting from, e.g., copy/paste-based reuse that are hard to maintain and analyze as the reuse hierarchy grows. Our approach introduces an iterative evaluation of goal models at different levels of the reuse hierarchy while keeping their modular structure. Additionally, our approach allows elements of the reused goal model to be differentiated from elements of the higher-level reusing goal model (both at the model and the metamodel level), resulting in a clear goal model reuse interface that hides elements of lower-level models and provides proper encapsulation.

C.6) Proof-of-concept implementations [13][32][33][34] exist in the TouchCORE [16] and jUCMNav [17] tools to provide evidence for the feasibility and scalability of the proposed improvements. As a result, support is available for reusable goal modeling activities identified by the key requirements.

The research activities planned for the remainder of this research and their expected contributions are:

FC.1) To support the delaying of decisions in the goal model reuse hierarchies, partial reuse of a reusable artifact while retaining trade-off analysis capabilities is to be provided with the help of a *range evaluation* mechanism. The idea is to evaluate the possible minimum and maximum satisfaction ranges for intentional elements considering both the decisions made and the ones that will be made at a later point in the reuse hierarchy (i.e., they are currently unknown). Having such an evaluation mechanism built on top of our current novel evaluation (i.e., taking additional constraints into account and spanning the whole reuse hierarchy) will address not only R.2, but also R.4 to a certain extent, as the information being introduced to the model at later stages may be contextual.

FC.2) To further ensure the compliance of our goal modeling approach with R.4, the GRL metamodel is to be extended with the identified concepts from the literature review to consequently support modeling context dependent information and the development of tool support for this purpose. Furthermore, the influence of representing contextual information in the goal model on the current novel evaluation mechanism needs to be investigated.

FC.3) The evaluation of reusable goal models is to be extended to support the evaluation of stakeholders and decomposition and dependency links in addition to already supported contribution links.

FC.4) In addition to scalability tests with large, artificial goal models, an example case study is to be conducted with the completed approach and implemented tool support to provide further evidence of feasibility to validate the second hypothesis.

VII. Conclusion

A. Research Summary

In the early requirements phase, goal models are used to capture hierarchical representations of stakeholder objectives, system qualities, potential solutions, and their relationships. With goal models, requirements engineers can understand the goals of stakeholders and explore alternative solutions with their impacts on goals. Reuse of goal models, however, has received limited attention. This work investigates the requirements for goal model reuse and proposes an approach that addresses five identified requirements: providing trade-off reasoning via goal model evaluation through reuse hierarchies, providing the means for delaying decisions to a later point when more information is available, taking constraints imposed by other models into account, allowing context dependent information to be modeled and analyzed, and providing a reuse interface.

The hypotheses of this research are defined as: *(a)* current goal modeling technologies do not fully satisfy the five requirements, and *(b)* it is feasible to determine the required goal model concepts and tool-supported algorithms for goal model reuse that satisfy the five requirements. This research is assessed via a systematic literature review to evaluate the findings qualitatively and a proof-of-concept implementation of proposed improvements with an example case study to evaluate the feasibility of the proposed approach quantitatively.

B. Progress and Publications

All of the author's publications to date that are related to this doctoral research are listed in this subsection. At this point, the systematic literature survey to analyze whether the existing goal model reuse approaches address the five requirements is completed and the findings indicate the originality of our work and the novelty of a goal model reuse approach that satisfies all five requirements.

To implement this novel approach, three of the five requirements for goal model reuse were investigated and the findings were published [13][32][33][34]. Relative contribution values were introduced and algorithms formulated for the normalization of relative goal model satisfactions (R.1) while taking into account the external constraints imposed by feature models [32] and workflow models [13] (R.3). Their feasibility and scalability were analyzed with proof-of-concept implementations in the modeling tools jUCMNav [17] and TouchCORE [16]. The GRL metamodel was extended with the constructs that allow the goal model evaluation to be conducted in reuse hierarchies (R.1) and help define a clear reuse interface (R.5) for reusable goal models [33][34].

C. Future Work

In future work, the remaining two requirements are to be addressed (R.2 and R.4). A range evaluation mechanism for trade-off analysis in goal model reuse hierarchies is being built on top of our current novel evaluation mechanism that takes additional constraints into account and spans the whole reuse hierarchy. Additional concepts will be introduced to the GRL metamodel to support modeling context dependent information. Furthermore, the influence of representing context dependent information on the novel evaluation mechanism will be investigated (e.g., in terms of scalability).

Our current approach covers the handling of contribution links in goal model evaluation. This work is to be extended to include decomposition and dependency links, in addition to the evaluation of stakeholders.

Last but not the least, for the evaluation of the feasibility and scalability of the proposed approach, an application of the reuse approach to an example case study is to be conducted with the tool support to be implemented in the TouchCORE [16] and jUCMNav [17] tools.

References

[1] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed requirements acquisition," *Sci. Comput. Program.*, vol. 20, no. 1-2, pp. 3–50, Apr. 1993.

[2] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, ser. International Series in Software Engineering. Springer, October 1999, vol. 5.

[3] E. S.-K. Yu, "Modelling strategic relationships for process reengineering," Ph.D. dissertation, Toronto, Ont., Canada, Canada, 1996, uMI Order No. GAXNN-02887 (Canadian dissertation).

[4] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An agent-oriented software development methodology," *Auton. Agents & Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, 2004.

[5] International Telecommunication Union (ITU-T), "Recommendation Z.151 (10/12): User Requirements Notation (URN) - Language Definition," approved October 2012.

[6] C. W. Krueger, "Software reuse," *ACM Comput. Surv.*, vol. 24, no. 2, pp. 131–183, Jun. 1992.

[7] NATO, "Standard for the Development of Reusable Software Components," GTE Gov-t Systems, Chantilly, VA, USA, Tech. Rep., 1994.

[8] G. Mussbacher and J. Kienzle, "A vision for generic concern-oriented requirements reuse RE@ 21," in *Requirements Engineering Conference (RE), 2013 21st IEEE International*. IEEE, 2013, pp. 238–249.

[9] J. Kienzle, W. A. Abed, and J. Klein, "Aspect-oriented multi-view modeling," in *AOSD 2009, Virginia, USA*, 2009, pp. 87–98.

[10] O. Alam, J. Kienzle, and G. Mussbacher, "Concern-oriented software design," in *MODELS 2013, Miami, FL, USA*, 2013, pp. 604–621.

[11] D. Amyot, S. Ghanavati, J. Horkoff, G. Mussbacher, L. Peyton, and E. Yu, "Evaluating goal models within the goal-oriented requirement language," *Int. J. Intell. Syst.*, vol. 25, no. 8, pp. 841–877, Aug. 2010.

[12] J. Kienzle, G. Mussbacher, P. Collet, and O. Alam, "Delaying decisions in variable concern hierarchies," in *2016 ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences*, ser. GPCE 2016. New York, NY, USA: ACM, 2016, pp. 93–103.

[13] M. B. Duran and G. Mussbacher, "Investigation of feature run-time conflicts on goal model-based reuse," *Information Systems Frontiers*, vol. 18, no. 5, pp. 855–875, 2016.

[14] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005.

[15] J. Kienzle, G. Mussbacher, O. Alam, M. Schöttle, N. Belloir, P. Collet, B. Combemale, J. DeAntoni, J. Klein, and B. Rumpe, "VCU: the three dimensions of reuse," in *Software Reuse: Bridging with Social-Awareness - 15th Intl. Conf., ICSR 2016, Limassol, Cyprus, June 5-7, 2016, Proceedings*, 2016, pp. 122–137.

[16] TouchCORE tool, "(version 6.4.0)," 2017, http://touchcore.cs.mcgill.ca.

[17] jUCMNav tool, "(version 7.0.0)," 2017, http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/JUCMNavRelease700.

[18] J.-F. Roy, "Requirement engineering with URN: Integrating goals and scenarios," Master's thesis, University of Ottawa, Canada, 2007.

[19] D. Colomer and J. Franch Gutiérrez, "i* modules: a jUCMNav implementation," in *iStar 5th International i* workshop: 29-30th August, 2011, Trento, Italy*. CEUR Workshop Proceedings, 2011, pp. 178–180.

[20] G. Mussbacher, D. Amyot, J. Araújo, A. Moreira, and M. Weiss, "Visualizing Aspect-Oriented Goal Models with AoGRL," in *Requirements Engineering Visualization, 2007. REV 2007. Second International Workshop on*. IEEE, Oct. 2007, pp. 1–1.

[21] R. Ali, R. Chitchyan, and P. Giorgini, "Context for goal-level product line derivation," in *3rd International Workshop on Dynamic Software Product Lines (DSPL09)*, 2009.

[22] D. Amyot and G. Mussbacher, "Development of telecommunications standards and services with the user requirements notation," in *Workshop on ITU System Design Languages 2008*, 2008, pp. 15–16.

[23] Z. Xu, H. Zhao, and L. Liu, "User's Requirements Driven Services Adaptation and Evolution," in *COMPSACW*. IEEE, 2012, pp. 13–19.

[24] Y. Lei, K. Ben, and Z. He, "A Framework for Self-Adaptive Software Based on Extended Tropos Goal Model," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015 7th International Conference on*, vol. 2. IEEE, 2015, pp. 533–536.

[25] J. Castro, M. Kolp, L. Liu, and A. Perini, "Dealing with Complexity Using Conceptual Models Based on Tropos," in *Conceptual Modeling: Foundations and Applications*, ser. Lecture Notes in Computer Science, A. T. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. Yu, Eds. Springer, 2009, no. 5600, pp. 335–362, dOI: 10.1007/978-3-642-02463-4_18.

[26] A. Lapouchnian and J. Mylopoulos, "Modeling Domain Variability in Requirements Engineering with Contexts," in *SpringerLink*. Springer, 2009, pp. 115–130, dOI: 10.1007/978-3-642-04840-1_11.

[27] A. Gil and J. Araujo, "AspectKAOS: integrating early-aspects into KAOS," in *15th workshop on Early aspects*. ACM, 2009, pp. 31–36.

[28] L. Chung and S. Supakkul, "Capturing and reusing functional and non-functional requirements knowledge: a goal-object pattern approach," in *IRI 2006*. IEEE, 2006, pp. 539–544.

[29] S. A. Behnam, D. Amyot, G. Mussbacher, E. Braun, N. Cartwright, and M. Saucier, "Using the Goal-oriented pattern family framework for modelling outcome-based regulations," in *Requirements Patterns (RePa), 2012 IEEE 2nd Intl. Workshop on*. IEEE, 2012, pp. 35–40.

[30] W. Cheah, L. Sterling, and K. Taveter, "Task Knowledge Patterns Reuse in Multi-Agent Systems Development," in *Principles and Practice of Multi-Agent Systems*, ser. LNCS. Springer Berlin Heidelberg, Nov. 2010, no. 7057, pp. 459–474, dOI: 10.1007/978-3-642-25920-3_33.

[31] Z. Yang and Z. Jin, "Modeling and specifying parametric adaptation mechanism for self-adaptive systems," in *Requirements Engineering*. Springer, 2014, pp. 105–119.

[32] M. B. Duran, G. Mussbacher, N. Thimmegowda, and J. Kienzle, "On the reuse of goal models." in *SDL Forum*, ser. Lect. N. in Comp. Sc., vol. 9369. Springer, 2015, pp. 141–158.

[33] M. B. Duran, A. N. Pina, and G. Mussbacher, "Evaluation of reusable concern-oriented goal models," in *2015 IEEE International Model-Driven Requirements Engineering Workshop, MoDRE, Ottawa, ON, Canada, August 24, 2015*, 2015, pp. 53–62.

[34] R. Alexandre, C. Camillieri, M. B. Duran, A. N. Pina, M. Schöttle, J. Kienzle, and G. Mussbacher, "Support for Evaluation of Impact Models in Reuse Hierarchies with jUCMNav and TouchCORE," in *MoDELS 2015 Demo and Poster Session, Ottawa, Canada*. CEUR-WS.org, 2015.