Information and Software Technology 110 (2019) 156–173

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Reusability in goal modeling: A systematic literature review

Mustafa Berk Duran*, Gunter Mussbacher

Department of Electrical and Computer Engineering, McGill University, Montréal, Canada

ARTICLE INFO

Keywords:
Goal model
Reuse
Context
Requirements reuse
Model-driven requirements engineering
Systematic literature review

ABSTRACT

Context: Goal modeling is an important instrument for the elicitation, specification, analysis, and validation of early requirements. Goal models capture hierarchical representations of stakeholder objectives, requirements, possible solutions, and their relationships to help requirements engineers understand stakeholder goals and explore solutions based on their impact on these goals. To reuse a goal model and benefit from the strengths of goal modeling, we argue that it is necessary (i) to make sure that analysis and validation of goal models is possible through reuse hierarchies, (ii) to provide the means to delay decision making to a later point in the reuse hierarchy, (iii) to take constraints imposed by other modeling notations into account during analysis, (iv) to allow context dependent information to be modeled so that the goal model can be used in various reuse contexts, and (v) to provide an interface for reuse.

Objective: In this two-part systematic literature review, we (i) evaluate how well existing goal modeling approaches support reusability with our five desired characteristics of contextual and reusable goal models, (ii) categorize these approaches based on language constructs for context modeling and connection to other modeling formalisms, and then (iii) draw our conclusions on future research themes.

Method: Following guidelines by Kitchenham, the review is conducted on seven major academic search engines. Research questions, inclusion criteria, and categorization criteria are specified, and threats to validity are discussed. A final list of 146 publications and 34 comparisons/assessments of goal modeling approaches is discussed in more detail.

Results: Five major research themes are derived to realize reusable goal models with context dependent information.

Conclusion: The results indicate that existing goal modeling approaches do not fully address the required capabilities for reusability in different contexts and that further research is needed to fill this gap in the landscape of goal modeling approaches.

1. Introduction

In requirements engineering, goal models have traditionally been used to capture requirements of the system with underlying stakeholder objectives and their relationships. Goal modeling helps requirements engineers to better understand the goals of stakeholders and explore alternative solutions and their potential impacts on the achievement of these goals during requirements engineering activities. The analysis capability provided by goal modeling can be used to evaluate the compliance of alternative solutions with the objectives of different stakeholders or to assess the achievement of the goals. In other words, goal model analysis helps capture the trade-offs in satisfaction of stakeholder objectives given a proposed solution.

Software reuse is powerful due to its potential benefits 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 [1] and as the use of existing artifacts in a new context [2]. Even though goal models are typically used in the early requirements phase to help better understand stakeholder objectives and compare potential solutions, they can also be used to describe reusable artifacts [3]. In this sense, goal models may express the impacts of the reusable artifact on high-level stakeholder goals and qualities. With the help of goal model analysis, stakeholders can answer why a reusable artifact should be chosen over another candidate reusable artifact.

The overall research problem is to improve the reusability of goal models. This paper is interested in the reuse of the actual goal model, not the reuse of knowledge or ontologies. Furthermore, the focus is not on reusing one goal model in another goal modeling notation. The focus is on reusing a goal model in the same notation as (a) a standalone artifact in possibly different contexts or (b) an artifact that is connected to other artifacts. Examples for the latter are Software Product Lines

* Corresponding author.
E-mail addresses: berk.duran@mail.mcgill.ca (M.B. Duran), gunter.mussbacher@mcgill.ca (G. Mussbacher).

https://doi.org/10.1016/j.infsof.2019.03.004
Received 10 July 2018; Received in revised form 16 January 2019; Accepted 8 March 2019
Available online 9 March 2019
0950-5849/

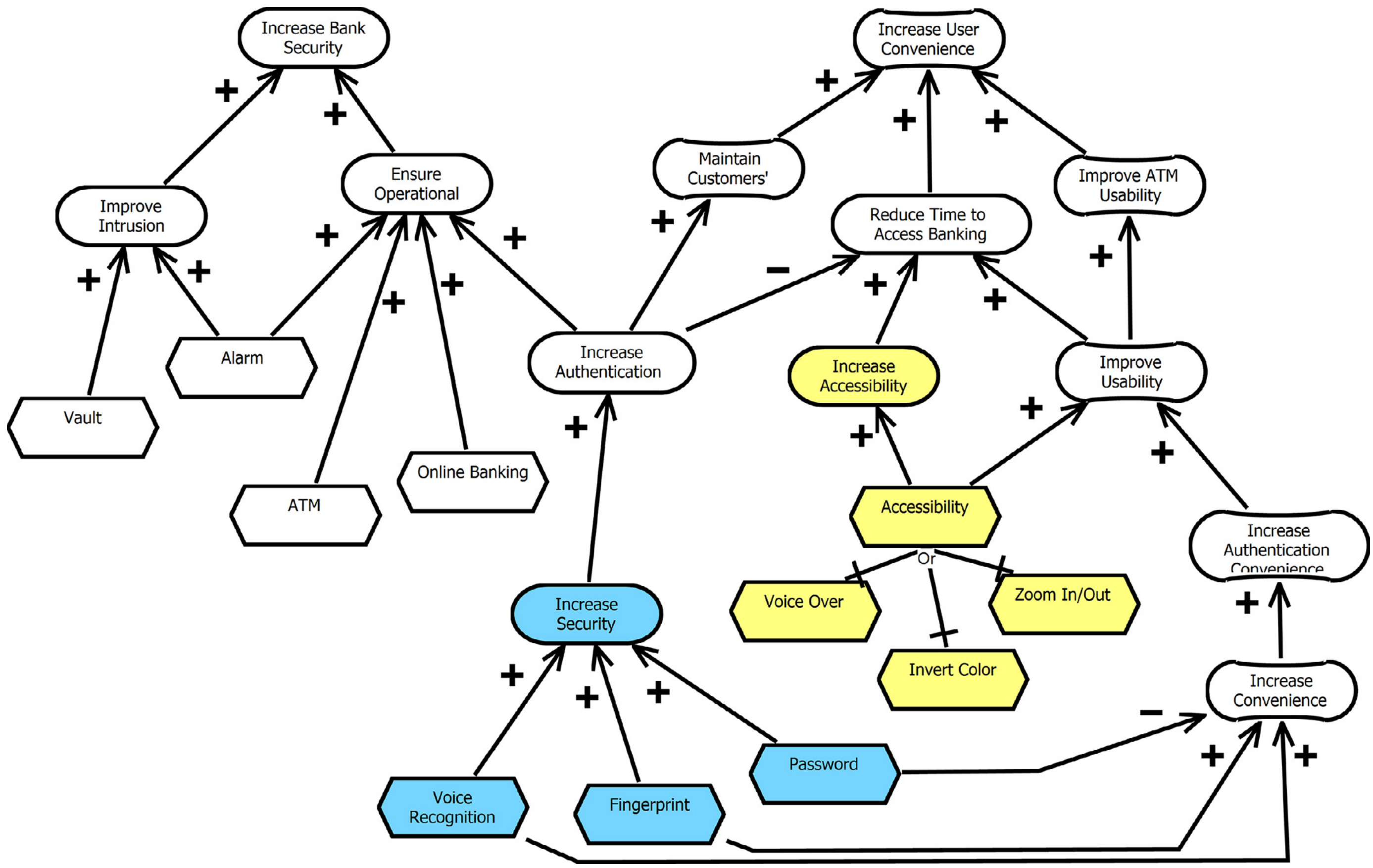

Fig. 1. An example scenario for goal model reuse with GRL.

and adaptive systems, where the goal model may be used to compare the advantages and disadvantages of various features in the Software Product Line and various solutions in adaptive systems.

The following illustrative example in Fig. 1 serves as motivation for why advanced support for reusable goal models is needed. Consider a banking application that needs to take into account security and convenience concerns. The requirements engineer starts with modeling the positive and negative impact of various parts of the bank (e.g., vault and ATM) on the high-level security goal, which is supported by many goal modeling notations.

Assume that a requirements engineer has access to reusable artifacts for authentication and accessibility. The requirements engineer proceeds by adding the goal model for authentication (highlighted in blue) and the goal model for accessibility (highlighted in yellow) to the required elements of the banking goal model. The result is a larger, monolithic goal model where it is not possible to differentiate reused elements from application-specific elements that belong to the banking domain. Note that goal model elements are only colored for illustration in this example; in real goal models all of these goal model elements look the same. Furthermore, note that many goal modeling notations also support AND/OR/XOR decompositions as shown for the accessibility element.

Several problems can be observed in this small example. First, reuse boundaries are not clearly visible, because a clear reuse interface does not exist. Second and consequently, any reasoning technique has to operate on the complete monolithic goal model, because it is not possible to distinguish the reused authentication and accessibility elements. Clear modules could reduce the complexity of these evaluations.

Third, at the time of reuse, the requirements engineer may not have all the information needed to make a decision on which accessibility technique to use (e.g., it may not be clear which specific language needs to be supported for voice over or whether the light conditions allow color inversion). In this case, it is typically only possible to select both options which does not allow the reasoning technique to take into account that only one will be used eventually. To avoid this problem, it should be possible to delay the decision to a point in time where more information is available while still being able to perform a partial evaluation.

Fourth, there may be constraints imposed on which security mechanisms can be combined with each other and which one cannot. These constraints may be too complex to be captured in a goal model and hence may have to be captured with different formalisms (e.g., feature models). Without this knowledge, however, a reasoning technique may result in an incorrect assessment.

Fifth and finally, elements in the goal model may depend on contextual information, e.g., the noise level at the location of the ATM may reduce the impact of voice recognition on the security goal and the impact of voice over on accessibility. Therefore, a reusable artifact is often designed to have a generic nature and kept unaware of the application under development, so that it can be used in many different application contexts [4]. However, creating a generic software artifact is not a simple task as it requires considering a wide range of scenarios in which the artifact could be reused. This often results in using placeholders instead of concrete model elements to represent unknown application-specific information to keep the reusable artifact generic [5]. More specifically, it is important to allow elements of a goal model to adapt to different circumstances, e.g., contribution values or link types may change between contexts because different actors/intentions may be relevant in different contexts. Consequently, a reusable goal model must allow context dependent information to be captured and properly encapsulated for the goal model to be reusable in various application contexts.

Goal models may benefit from the strengths of reuse just like any other modeling notation [5–8] (e.g., class diagrams, use case diagrams, sequence diagrams, etc.). Examples of reusable models can be found in the areas of aspect-oriented modeling [9], concern orientation [5], Software Product Lines [10], and component-based development [11,12]. To facilitate reuse of goal models, this systematic literature review aims to explore reusable goal models that support key characteristics of reusable artifacts as discussed in literature and from our experience. The long-term goal is to reuse a goal model without having to make premature decisions and violate constraints imposed on goal model elements, while at the same time allowing the goal model to be customized to the reuse context and maintaining clear reuse interfaces. The contribution of this work is twofold. First, five required characteristics of reusable goal models are identified and discussed. Second, a two-part systematic literature review identifies further research needed to fill a gap in the landscape of goal modeling approaches to support reusability of goal models with context dependent information.

This work expands on our previous work [13] which presented an overview of the systematic literature review covering publications up to the year 2016. This work includes the full details and publications up to the year 2018. The remainder of this paper presents background information on goal model reuse and the representation of contextual information in goal models in Section 2. The research methodology used to conduct the systematic review is explained in detail in Section 3 including research questions, selection of publications, and threats to validity. Section 4 summarizes the findings of the two-part systematic review and identifies five major themes for future research to support reuse of goal models with context dependent information. Section 5 concludes the paper.

2. Goal model reuse

Goal modeling is important to understand the requirements and communicate the needs and expectations of stakeholders by abstracting away unnecessary information (e.g., implementation details). Key examples for the various goal modeling languages that exist in the literature include Knowledge Acquisition in Automated Specification (KAOS) [14], i* [15], the Non-functional Requirements (NFR) framework [16], Tropos [17], and the Goal-oriented Requirement Language (GRL) [18]. Even though there are differences among these languages in terms of their supported concepts, there are also a lot of similarities. In a broad sense, the concept of a *goal* exists in all languages. i* and GRL further classify intentional elements into *goals, softgoals, tasks*, and *resources* while the NFR framework only uses *softgoals* (i.e., goals for which there is no objective measure of satisfaction). All languages except KAOS cover *contribution links*, although the means to express contributions differ. A contribution captures the impact of lower-level goal model elements (often potential solutions) on higher-level goal model elements (often goals). AND and OR *decompositions* among goals are possible in all languages.

When goal models are considered in the context of reuse, *contribution values* and how they are assigned become critical. Contribution values in goal models may take four types of values: qualitative, global quantitative, relative quantitative, and real-life (measured) values [19,20]. These values may also be crisp or fuzzy. In any case, qualitative values are considered too vague for a detailed assessment of the advantages and disadvantages of candidate solutions. Global quantitative values require a common understanding of what the values mean for consistent use of those values by independent requirements engineers. Reusable artifacts are often created independently by geographically distributed teams. It is very unlikely that these teams come to a common understanding when describing their reusable artifacts with goal models, especially for hard to measure contributions to goals such as user friendliness or convenience. Therefore, global quantitative values are not appropriate. Consequently, relative quantitative contribution values should be used in a reuse context unless objective, real-life measurements are available. Relative quantitative values allow software engineers to specify contribution values relative to all locally known impacts on a goal without having to know anything about the design of other reusable artifacts [19].

Goal model reuse has received limited attention in the goal modeling community. However, just like any other modeling notation, goal modeling can benefit from reuse to increase quality and productivity [7,8].

In the context of reuse, goal models may be used either as:

- *standalone goal models* that are used to describe a problem situation and that can be reused in different contexts by themselves while maintaining the provided analysis capabilities to better understand the needs of different stakeholders (e.g., goal catalogues [21] or patterns [22]), or
- *connected goal models* that, 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 (e.g., concerns [5]). 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. In this case, goal models guide the exploration of design choices among alternative reusable artifacts and solutions.

A comprehensive goal modeling language should support both cases (i.e., reuse of standalone and connected goal models). In any case whether a goal model describes a problem situation or reusable artifact, the focus of this work is on the reuse of the actual goal model. It is also possible to reuse guiding questions that could be derived from goal models and help build goal models in a new context, but this is not in the scope of this work.

Most catalogue and pattern based goal model reuse techniques (e.g., GRL Catalogues [21], Patterns in i* [22]) employ a copy-paste-like reuse of standalone goal models which results in large, monolithic goal models that are potentially difficult to understand, evaluate, and analyze. Furthermore, boundaries between individual copied goal models are usually lost upon their composition.

On the other hand, connected goal models bring additional challenges to the subject of goal model reuse. Concern-Orientation [5], for instance, uses goal models to express the variation interface of their modular reusable artifacts called *concerns* in building large reuse hierarchies. In the variation interface, a feature model captures the available design choices and functional variants offered by the concern whereas the impact of those design choices (i.e., features) on high-level stakeholder goals is captured in a goal model (referred to as an *impact model* in this context). This connection between modeling formalisms requires a goal model reuse technique to deal with external constraints that may be imposed by the feature model as it defines the relationships between features such as OR and XOR relationships as well as cross-tree constraints that can potentially limit the assignment of initial satisfaction values to features in the goal model.

Last, but not the least, building model hierarchies with the help of smaller, generic goal models, whether they are standalone or connected goal models, calls for the capability to represent contextual information. However, this contextual information is unavailable during design and analysis of the reusable goal model and only available later when the goal model is actually reused and the context is known. Such capability is not always provided by a goal model reuse technique.

Following these challenges and shortcomings of goal modeling approaches on reuse and to assess the extent of support of existing goal modeling approaches, the requirements that need to be fulfilled for the reuse of goal models are identified as follows:

Requirement 1 (R.1): Reuse of goal models, as for any other reusable artifact, results in reuse hierarchies, i.e., goal models describing lower-level reusable artifacts are combined together to form a higher-level artifact, which in turn is reused at the next-higher level in the reuse hierarchy. As the hierarchy is built, the goal models must also be composed to support analysis and validation from small, low-level artifacts to large, system-level artifacts. In the context of GRL, for instance, analysis in the form of trade-off reasoning is performed by a propagation-based evaluation mechanism that 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 the two [23]. As the reuse hierarchy grows with each reuse, the complexity of the evaluation of the composed models must be managed across the reuse hierarchy for the evaluation to remain feasible, because the eventual purpose of goal modeling is to support goal-based reasoning [24]. Therefore, to reuse goal models and maintain their capability for analysis and validation, it is important to*make sure that trade-off reasoning via goal model evaluation is possible through the reuse hierarchy*.

Requirement 2 (R.2): The context in which a reusable artifact is going to be used is typically unknown in advance and consequently the exact requirements for the reusable artifact are either unknown or variable [25]. Two reuse scenarios exist in general, and they are also applicable to goal model reuse. In the first scenario, the modeler has all the information about the system, its requirements, and the context in which the reusable artifact is going to operate. In this case, the modeler simply proceeds with the complete reuse of the most appropriate reusable artifact. In the second scenario, however, the modeler lacks some of the information. Nevertheless, it is desirable to allow the modeler to proceed with the limited information at hand, and support the partial reuse of a reusable artifact while at the same time leaving some reuse decisions open to be addressed later in the development. A goal model essentially describes a set of choices that are evaluated against stakeholder objectives. In the first reuse scenario a decision has been made for each choice, while in the second reuse scenario, some choices have not yet been made. Consequently, a reusable goal model requires the goal modeling approach to *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 partially composed with the application under development.*

Requirement 3 (R.3): Goal models cannot always be considered in isolation due to additional constraints on some goal model elements (typically tasks) that may be expressed in other modeling notations. For example, causal relationships between tasks at runtime may be expressed with a workflow model, while a feature model may describe which tasks can be selected together to build a functionality or a system [19]. Ignoring these constraints during goal model analysis may result in an incorrect evaluation (e.g., two tasks that cannot be selected together according to the feature model should not be part of the same solution in the goal model). Using different requirements models in collaboration with goal models benefits the modeler as each notation has their own strengths for requirements modeling. An attempt to represent all of the aforementioned information in the goal model itself results in an overly complex goal modeling notation that is hard to understand and analyze [26]. Therefore, it is necessary to *take the constraints imposed by other modeling notations into account when evaluating goal models.* Note that such constraints make the evaluation of goal models in reuse hierarchies (see R.1) much more challenging. E.g., rather than employing the usual straightforward propagation algorithms, goal model analysis may require backtracking or other forms of non-linear exploration of the goal model [19].

Requirement 4 (R.4): Creating a generic software artifact has its challenges. It requires considering a wide range of scenarios in which the artifact could be reused. A generic software artifact often uses placeholders instead of concrete model elements to represent unknown application-specific information to keep the reusable artifact generic. However, accurate representation of context is necessary as it influences user's goals and their choices to reach those goals [27]. Hence, it is important to allow elements of a goal model to adapt to different circumstances, e.g., contribution values or link types may change between contexts. This requires the goal modeling approach to*allow context dependent information to be modeled so that the goal model can be used and analyzed in various application contexts.*

Requirement 5 (R.5): Many successful reusable artifacts have clearly defined interfaces to increase encapsulation and modularity as well as strengthen the reuse hierarchy [3,28]. Goal models are no exception. In a reuse hierarchy, this interface allows lower-level goal model elements of the current artifact to be hidden from the next higher-level artifact wanting to reuse this current artifact. Modelers can also differentiate model elements of a reused lower-level artifact from model elements of the reusing artifact both at the model and the metamodel levels. Consequently, it becomes essential to*provide a well-defined interface for reuse.*

To summarize, an approach that supports goal model reuse must address these five requirements. To understand the existing landscape of goal modeling approaches with respect to these five requirements, we conduct a two-part systematic literature review that addresses (i) goal model reuse and (ii) contextual goal models. The research methodology for this two-part systematic literature review including threats to validity is described in the following section.

3. Research methodology

For the investigation of the existing body of knowledge on goal model reuse and representation of context related information in goal models (i.e., contextual goal models), the systematic literature review is conducted following the guidelines proposed by Kitchenham [29]. The procedure followed for the two-part systematic literature review and the threats to validity are described in this section.

3.1. Research questions

Emerging from the previously stated requirements for the reuse of goal models with contextual information, the following research questions are articulated:

(Q.1) To what extent do goal modeling approaches satisfy the five requirements R.1 to R.5: analysis and validation across the reuse hierarchy, delaying decisions, handling constraints imposed by other modeling notations, means to model context dependent information, and a defined reuse interface.

(Q.2) Which goal modeling concepts are used or introduced to represent context in a goal modeling notation?

(Q.3) What other requirements models, if any, are used in collaboration with reusable goal models?

(Q.4) For which reusable goal modeling approaches do tool support exist?

(Q.5) 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).

3.2. Selection of publications

To answer the research questions, the initial publication set is built upon two searches on seven major academic search engines (i.e., ACM Digital Library, Google Scholar, IEEE Xplore, ScienceDirect, Scopus, SpringerLink, and Web of Science) using the search strings:

1. *goal model* AND (*reuse* OR *reusability*)
2. *goal model* AND (*context aware* OR *context oriented* OR *context based* OR *context dependent* OR *context driven*)

with the proper syntax for each search engine as shown in Table 1 (for the search terms with context, terms with both space and dash are used). The search for goal model reuse (i.e., search string 1) was first performed on May 12, 2016 and the search for context (i.e., search string 2) on November 13, 2016. On September 18, 2018, both searches were performed again to take into consideration the publications published after the initial search.

The reasoning behind performing a second, separate search for context stems from the initial results of the first search for goal model reuse. Initially, the publications in the field of goal model reuse were targeted very broadly to get an overview of existing reuse techniques with the first search string. While the compliance of the approaches discovered

Table 1
Search syntax used per search engine.

Search Engine	String	Exact Syntax
ACM Digital Library	1	+("goal model") + (reuse reusability)
	2	+("goal model") + ("context aware" "context-aware" "context oriented" "context-oriented" "context based" "context-based" "context dependent" "context-dependent" "context driven" "context-driven")
Google Scholar	1	"goal model" reuse OR reusability
	2	"goal model" AND ("context aware" OR "context oriented" OR "context based" OR "context dependent" OR "context driven")
IEEE Xplore ScienceDirect Scopus SpringerLink Web of Science	1	"goal model" AND (reuse OR reusability)
	2	"goal model" AND ("context aware" OR "context-aware" OR "context oriented" OR "context-oriented" OR "context based" OR "context-based" OR "context dependent" OR "context-dependent" OR "context driven" OR "context-driven")

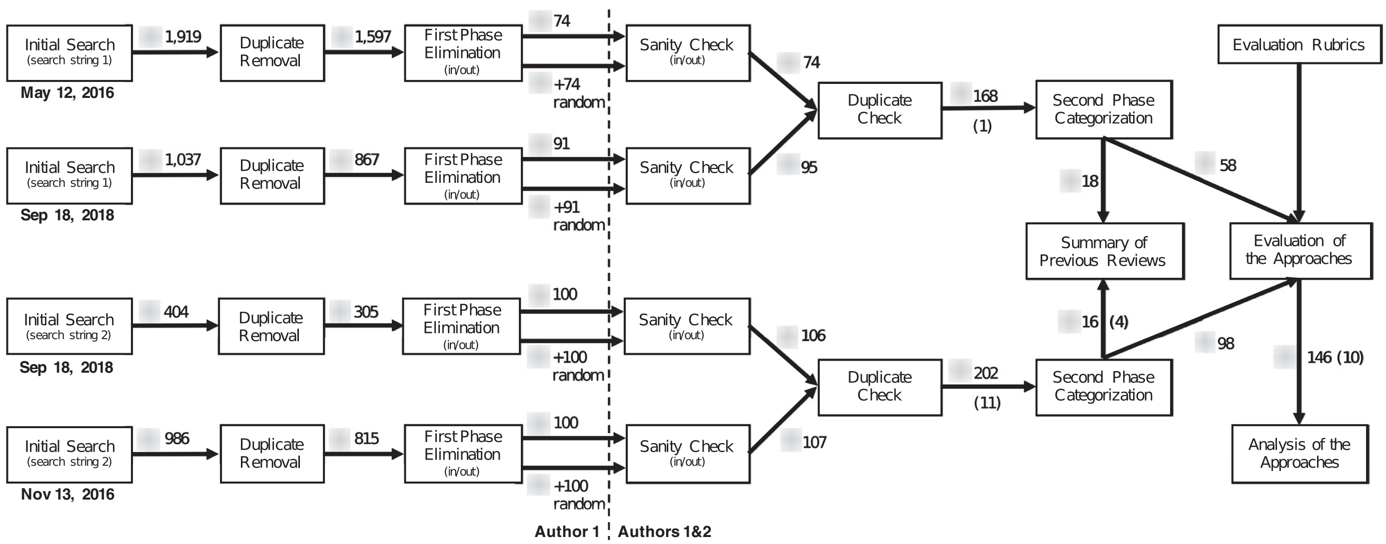

Fig. 2. Summary of the review process.

as a result of the first search with four of the requirements (i.e., R1-3 and R5) were satisfactory, the requirement related to contextual information (i.e., R4) called for further investigation. From experience, and by looking into the references that were discovered, we knew that a substantial body of work in this area is not covered by goal model reuse and our intuition was that authors do not talk about reuse explicitly while working on context. However, for reuse, it is essential to understand the context first and that is the kind of reasoning widely used by self-adaptive and context-aware systems. Consequently, the second search string was used to further explore the existing landscape of context aware goal models for their ability to support goal model reuse.

To ensure repeatability of this systematic literature review, every step in the process and every decision made are clearly documented to keep track of all changes from one step to the next for transparency and verification purposes with the help of a reference management tool [30]. An overview of the search results is given in Fig. 2.

The search performed in 2016 for goal model reuse (i.e., search string 1) on Google Scholar alone resulted in 2100 results. Considering that investigating all of them would be impractical, we limited the results from Google Scholar to the first 1000 results listed according to the relevance criteria of the search engine. With the addition of 919 hits from the other six academic search engines, we ended up with a total of 1919 raw search hits, which was brought down to 1597 after the removal of duplicate results (i.e., the publications that occur multiple times in the combined results). The search performed in 2018, with the time filter to cover the publications made available after December 31, 2015, resulted in 1037 raw hits which was brought down to 867 after duplicate removal.

The search performed in 2016 for context (i.e., search string 2) yielded a total of 986 raw search hits which was brought down to 815 after the removal of duplicate results whereas the search performed in 2018, with the time filter to cover the publications made available after December 31, 2015, resulted in 404 raw hits which was brought down to 305 after duplicate removal. Hence, all search hits for context were examined in the subsequent phase.

3.2.1. First phase

For the first phase of evaluation of the initial set of publications, inclusion criteria are developed as follows:

1. Publication contains information on goal models
2. Publication addresses reuse (for the result set of the search for goal model reuse) *OR* contextual information (for the result set of the search for context)
3. There is correlation between goal modeling and reuse (for the result set of the search for goal model reuse) *OR* goal modeling and context (for the result set of the search for context)
4. Publication is written in English
5. Type of the publication is one of the following: conference paper, workshop paper, journal article, book chapter, book, or thesis

For a publication to be included in the second phase, it is required to comply with all of the inclusion criteria listed (i.e., there is an *AND* relationship among the five inclusion criteria).

The first phase evaluation was conducted in two stages for both search strings. Due to the large number of search results, only the first author assessed the complete sets of 1597, 867, 815, and 305 publications from the four searches in 2016 and 2018 with respect to the inclusion criteria. The second author performed the same assessment on a smaller set to check and verify the accuracy of the selection. The control sets for the second author were prepared by doubling the amount of publications from the first author's assessment by including the same number of randomly picked publications from the excluded ones.

Consequently, in the first stage, the first author assessed the 2464 (i.e., 1597 from the search in 2016 and 867 from the search in 2018) publications in the result sets of the search for goal model reuse and concluded that 165 (i.e., 74 from the search in 2016 and 91 from the search in 2018) publications fit all of the inclusion criteria whereas 2299 were to be excluded. For the second stage, second sets of 74 and 91 publications were picked randomly out of the 1523 and 776 that were excluded by the first author and were merged into the original sets of 74 and 91 included publications, without leaving any indication of which papers were included by the first author. The second author then reviewed and classified these new groups of 148 and 182 publications according to the inclusion criteria.

After the second author's classification, a discussion session was held to revisit the disagreements. For the search performed in 2016, the second author initially decided to include 2 new publications that were not included by the first author. The classification of those 2 publications were changed, however, in favor of excluding them after the discussion between the authors. Following the discussion session, both authors agreed to move all 74 publications that were initially included by the first author on to the second phase. For the search performed in 2018, the second author decided to include 4 new publications in addition to the 91 that were included by the first author and following the discussion session, both authors agreed to include those 4 new publications. As both searches from 2016 and 2018 covered publications made available in 2016, an additional check for duplicates was performed on these 74 and 95 publications and 1 publication was found in common for both results. Consequently, a total of 168 publications were moved to on to the second phase. Given the large overlap of the assessment results of both authors, the confidence in the accuracy of the list moved to the second phase is deemed high enough to continue with the second phase.

Similarly, for the result sets of the search for context, two sets of 100 publications out of the initial publication sets of 815 and 305 were included by the first author in the first stage. In the second stage, two verification sets of 200 publications were prepared by randomly picking additional two sets of 100 out of the 715 and 205 publications that were excluded by the first author in the first stage. Following the classification of the second author and the discussion session, 7 new publications were included in addition to the 100 that were selected in the first stage, resulting in 107 publications from the search performed in 2016 to be moved on to the second phase in accordance with the inclusion criteria. Among the 7 newly included publications, 5 had different publications by the same authors that were already included in the initial 100. For the search performed in 2018, 6 new publications were included in addition to the 100 that were selected by the first author. Among these 6 newly included publications, 3 had different publications by the same authors that were already included either in the initial 100 or in the search for reuse. The additional check for duplicates was performed on these 107 and 106 publications and 11 publications were found in common for both results. Consequently, a total of 202 publications were moved to on to the second phase.

In total, 8 publications were missed by the first author but had authors with different publications already included in the initial list of 200 publications. The primary reason for those 8 publications being missed by the first author stems from the interpretation of inclusion criterion 3 (i.e., correlation between goal modeling and context). The second author included publications with a weaker correlation. Nevertheless, due to the large overlap of the assessment results of both authors (only 5 entirely new publications), the confidence in the accuracy of the list moved to the second phase is deemed high enough to continue with the second phase.

The large number of publications being excluded in the first phase is mainly due to not restricting the search to titles or abstracts of the publications. As a result of this, the majority of the results were excluded due to the occurrence of the search terms only in the meta-data, name of the conference, the title of the book containing the publication, or in the list of references, and not in the body, or there was no correlation between the two concepts (i.e., a violation of the third inclusion criterion, e.g., different meanings of the words "goal" and "context" were used, words "reuse" and "context" referred to other concepts in the publication).

3.2.2. Second phase

Answering the research questions requires focusing on the publications that contain a goal model reuse approach. Therefore, the 168 publications from the first phase related to goal model reuse are categorized according to their content in the second phase. This categorization is not based on a predefined set of labels, but rather emerged from the examination of the publications:

- 58 contain a goal model reuse approach (see Tables 2 and 3 Search String 1),
- 72 use goal modeling in a wider context, as part of a proposed approach, and do not focus or elaborate on goal models and their reuse[1],
- 15 introduce a new goal modeling language or an extension to an existing one but reusability is not investigated and a reuse approach is not specified[1],
- 5 propose an adaptation of an existing goal modeling notation to a specific domain, while not focusing on reuse[1],
- 18 provide a comparison of approaches or an assessment of existing goal modeling languages[1].

A similar categorization is performed for the 202 publications from the first phase related to context:

- 98 introduce an approach for modeling contextual information in goal models (see Tables 3–5 - Search String 2),
- 39 use one of the contextual goal modeling approaches in the first category as is[1],
- 45 use goal models in a wider context, as part of a proposed approach, and do not focus or elaborate on how context is/should be represented in goal models,
- 20 provide a comparison of approaches or an assessment of existing contextual goal modeling approaches[1].

The complete sets of 168 and 202 publications that were moved on to the second phase were categorized together by the authors through a discussion session, i.e., each author individually assessed the publications, and the results were then compared and discussed until consensus was reached.

The comparisons of approaches and assessments of existing goal modeling languages are investigated further to check whether the research questions posed by this systematic literature review have been addressed before (see Section 4.1.1 for the 18 comparisons from the search for goal model reuse and Section 4.1.2 for the 20 comparisons from the search for context - four appeared in both search results). The 58 publications on a goal model reuse approach and the 98 publications on a contextual goal modeling approach are analyzed further in Section 4.2 to address the research questions identified in Section 3.1. Ten publications appeared in both search results. The remaining publications are discarded (i.e., 92 related to goal model reuse and 84 related to contextual goal models). A complete overview of the results of the systematic review process is given in Fig. 2.

[1] For space reasons, we provide the full list of publications in each category online: http://www.ece.mcgill.ca/gmussb1/ReusabilityInGoalModelsSLR/Categorizations.pdf.

Table 2
Summary of goal model reuse approaches.

Approach	Search String	R.1	R.2	R.3	R.4	R.5	Context Representation	Supplementary Models	Tool Support
GRL Catalogues [21]	1	◑	◑	○	◑	◑	strategies and beliefs	use case maps	●
i* Modules [76]	1	◑	◑	○	○	◑	–	–	●
AoGRL [77–81]	1	◑	◑	◑	◑	◑	strategies and beliefs	use case maps	F
Goal Aspects [82,83]	1	◑	◑	◑	○	◑	–	sequence d.	●
Aspects with Tropos [84]	1	F	○	◑	○	●	–	use case maps	F
Aspects with i* [85]	1	F	○	○	○	●	–	–	F
Aspects with V-graph [86,87]	1	○	○	○	◕	◑	topics	–	F
AspectKAOS [88,89]	1	○	○	○	○	◑	–	–	F
ExtendedKAOS [90,91]	1	○	○	◑	○	○	–	responsibility, object, operation	●
GoPF [92–95]	1	◑	◑	◑	◑	●	strategies	use case maps	F
Task Knowledge Patterns [96]	1	○	○	◑	○	◑	–	role model, organization m.	○
Patterns in i* [22]	1	◑	◑	○	○	◑	–	–	○
Attack Patterns [97]	1	○	○	○	○	◑	–	–	F
Patterns in NFR [98]	1	◑	◑	◑	○	◑	–	use case model, sequence d.	F
GOPCSD [99,100]	1	◑	○	◑	○	◑	–	state diagram	●
Variability in Goal Models [101]	1	◑	◑	○	○	●	–	–	F
KAOS Templates and Patterns [102]	1	◑	○	◑	○	◑	–	agent, object, and operation model	●
Goal-Oriented Interaction Diagrams [103]	1	○	○	○	○	◑	–	–	○
GoalSPEC [104]	1	◑	○	○	○	○	–	–	●

Table 3
Summary of goal model reuse and contextual goal modeling approaches.

Approach	Search String	R.1	R.2	R.3	R.4	R.5	Context Representation	Supplementary Models	Tool Support
Reusable Goal Models [3,13,19,20,105–111]	1, 2	●	●	●	◑	●	strategies and re-exposition	feature model, workflow model	●
Security Patterns [112–114] [115–117]	1, 2	◑	◑	○	●	◑	domain properties	–	●
Context Transformations [118]	1, 2	○	○	○	○	◑	–	–	F
Goal-Oriented Holonic Systems [119–121]	1, 2	○	○	○	◕	●	context monitoring, required evaluation, and required goals	component models	○
Capability-Driven Development [122–141]	1, 2	○	◑	○	◕	○	context element & KPI	–	●
STREAM-A [142,143]	1, 2*	◑	○	◑	◕	●	context annotations	acme	F
Xipho [144,145]	1**, 2	○	○	○	◕	○	contextual beliefs and resources	–	F
Contextual Goal Models [27,69,146–155]	1***, 2	◑	◑	◑	●	○	decompositions, dependencies, contributions, goal activation	feature m. & problem frames	●

* not categorized as one of the 98 contextual goal modeling approaches but as one of the 39 that use a contextual goal modeling approach as is
** not categorized as one of the 58 goal model reuse approaches but as one of the 72 that use goal modeling in wider context
*** not categorized as one of the 58 goal model reuse approaches but as one of the 15 that introduce a new goal modeling language or an extension to an existing one

Table 4
Summary of contextual goal modeling approaches.

Approach	Search String	R.1	R.2	R.3	R.4	R.5	Context Representation	Supplementary Models	Tool Support
FLAGS [156–158]	2	◑	◑	○	●	○	adaptive goals	–	F
Claim Refinement Models [159–161]	2	◑	◑	○	●	○	claims, contributions	–	●
Agents with Context-dependent Goal Models [162]	2	◑	◑	○	●	○	conditional goals, decompositions, contributions	–	○
Pragmatic CGM [163,164]	2	◑	◑	○	●	○	pragmatic goals (contextual interpretation)	–	F
Context-enriched Goal Models [165–168]	2	◑	◑	○	●	○	domain properties & contextual tags	–	F
Context Sensing Goal Models [169]	2	◑	◑	○	●	○	context variables	ERD, state d.	○
Extended Tropos Goal Model [170,171]	2	◑	◑	○	●	○	context conditions & internal event conditions	–	F
Goal-oriented Context Requirement Model [172–174]	2	○	◑	○	◕	○	context conditions, decompositions	process model	●
URN [175–177]	2	◑	◑	◑	●	○	strategies and KPIs	use case maps	●
Context Feature Model [178]	2	◑	◑	○	●	○	feature model	context feature m. & feature m.	○
Adaptive RML [62,179,180]	2	◑	◑	○	●	○	context and resources	–	●
AwReqs [181,182]	2	◑	◑	○	●	○	variation points, control variables, indicators, awareness requirements	–	●
Generic Quantitative Goal Model [183]	2	◑	◑	○	●	○	context-dependent weights	–	F
Contextual Preference Model [184]	2	◑	◑	○	●	○	contextual preference specification	–	●
Quality Constraint Templates [185]	2	◑	◑	○	●	○	quality constraint template	–	F

Table 5
Summary of contextual goal modeling approaches (cont.d).

Approach	Search String	R.1	R.2	R.3	R.4	R.5	Context Representation	Supplementary Models	Tool Support
GCPD [186]	2	◑	◑	○	●	○	contextual factors & constraints	–	●
CARGO [187–189]	2	◑	◑	○	●	○	KPIs	use case maps	●
i*-Context [190]	2	◑	◑	○	●	○	agent roles and role playing relations	–	○
Adaptation Goal Model [191–194]	2	◑	◑	○	●	○	adaptation goals, atomic contexts, tags, uncertain entities	context model	F
SecuriTAS [195,196]	2	◑	◑	○	●	○	contextual factors	–	●
REFAS [197,198]	2	◑	◑	○	●	○	context variables & claims	–	●
RBUIS [199]	2	◑	◑	○	●	○	context element	–	●
Feature-Oriented NFRs [200]	2	◑	◑	○	●	○	feature context model	feature model	F
Tropos4AS [201]	2	◑	◑	○	●	○	goal conditions	environment m. & failure m.	●
Event-based Goal Adaptation [202]	2	◑	◑	○	●	○	event & indicator	–	●
G+ [203]	2	○	○	○	◕	○	goal functional context	–	○
Agon [204–206]	2	◑	◑	○	●	○	user context model & context dimension trees	–	F
Tropos with Control Model [207–210]	2	◑	◑	○	●	○	ECA rules & domain properties	control model	○
IRIS [211]	2	◑	◑	○	●	○	big queries, big analytics, big data, and KPIs	process model	●
ConG4DaS [212]	2	◑	◑	○	●	○	context model & context annotations	–	F

3.3. *Threats to validity*

Threats to the validity of the systematic literature review are discussed following the classification of threats proposed by Petersen and Gencel [31] according to the definition of Maxwell [32].

3.3.1. *Descriptive validity*

Descriptive validity refers to the factual accuracy of the research and its ability to correctly explain objective/subjective truth [31].

A threat to descriptive validity in this work is that the first phase assessment is performed predominately by the first author, which could lead to bias. As countermeasures, the second author performed the same assessment on a smaller, randomly generated set to check and verify the accuracy of the decisions made by the first author as explained in detail in Section 3.2. Results of this verification showed that for both search strings, the complete sets of 74, 91, 100, and 100 publications that were included by the first author were moved to the second phase. 17 new publications were included by the second author, where only 9 out of the 17 did not have any related publications included initially (i.e., 8 of the 17 publications already had similar publications included by the first author). Furthermore, clear inclusion criteria are defined and agreed upon by both authors to reduce room for interpretation, and all decisions made in the process are documented. The state of the publication set at each step is stored transparently with the help of a reference management tool [30]. To not risk missing an approach that might be relevant to the purpose of the literature review, all of the publications that fit the inclusion criteria and contain a clear correlation between the concepts used in the search query are taken into consideration for the second phase.

3.3.2. *Theoretical validity*

Investigation of confounding factors and the capability of the researcher to capture what was intended, addresses theoretical validity [31].

A threat to theoretical validity is conducting the search in two stages through extending the search for goal model reuse approaches with context and goal modeling to further investigate the extent to which existing goal modeling approaches satisfy the fourth requirement (i.e., R4). The reason why no other tailored search strings were used for the other requirements (R1-3 and R5) is due to our experience in the area of goal model reuse regarding those four requirements compared to our more limited experience with context aware goal models. Our intuition was that reuse is not explicitly stated by the authors working on context and it turned out to be accurate considering that there are only eight publication groups related to search strings 1 and 2 (see Table 3), while there are 38 publication groups in total related to search string 2 (see Table 3–5).

Another threat to theoretical validity arises from the search terms not being sufficient. To address this threat, the search term is varied to be inclusive for possible variations of the use of terms (e.g., terms such as context-aware are included both with dash and space in the search phrase). Furthermore, seven different academic search engines are used to keep the sample size big enough in addition to defining the inclusion criteria in a broad way. Snowballing could have also been used as a countermeasure for this threat, but was not performed because of the broadness of the search terms and inclusion criteria.

3.3.3. *Generalizability*

Generalizability is concerned with the degree to which the results of the research can be generalized to different situations [31].

One of the threats to generalizability is the inclusion of the first 1,000 results out of 2100 according to the relevance ranking of Google Scholar, for the search for goal model reuse performed in 2016. To address this threat, all of the search results from the remaining six academic search engines are taken into consideration. Furthermore, to determine the likelihood of missing a relevant goal model reuse approach, we look at the 15 publications in the final publication set that only appeared in Google Scholar results in the search performed in 2016. 13 out of 15 publications (i.e., 87%) are among the first 600 hits and the 15th publication is still in the first 900 hits. Therefore, it is not very likely that a relevant publication is not among the top 1000 hits.

To make the results of this review more generalizable, the publication set is kept broad by conducting separate searches for reuse and context.

3.3.4. *Interpretive validity*

Interpretive validity refers to the investigation of conclusions drawn to see whether they are reasonable according to the data depicting objective/subjective truth [31].

Formalization of the inclusion criteria and categorization rules is done by both authors to address the threat to interpretive validity and to prevent bias in the evaluation of the approaches. Furthermore, the result sets at each step are verified by the second author.

3.3.5. *Repeatability*

Repeatability (or reproducibility/dependability) relies on a clear definition of steps followed for the research to be repeatable and is argued to follow from the previous categories listed [31].

To make sure that it is possible to repeat this systematic literature review, following the same guidelines and reproducing the same results, all steps followed at each phase of the review are diligently recorded. A reference management tool [30] is used to keep track of all of the decisions made and consequently the results of each step are clearly traceable.

4. Results and analysis

This section presents the results of the second phase categorizations and analysis of the approaches. A summary of the 34 publications (4 publications [33–36] appear in both sets of 18 and 20 publications) that provide a comparison of approaches or an assessment of existing *(i)* goal modeling languages and *(ii)* contextual goal modeling approaches is given in Section 4.1. This is followed by an analysis of the 146 publications in the combined, final publication set (10 publications appear in both sets of 58 and 98 publications) in Section 4.2 to answer the research questions Q.1 to Q.4 from Section 3.1. Finally, a discussion of future research themes (research question Q.5) is provided in Section 4.3.

4.1. Summary of previous reviews

According to the guidelines proposed by Kitchenham [29], a summary of previous reviews is needed for the justification of the need for the review. Therefore, we review existing comparisons and assessments related to goal model reuse and contextual goal models and present our analysis on them.

4.1.1. Reviews of goal model reuse

A comparison of goal based (i*-based) and use-case based approaches in terms of requirements reuse, reuse of provided artifacts in a later phase, and the types of reusable artifacts for security requirements modeling [37] concludes that for i*-based approaches, the provided artifacts are not reusable in later phases. However, most of the ideas and concepts used during the early and late requirements phases are considered reusable and classified as consistent, modifiable, and appropriate for requirements reuse.

Comparison of i* and KAOS in terms of quality [38] derives that both languages and tools suggest means for reuse, yet they lack methodological guidelines and their tools are not sufficient to ensure the semantic quality of models. Semantic quality examines the correspondence between the information captured by the model and the modeled domain [39]. i* models are considered difficult to reuse [40] due to their reusability being confined to a basic copy and paste of designated elements. Similarly, the evaluation of KAOS and its associated tool Objectiver [41] points out that KAOS does not provide support for libraries of goal models, and hence reuse.

Muñoz-Fernández et al. [35] introduce 10 challenges they identified for the specification of self-adaptive software (SAS) and claim that reusable patterns help modelers define optimization models for SAS, as well as their evaluation and verification. Similarly, Stirna [42] puts an emphasis on pattern reuse in their assessment of capability design concepts while Akhigbe et al. [43] highlight the use of model and pattern families in a catalogue-based manner to enable goal model reuse.

In a study, Bombonatti et al. [44] conduct a systematic mapping to discover (i) the non-functional requirements that are more often related to reusability and (ii) the context factors that are tightly associated with software reuse. While the former aspect of their study is not in our scope for goal model reuse, results for the latter show that incorporating domain knowledge in goal modeling is an important facilitator of reuse.

Focusing on the knowledge reuse in security requirements engineering, Souag et al. [45] derive that security requirements engineering approaches reuse knowledge in forms of security patterns, taxonomies and ontologies, templates and profiles, catalogs and generic models, and as a mixture of these. While the study is targeted at security requirements and knowledge is not necessarily represented as a goal model, their findings on the commonly used forms of knowledge reuse are aligned with ours for the majority of goal model reuse approaches that we investigate in this paper.

The remaining 9 publications [33,34,36,46–51] classified as a comparison of approaches or an assessment of existing goal modeling languages are excluded from this summary upon closer inspection, because they do not cover and report on goal model reuse in their assessments.

To summarize, existing literature reviews in the area of goal model reuse show that despite knowledge and expertise encapsulated in goal models being valuable and reusable, existing approaches that promote goal model reuse fall short of producing fully reusable artifacts in the end. Furthermore, the body of work targeted by the previous reviews covers only a limited portion of the corpus of goal model reuse. Therefore, this systematic literature review is novel for exploring reusability in goal models that support key characteristics of reusable artifacts as discussed in literature and from our experience.

4.1.2. Reviews of contextual goal models

Modeling and using contextual information in model driven engineering (MDE) practices has always received attention from the requirements engineering community. Several approaches have been proposed and used for the representation of context in goal models in the literature. As a result of this, substantial research has been done on the evaluation and comparison of those approaches, taking multiple aspects of modeling in consideration.

Paja et al. [52] show, through a comparative study, that goal modeling is insufficient to reason about the best alternative without the capability for context modeling as the optimal decision varies depending on the context.

Kolos-Mazuryk et al. [53] analyze how the Goal-Based Requirements Analysis Method (GBRAM) [54], KAOS [55], and i* [56] allow modeling of system context through a comparison of three goal-oriented requirements engineering (GORE) methods from the perspective of pervasive services and context modeling. Their results on context modeling affirm that, among the three, only i* partly covers representation of the context and identification of context attributes. Furthermore, studies of i* extensions by Gonçalves et al. [57,58] point out an increase in the amount of publications oriented around modeling contextual information in i*. Context attributes whose values may change, activities that may initiate the aforementioned attribute changes, actors that perform the activities, and correlations between context changes are considered necessary for pervasive systems.

In a study, Alegre et al. [59] conduct a survey targeted at the development of context-aware systems, methodologies used, challenges involved, in addition to the reasons for their lack of adoption. They investigated requirements elicitation techniques that have been tailored for meeting the needs of context-aware system development. Adaptive and goal oriented requirements elicitation techniques are claimed to provide means to capture and analyze variability and open the way to requirements evolution with the disadvantage of not being able to easily detect when the system behavior meets the requirements.

Requirements engineering practices also play an important part for self-adaptive systems (SAS), which are context-aware. Soares et al. [60] provide a comparative study to evaluate the support of goal modeling approaches for self adaptive systems. The goal-oriented approaches studied are Tropos4AS [61], Adaptive RML [62], and Design Goal Model (DGM) [63]. These three approaches are not used for general purpose goal modeling, but are developed to allow modeling the requirements of SAS. As the name suggests, a self-adaptive system needs to decide whether it is necessary to adapt to a change in system's context. The comparative study concludes that the three approaches have difficulties in representing the mechanisms for the adaptation and it is not clear whether they support modeling the evolution of the system. As mentioned in the previous section, Muñoz-Fernández et al. [35] point towards reusable patterns for the reuse of contextual information in their study on SAS. Furthermore, a survey of engineering approaches for SAS [64] highlights that goal-oriented modeling techniques play an important role in designing and influencing how an adaptive system should behave. On the other hand, in their studies on requirements modeling and analysis for SAS, Yang et al. [65,66] associate reuse primarily with domain modeling rather than goal modeling.

Various modeling notations exist for modeling the system requirements and as a consequence of this, it is possible to observe the need for representing contextual information during requirements modeling in other modeling notations, too. Systematic reviews targeting variability modeling in dynamic software product lines (DSPL) highlight one common message, i.e., the dominant modeling notation in the field is feature models. Guedes et al. [67] presents the results of a systematic mapping study to determine how variability is modeled in DSPL approaches, where two thirds of the investigated publications use feature models. Goal models are claimed to describe the variability in terms of alternatives to stakeholder or system goals.

A ranking of DSPL modeling techniques [68] places a technique using Tropos goal models integrated with context information [69] in the second place, being the only goal modeling notation among ten modeling techniques included in the survey. Shortcomings of the goal modeling technique are stated as the lack of a real case study, not using multiple layers of abstraction, not completely supporting constraints (aside from dependency links), not supporting modeling unforeseen adaptations at development time, and not supporting decision point activation and modification. A further evaluation of variability modeling techniques for dynamic software product lines (DSPL) [70] states that, although the technique using Tropos goal models integrated with context information [69] is less effective than the Context-aware Feature Model technique [71] in terms of precision, both techniques are considered effective in terms of recall.

The remaining 7 publications [33,34,36,72–75] classified as a comparison of approaches or an assessment of existing contextual goal modeling approaches are excluded from this summary upon closer inspection, because they do not cover and report on reuse of contextual goal models in their assessments.

In summary, existing literature reviews in the area of contextual goal models show that the existing body of work, although rich, is composed of research in different application domains and hence often restricted to the approaches in that particular domain. The systematic literature review performed in this work is novel for targeting specifically goal modeling and how contextual goal models are/can be used in the context of reuse and in the presence of reuse hierarchies.

4.2. Analysis of the approaches

The 58 publications on goal model reuse approaches and the 98 publications on contextual goal modeling approaches are taken into further consideration as described in Section 3. As 10 of those publications appear in both final sets, the combined 146 publications are grouped under the name of the approach they present and compared in Tables 2–5. This resulted in 27 approaches for goal model reuse (Tables 2 and 3) and 38 for contextual goal models (Tables 3–5) where the 8 approaches in Table 3 are in common.

The column labeled "*Search String*" shows in which part of the two-part search the approach appeared (i.e., 1 corresponds to the search for goal model reuse, 2 corresponds to the search for context and goal models, and 1, 2 means the approach appeared in both).

The five columns labeled R.1 to R.5 correspond to the five requirements identified in Section 2 and show the degree of satisfaction of their corresponding requirement with respect to the predetermined evaluation criteria defined in Tables 6–10. These five columns address the research question Q.1 from Section 3.

If an approach contains modeling constructs to represent context in goal models, and/or makes use of other requirements modeling notations in collaboration with goal models, they are shown in the corresponding columns labeled "*Context Representation*" and "*Supplementary Models*", respectively. The "*Context Representation*" column addresses research question Q.2 whereas the "*Supplementary Models*" column addresses research question Q.3 from Section 3.

The last column shows whether tool support exists (●) for the corresponding approach or not (○) in order to answer research question Q.4 from Section 3. If a criterion in one of the columns is stated as future work in the publication, it is shown with the letter F.

In order to maintain the readability of the analysis, only the names of the approaches are used in the remainder of this section. For the references to their corresponding publications, we point the reader to the "*Approach*" columns of Tables 2–5.

4.2.1. Trade-off reasoning

As shown in the evaluation rubric for R.1 in Table 6, for an approach to fully satisfy the first requirement of providing support for analysis or validation through reuse hierarchy, the reasoning mechanism provided should also deal with complexity resulting from large reuse hierarchies.

In Tables 2–5, R.1 is partially satisfied by 42 approaches. Even though goal model evaluation is possible with 37 of these approaches (GRL Catalogues, i* Modules, AoGRL, Goal Aspects, Patterns in i*, Patterns in NFR, GOPCSD, GoalSPEC, Security Patterns, Contextual Goal Models, FLAGS, Claim Refinement Models, Agents with Context-dependent Goal Models, Pragmatic CGM, Context-enriched Goal Models, Context Sensing Goal Models, Extended Tropos Goal Model, URN, Context Feature Model, Adaptive RML, AwReqs, Generic Quantitative Goal Model, Contextual Preference Model, GCPD, CARGO, i*-Context, Adaptation Goal Model, SecuriTAS, REFAS, RBUIS, Feature-Oriented NFRs, Tropos4AS, Event-based Goal Adaptation, Agon, Tropos with Control Model, IRIS, and ConG4DaS), the absence of a reuse hierarchy (reuse is done via copy and paste operation) leads to one big monolithic goal model which, in turn, is complex to evaluate when considering additional constraints imposed by other modeling notations on the goal model.

The STREAM-A approach modularizes i* models via grouping the model elements into actors that can be reused in other domains. The GoPF, Quality Constraint Templates, and KAOS Templates and Patterns approaches also define the boundaries of the reusable artifacts (goal templates) well. The Variability in Goal Models approach uses goal models in the decision-making process of choosing variants and divides a system into smaller aspects. These five approaches allow the evaluation to be done on the composed model, although not incrementally; due to which, the complexity issue persists.

The Reusable Goal Models approach satisfies this requirement by allowing the goal model evaluation to span the entire reuse hierarchy and incrementally evaluating levels in the reuse hierarchy. Furthermore, the incremental evaluation mechanism is taken a step further to leverage the reduced complexity in a top-down evaluation of goal models.

Two approaches, Aspects with Tropos and Aspects with i* state that a reasoning mechanism is planned for their future work while the remaining twelve approaches do not specify any means for analysis or validation of the goal model.

4.2.2. Delaying decisions

In order to satisfy the second requirement of providing the means to delay decisions to a later point in the reuse hierarchy, a reusable goal modeling approach shall incorporate delayed decisions and associated uncertainty into the composition and evaluation of reusable artifacts.

As shown in the evaluation rubric in Table 7, not being able to make use of the uncertainty represented in a goal model in the composition and evaluation, results in the partial satisfaction of this requirement.

Consequently, the second requirement of providing support for delaying reuse decisions to a later point (R.2) is partially satisfied by 40 approaches as they allow use of data obtained at runtime for re-evaluation of goal satisfactions or system re-configuration (Capability-Driven Development, Contextual Goal Models, FLAGS, Claim Refinement Models, Pragmatic CGM, Context Sensing Goal Models, Extended Tropos Goal Model, Goal-oriented Context Requirement Model, URN,

Table 6
Evaluation Rubric for R.1.

Criteria	Evaluation
Does not specify any means for analysis or validation of the goal model (i.e., reasoning mechanism).	○
States that a reasoning mechanism is planned for future work.	F
Describes a reasoning mechanism.	◑
For the described reasoning mechanism, also deals with complexity resulting from large reuse hierarchies.	●

Table 7
Evaluation Rubric for R.2.

Criteria	Evaluation
Does not specify any means for delaying decisions.	○
States that support for delaying decisions is planned for future work.	F
Describes means to delay decisions and enable representation of uncertain input values.	◑
Incorporates delayed decisions and associated uncertainty into the composition and evaluation of reusable artifacts.	●

Table 8
Evaluation rubric for R.3.

Criteria	Evaluation
Does not mention any other modeling notation that imposes external constraints on goal models.	○
States that support for external constraints is planned for future work.	F
Describes means to capture constraints imposed by other modeling formalisms on goal models	◑
Takes external constraints into account for the evaluation of goal models.	●

Table 9
Evaluation rubric for R.4.

Criteria	Evaluation
Does not specify any means for modeling context dependent information.	○
States that support for context dependent information is planned for future work.	F
Describes means to capture contextual information.	◑
Uses explicit language constructs to represent context related information.	◕
Uses contextual information that is explicitly modeled during the analysis of the goal model.	●

Table 10
Evaluation Rubric for R.5.

Criteria	Evaluation
Does not specify any means to define module boundaries (e.g., reuse interfaces).	○
States that support for clear module boundaries is planned for future work.	F
Describes clear module boundaries.	◑
Maintains the modularity and clear model boundaries after reuse of the goal model.	●

Context Feature Model, Adaptive RML, Quality Constraint Templates, GCPD, CARGO, i*-Context, Adaptation Goal Model, SecuriTAS, REFAS, RBUIS, Tropos4AS, Event-based Goal Adaptation, Agon, Tropos with Control Model, and ConG4Das), or use qualitative reasoning to cope with uncertainty (GRL Catalogues, i* Modules, AoGRL, Goal Aspects, GoPF, Patterns in i*, Patterns in NFR, Variability in Goal Models, Security Patterns, Agents with Context-dependent Goal Models, Context-enriched Goal Models, AwReqs, Generic Quantitative Goal Model, Contextual Preference Model, Feature-Oriented NFRs, and IRIS). However, these approaches do not address reuse hierarchies, require knowledge of the whole system, and hence do not explicitly specify mechanisms to delay decision making to a later point in the reuse hierarchy (e.g., indicating which decisions still need to be taken or are allowed to be delayed and the impact of doing so).

The Reusable Goal Models approach makes use of re-exposed features, which explicitly indicate those goal model elements in the reused goal model that still can be included in or excluded from the system at a later time in the reuse hierarchy. Furthermore, with the help of a range evaluation mechanism, this approach makes full use of re-exposition and the associated uncertainty in the evaluation of the goal model by reporting the still possible evaluation results given the uncertain elements. Consequently, it is the only approach that fully satisfies this requirement.

4.2.3. External constraints and supplementary models

In this section, the satisfaction of the third requirement is discussed together with the supplementary models used by goal modeling approaches. The rubric used in the evaluation of the third requirement of taking constraints imposed by other modeling notations into account (R.3) is shown in Table 8. The column labeled *Supplementary Models* in Tables 2–5 addresses research question Q.3 from Section 3 and is discussed together with the models that are used in collaboration with goal models in this section.

The third requirement (R.3) is partially satisfied by 12 of the 57 approaches. STREAM-A uses i* models in collaboration with Acme architectural models. While the Goal Aspects approach explicitly uses sequence diagrams, Patterns in NFR use both use case and sequence diagrams in collaboration with goal models. Aspects with Tropos explicitly allow modelers to express workflow models with Use Case Maps. Responsibility, object, and operation sub-models of KAOS are also used in the ExtendedKAOS approach. Task Knowledge Patterns use role and organizational models, KAOS Templates and Patterns use agent, object, and operation models, whereas the GOPCSD approach uses state diagrams to capture external constraints. Contextual Goal Models, on the other hand, allow modelers to use feature models and problem frames to complement goal models. AoGRL, GoPF, and URN use links between Use Case Maps and GRL goal models to capture constraints. However, these approaches do not leverage the constraints imposed by the other modeling languages in the evaluation of the goal model, which prevents the full satisfaction of the third requirement.

The Reusable Goal Models approach satisfies this requirement with an updated evaluation mechanism that handles external constraints imposed by both feature models and workflow models.

This requirement is not satisfied by any other approach (i.e., no publication explicitly mentions any other modeling notation that imposes constraints on goal models and their evaluation).

In addition to the 13 approaches that either partially or fully satisfy the third requirement (R.3), 11 of the approaches in the final set make use of supplementary models in addition to their goal models. Although the supplementary models used by these 11 approaches do not impose any external constraints on goal models, it is important to highlight them to answer the third research question (i.e., What other requirements models, if any, are used in collaboration with reusable goal models?).

Overall, behavioral models in the form of workflow models (i.e., Use Case Maps and process models), sequence diagrams, and state diagrams make up the majority of the supplementary models. Furthermore, several structural modeling notations such as role models, component models, organization models, use case models, Entity Relation Diagrams (ERD), environment and failure models, problem frames, control models, and context models are also used in collaboration with goal models. Finally, feature models are used by four approaches in connection with goal models.

4.2.4. Contextual information

The fourth requirement to allow context dependent information to be modeled (R.4) is evaluated according to the rubric given in Table 9.

A difference in the rubric for this requirement is the use of three-quarters (◕) as a measure of partial satisfaction in addition to the lesser one-half (◑) measure of partial satisfaction. This emerges from the need for a distinction between the approaches that use explicit modeling constructs to capture contextual information and those that do not. For the complete satisfaction of this requirement (●), an approach is required to use explicitly captured contextual information in the analysis of the goal model.

GRL possibly captures contextual information in strategies and beliefs without explicitly defining the concept of context. Therefore, GRL-based approaches that only make use of strategies and beliefs such as GRL Catalogues, AoGRL, GoPF, and Reusable Goal Models do not meet the three-quarters partial satisfaction criterion and only fulfill the one-half partial satisfaction criterion.

Seven of the approaches meet the three-quarters partial satisfaction criterion whereas the fourth requirement is only fully satisfied by Security Patterns and Contextual Goal Models from the search for goal model reuse (i.e., search string 1 in Tables 2 and 3). In addition, the majority (28 out of 38 approaches) that appeared in the search for context and goal models (i.e., search string 2 in Tables 3 and 5) fully satisfy the fourth requirement.

As shown in the column labeled "*Context Representation*" to address research question Q.2 from Section 3, these approaches provide the following constructs for modeling contextual information. Starting with the seven approaches that do capture context explicitly without incorporating this information into analysis, Aspects with *V*-graph uses the concept of topic in *V*-graph model to capture contextual information for goals, tasks, and softgoals whereas Goal-Oriented Holonic Systems introduce holons with interfaces that cover the concepts of context monitoring, required evaluation, and required goals. Capability-Driven Development uses context KPIs in their run-time adjustments. STREAM-A adopts the context representation of Contextual Goal Models with their use of context annotations. Xipho, an agent-oriented methodology for engineering context-aware personal agents (CPA), extends Tropos with contextual beliefs and resources for the representation of contextual information and to support CPA development. Goal-oriented Context Requirement Model uses a goal refinement model with and/or decompositions and context conditions whereas G+ makes use of goal functional context to represent context in the form of pre/post-condition constraints for goals.

Out of the 30 approaches overall that fully satisfy the fourth requirement, Security Patterns employ contextual goal models for their representation and specifies context in terms of domain properties. On the other hand, Contextual Goal Models identify variation points in their goal models as or-decomposition, means-end, actor dependency, root goals, and-decomposed goals, and contribution to softgoals.

FLAGS adds adaptive goals to KAOS models, specifying how the system should handle its self-adaptation through application of a suitable countermeasure, while Claim Refinement Models use claims and changing contribution values. Agents with Context-dependent Goal Models use conditional goals, decompositions, and contributions to answer whether agents can support their goals within a given context. Pragmatic CGM introduces pragmatic goals that have a dynamic satisfaction criterion and an algorithm for the evaluation of their achievability.

The Context-enriched Goal Models and ConG4DaS approaches define a set of contextual tags or context annotations and are concerned with the visibility of model elements, whereas the Context Sensing Goal Models and REFAS approaches identify context variables that are later utilized in order to achieve an objective. Extended Tropos Goal Model introduces context conditions and internal event conditions to address the shortcomings of Tropos goal model in specifying software variations. The Contextual Preference Model approach, on the other hand, allows modelers to specify contextual preferences of stakeholders.

URN makes use of key performance indicators (KPI) and beliefs for the representation of context, whereas the AwReqs approach introduces variation points and control variables that carry resemblance to KPIs to their models. In addition to their use of KPIs, IRIS uses big queries, big analytics, and big data to represent contextual information. Context-aware Reasoning Using Goal-orientation (CARGO) also uses KPIs while Event-based Goal Adaptation uses events together with indicators.

Adaptive RML builds on the abstract early-requirements modeling language Techne [213] by adding the concepts of context and resource. While the Generic Quantitative Goal Model approach employs context-dependent weights, the Quality Constraint Templates allow modelers to define quality constraint templates for the explicit representation of contextual information.

Goal-based Contextual Problem Detection (GCPD) and SecuriTAS attempt to detect contextual changes via defining contextual factors while the RBUIS approach uses context elements that are monitored during execution. Another approach focusing on adaptive behavior, the self-adaptation framework for Tropos (Tropos4AS) uses goal conditions. Tropos with Control Model approach uses event-condition-action (ECA) rules, whereas Adaptation Goal Model makes use of adaptation goals and atomic contexts to represent context dependent information.

The Feature-Oriented NFRs and Context Feature Model approaches capture context with the help of feature models where Feature-Oriented NFRs take a step further to incorporate NFRs into the feature model. The Agon methodology makes use of user context models and context dimension trees to characterize intended users with context variables. Lastly, i*-Context extends i* with categorization of agent roles and role playing relations for the representation of context dependent information.

The remaining 16 approaches do not specify any means for modeling context dependent information. The fact that all of these 16 approaches appear in the result set of the search for goal model reuse (i.e., search string 1) reconfirms the need for the search for context and goal models (i.e., search string 2) and the lack of context representation in goal modeling approaches predominantly focusing on reuse.

4.2.5. Reuse interface

In order to fully satisfy the last requirement of providing an interface for reuse (R.5), an approach shall maintain the modularity and clear model boundaries even after reuse of the goal model. Therefore, loss of module boundaries upon reusing the goal model results in only partial satisfaction of this requirement as shown in Table 10.

As shown in Tables 3–5, none of the contextual goal modeling approaches except the five in common for both searches (i.e., Reusable Goal Models, Security Patterns, Context Transformations, Goal-Oriented Holonic Systems, and STREAM-A) focus on defining model boundaries and hence fail to satisfy this requirement.

The fifth requirement is partially satisfied by fifteen of the approaches (GRL Catalogues, i* Modules, AoGRL, Goal Aspects, Aspects with V-graph, AspectKAOS, Task Knowledge Patterns, Patterns in i*, Attack Patterns, Patterns in NFR, GOPCSD, KAOS Templates and Patterns, Goal-Oriented Interaction Diagrams, Security Patterns, and Context Transformations), because, even though they provide the necessary module boundaries (e.g., actor, pointcut and advice, template, etc.), modularization and hence the reuse hierarchy is lost upon composing the models and it is not possible for the user of the reusable artifact to identify model elements at different levels after that point.

Overall, seven approaches fully satisfy the fifth requirement. STREAM-A groups i* model elements into actors so that the actor can be reused in another domain, and hence maintains the boundaries of an actor. Two approaches to create modular i* and Tropos models with the help of Aspect Oriented Software Development, Aspects with i* and Aspects with Tropos, introduce *aspects* and*aspectual elements* for this modularization. Similar to the use of actors, intentional elements are contained in these aspects and aspectual elements that can be composed through crosscut relationships. As previously mentioned for R.1, goal templates used in the GoPF approach maintain clear model boundaries after they are reused and hence this approach satisfies R.5. The composition mechanism used in the Variability in Goal Models approach keeps the modules separate and uses composition labels to attach models. Holons introduced by Goal-Oriented Holonic Systems as their unit of encapsulation ensure clear boundaries between modules even after composition. The Reusable Goal Models approach satisfies this requirement as it provides a clearly defined reuse interface with the necessary metamodel extensions, and the model boundaries are maintained after reuse.

4.2.6. Tool support

The last columns of Tables 2–5 address research question Q.4 from Section 3 and provide some evidence that tool support is considered important in the modeling community as 25 out of 57 approaches provide tool support and a further 22 out of 57 state it as their future work.

4.3. Future research themes

The approach that scored the highest in the assessment performed in Section 4.2 is Reusable Goal Models which received full scores for R.1 to R.3 and R.5 as well as partial scores for R.4. Hence, the support for the explicit capturing and evaluation of context-dependent information is not as strong as it could be. As is evident from Tables 3–5, many different approaches exist that provide such support for goal models and it is not yet clear which approach is best combined with Reusable Goal Models to further improve support for reuse of context-oriented goal models.

The results of the systematic literature review point to five major research themes to enhance the capabilities of existing goal modeling approaches with more advanced support for reuse and context dependent information. The future research themes align with the five requirements from Section 2 and the results of the systematic literature review for these requirements (i.e., the five R.n columns in Tables 2–5).

Theme 1: constraint handling framework

The R.3 columns in Tables 2–5 show that very little support is available to take into account additional constraints imposed by models that are connected to goal models but expressed with different modeling formalisms. Only 12 out of 57 approaches consider constraints imposed by other modeling formalisms on goal models and only one approach out of 57 actually takes these constraints into account during goal model evaluation. These constraints may significantly influence goal model evaluation as they may exclude otherwise permissible candidate solutions. The existing approach that takes such constraints into account is tailored to specific modeling formalisms (i.e., feature models [20] or workflow models [19]) and does not necessarily apply to the constraints imposed by yet another modeling formalism. Techniques have not yet been developed for other modeling formalisms connected with goal models (e.g., see organizational models, structural models, and state-based models in the "*Supplementary Models*" columns of Tables 2–5). Instead of developing individual solutions for each modeling formalism, a general framework should be developed that allows general constraints on goal models to be expressed and linked to modeling constructs in other modeling formalisms. Based on this specification, a general evaluation mechanism could then be developed, allowing different modeling formalisms to be flexibly combined with goal models while respecting imposed constraints.

Theme 2: reuse interface of a goal model

Based on the results in the R.5 columns in Tables 2–5, it seems that the goal modeling community has not yet converged on a generally accepted definition of what constitutes the reuse interface of a goal model. Only seven out of 57 approaches consider the definition of an explicit reuse interface to encapsulate internal goal model structures. The majority of these approaches favors a component-based approach (e.g., actors, aspects, or holons) (STREAM-A [142,143], Aspects with i* [85], Aspects with Tropos [84], Variability in Goal Models [101], and Goal-Oriented Holonic Systems [119–121]). A template-based approach is suggested by GoPF [92–95], while Reusable Goal Models [3,105,106,110,111] employ an element-based approach, i.e., a dedicated model element is used in the reusing model to identify elements from the reused model. However, a thorough comparison of these three techniques should be performed as it is not clear which of these approaches is the most effective for goal models. In addition, an analysis of existing, common goal model constructs should be performed to determine which constructs may be visible to a reusing party and which should not be visible. Furthermore, techniques should be developed that govern under which circumstances the visibility of a modeling element is allowed to change. E.g., what happens to the visibility of modeling elements of a reused goal model in the reusing model?

Theme 3: efficient evaluation across reuse hierarchy

As seen from the R.1 columns in Tables 2–5, while the majority of the 57 approaches reports on a reasoning mechanism for goal models, only Reusable Goal Models [19,20,105,106,110,111] uses reuse hierarchies to improve the performance of its reasoning mechanism. The approach of Reusable Goal Models is applicable to all propagation-based techniques. While its principles may be applied to other techniques, too, it is not clear to what extent this is feasible. Furthermore, goal model evaluations for the existing component-based and template-based approaches (see Theme 2) have not yet taken full advantage of the existence of reuse interfaces (in particular when constraints imposed by external models have to be taken into account). Hence, other reasoning techniques should be extended to work across a reuse hierarchy while taking external constraints into account and the results compared against Reusable Goal Models.

Theme 4: dealing with uncertainty

With the exception of Reusable Goal Models [110,111], none of the existing goal modeling approaches provides a complete mechanism to delay decisions in the reuse hierarchy, as indicated by the R.2 columns in Tables 2–5. Existing approaches go as far as enabling the specification of where and when decisions may be delayed or supporting the re-evaluation of goal models at runtime to reassess a possibly changed reuse context. However, delaying of decisions introduces an element of uncertainty or variability that needs to be considered when evaluating goal models to determine the best trade-off for stakeholders. E.g., instead of dealing with concrete results for a goal model evaluation, possible ranges of evaluation values may have to be considered. Reusable Goal Models [110,111] provide the ability to explicitly define elements that are uncertain and take this information into account when evaluating goal models by providing a range evaluation (i.e., the result of the evaluation is a range of possible outcomes instead of a single concrete value). The evaluation technique used by Reusable Goal Models [110,111] is applicable to any propagation-based technique, but may have to be adjusted significantly for other techniques or propagation-based techniques that do not use relative, quantitative values in goal models. As is evident from the 40 out of 57 approaches that partially satisfy R.2, a large number of techniques exist that describe means to delay decisions and enable representation of uncertain input values. Therefore, new evaluation mechanisms could be developed for these various goal modeling approaches investigated by this systematic literature review to address the evaluation of uncertain elements. Once additional techniques exist, a thorough comparison should be undertaken to evaluate the performance of the new evaluation mechanisms against the one from Reusable Goal Models [110,111].

Theme 5: A holistic goal modeling environment

The overall challenge is to provide a holistic goal modeling environment that supports all five requirements stated in Section 2. Various goal modeling approaches address individual requirements. Many different goal modeling approaches support context dependent information as indicated in the R.4 columns in Tables 2–5, using a variety of modeling constructs to express context as shown in the "*Context Representation*" columns in Tables 2–5. Despite the close relationship between reuse and context, there exists very little overlap between the two concepts (as indicated by only five out of 57 approaches that were classified as an approach in both result sets). Therefore, it is not clear which of the modeling constructs to express context is the most appropriate for reuse hierarchies, for which some support exists as shown in the R.1 columns in Tables 2–5. For example, Reusable Goal Models [3,108,110,111] only allow features in a goal model to be included or excluded based on context, but do not allow contributions or other model elements to vary from one context to the next. Hence, each proposed modeling construct for context should be assessed in terms of how it could fully support reuse hierarchies, a reuse interface, and constraints imposed by other modeling formalisms. Based on this assessment, new concepts may have to be added to existing goal modeling approaches and new evaluation mechanisms may have to be developed to support contextual and reusable goal models.

5. Conclusion

This work first identifies and discusses five required characteristics for supporting reusability in goal models with context dependent information and then performs a two-part systematic literature review to investigate the current landscape of such goal modeling approaches. The review was performed in two rounds, the first covering publications until mid to late 2016 and the second covering publications available as of September 2018.

The required characteristics are (i) to enable analysis and validation of goal models through reuse hierarchies, (ii) to provide the means to delay decision making to a later point in the reuse hierarchy, (iii) to take constraints imposed by other modeling notations into account during goal model analysis, (iv) to allow context dependent information to be modeled so that the goal model can be used in various reuse contexts, and (v) to provide an interface for reuse.

The two-part systematic literature review covers both reuse and context in goal models. Research questions and inclusion criteria are specified, as well as threats to validity are discussed. Each step of the review is documented, and all results are tracked with the help of a reference management tool. While the initial publication list retrieved from seven major academic search engines for both review rounds combined included more than 4300 unique hits, a final list of 146 publications and 34 comparisons/assessments of goal modeling approaches are discussed in more detail.

Based on the examination of these publications, five major research themes are derived to support reusability in goal models with context dependent information: (i) development of a general framework for constraints imposed on goal models by other models, (ii) definition of a generally accepted reuse interface for goal models, (iii) development of efficient reasoning mechanisms across the whole reuse hierarchy, (iv) development of mechanisms in goal models to support delaying of decision and uncertain elements in reuse hierarchies and their evaluation, and (v) development of a holistic goal modeling environment to support all five characteristics of reusable and contextual goal models.

Acknowledgments

Supported by NSERC Canada Discovery Grants.

References

[1] C.W. Krueger, Software reuse, ACM Comput. Surv. 24 (2) (1992) 131–183, doi:10.1145/130844.130856.
[2] NATO, Standard for the Development of Reusable Software Components, Technical Report, GTE Gov-t Systems, Chantilly, VA, USA, 1994.
[3] J. Kienzle, G. Mussbacher, O. Alam, M. Schöttle, N. Belloir, P. Collet, B. Combemale, J. DeAntoni, J. Klein, B. Rumpe, VCU: the three dimensions of reuse, in: Proceedings of the 15th International Conference Software Reuse: Bridging with Social-Awareness, ICSR 2016, Limassol, Cyprus, 2016, pp. 122–137, doi:10.1007/978-3-319-35122-3_9. June 5–7, 2016.
[4] G. Mussbacher, J. Kienzle, A vision for generic concern-oriented requirements reuse re@21, in: Proceedings of the 21st IEEE International Requirements Engineering Conference, RE 2013, Rio de Janeiro-RJ, Brazil, IEEE Computer Society, 2013, pp. 238–249, doi:10.1109/RE.2013.6636724. July 15–19, 2013.
[5] O. Alam, J. Kienzle, G. Mussbacher, Concern-oriented software design, in: Proceedings of the 16th International Conference Model-Driven Engineering Languages and Systems MODELS 2013, Miami, FL, USA, 2013, pp. 604–621, doi:10.1007/978-3-642-41533-3_37.
[6] W. Weber, P. Metz, Reuse of models and diagrams of the uml and implementation concepts regarding dynamic modeling, in: M. Schader, A. Korthaus (Eds.), The Unified Modeling Language, Physica-Verlag HD, Heidelberg, 1998, pp. 190–203.
[7] V.R. Basili, L.C. Briand, W.L. Melo, How reuse influences productivity in object-oriented systems, Commun. ACM 39 (10) (1996) 104–116, doi:10.1145/236156.236184.
[8] W.C. Lim, Effects of reuse on quality, productivity, and economics, IEEE Softw. 11 (5) (1994) 23–30, doi:10.1109/52.311048.
[9] S. Katz, M. Mezini, J. Kienzle (Eds.), Transactions on aspect-oriented software development VII - a common case study for aspect-oriented modeling, Lecture Notes in Computer Science, Vol. 6210, Springer, 2010.
[10] K. Pohl, G. Böckle, F. van der Linden, Software Product Line engineering - Foundations, Principles, and Techniques, Springer, 2005.
[11] J. Pérez, J. Díaz, C.C. Soria, J. Garbajosa, Plastic partial components: A solution to support variability in architectural components, in: Proceedings of the WICSA/ECSA, IEEE, 2009, pp. 221–230.
[12] T. van der Storm, Variability and component composition, in: Proceedings of the ICSR, in: Lecture Notes in Computer Science, 3107, Springer, 2004, pp. 157–166.
[13] M.B. Duran, Reusable Goal Models, in: Proceedings of the IEEE 25th International Requirements Engineering Conference, IEEE, New York, 2017, pp. 532–537, doi:10.1109/RE.2017.34. WOS:000427159100074.
[14] A. Dardenne, A. van Lamsweerde, S. Fickas, Goal-directed requirements acquisition, Sci. Comput. Program. 20 (1–2) (1993) 3–50, doi:10.1016/0167-6423(93)90021-G.
[15] E.S.-K. Yu, Modelling Strategic Relationships for Process Reengineering, 1996 Ph.D. thesis. Toronto, Ont., Canada, Canada, UMI Order No. GAXNN-02887 (Canadian dissertation).
[16] L. Chung, B.A. Nixon, E. Yu, J. Mylopoulos, Non-Functional requirements in software engineering, International Series in Software Engineering, 5, Springer, 1999.
[17] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, J. Mylopoulos, Tropos: an agent-oriented software development methodology, Auton. Agents Multi Agent Syst. 8 (3) (2004) 203–236, doi:10.1023/B:AGNT.0000018806.20944.ef.

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

[19] M.B. Duran, G. Mussbacher, Investigation of feature run-time conflicts on goal model-based reuse, Inf. Syst. Front. (2016) 1–21, doi:10.1007/s10796-016-9657-7.

[20] M.B. Duran, G. Mussbacher, N. Thimmegowda, J. Kienzle, On the reuse of goal models., in: J. Fischer, M. Scheidgen, I. Schieferdecker, R. Reed (Eds.), Proceedings of the SDL Forum, Lecture Notes in Computer Science, 9369, Springer, 2015, pp. 141–158.

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

[22] M. Strohmaier, J. Horkoff, E. Yu, J. Aranda, S. Easterbrook, Can Patterns Improve i* Modeling? Two Exploratory Studies, in: B. Paech, C. Rolland (Eds.), Requirements Engineering: Foundation for Software Quality, Lecture Notes in Computer Science, Springer, 2008, pp. 153–167, doi:10.1007/978-3-540-69062-7_16. 5025.

[23] D. Amyot, S. Ghanavati, J. Horkoff, G. Mussbacher, L. Peyton, E. Yu, Evaluating goal models within the goal-oriented requirement language, Int. J. Intell. Syst. 25 (8) (2010) 841–877, doi:10.1002/int.v25:8.

[24] A. van Lamsweerde, Goal-oriented requirements engineering: A guided tour, in: Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE 2001), Toronto, Canada, 2001, p. 249, doi:10.1109/ISRE.2001.948567.

[25] J. Kienzle, G. Mussbacher, P. Collet, O. Alam, Delaying decisions in variable concern hierarchies, in: Proceedings of the ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences, in: GPCE 2016, ACM, New York, NY, USA, 2016, pp. 93–103, doi:10.1145/2993236.2993246.

[26] A. Palmieri, P. Collet, D. Amyot, Handling regulatory goal model families as software product lines, in: Proceedings of the 27th International Conference Advanced Information Systems Engineering CAiSE 2015, Stockholm, Sweden, 2015, pp. 181–196, doi:10.1007/978-3-319-19069-3_12.

[27] R. Ali, F. Dalpiaz, P. Giorgini, Contextual Goal Models, Technical Report, disi-10-020, University of Trento, 2010.

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

[29] B. Kitchenham, Procedures for Performing Systematic Reviews, Technical Report TR/SE-0401, Department of Computer Science, Keele University, UK, 2004.

[30] Roy Rosenzweig Center for History and New Media at George Mason University and the Corporation for Digital Scholarship, Zotero,. https://www.zotero.org/.

[31] K. Petersen, C. Gencel, Worldviews, research methods, and their relationship to validity in empirical software engineering research, in: Proceedings of the Joint Conference of the 23rd International Workshop on Software Measurement (IWSM) and the 8th International Conference on Software Process and Product Measurement, in: IWSM-MENSURA '13, IEEE Computer Society, Washington, DC, USA, 2013, pp. 81–89, doi:10.1109/IWSM-Mensura.2013.22.

[32] J. Maxwell, Understanding and validity in qualitative research, Harv. Edu. Rev. 62 (3) (1992) 279–301.

[33] B.B. Duarte, A.L. de Castro Leal, R. de Almeida Falbo, G. Guizzardi, R.S. Guizzardi, V.E.S. Souza, Ontological foundations for software requirements with a focus on requirements at runtime, Appl. Ontol. (Preprint) (2017) 1–33.

[34] M. Gharib, P. Giorgini, J. Mylopoulos, Ontologies for privacy requirements engineering: a systematic literature review, arXiv:1611.10097 (2016).

[35] J.C. Muñoz-Fernández, R. Mazo, C. Salinesi, G. Tamura, 10 challenges for the specification of self-adaptive software, in: Proceedings of the RCIS, IEEE, 2018, pp. 1–12.

[36] J. Zdravkovic, J. Stirna, J. Grabis, A comparative analysis of using the capability notion for congruent business and information systems engineering, Compl. Syst. Inf. Model. Quart. (10) (2017) 1–20.

[37] O. Daramola, Y. Pan, P. Karpati, G. Sindre, A comparative review of i*-based and use case-based security modelling initiatives, in: Proceedings of the Sixth International Conference on Research Challenges in Information Science (RCIS), IEEE, 2012, pp. 1–12.

[38] R. Matulevičius, P. Heymans, Comparing Goal Modelling Languages: an Experiment, in: P. Sawyer, B. Paech, P. Heymans (Eds.), Requirements Engineering: Foundation for Software Quality, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 18–32, doi:10.1007/978-3-540-73031-6_2. 4542.

[39] G. Poels, A. Maes, F. Gailly, R. Paemeleire, Measuring the Perceived Semantic Quality of Information Models, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 376–385. doi:10.1007/11568346_41.

[40] X. Franch, Fostering the Adoption of i* by practitioners: some challenges and research directions, in: S. Nurcan, C. Salinesi, C. Souveyet, J. Ralyté (Eds.), Intentional Perspectives on Information Systems Engineering, Springer, 2010, pp. 177–193, doi:10.1007/978-3-642-12544-7_10.

[41] H.S. Al-Subaie, T.S. Maibaum, Evaluating the effectiveness of a goal-oriented requirements engineering method, in: Proceedings of the Fourth International Workshop on Comparative Evaluation in Requirements Engineering CERE, IEEE, 2006, pp. 8–19.

[42] J. Stirna, A comparative analysis of concepts for capability design used in capability driven development and the nato architecture framework, in: A. Metzger, A. Persson (Eds.), Advanced Information Systems Engineering Workshops, Lecture Notes in Business Information Processing, Springer International Publishing, 2017, pp. 27–38.

[43] O. Akhigbe, D. Amyot, G. Richards, A systematic literature mapping of goal and non-goal modelling methods for legal and regulatory compliance, Requ. Eng. (2018), doi:10.1007/s00766-018-0294-1.

[44] D. Bombonatti, M. Goulão, A. Moreira, Synergies and tradeoffs in software reuse - a systematic mapping study, Softw., Pract. Exper. 47 (7) (2017) 943–957.

[45] A. Souag, R. Mazo, C. Salinesi, I. Comyn-Wattiau, Reusable knowledge in security requirements engineering: a systematic mapping study, Requ. Eng. 21 (2) (2016) 251–283, doi:10.1007/s00766-015-0220-8.

[46] S.T. Bulusu, R. Laborde, A.S. Wazan, F. Barrère, A. Benzekri, A requirements engineering-based approach for evaluating security requirements engineering methodologies, in: S. Latifi (Ed.), Information Technology - New Generations, Advances in Intelligent Systems and Computing, Springer International Publishing, 2018, pp. 517–525.

[47] J. Horkoff, F.B. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja, M. Salnitri, L. Piras, J. Mylopoulos, P. Giorgini, Goal-oriented requirements engineering: an extended systematic mapping study, Requ. Eng. (2017), doi:10.1007/s00766-017-0280-z.

[48] L.R. Soares, P.-Y. Schobbens, I. do Carmo Machado, E.S. de Almeida, Feature interaction in software product line engineering: a systematic mapping study, Inf. Softw. Technol. 98 (2018) 44–58, doi:10.1016/j.infsof.2018.01.016.

[49] M.L.d.J. Souza, A.R. Santos, I.D.C. Machado, E.S.d. Almeida, G.S.d.S. Gomes, Evaluating variability modeling techniques for dynamic software product lines: a controlled experiment, in: Proceedings of the X Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS), IEEE, 2016, pp. 1–10, doi:10.1109/SBCARS.2016.15.

[50] M. Szvetits, U. Zdun, Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime, Softw. Syst. Model. 15 (1) (2016) 31–69, doi:10.1007/s10270-013-0394-9.

[51] R. Tõnisson, R. Matulevicius, A coarse-grained comparison of modelling languages for business motivation and intentional distribution, in: Proceedings of the BIR, in: Lecture Notes in Business Information Processing, 261, Springer, 2016, pp. 80–95.

[52] E. Paja, A. Maté, C. Woo, J. Mylopoulos, Can goal reasoning techniques be used for strategic decision-making?, in: I. Comyn-Wattiau, K. Tanaka, I.Y. Song, S. Yamamoto, M. Saeki (Eds.), Conceptual Modeling. ER 2016. Lecture Notes in Computer Science, 9974, Springer, Cham, 2016.

[53] L. Kolos-Mazuryk, P. Eck, R. Wieringa, A Survey of Requirements Engineering Methods for Pervasive Services, Freeband A-MUSE Deliverable D5.7, 2006.

[54] A.I. Anton, Goal Identification and Refinement in the Specification of Software-based Information Systems, 1997. Ph.D. thesis, Atlanta, GA, USA, UMI Order No. GAX97-35409.

[55] R. Darimont, E. Delor, P. Massonet, A. van Lamsweerde, GRAIL/KAOS: an environment for goal-driven requirements engineering, in: W.R. Adrion, A. Fuggetta, R.N. Taylor, A.I. Wasserman (Eds.), Proceedings of the 19th International Conference on Software Engineering Pulling Together, Boston, Massachusetts, USA, ACM, 1997, pp. 612–613, doi:10.1145/253228.253499.

[56] J. Mylopoulos, L. Chung, E. Yu, From object-oriented to goal-oriented requirements analysis, Commun. ACM 42 (1) (1999) 31–37, doi:10.1145/291469.293165.

[57] E. Gonçalves, J. Castro, J. Araújo, T. Heineck, A systematic literature review of Istar extensions, J. Syst. Softw. 137 (2018) 1–33.

[58] E. Gonçalves, M. Oliveira, I. Monteiro, J. Castro, J. Araújo, Understanding what is important in Istar extension proposals: the viewpoint of researchers, Req. Eng. (2018), doi:10.1007/s00766-018-0302-5.

[59] U. Alegre, J.C. Augusto, T. Clark, Engineering context-aware systems and applications: a survey, J. Syst. Softw. 117 (2016) 55–83, doi:10.1016/j.jss.2016.02.010.

[60] M. Soares, J. Vilela, G. Guedes, C. Silva, J. Castro, Core ontology to aid the goal oriented specification for self-adaptive systems, in: Proceedings of the New Advances in Information Systems and Technologies, Springer, 2016, pp. 609–618.

[61] M. Morandini, L. Penserini, A. Perini, Towards goal-oriented development of self-adaptive systems, in: Proceedings of the International Workshop on Software Engineering for Adaptive and Self-managing Systems, in: SEAMS '08, ACM, New York, NY, USA, 2008, pp. 9–16, doi:10.1145/1370018.1370021.

[62] N.A. Qureshi, I.J. Jureta, A. Perini, Towards a Requirements Modeling Language for Self-Adaptive Systems, Springer Berlin Heidelberg, 2012, pp. 263–279, doi:10.1007/978-3-642-28714-5_24.

[63] J.a. Pimentel, J. Castro, J. Mylopoulos, K. Angelopoulos, V.E.S. Souza, From requirements to statecharts via design refinement, in: Proceedings of the 29th Annual ACM Symposium on Applied Computing, in: SAC '14, ACM, New York, NY, USA, 2014, pp. 995–1000, doi:10.1145/2554850.2555056.

[64] C. Krupitzer, M. Breitbach, F.M. Roth, S. VanSyckel, G. Schiele, C. Becker, A survey on engineering approaches for self-adaptive systems (extended version), (2018).

[65] Z. Yang, Z. Li, Z. Jin, A thematic study of requirements modeling and analysis for self-adaptive systems, CoRR, 2017. http://arxiv.org/abs/1704.00420.

[66] Z. Yang, Z. Li, Z. Jin, H. Zhang, Review on requirements modeling and analysis for self-adaptive systems: a ten-year perspective, arXiv:1704.00421 (2017b).

[67] G. Guedes, C. Silva, M. Soares, J. Castro, Variability Management in Dynamic Software Product Lines: A systematic mapping, in: Proceedings of the IX Brazilian Symposium on Components, Architectures and Reuse Software (SBCARS),, IEEE, 2015, pp. 90–99.

[68] M.L. de Jesus Souza, A.R. Santos, E.S. de Almeida, Towards the selection of modeling techniques for dynamic software product lines, in: Proceedings of the Fifth International Workshop on Product LinE Approaches in Software Engineering, IEEE Press, 2015, pp. 19–22.

[69] R. Ali, R. Chitchyan, P. Giorgini, Context for goal-level product line derivation, in: Proceedings of the 3rd International Workshop on Dynamic Software Product Lines (DSPL09), 2009.

[70] M.L. de Jesus Souza, A.R. Santos, I. do Carmo Machado, E.S. de Almeida, G.S. da Silva Gomes, Evaluating variability modeling techniques for dynamic software product lines: A controlled experiment, in: Proceedings of the X Brazilian Symposium on Software Components, Architectures and Reuse, SBCARS Maringá, Brazil, 2016, pp. 1–10, doi:10.1109/SBCARS.2016.15.

[71] K. Saller, M. Lochau, I. Reimund, Context-aware dspls: Model-based runtime adaptation for resource-constrained systems, in: Proceedings of the 17th International Software Product Line Conference Co-located Workshops, in: SPLC '13 Workshops, ACM, New York, NY, USA, 2013, pp. 106–113, doi:10.1145/2499777.2500716.

[72] D.B. Abeywickrama, E. Ovaska, A survey of autonomic computing methods in digital service ecosystems, Serv. Oriented Comput. Appl. 11 (1) (2017) 1–31, doi:10.1007/s11761-016-0203-8.
[73] C. Fastnacht, H. Koç, D. Nesterenko, K. Sandkuhl, Comparison of tool support for goal modelling in capability management, in: Proceedings of the CAiSE Workshops, in: Lecture Notes in Business Information Processing, 249, Springer, 2016, pp. 29–39.
[74] M. Salama, R. Bahsoon, N. Bencomo, Chapter 11 - managing trade-offs in self-adaptive software architectures: a systematic mapping study, in: I. Mistrik, N. Ali, R. Kazman, J. Grundy, B. Schmerl (Eds.), Managing Trade-Offs in Adaptable Software Architectures, Morgan Kaufmann, Boston, 2017, pp. 249–297, doi:10.1016/B978-0-12-802855-1.00011-3.
[75] D. Weyns, Software engineering of self-adaptive systems: an organised tour and future challenges, Chapter in Handbook of Software Engineering, 2017.
[76] D. Colomer, J. Franch Gutiérrez, i* modules: a jUCMNav implementation, in: Proceedings of the iStar 5th International i* Workshop: 29-30th Trento, Italy, CEUR Workshop Proceedings, 2011, pp. 178–180.
[77] G. Mussbacher, D. Amyot, J. Araújo, A. Moreira, M. Weiss, Visualizing aspect-oriented goal models with AoGRL, in: Proceedings of the Second International Workshop on Requirements Engineering Visualization REV, IEEE, 2007, doi:10.1109/REV.2007.11. 1–1.
[78] G. Mussbacher, Aspect-Oriented User Requirements Notation: Aspects in Goal and Scenario Models, in: H. Giese (Ed.), Models in Software Engineering, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 305–316, doi:10.1007/978-3-540-69073-3_32. 5002.
[79] G. Mussbacher, D. Amyot, Extending the user requirements notation with aspect-oriented concepts, in: R. Reed, A. Bilgic, R. Gotzhein (Eds.), SDL 2009: Design for Motes and Mobiles, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 115–132, doi:10.1007/978-3-642-04554-7_8.
[80] G. Mussbacher, D. Amyot, J. Araújo, A. Moreira, Modeling Software Product Lines with AoURN, in: Proceedings of the AOSD Workshop on Early Aspects, in: EA '08, ACM, New York, NY, USA, 2008, pp. 2:1–2:8, doi:10.1145/1404946.1404948.
[81] G. Mussbacher, D. Amyot, J. Whittle, Composing goal and scenario models with the aspect-oriented user requirements notation based on syntax and semantics, in: A. Moreira, R. Chitchyan, J. Araújo, A. Rashid (Eds.), Aspect-Oriented Requirements Engineering, Springer, 2013, pp. 77–99. 10.1007/978-3-642-38640-4_5.
[82] J.C.S.d.P. Leite, Y. Yu, L. Liu, E.S.K. Yu, J. Mylopoulos, Quality-based software reuse, in: O. Pastor, J.F.e. Cunha (Eds.), Advanced Information Systems Engineering, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 535–550, doi:10.1007/11431855_37. 3520.
[83] N. Niu, Y. Yu, B. González-Baixauli, N. Ernst, J.C.S.d.P. Leite, J. Mylopoulos, Aspects across software life cycle: a goal-driven approach, in: S. Katz, H. Ossher, R. France, J.-M. Jézéquel (Eds.), Transactions on Aspect-Oriented Software Development VI, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 83–110, doi:10.1007/978-3-642-03764-1_3.
[84] J. Castro, M. Kolp, L. Liu, A. Perini, Dealing with complexity using conceptual models based on Tropos, in: A.T. Borgida, V.K. Chaudhri, P. Giorgini, E.S. Yu (Eds.), Conceptual Modeling: Foundations and Applications, Lecture Notes in Computer Science, Springer, 2009, pp. 335–362, doi:10.1007/978-3-642-02463-4_18.
[85] F. Alencar, J. Castro, A. Moreira, J. Araújo, C. Silva, R. Ramos, J. Mylopoulos, Integration of aspects with I* models, in: M. Kolp, B. Henderson-Sellers, H. Mouratidis, A. Garcia, A.K. Ghose, P. Bresciani (Eds.), Agent-Oriented Information Systems IV, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 183–201, doi:10.1007/978-3-540-77990-2_11.
[86] L.F. da Silva, An aspect-oriented approach to model requirements, in: Proceedings of the IEEE International Conference on Requirements Engineering, Doctoral Consortium, 2005, p. 24.
[87] L.F.d. Silva, J.C.S.d.P. Leite, Aspect-Oriented Goal Modeling and Composition with AOV-Graph, in: A. Moreira, R. Chitchyan, J. Araújo, A. Rashid (Eds.), Aspect-Oriented Requirements Engineering, Springer Berlin Heidelberg, 2013, pp. 101–120, doi:10.1007/978-3-642-38640-4_6.
[88] A. Gil, J. Araujo, AspectKAOS: integrating early-aspects into KAOS, in: Proceedings of the 15th Workshop on Early Aspects, ACM, 2009, pp. 31–36.
[89] A.T.V. Gil, Integrating early aspects with goal-oriented requirements engineering, FCT-UNL, 2008 Ph.D. thesis.
[90] F. Semmak, C. Gnaho, R. Laleau, Extended kaos to support variability for goal oriented requirements reuse, in: Proceedings of the MoDISE-EUS, in: CEUR Workshop Proceedings, 341, CEUR-WS.org, 2008, pp. 22–33.
[91] F. Semmak, R. Laleau, C. Gnaho, Supporting variability in goal-based requirements, in: A. Flory, M. Collard (Eds.), Proceedings of the Third IEEE International Conference on Research Challenges in Information Science, RCIS Fès, Morocco, IEEE, 2009, pp. 237–246, doi:10.1109/RCIS.2009.5089287.
[92] S.A. Behnam, D. Amyot, G. Mussbacher, E. Braun, N. Cartwright, M. Saucier, Using the Goal-oriented pattern family framework for modelling outcome-based regulations, in: Proceedings of the IEEE 2nd International Workshop on Requirements Patterns (RePa), IEEE, 2012, pp. 35–40.
[93] S.A. Behnam, D. Amyot, Evolution mechanisms for goal-driven pattern families used in business process modelling, Int. J. Electron. Bus. 10 (3) (2013) 254–291.
[94] S.A. Behnam, D. Amyot, G. Mussbacher, Towards a Pattern-Based Framework for Goal-Driven Business Process Modeling, in: Proceedings of the Eighth ACIS International Conference on Software Engineering Research, Management and Applications (SERA), IEEE, 2010, pp. 137–145.
[95] S.A. Behnam, Goal-oriented Pattern Family Framework for Business Process Modeling, University of Ottawa, 2012 Ph.D. thesis.
[96] W. Cheah, L. Sterling, K. Taveter, Task Knowledge Patterns Reuse in Multi-Agent Systems Development, in: N. Desai, A. Liu, M. Winikoff (Eds.), Principles and Practice of Multi-Agent Systems, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 459–474, doi:10.1007/978-3-642-25920-3_33. 7057.
[97] L.A. Hermoye, A. Van Lamsweerde, D.E. Perry, Attack patterns for security requirements engineering, 2006.
[98] L. Chung, S. Supakkul, Capturing and reusing functional and non-functional requirements knowledge: a goal-object pattern approach, in: Proceedings of the IEEE International Conference on Information Reuse and Integration, IEEE, 2006, pp. 539–544.
[99] I.A.M. El-Maddah, T.S.E. Maibaum, Goal-oriented requirements analysis for process control systems design, in: Proceedings of the 1st ACM & IEEE International Conference on Formal Methods and Models for Co-Design (MEMOCODE 2003) Mont Saint-Michel, France, 2003, pp. 45–46, doi:10.1109/MEMCOD.2003.1210085.
[100] I.A.M. El-Maddah, T.S.E. Maibaum, Requirements-Reuse Using GOPCSD: component-based development of process control systems, in: J. Bosch, C. Krueger (Eds.), Software Reuse: Methods, Techniques, and Tools, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 318–328, doi:10.1007/978-3-540-27799-6_27.
[101] B. González-Baixauli, M.A. Laguna, J.C.S. do Prado Leite, Using goal-models to analyze variability., in: Proceedings of the VaMoS, 2007, pp. 101–107.
[102] R. Darimont, W. Zhao, C. Ponsard, A. Michot, Deploying a Template and Pattern Library for Improved Reuse of Requirements Across Projects, in: Proceedings of the IEEE 25th International Requirements Engineering Conference (RE), IEEE, 2017, pp. 456–457.
[103] L. Xiao, J. Fox, Goal Modelling in Clinical Decision Support, in: Proceedings of the IEEE 24th International Requirements Engineering Conference Workshops (REW), IEEE, 2016, pp. 135–144.
[104] L. Sabatucci, M. Cossentino, Self-adaptive smart spaces by proactive means-end reasoning, J. Reliable Intell. Environ. 3 (3) (2017) 159–175, doi:10.1007/s40860-017-0047-9.
[105] R. Alexandre, C. Camillieri, M.B. Duran, A.N. Pina, M. Schöttle, J. Kienzle, G. Mussbacher, Support for evaluation of impact models in reuse hierarchies with jUCMNav and TouchCORE, in: Proceedings of the 2nd MoDELS 2015 Demo and Poster Session co-located with ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2015), Ottawa, Canada, CEUR-WS.org, 2015.
[106] M.B. Duran, A.N. Pina, G. Mussbacher, Evaluation of reusable concern-oriented goal models, in: Proceedings of the IEEE International Model-Driven Requirements Engineering Workshop, MoDRE, Ottawa, ON, Canada, 2015, pp. 53–62, doi:10.1109/MoDRE.2015.7343876.
[107] C. Bensoussan, M. Schöttle, J. Kienzle, Associations in MDE: a concern-oriented, reusable solution, in: Modelling Foundations and Applications, in: Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 121–137.
[108] O. Alam, Concern oriented reuse: a software reuse paradigm, McGill University Libraries, 2016 Ph.D. thesis.
[109] B. Combemale, J. Kienzle, G. Mussbacher, O. Barais, E. Bousse, W. Cazzola, P. Collet, T. Degueule, R. Heinrich, J.-M. Jézéquel, M. Leduc, T. Mayerhofer, S. Mosser, M. Schöttle, M. Strittmatter, A. Wortmann, Concern-Oriented language development (COLD): fostering reuse in language engineering, Comput. Lang. Syst. Struct. 54 (2018) 139–155.
[110] M.B. Duran, G. Mussbacher, Evaluation of Goal Models in Reuse Hierarchies with Delayed Decisions, in: Proceedings of the IEEE 25th International Requirements Engineering Conference Workshops (REW), IEEE, 2017, pp. 6–15, doi:10.1109/REW.2017.66. WOS:000427148000002.
[111] M.B. Duran, G. Mussbacher, Top-down evaluation of reusable goal models, in: R. Capilla, B. Gallina, C. Cetina (Eds.), New Opportunities for Software Reuse, Springer International Publishing, Cham, 2018, pp. 76–92.
[112] T. Li, J. Horkoff, J. Mylopoulos, Integrating Security Patterns with Security Requirements Analysis Using Contextual Goal Models, Springer Berlin Heidelberg, 2014, pp. 208–223, doi:10.1007/978-3-662-45501-2_15.
[113] T. Li, J. Mylopoulos, Modeling and applying security patterns using contextual goal models, in: F. Dalpiaz, J. Horkoff (Eds.), Proceedings of the Seventh International i* Workshop co-located with the 26th International Conference on Advanced Information Systems Engineering (CAiSE 2014), Thessaloniki, Greece, CEUR Workshop Proceedings, 1157, CEUR-WS.org, 2014.
[114] T. Li, J. Horkoff, J. Mylopoulos, Analyzing and Enforcing Security Mechanisms on Requirements Specifications, Springer International Publishing, 2015, pp. 115–131, doi:10.1007/978-3-319-16101-3_8.
[115] T. Li, E. Paja, J. Mylopoulos, J. Horkoff, K. Beckers, Security attack analysis using attack patterns, in: Proceedings of the IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), IEEE, 2016, pp. 1–13.
[116] T. Li, Holistic Security Requirements Engineering for Socio-Technical Systems, University of Trento, 2016 Ph.D. thesis.
[117] T. Li, J. Horkoff, J. Mylopoulos, Holistic security requirements analysis for socio-technical systems, Softw. Syst. Model. 17 (4) (2018) 1253–1285, doi:10.1007/s10270-016-0560-y.
[118] P. Spoletini, A. Ferrari, S. Gnesi, Context transformations for goal models, in: Proceedings of the IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE), IEEE, 2014, pp. 17–26.
[119] A. Diaconescu, Goal-oriented holonic systems, in: C. Müller-Schloer, S. Tomforde (Eds.), Organic Computing – Technical Systems for Survival in the Real World, 2017, pp. 223–276.
[120] A. Diaconescu, S. Frey, C. Müller-Schloer, J. Pitt, S. Tomforde, Goal-oriented holonics for complex system (self-) integration: concepts and case studies, in: IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2016, pp. 100–109.

[121] A. Diaconescu, K.L. Bellman, L. Esterle, H. Giese, S. Götz, P.R. Lewis, A. Zisman, Architectures for collective self-aware computing systems, in: Self-Aware Computing Systems, Springer International Publishing, 2017, pp. 191–235.
[122] J. Stirna, J. Grabis, M. Henkel, J. Zdravkovic, Capability Driven Development – an Approach to Support Evolving Organizations, Springer Berlin Heidelberg, 2012, pp. 117–131, doi:10.1007/978-3-642-34549-4_9.
[123] S. Bērziša, G. Bravos, T.C. Gonzalez, U. Czubayko, S. España, J. Grabis, M. Henkel, L. Jokste, J. Kampars, H. Koç, J.-C. Kuhr, C. Llorca, P. Loucopoulos, R.J. Pascual, O. Pastor, K. Sandkuhl, H. Simic, J. Stirna, F.G. Valverde, J. Zdravkovic, Capability driven development: an approach to designing digital enterprises, in: Business & Information Systems Engineering, 57, Springer Fachmedien Wiesbaden, 2015, pp. 15–25, doi:10.1007/s12599-014-0362-0.
[124] H. Koç, J. Kuhr, K. Sandkuhl, F. Timm, Capability-driven development - a novel approach to design enterprise capabilities, in: E. El-Sheikh, A. Zimmermann, L.C. Jain (Eds.), Emerging Trends in the Evolution of Service-Oriented and Enterprise Architectures, Intelligent Systems Reference Library, 111, 2016, pp. 151–177, doi:10.1007/978-3-319-40564-3_9.
[125] J. Stirna, J. Zdravkovic, M. Henkel, P. Loucopoulos, C. Stratigaki, Modeling Organizational Capabilities on a Strategic Level, in: Proceedings of the IFIP Working Conference on The Practice of Enterprise Modeling, Springer, 2016, pp. 257–271.
[126] J. Stirna, J. Zdravkovic, Supporting perspectives of business capabilities by enterprise modeling, context, and patterns, in: Proceedings of the International Conference on Business Informatics Research, Springer, 2016, pp. 262–277.
[127] S. España, J. Grabis, M. Henkel, H. Koç, K. Sandkuhl, J. Stirna, J. Zdravkovic, Strategies for capability modelling: Analysis based on initial experiences, in: Proceedings of the International Workshops Advanced Information Systems Engineering Workshops - CAiSE Stockholm, Sweden, 2015, pp. 40–52, doi:10.1007/978-3-319-19243-7_4.
[128] J. Grabis, J. Kampars, Capability Management in the Cloud, in: Capability Management in Digital Enterprises, Springer, 2018, pp. 175–188.
[129] J. Stirna, J. Zdravkovic, Development of a modeling language for capability driven development: experiences from meta-modeling, in: I. Comyn-Wattiau, K. Tanaka, I.-Y. Song, S. Yamamoto, M. Saeki (Eds.), Conceptual Modeling, Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 396–403.
[130] J. Stirna, J. Zdravkovic, J. Grabis, K. Sandkuhl, Development of capability driven development methodology: experiences and recommendations, in: G. Poels, F. Gailly, E. Serral Asensio, M. Snoeck (Eds.), The Practice of Enterprise Modeling, Lecture Notes in Business Information Processing, Springer International Publishing, 2017, pp. 251–266.
[131] J. Grabis, J. Zdravkovic, J. Stirna, Overview of capability-driven development methodology, in: K. Sandkuhl, J. Stirna (Eds.), Capability Management in Digital Enterprises, Springer International Publishing, Cham, 2018, pp. 59–84, doi:10.1007/978-3-319-90424-5_4.
[132] K. Sandkuhl, J. Stirna, Organizational adoption of capability management, in: K. Sandkuhl, J. Stirna (Eds.), Capability Management in Digital Enterprises, Springer International Publishing, Cham, 2018, pp. 209–230, doi:10.1007/978-3-319-90424-5_12.
[133] K. Sandkuhl, J. Stirna, Capability Thinking, in: K. Sandkuhl, J. Stirna (Eds.), Capability Management in Digital Enterprises, Springer International Publishing, Cham, 2018, pp. 1–24, doi:10.1007/978-3-319-90424-5_1.
[134] K. Sandkuhl, Integrating local and global optimization in capability delivery, in: G. Poels, F. Gailly, E. Serral Asensio, M. Snoeck (Eds.), The Practice of Enterprise Modeling, Lecture Notes in Business Information Processing, Springer International Publishing, 2017, pp. 341–351.
[135] H. Koç, K. Sandkuhl, J. Stirna, J. Kuhr, Capability as a service: method and tool support for context-aware business services, IJISSS 10 (3) (2018) 64–84, doi:10.4018/IJISSS.2018070104.
[136] H. Koç, K. Sandkuhl, Capability-driven digital service innovation: Implications from business model and service process perspectives, in: Proceedings of the Practice of Enterprise Modeling - 10th IFIP WG 8.1. Working Conference, PoEM 2017, Leuven, Belgium, 2017, pp. 126–140, doi:10.1007/978-3-319-70241-4_9.
[137] H. Koç, M. Ruiz, S. España, Lightcdd: A lightweight capability-driven development method for start-ups, in: Proceedings of the Advanced Information Systems Engineering Workshops - CAiSE International Workshops, Ljubljana, Slovenia, 2016, pp. 15–26, doi:10.1007/978-3-319-39564-7_2.
[138] H. Koç, M. Ruiz, S. España, Lightcdd: application of a capability-driven development method for start-ups development, CSIMQ 10 (2017) 53–74, doi:10.7250/csimq.2017-10.04.
[139] M. Henkel, C. Stratigaki, J. Stirna, P. Loucopoulos, Y. Zorgios, A. Migiakis, Extending capabilities with context awareness, in: J. Krogstie, H. Mouratidis, J. Su (Eds.), Advanced Information Systems Engineering Workshops, Lecture Notes in Business Information Processing, Springer International Publishing, 2016, pp. 40–51.
[140] J. Grabis, J. Kampars, Adjustment of Capabilities: How to Add Dynamics, in: K. Sandkuhl, J. Stirna (Eds.), Capability Management in Digital Enterprises, Springer International Publishing, Cham, 2018, pp. 139–158, doi:10.1007/978-3-319-90424-5_8.
[141] S. España, H. Koç, M. Ruiz, O. Pastor, Capability support for entrepreneurial ventures, in: K. Sandkuhl, J. Stirna (Eds.), Capability Management in Digital Enterprises, Springer, Cham, 2018, pp. 311–325, doi:10.1007/978-3-319-90424-5_16.
[142] J. Castro, M. Lucena, C. Silva, F. Alencar, E. Santos, J. Pimentel, Changing attitudes towards the generation of architectural models, J. Syst. Softw. 85 (3) (2012) 463–479, doi:10.1016/j.jss.2011.05.047.
[143] J. Pimentel, M. Lucena, J. Castro, C. Silva, E. Santos, F. Alencar, Deriving software architectural models from requirements models for adaptive systems: the STREAM-a Approach, in: Requirements Engineering, 17, Springer-Verlag, 2011, pp. 259–281, doi:10.1007/s00766-011-0126-z.
[144] P.K. Murukannaiah, M.P. Singh, Xipho: Extending Tropos to engineer context-aware personal agents, in: Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 309–316.
[145] P.K. Murukannaiah, Engineering Personal Agents: Toward Personalized, Context-Aware, and Privacy-Preserving Applications., 2016 Ph.D. thesis.
[146] R. Ali, F. Dalpiaz, P. Giorgini, A Goal Modeling Framework for Self-contextualizable Software, in: Enterprise, Business-Process and Information Systems Modeling, Springer, 2009, pp. 326–338.
[147] R. Ali, F. Dalpiaz, P. Giorgini, Goal-based self-contextualization, in: E.S.K. Yu, J. Eder, C. Rolland (Eds.), Proceedings of the Forum at the CAiSE Conference, Amsterdam, The Netherlands, CEUR Workshop Proceedings, 453, CEUR-WS.org, 2009.
[148] R. Ali, F. Dalpiaz, P. Giorgini, A goal-based framework for contextual requirements modeling and analysis, Requ. Eng. 15 (2010) 439–458, doi:10.1007/s00766-010-0110-z.
[149] R. Ali, F. Dalpiaz, P. Giorgini, Reasoning about Contextual Requirements for Mobile Information Systems: a Goal-based Approach, Technical Report, disi-10-029, University of Trento, 2010.
[150] R. Ali, F. Dalpiaz, P. Giorgini, Reasoning with contextual requirements: detecting inconsistency and conflicts, Inf. Softw. Technol. 55 (1) (2013) 35–57, doi:10.1016/j.infsof.2012.06.013.
[151] R. Ali, P. Giorgini, Requirements-driven deployment, Softw. Syst. Model. 13 (2014) 433–456, doi:10.1007/s10270-012-0255-y.
[152] R. Ali, F. Dalpiaz, P. Giorgini, V.E.S. Souza, Requirements evolution: from assumptions to reality, in: Enterprise, Business-Process and Information Systems Modeling, Springer, 2011, pp. 372–382, doi:10.1007/978-3-642-21759-3_27.
[153] R. Ali, A. Franzen, A. Griggio, P. Giorgini, Modeling and Analyzing Contextual Requirements, Technical Report, disi-09-019, University of Trento, 2009.
[154] R. Ali, Modeling and reasoning about contextual requirements: Goal-based framework, University of Trento, 2010 Ph.D. thesis.
[155] R. Ali, Y. Yu, R. Chitchyan, A. Nhlabatsi, P. Giorgini, Towards a unified framework for contextual variability in requirements, in: Proceedings of the Third International Workshop on Software Product Management (IWSPM), IEEE, 2009, pp. 31–34.
[156] L. Baresi, L. Pasquale, Live goals for adaptive service compositions, in: Proceedings of the ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2010, pp. 114–123.
[157] L. Baresi, L. Pasquale, Adaptation goals for adaptive service-oriented architectures, in: P. Avgeriou, J. Grundy, J.G. Hall, P. Lago, I. Mistrík (Eds.), Relating Software Requirements and Architectures, Springer Berlin Heidelberg, 2011, pp. 161–181, doi:10.1007/978-3-642-21001-3_10.
[158] L. Baresi, L. Pasquale, P. Spoletini, Fuzzy goals for requirements-driven adaptation, in: Proceedings of the 18th IEEE International Requirements Engineering Conference, IEEE, 2010, pp. 125–134.
[159] N. Bencomo, Requirements for self-adaptation, in: Generative and Transformational Techniques in Software Engineering IV, International Summer School, GTTSE 2011, Braga, Portugal Revised Papers, 2011, pp. 271–296, doi:10.1007/978-3-642-35992-7_7.
[160] N. Bencomo, K. Welsh, P. Sawyer, J. Whittle, Self-explanation in adaptive systems, in: Proceedings of the 17th International Conference on Engineering of Complex Computer Systems (ICECCS), IEEE, 2012, pp. 157–166.
[161] K. Welsh, N. Bencomo, P. Sawyer, J. Whittle, Self-explanation in adaptive systems based on runtime goal-based models, Trans. Comput. Collect. Intell. 16 (2014) 122–145, doi:10.1007/978-3-662-44871-7_5.
[162] G. Chatzikonstantinou, K. Kontogiannis, Model contextual variability for agents using goals and commitments, in: Proceedings of the 6th International *i** Workshop Valencia, Spain, 2013, pp. 103–108.
[163] F. Guimaraes, G. Rodrigues, D. Batista, R. Ali, Pragmatic requirements for adaptive systems: a goal-driven modeling and analysis approach 9381 (2015) 50–64, doi:10.1007/978-3-319-25264-3_4.
[164] F.P. Guimarães, G.N. Rodrigues, R. Ali, D.M. Batista, Planning runtime software adaptation through pragmatic goal model, Data Knowl. Eng. 109 (2017) 25–40, doi:10.1016/j.datak.2017.03.003.
[165] A. Lapouchnian, J. Mylopoulos, Modeling Domain Variability in Requirements Engineering with Contexts, Springer, 2009, pp. 115–130, doi:10.1007/978-3-642-04840-1_11.
[166] A. Lapouchnian, J. Mylopoulos, Capturing contextual variability in *i** models, in: J.B. de Castro, X. Franch, J. Mylopoulos, E.S.K. Yu (Eds.), Proceedings of the 5th International *i** Workshop 2011, Trento, Italy, CEUR Workshop Proceedings, 766, CEUR-WS.org, 2011, pp. 96–101.
[167] A. Lapouchnian, Exploiting Requirements Variability for Software Customization and Adaptation, University of Toronto, 2011 Ph.D. thesis.
[168] Souza, Mylopoulos, Monitoring and diagnosing malicious attacks with autonomic software, in: Proceedings of the International Conference on Conceptual Modeling, 5829 LNCS, 2009, pp. 84–98, doi:10.1007/978-3-642-04840-1_9.
[169] A. Lapouchnian, E.S.K. Yu, Exploring context sensing in the goal-driven design of business processes, in: Proceedings of the 18th IEEE Conference on Business Informatics, 29th CBI Paris, France, Volume 1 - Conference Papers, 2016, pp. 45–54, doi:10.1109/CBI.2016.14.
[170] Y. Lei, K. Ben, Z. He, A framework for self-adaptive software based on extended tropos goal model, in: Proceedings of the 7th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015, 2, IEEE, 2015, pp. 533–536.

[171] Y. Lei, K. Ben, Z. He, A model driven agent-oriented self-adaptive software development method, in: Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), IEEE, 2015, pp. 2242–2246.
[172] W. Liu, Z. Feng, Context-based requirement modeling for self-adaptive service software, Binary Inf. Press (2012).
[173] W. Liu, Z. Feng, Requirement uncertainty analysis for service-oriented self-adaptation software, in: Network Computing and Information Security, Springer, 2012, pp. 156–163, doi:10.1007/978-3-642-35211-9_20.
[174] W. Liu, C.W. He, Z.W. Feng, Requirement uncertainty modeling for service oriented self-adaptive software, in: Advanced Materials Research, 433, Trans Tech Publ, 2012, pp. 4798–4801.
[175] D. Amyot, G. Mussbacher, Development of Telecommunications Standards and Services with the User Requirements Notation, in: Proceedings of the Workshop on ITU System Design Languages, 2008, pp. 15–16.
[176] A. Pourshahid, L. Peyton, S. Ghanavati, D. Amyot, P. Chen, M. Weiss, Model-based validation of business processes, Bus. Process Mgmt. Concepts, Tech. Appl. (2012) 165–183.
[177] A. Pourshahid, A Framework for Monitoring and Adapting Business Processes Using Aspect-Oriented URN, Université d'Ottawa/University of Ottawa, 2014 Ph.D. thesis.
[178] T. Zhao, H. Zhao, W. Zhang, Z. Jin, User preference based autonomic generation of self-adaptive rules, in: Proceedings of the 6th Asia-Pacific Symposium on Internetware on Internetware, ACM, 2014, pp. 25–34.
[179] N.A. Qureshi, I. Jureta, A. Perini, Adaptive RML: A Requirements Modeling Language for Self-Adaptive Systems, Technical Report, 2011.
[180] N.A. Qureshi, Requirements Engineering for Self-Adaptive Software: Bridging the Gap between Design-Time and Run-Time, University of Trento, 2011 Ph.D. thesis.
[181] V.E.S. Souza, A. Lapouchnian, J. Mylopoulos, System Identification for Adaptive Software Systems: A Requirements Engineering Perspective, Springer Berlin Heidelberg, 2011, pp. 346–361, doi:10.1007/978-3-642-24606-7_26.
[182] V.E. Silva Souza, Requirements-based software system adaptation, University of Trento, 2012 Ph.D. thesis.
[183] V. Klös, T. Göthel, S. Glesner, Runtime management and quantitative evaluation of changing system goals in complex autonomous systems, J. Syst. Softw. 144 (2018) 314–327.
[184] K.A. Botangen, J. Yu, S. Yongchareon, L.H. Yang, Q. Bai, Specifying and reasoning about contextual preferences in the goal-oriented requirements modelling, in: Proceedings of the Australasian Computer Science Week Multiconference, in: ACSW '18, ACM, New York, NY, USA, 2018, pp. 47:1–47:10, doi:10.1145/3167918.3167945.
[185] A. Vialon, K. Tei, S. Aknine, Soft-Goal Approximation Context Awareness of Goal-Driven Self-Adaptive Systems, in: Proceedings of the IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2017, pp. 233–238, doi:10.1109/ICAC.2017.25.
[186] J. Sun Kim, S. Park, V. Sugumaran, Contextual problem detection and management during software execution in complex environments, Ind. Mgmt. Data Syst. 106 (4) (2006) 540–561.
[187] M. Vrbaski, G. Mussbacher, D. Petriu, D. Amyot, Goal Models As Run-time Entities in Context-aware Systems, in: Proceedings of the 7th Workshop on Models@Run.Time, in: MRT '12, ACM, New York, NY, USA, 2012, pp. 3–8, doi:10.1145/2422518.2422520.
[188] M. Vrbaski, D. Petriu, D. Amyot, Tool support for combined rule-based and goal-based reasoning in Context-Aware systems, in: Proceedings of the 20th IEEE International Requirements Engineering Conference (RE), IEEE, 2012, pp. 335–336.
[189] M. Vrbaski, Domain Independent Context Awareness Framework, Carleton University Ottawa, 2012 Ph.D. thesis.
[190] Z. Xu, H. Zhao, L. Liu, User's Requirements Driven Services Adaptation and Evolution, in: Proceedings of the IEEE 36th Annual Computer Software and Applications Conference Workshops (COMPSACW), IEEE, 2012, pp. 13–19.
[191] Z. Yang, Z. Jin, Modeling and Specifying Parametric Adaptation Mechanism for Self-adaptive Systems, in: Requirements Engineering, Springer, 2014, pp. 105–119.
[192] Z. Yang, Z. Jin, Z. Li, A model-based fuzzy control approach to achieving adaptation with contextual uncertainties, CoRR, 2017. http://arxiv.org/abs/1704.00417.
[193] Z. Yang, Z. Jin, Z. Li, Achieving adaptation for adaptive systems via runtime verification: A model-driven approach, CoRR, 2017. http://arxiv.org/abs/1704.00869.
[194] Z. Yang, Z. Jin, Z. Li, Modeling uncertainty and evolving self-adaptive software: a fuzzy theory based requirements engineering approach, arXiv:1704.00873 (2017c).
[195] C. Menghi, Contextual, requirements driven, adaptive access control, Politecnico di Milano, 2012 Ph.D. thesis.
[196] L. Pasquale, C. Menghi, M. Salehie, L. Cavallaro, I. Omoronyia, B. Nuseibeh, Securitas: a tool for engineering adaptive security, in: Proceedings of the 20th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-20), SIGSOFT/FSE'12, Cary, NC, USA, 2012, p. 19, doi:10.1145/2393596.2393618.
[197] L. Dounas, R. Mazo, C. Salinesi, O. El Beqqali, Continuous monitoring of adaptive e-learning systems requirements, in: Proceedings of the 12th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2015), 2015.
[198] L. Dounas, R. Mazo, C. Salinesi, O. El Beqqali, Runtime requirements monitoring framework for adaptive e-learning systems, in: Proceedings of the International Conference on Software & Systems Engineering and their Applications (ICSSEA'15), 2015.
[199] P. Akiki, Engineering adaptive model-driven user interfaces for enterprise applications, The Open University, 2014 Ph.D. thesis.
[200] X. Peng, S.-W. Lee, W.-Y. Zhao, Feature-oriented nonfunctional requirement analysis for software product line, J. Comput. Sci. Technol. 24 (2009) 319–338, doi:10.1007/s11390-009-9227-2.
[201] M. Morandini, Goal-oriented development of self-adaptive systems, University of Trento, 2011 Ph.D. thesis.
[202] A. Mello Ferreira, Energy aware service based information systems, Politecnico di Milano, 2013 Ph.D. thesis.
[203] I. Elgedawy, Z. Tari, M. Winikoff, Exact functional context matching for web services, in: Proceedings of the Service-Oriented Computing - ICSOC Second International Conference, New York, NY, USA, 2004, pp. 143–152, doi:10.1145/1035167.1035189.
[204] L. Piras, P. Giorgini, J. Mylopoulos, Acceptance requirements and their gamification solutions, in: Proceedings of the IEEE 24th International Requirements Engineering Conference (RE), IEEE, 2016, pp. 365–370.
[205] L. Piras, E. Paja, P. Giorgini, J. Mylopoulos, R. Cuel, D. Ponte, Gamification solutions for software acceptance: a comparative study of requirements engineering and organizational behavior techniques, in: Proceedings of the 11th International Conference on Research Challenges in Information Science (RCIS), IEEE, 2017, pp. 255–265.
[206] L. Piras, E. Paja, P. Giorgini, J. Mylopoulos, Goal models for acceptance requirements analysis and gamification design, in: H.C. Mayr, G. Guizzardi, H. Ma, O. Pastor (Eds.), Conceptual Modeling, Lecture Notes in Computer Science, Springer International Publishing, 2017, pp. 223–230.
[207] Aradea, I. Supriana, K. Surendro, I. Darmawan, Integration of self-adaptation approach on requirements modeling, in: T. Herawan, R. Ghazali, N.M. Nawi, M.M. Deris (Eds.), Recent Advances on Soft Computing and Data Mining, Advances in Intelligent Systems and Computing, 549, Springer International Publishing, 2017, pp. 233–243.
[208] I. Supriana, K. Surendro, Self-adaptive software modeling based on contextual requirements., Telkomnika 16 (3) (2018).
[209] I. Supriana, K. Surendro, E. Ramadhan, Self-adaptive cyber city system, in: Proceedings of the International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), IEEE, 2016, pp. 1–6.
[210] K. Surendro, A. Supriana, I. Supriana, requirements engineering for cloud computing adaptive model., J. Inf. Commun. Technol. 15 (2) (2016).
[211] E. Park, IRIS: A Goal-Oriented Big Data Business Analytics Framework, Ph.D. thesis, 2017.
[212] G. Guedes, C. Silva, M. Soares, Comparing configuration approaches for dynamic software product lines, in: Proceedings of the 31st Brazilian Symposium on Software Engineering, ACM, 2017, pp. 134–143.
[213] I.J. Jureta, A. Borgida, N.A. Ernst, J. Mylopoulos, Techne: Towards a new generation of requirements modeling languages with goals, preferences, and inconsistency handling, in: Proceedings of the 18th IEEE International Requirements Engineering Conference, in: RE '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 115–124, doi:10.1109/RE.2010.24.