Using Goal Models Downstream: A Systematic Roadmap and Literature Review

Jennifer Horkoff, Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Tong Li, Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Feng-Lin Li, Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Mattia Salnitri, Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Evellin Cardoso, Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Paolo Giorgini, Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

John Mylopoulos, Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

ABSTRACT

Goal models have proven useful for capturing, understanding, and communicating requirements during early stages of software development. However, the utility of goal models is greatly enhanced when they can be exploited during downstream stages of the requirements analysis process (e.g. requirements elaboration, validation, planning), and can be used as part of the entire system life cycle (e.g., architectural and behavioral process design, coding, testing, monitoring, adaptation, and evolution). In order to better understand the progress that has been made in integrating goal models with downstream system development, the authors ask: what approaches exist that map/integrate/transform goal models to later stage software artifacts? To answer this question, they conduct a systematic survey, producing a roadmap of work summarizing 243 publications. Results include a categorization of the "why?" and "how?" for each approach. Furthermore, they select the 50 most prominent publications, based on citation numbers, in order to perform an in-depth literature review. Findings show that there is a wide variety of proposals with a variety of proposed goal models and targets, covering multiple paradigms, motivated by a variety of purposes. The authors conclude that although much work has been done in this area, the work is fragmented, following multiple separate strands of goal-orientation, and is often still in early stages of maturity.

Keywords. *Evidence-Based Requirements Engineering, Goal Model, Model Transformation, Requirements Engineering, Systematic Literature Map, Systematic Literature Survey*

DOI: 10.4018/IJISMD.2015040101

1. INTRODUCTION

The quality of a software system critically depends on the degree to which it fulfills its requirements. Such requirements are often captured, modeled and analyzed as (stakeholder) goals. Over the past two decades, goal modeling has received much attention in Requirements Engineering (RE) research, but also in Software Engineering (SE), Information Systems (IS), Conceptual Modeling (CM), and Enterprise Modeling. Goal models have been used as an effective means for capturing the interactions and tradeoffs between requirements.

Although goal models per se have proven their worth during requirements analysis, their utility would be greatly enhanced if these models were also used during other phases of the development process (downstream development), such as architectural and process design, code development, testing, monitoring, adaptation, and system evolution. This integration of goal modeling into downstream activities is challenging due to the qualitative, social nature of goal models. After all, it is difficult to take "fuzzy" concepts such as softgoals, roles, and actor dependencies and map or transform them to concrete functional system elements.

Much existing work has addressed aspects of this integration problem along different dimensions. In this work, we aim to understand the landscape and status of such existing work, evaluating the progress and maturity of efforts in this area, drilling down into the details of prominent publications. In particular, we want to understand the nature of existing proposals for goal model integration, including the type of transformations deployed, the type of goal models used, the motivations for such techniques, the common targets of the transformations, the venues where this work was published, the network of paper authors and citations, and the trends in such approaches.

Our study is shaped in the form of a roadmap[1] of approaches that map, transform or otherwise integrate goal-oriented languages to or with other artifacts or models related to the software or system lifecycle. The study focuses on the top 50 cited papers within our roadmap, providing a deeper analysis and literature review for these papers.

Kitchenham et al. have advocated Evidence-based Software Engineering (2004), inspired by Evidence-based Medicine. In their work, finding and assessing available evidence to address software engineering questions for researchers and practitioners is done in a systematic method-based fashion. In our study, we perform Evidence-based Requirements Engineering (EBRE), systematically finding and summarizing available publications in order to answer goal model-related research questions.

Specifically, we first produce a roadmap summarizing publications falling under the scope of our study without considering their quality (Kitchenham, Budgen, & Brereton, 2011). We then examine more carefully selected publications, in order to better understand the details of and motivations behind proposed mappings/transformations/integrations. We place particular emphasis on publications classified under Software or Requirements Engineering.

This roadmap and literature review can be beneficial for several types of readers. For researchers interested in using goal-orientated system development, the roadmap helps one to build upon existing work, avoiding the proverbial 'reinvention of the wheel', helping to understand trends, and guiding efforts towards new directions. For practitioners, this roadmap helps demonstrate the ways in which goal-oriented approaches can be integrated into existing system development paradigms, offering ideas on how goal-orientation can be adopted in practice, including pointers to work containing further details.

This work builds upon work presented by Horkoff et al. (2014). Specifically, we have expanded the coverage of our roadmap, covering an additional 69 papers, including papers appearing from September 2013 to June 2014. As part of our roadmap, we have added an analysis of cross-citations between included papers. Finally, we have selected the most prominent papers for a deeper analysis, performing a systematic literature review and reporting on the results.

The rest of the paper is organized as follows. Section 2 describes the scope of our study, while Section 3 presents the research questions and methodology followed. Section 4 summarizes and analyzes the results of the roadmap, while Section 5 gives details of our in-depth review. Section 6 discusses results of both the roadmap and deeper literature review, while Section 7 lists threats to the validity of the study. Section 8 reviews related approaches, while Section 9 offers conclusions and sketches future work.

2. SCOPE AND PRELIMINARIES

In this section, we circumscribe the scope of our investigation, by defining key survey concepts. Definitions for Language and Goal-Oriented Language (GOL) are provided in the first two rows of Table 1.

The study focuses on techniques that introduce a transformation, mapping, or integration scheme to or from a GOL to a system-related artifact. In order to allow our roadmap to cover a variety of approaches, we deliberately leave the definition of system-related artifacts broad. Such artifacts can include, for example, models, conceptual artifacts (features, services, agents), and processes.

Table 1. Definitions used to define study scope

Language	"A **Language** consists of a syntactic notation (syntax), which is a possibly infinite set of legal elements, together with the meaning of those elements, which is expressed by relating the syntax to a semantic domain. ... Depending on the language type, syntactic elements can be words, sentences, statements, boxes, diagrams, terms, models, clauses, modules, and so on" (Harel & Rumpe, 2004). Languages can be graphical or textual, and the semantics (meaning) can be formally or informally defined.
Goal-Oriented Language (GOL)	A language which includes the concept of goal as a first class object. Goal-Oriented Languages are often graphical, having a visual syntax (e.g. Tropos (Bresciani, Perini, Giorgini, Giunchiglia, & Mylopoulos, 2004), i* (E. Yu, 1997), KAOS (Dardenne, Van Lamsweerde, & Fickas, 1993), NFR (Chung, Nixon, Yu, & Mylopoulos, 2000), GRL (ITU-T & Intern. Telecom. Union: Recommendation Z.151 (11/08), 2008), etc.) but may also be textual (e.g., GBRAM (Anton & Potts, 1998)).
Transformation	A process that takes one or more source models as input and produces one or more target models as output by following a set of transformation rules (Kleppe, Warmer, & Bast, 2003; Mens & Van Gorp, 2006).
Mapping	A set of rules that describes how one or more constructs in the source modeling language can be connected to one or more constructs in the target modeling language (Kleppe et al., 2003; Mens & Van Gorp, 2006).
Integration	The creation of a new modeling language which is made up of constructs and relations from the source and target modeling languages.
Exogenous Transformation	A transformation between models expressed in different languages (Mens & Van Gorp, 2006).
Endogenous Transformation	A transformation between models expressed in the same language (Mens & Van Gorp, 2006).
Vertical Transformation	A transformation where the source and target models reside at different abstraction levels (Mens & Van Gorp, 2006).
Horizontal Transformation	A transformation where the source and target models reside at the same abstraction level (Mens & Van Gorp, 2006).

In order to be included in our map, one source/target of the transformation/mapping/integration must be a GOL. We focus on GOLs which include a structured language, either textual or graphical, formally or informally defined.

We provide definitions for model transformations, mappings, and integrations in Table 1, including types of transformations (exogenous/endogenous, horizontal/vertical). When classifying the transformation types of publications, we capture the classifications used by the publication authors whenever possible (e.g. the authors describe their approach as "transforming", "mapping", "integrating") and otherwise make judgments based on the Table 1 definitions.

As we are interested in transformations and integrations, moving from one phase or view in the system lifecycle to another, we exclude from our study model extensions, where additional concepts are added to an existing language. In our view, model extensions do not make sufficient transitions from one phase or view of the system to another, while integrations are more likely to bridge across conceptual spaces. We differentiate extension from integration by specifying that integrations must integrate two different existing languages. For example, adding a security concept to *i** is an extension, and is excluded (e.g., Giorgini, Massacci, Mylopoulos, and Zannone (2005)), while combining *i** and problems frames is an integration, and is included (e.g., Liu and Jin (2006)).

We include those papers published as part of an international journal, conference, symposium or book. We omit theses, focusing on work which has been published. We include workshop publications and regional conferences which match particular criteria. Specifically, workshop papers must have more than three citations per year since their publication. In this way, we cover prominent or influential workshop papers, but exclude those with low impact. We include only work published in English. Our scoping criteria is summarized in Table 2.

We can illustrate our scoping rules using example goal-oriented approaches. An example exogenous vertical transformation would be a transformation from goal models to class diagrams (e.g., Alencar, Marin, et al. (2009)). An exogenous horizontal transformation may include a method which transforms a goal model to another high-level requirements modeling language, such as UML use cases (e.g., Estrada, Martínez, and Pastor (2003)). An endogenous vertical transformation may include methods which transform a goal model into another goal model with a lower level of abstraction, for example, from requirements to architecture (e.g., Bastos and Castro (2005)). Endogenous horizontal transformation would include, for example, goal model visualization techniques (views, slices), reasoning approaches, refactoring, or syntactical analysis (e.g., Sebastiani, Giorgini, and Mylopoulos (2004)). As our focus is on the link from goal models to the system lifecycle we omit approaches which perform only endogenous horizontal transformations. Such approaches are typically aimed to improve use of goal models as part of only the RE stage of system development.

Other transformation classifications, such as syntactical vs. semantical or different technical spaces (Mens & Van Gorp, 2006), do not act as inclusion or exclusion criteria in our survey. We are interested in both automatic and manual, bidirectional and unidirectional transformations, as per Czarnecki and Helsen (2003).

3. SURVEY METHODOLOGY

As per Petersen et al. (2008), we articulate the specific research questions (RQs) guiding our study. We can identify an overarching research question (**RQ0**), namely: What approaches exist which map/integrate/transform goal-oriented languages to/from other RE/SE software artifacts or languages? Once we have identified approaches, using our scoping criteria from Section 2, we ask further, more detailed questions, as listed in Table 3.

Our process for finding and including or excluding papers is adapted from the processes presented by Petersen et al. (2008), summarized in Figure 1. To increase our coverage, we

Table 2. Publication inclusion and exclusion criteria

Inclusion Criteria	Exclusion Criteria
Transforms (maps/integrates) to/from/with a GOL to/from/with an RE or SE artifact or model, and	Describes only model extensions, or
Describes exogenous vertical or horizontal, or endogenous vertical transformations, and	Describes only endogenous horizontal transformations, or
If GOL is formalized, uses formalisms as part of downstream development, and	Formalizes a GOL without using formalisms as part of downstream development, or
In conference, journal, or in/is a book, and	Is a theses, or
Workshop or regional conference with more than three citations per year, and	Workshop or regional conference with less than three citations per year, or
Is published in English.	Is published in another language.

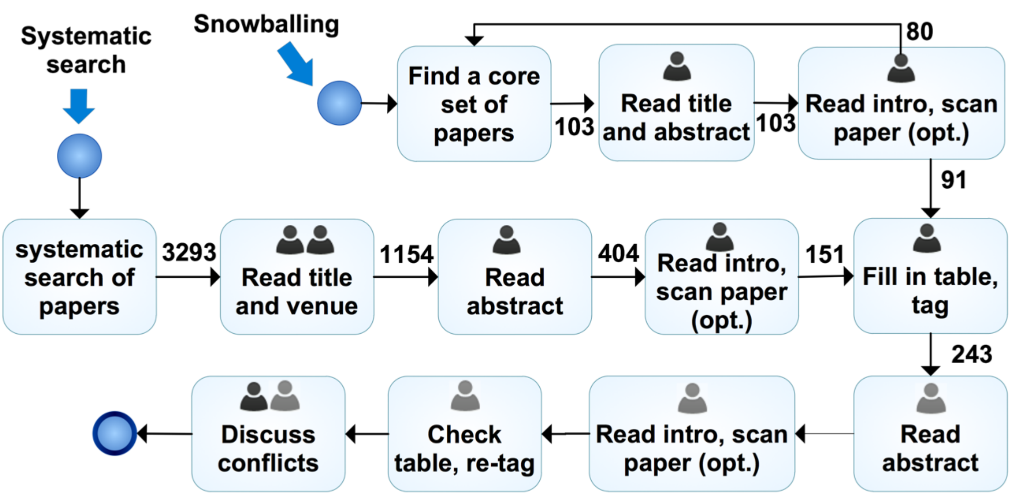
Table 3. Research questions

RQ0	What approaches exist which map/integrate/transform goal-oriented languages to/from other RE/SE software artifacts or languages?
RQ1	What types of transformations are used ([mapping/transformation/integration], [horizontal/vertical], [endogenous/exogenous])?
RQ2	What goal modeling frameworks are used most frequently?
RQ3	What sources or targets are goal models mapping/transformed/integration to/from/with? Are there trends in these choices?
RQ4	What are the motivations for the approaches? Are there trends in these motivations?
RQ5	What type of research papers focus on these approaches (validation/evaluation/solution/philosophical/opinion/experience as per (Wieringa, Maiden, Mead, & Rolland, 2006))?
RQ6	In what journals or conferences do approaches typically appear?
RQ7	What techniques are most widely cited? Are citations equally distributed?
RQ8	What techniques are most widely cited? Are citations equally distributed?
RQ9	Which papers reference each other? What does the network of cross-citations look like?
RQ10	Is interest in goal model integration increasing or decreasing?
RQ11	What are the details of the most prominent papers?

searched for relevant papers by conducting both a systematic search of available research paper databases and by "snowballing", starting with a set of core papers believed to be in-scope, and expanding our set of consideration based on papers referenced by these papers.

Snowballing. We started with a set of 103 core papers found as part of the authors' previous work, and believed to be included by our criteria. Candidate papers were assessed by reading the title and abstract. The reader could optionally look at details in the paper, reading the introduction or scanning the paper looking at figures and sections titles. A decision was made to include or exclude the papers based on our criteria (Table 2). Further papers were found by looking through the references of core papers, looking for candidate papers based on the paper title and publication venue. In order to ensure the snowballing process ended, we limited our reference search to a depth of two. Future efforts could extend this limit. Overall,

Figure 1. A Summary of the publication finding, exclusion, summary and tagging process (counts indicate included papers after each step)



at the time of paper submission, we have considered 183 papers in the snowballing process, including 92 papers in our survey.

Systematic Search. In addition, we performed a systematic literature search, searching for publications in several research databases (IEEE, Springer, ACM), published in the last 11 years (2003-2014). We derived our search string from our scope and research questions, searching for:

(“requirements engineering” OR “software engineering”) AND (“goal model”) AND (transformation OR mapping OR derivation OR alignment OR integration OR link)

where we replaced “goal model” with a variety of common forms (e.g., “goal modeling”, “goal-oriented requirements”).

We performed the systematic search in two stages, the first as part of Horkoff et al. (2014), where the initial search produced 2914 results. As part of the current roadmap expansion, we repeated the search covering September of 2013 to June of 2014, finding and additional 379 results. We divided the total 3293 results such that the paper title and venue were read by at least two people, marking the paper as relevant, irrelevant or possibly relevant. Papers which the title readers agreed were not relevant were discarded, while the rest were moved on to the next stage, with a single reader reading the abstract for relevancy. In total 1154 abstracts were read, with readers deciding that 404 of these papers were relevant or possibly relevant. A further round examined the paper introduction and optionally further paper details.

Papers included by both the snowballing process and systematic search were summarized in a shared table, recording source and target language, type of transformation, purpose of the paper, and research classification as per Wieringa et al. (2006). The summary included a set of tags (described in Section 4) summarizing the purpose and source/targets of the approach, derived via a Grounded Theory, grouping qualitative data according to relevant categories or codes relating to potentially interesting observations or theories (Seaman, 1999). For each type of paper, an additional reader was assigned to re-read the title, abstract, and summary of included papers, optionally looking at further paper details in order to re-tag the papers. Differences between tags were identified and discussed, resulting in a final set of tags. During this process, 46 duplicate publications, including overlaps between snowballing and systematic search papers, were identified. In total, our roadmap summarizes 243 papers. Roadmap results, answering RQ1 to 10, can be found in the next section. In order to have a more detailed view of the research landscape, and to answer **RQ11**, we performed a systematic literature review of the 50 most prominent papers. In this case we determined prominence by citations, taking into account the number of years since initial publication. Gathering citation counts from Google Scholar as of June 2014, we took the ratio of citation counts per year, determining those papers with the highest ratio. Results of this process can be found in Section 5.

4. RESULTS

In this section we present the results of our roadmap, using them to answer *RQ1-10.* We answer our overarching research question *RQ0* (What approaches exist which map/integrate/transform goal-oriented languages to/from other RE/SE software artifacts or languages?) in two ways. First, we provide the full list of 243 publications, including paper name, authors, venue, publication year, and other summary data. We make this information available in an online spreadsheet[2] and downloadable DB format[3]. Second, we provide a deeper review of the 50 most cited papers in Section 5. We use our roadmap results to answer the more detailed *RQ1-10* in the following.

RQ1: *Transformation Type.* We report the number of papers which were classified under mapping, transformation or integration in Table 4, including a classification of horizontal vs. vertical. Counts for endogenous/exogenous vs. vertical/horizontal are in Table 5. Note that it is possible for a paper to fall under more than one category, in which case it is counted for each category.

Examining the data in Table 4, mappings seem to be the preferred approach, particularly in the horizontal direction. In general, as creating a mapping is a subset of developing a transformation, mapping is the easier route, so these results are not surprising. However, there are still a significant amount of transformations, particularly in the vertical direction, where there are almost as many transformations as mappings. This shows that transformations are feasible even with highly social models such as goal models, although we do not have a clear sense of what concepts in the model are being transformed. The increase in transformations in the vertical direction, as compared to the horizontal direction, may indicate that it is more feasible to develop transformations when moving downstream, decomposing models into further detail, than when moving horizontally, linking models to a view at the same level of abstraction. We have also found 21 papers in-

Table 4. Technique count classified as mapping or transformation vs. horizontal or vertical

	Mapping	Transformation
Vertical	74	67
Horizontal	39	27

Table 5. Technique count classified as exogenous or endogenous vs. horizontal or vertical

	Endogenous	Exogenous
Vertical	7	154
Horizontal	18	60

cluding some form of integration. We can see that most approaches focus on transformations or mappings, avoiding language integrations, likely in order to avoid creation of overly complex languages.

Examining Table 5, transformations (including mappings/ integrations) are mostly vertical exogenous, with an apparent focus on moving goal models downstream via transformations to other artifacts. Despite the predominance of vertical exogenous transformations, we see a significant number of horizontal exogenous approaches, transforming goal models to other models at the same level of abstraction. Further exploration of horizontal transformations is provided as part of **RQ3**. Noting the predominance of exogenous transformations, we may surmise that although the intentional view provided by goal models is useful, most techniques see value in multiple conceptual views of the system, captured by multiple types of models. Issues encountered when classifying techniques as horizontal or vertical are discussed further in Section 7.

RQ2: *Goal model Source/Targets*. In Table 6 we list the top eight types of goal models used as transformation sources, including the counts of the number of publications using each language. Note that we found that many papers used "goal models" in general, without referring to a specific existing language. In some cases this is as the paper authors proposed their own, unnamed goal modeling framework. We could interpret these counts as indicating whether or not a particular goal model framework is more or less amenable to transformations. Alternatively, these counts could attest to the popularity or level of adoption of various goal modeling frameworks.

RQ3: *Non-Goal Model Source/Targets*. As described in Section 3, we undertook a grounded theory process in order to classify and tag publications. Our purpose was to summarize the purpose and motivation and the means associated with the included publications. This process resulted in the creation of a taxonomy of tags, under the general category of "how", i.e. the source/target of the transformation and "why", the general purpose or paradigm of the approach.

Our results include 170 unique non-goal model source and targets (one publication can be classified as having multiple sources and targets). Sources and targets are listed as part of the "how" tag taxonomies in Figures 2 and 3. The "how" taxonomy is divided into two views, focusing on approaches which integrate with or transform to goal models (Figure 2) and which transform from goal models to some other target (Figure 3). To help summarize our findings, we cluster detailed tags into more abstract categories (dashed boxes), e.g., Software Artifact, Requirements, and Business Artifact. As such classifications are subjective; our purpose is not to propose a rigorous hierarchy for system development, but instead to provide a higher-level, grounded summary of our results.

Counts are included in the upper right or in parentheses. Tags without a count have a default count of (1). Taggers could choose to classify leaf-level tags into more than one category, e.g., Activity Diagrams are UML Models and Behavioral Models. Words or phrases considered as pseudonyms for our tags are shown in parentheses, e.g., Business Rule is included under Informal Constraint.

Our taxonomies show that goal models have been transformed to and from a wide variety of languages and artifacts. Much activity

Table 6. The top 8 goal model source languages

Unidentified Goal Model	i*	KAOS	Tropos	GRL	NFR	SIG	Map
55	49	28	19	12	10	4	4

Figure 2. Taxonomy of "how" tags used to classify publications transforming to goal models (default count of (1))

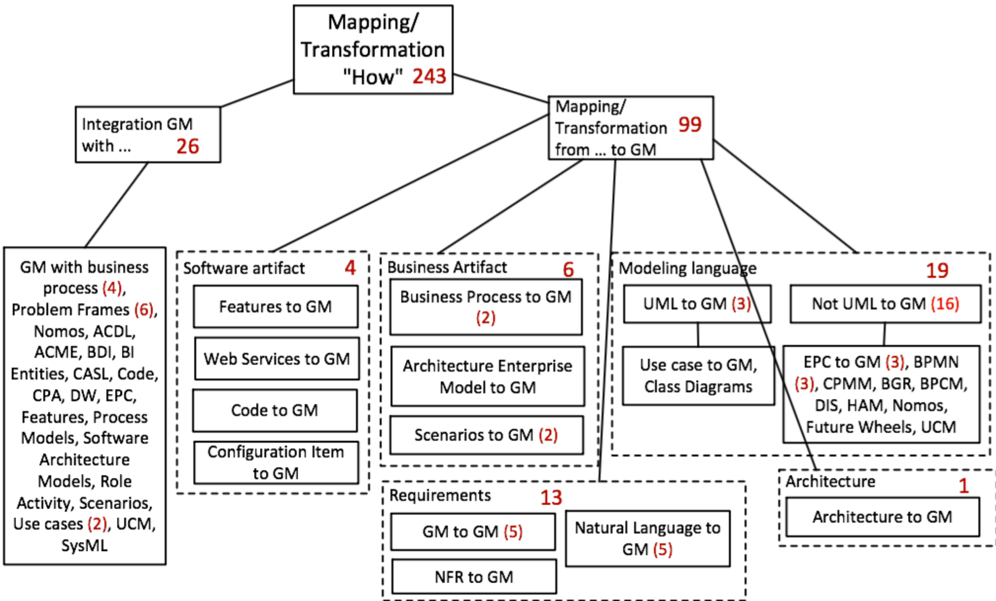
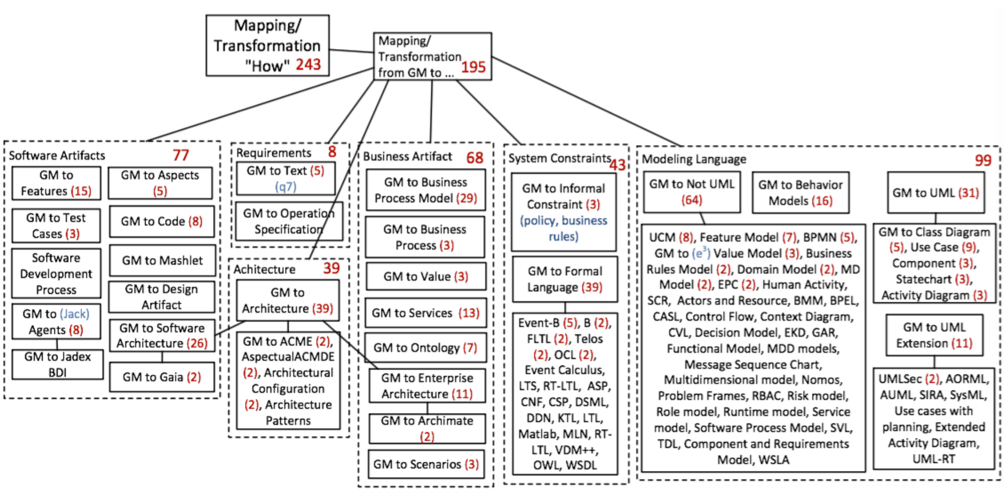


Figure 3. Taxonomy of "how" tags used to classify publications transforming from goal models (synonyms in parentheses, default count of (1))



has focused on transforming goal models to other modeling languages, particularly outside of UML (64 tag counts, listed in the middle left of in Figure 3). 31 papers capture transformations from GOL to some form of UML. Several other approaches have focused on formalizing goal models (39 papers), producing system constraints (43 papers). We can see that Business processes models and architecture are other popular targets, with 29 and 39 papers, respectively.

Most activity (61%) transforms goal models to another artifact, consistent with the view of goal models as an artifact for early requirements. However, it is interesting to note, as shown in Figure 2, that several techniques integrate goal models with other languages or use goal models as a target language. We may guess that most of these techniques would involve horizontal transformations, better exploring the requirements space. However, we can examine horizontal vs. vertical transformations in techniques which transform to and from goal models (Figure 4, left and right, respectively). We can see that techniques transforming to goal models use vertical transformation nearly as often as those transforming from goal models (63% vs. 68%). This indicates that techniques are using goal models for purposes beyond early requirements, as artifacts downstream from other artifacts. Given the focus of many GOL on early RE, this is a somewhat surprising result.

We can examine trends in the non-goal model source/targets of the approaches over time. Figure 5, Figure 6, Figure 7, and Figure 8 show the frequency of the source/target mid-level categories over the last eleven years (integrations with goal models and transformation to goal models in the top two charts, transformations from goal models in the bottom two charts). Note that the taxonomies in Figures 2 and 3 cover several papers published before 2003 arising from the snowballing process, while Figures 5-8 covers only 2003-2014. As such, the counts between the figures do not match exactly.

Looking at Figures 5 through 8, we can observe some trends. In Figure 5, it seems that publications integrating another artifact or language with goal models peaked in 2009, but is now beginning to rise again. The same could be said to a lesser degree for techniques transforming business artifacts to goal models. Transformations from software artifacts to GM (GOL) have remained relatively low. Looking at Figure 6, transformations from UML to GOL have declined, while Requirements to goal models had declined, but has begun to peak again. There is recent a spike in techniques transforming non-UML models to goal models. However, as the counts for integrations and transformations to goal models captured in these charts are very low overall, all trends may not be significant.

In Figure 7, there is a recent increase in techniques transforming GOL to business artifacts. In comparison, GM to software artifacts and architecture appear to in decline when compared to 2011. GM to system constraints appears to have peaked and was in decline, but has begun to rise again in 2013.

Figure 4. Horizontal vs. vertical transformations in techniques transforming to/from goal models

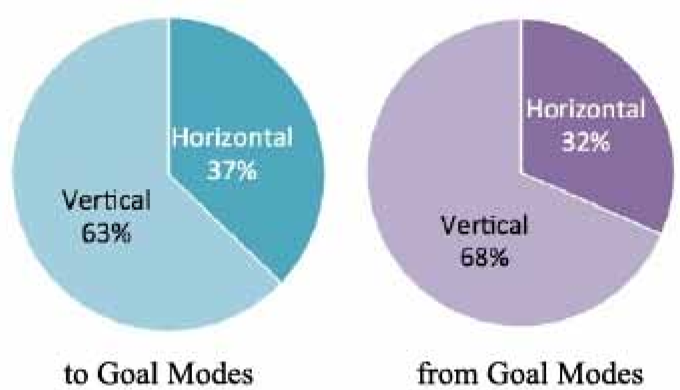


Figure 5. Frequency of "how", non goal-model source/target categories over time-integrations with goal models

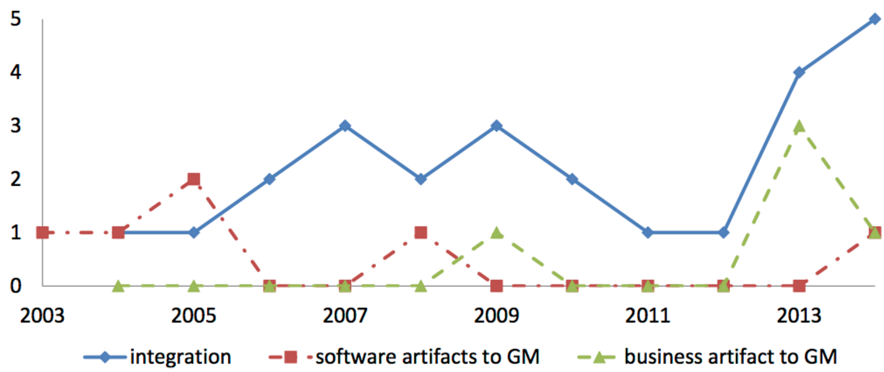
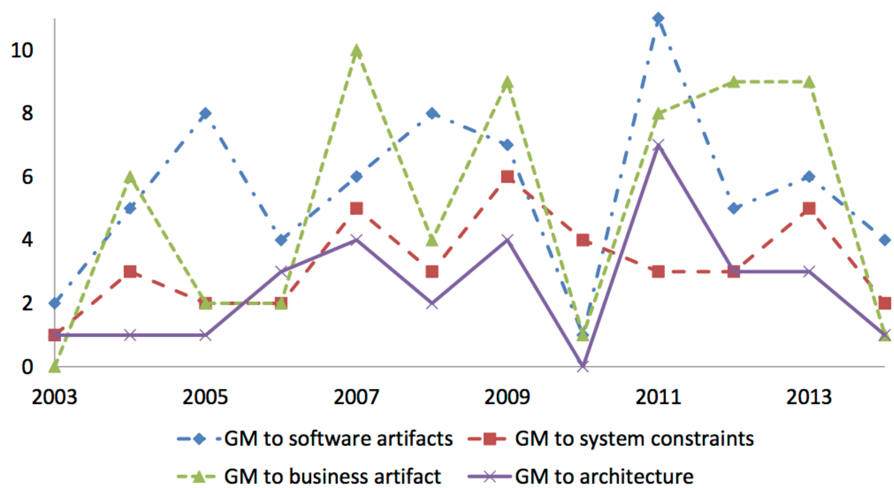


Figure 6. Frequency of "how", non goal-model source/target categories over time- transformations from goal models



In Figure 8, transformations from GM to modeling languages outside of UML are frequent compared to the other categories. After a sharp increase in 2011, such techniques appear to be on the rise again (data for 2014 is incomplete). Effort in transformations to UML models seems to be in general decline. The other two data series, GM to requirements and GM to UML extensions hold relatively steady, low frequencies.

RQ4: *Technique Motivations.* We show our "why" taxonomy in Figure 9. Several of the publications used such paradigms in order to motivate work, e.g. "taking advantage of the benefits of aspect-orientation", with the assumption that the benefits associated with a particular paradigm were well-known. As such, tags in this taxonomy are not clearly motivations, but often describe general paradigms, e.g. service-orientation, aspect-orientation. As it is not our purpose to describe the potential benefits

Figure 7. Frequency of "how", non goal-model source/target categories over time- transformations to goal models

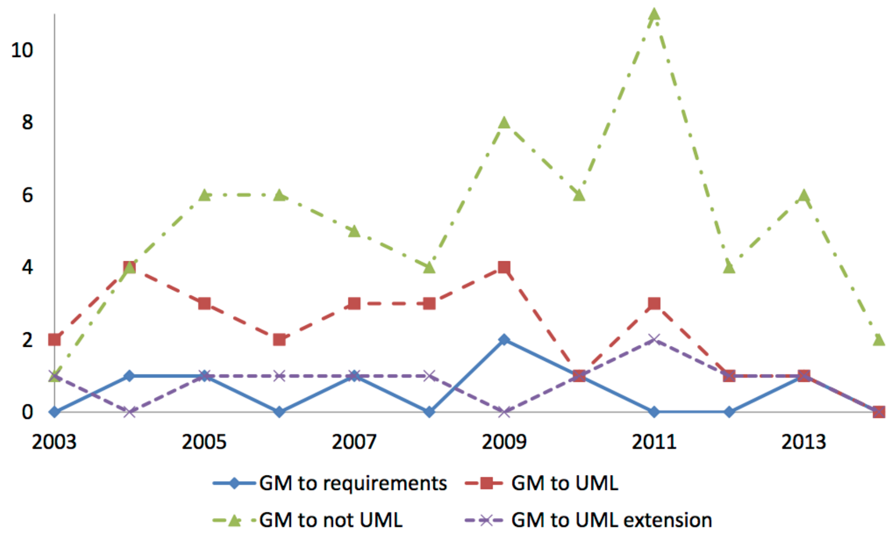
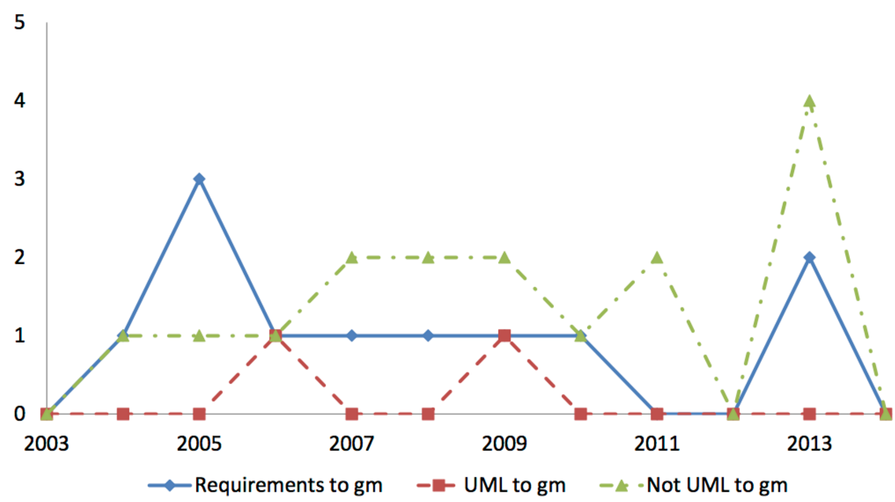


Figure 8. Frequency of "how", non goal-model source/target categories over time- transformations to goal models



of such paradigms, we stop our "why" analysis at this level. As with the "how" taxonomy, we classify the "why" taxonomy into higher-level categories (dashed boxes), e.g., Enhanced RE, Business Analysis, and Decision Making.

We examine the frequency of leaf-level "why" tags over time in Figures 10 through 13. Again, we note that the data for 2014 is not yet complete. Here we only show the most frequent 12 tags. We can note that use of goal model transformations for Enhanced RE was decreasing but has peaked again in 2013. Techniques aimed for alignment have peaked in 2011 and are now in decline. Service-oriented techniques are on the rise, peaking in 2013. Techniques addressing business process and adaptation are also on the upswing, rising in 2013. Techniques for MDD, variability and security show a small decline, while the remainder of the categories do not show obvious trends.

RQ5: *Research Classification.* We have classified the publications under the research classifications described by Wieringa et al. (2006), as follows: validation 4 papers (2%), evaluation 3 (1%), solution 235 (92%), philosophical 2 (1%), opinion 3 (1%), and experience 8 (3%), summarized in Figure 14. Publications could be classified into more than one category. In addition we found ten papers which we classified as related surveys, described in Section 8, and excluded from other counts. The high presence of solution papers was due in part to our practice of classifying papers as evaluation or experience only if this was the primary purpose of the paper, i.e. papers which presented a new method with some evaluation were classified only as solution. Even so, the prominence of solution papers can be interpreted as an indication of the immaturity of the field, with many proposals lacking extensive application or evaluation.

RQ6: *Venue.* We list the top venues in Figure 15. These are all the venues which have published three or more papers included in our roadmap, with 46% of the included publications appearing in these top 21 venues. Overall, included publications appear in a total of 137 unique conference/book/journal venues. This wide spread in venues may hinder consolidation of research results, while helping dissemination to a wider audience.

RQ7: *Citations.* We show citations numbers for the top 50 cited papers in Figure 16, listing the top five cited papers in the figure. Other papers can be identified via their identifier by looking at our online list of papers[4]. We see that all of the top five cited papers have been published at least nine years ago. Time, not surprisingly, is a significant factor in accumulating citations. Citations numbers are heavily weighted towards a few papers, although the average number of citations is 42, the top ten cited papers have 53.4% of the total citations. We can conclude that although many approaches are proposed, most have not been extensively re-applied in a research context.

RQ8: *Authors.* Examining the 243 included publications, we see a total of 484 authors. Among these authors, 149 have at least two publications included in our roadmap, while 72 have at least three publications. We can use

Figure 9. Taxonomy of "why" tags used to classify publications (synonyms in parentheses, default count of (1))

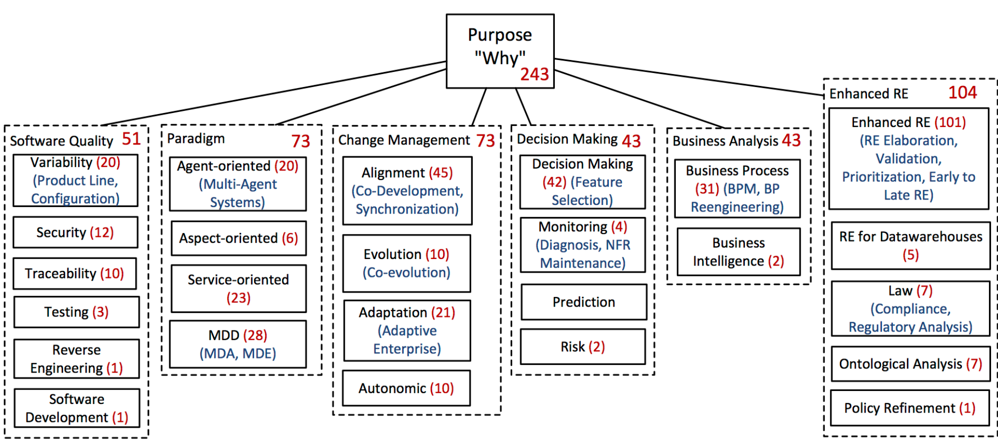


Figure 10. Frequency of most common "why" leaf tags over time

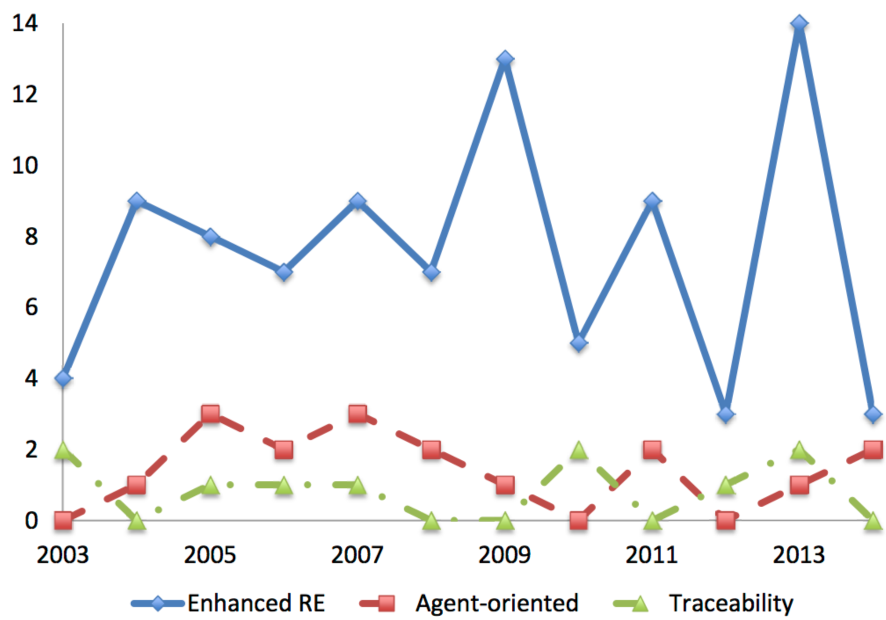


Figure 11. Frequency of most common "why" leaf tags over time

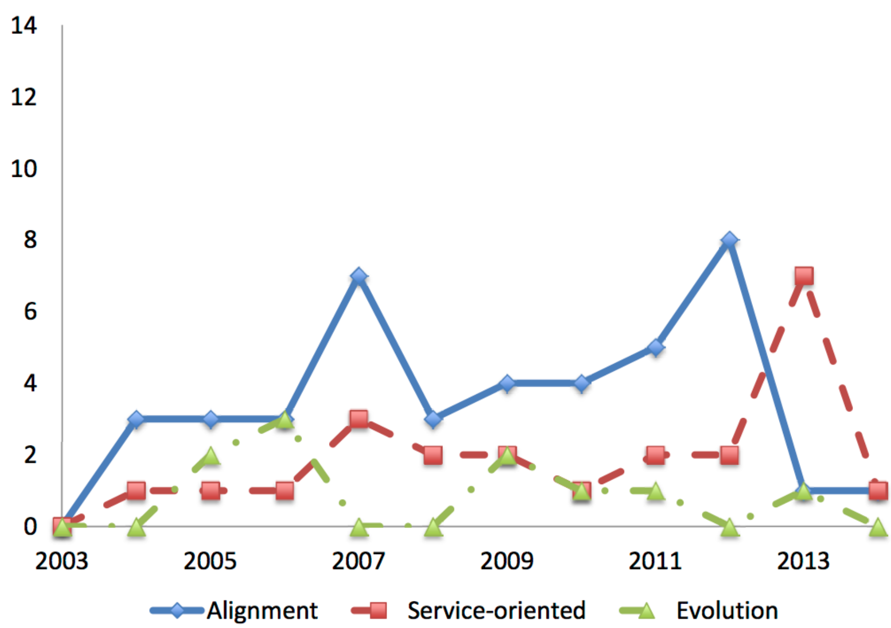


Figure 12. Frequency of most common "why" leaf tags over time

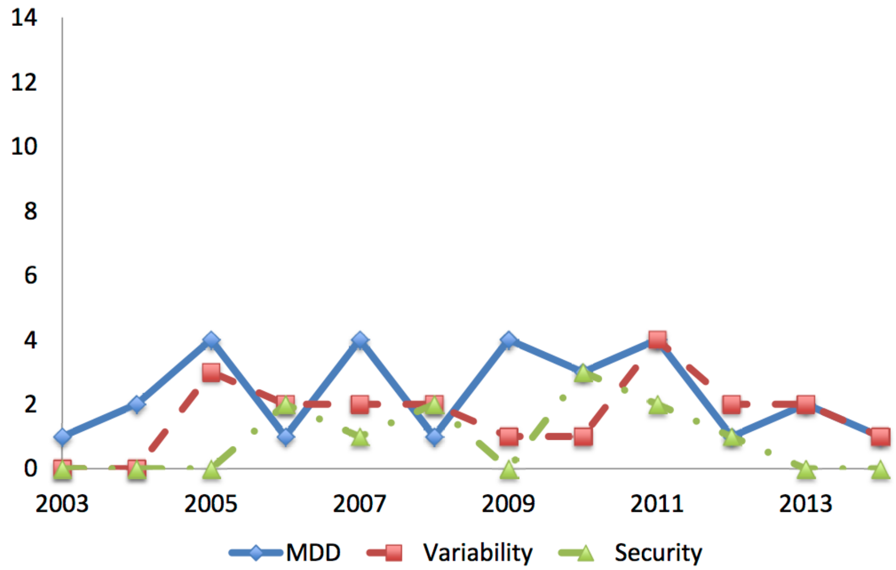


Figure 13. Frequency of most common "why" leaf tags over time

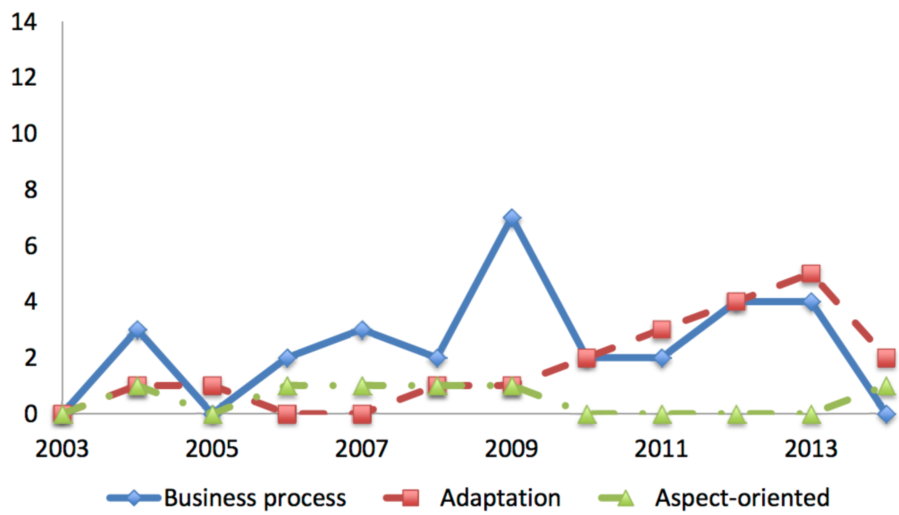


Figure 14. Breakdown of research classifications

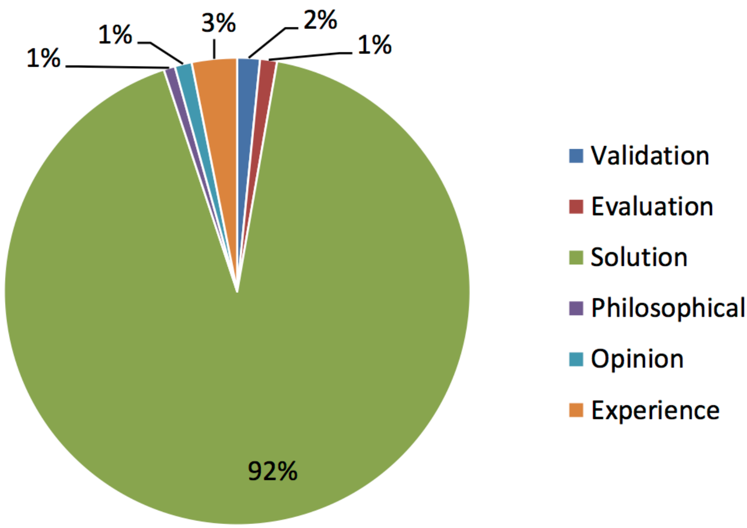
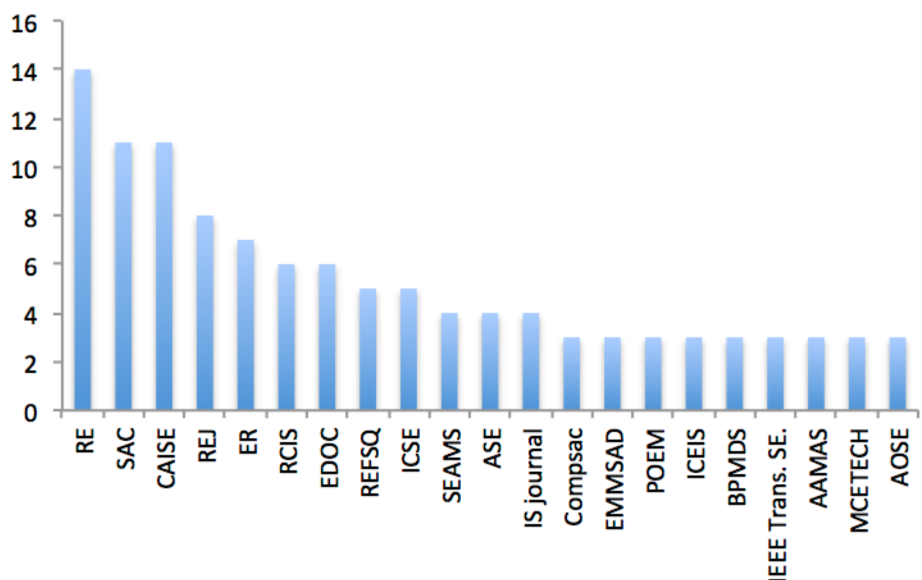
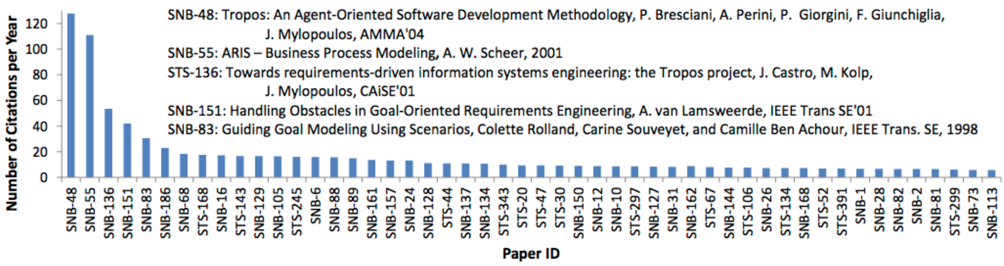


Figure 15. Top 15 publication venues



data on the included papers to create views of the co-author networks of paper authors. We show a high-level complete view of the authors of all 243 papers in Figure 17, a more readable version of this model can be found online[5]. One can note a few large, strongly-connected clusters of co-authors and the presence of many small clusters. A more detailed view of authors who

Figure 16. Citations counts for the top 50 publications



have more than three included publications is shown in Figure 18. Here authors are identified by their surname plus the first character of their first given name. From these figures we can see that the level of collaboration is significant, but not complete. However, it may not be desirable for such a network to converge in a particular topic, as it may hinder diversity and innovative ideas. We examine the cross citation diagram in the next section for hints as to the integration of the various approaches.

RQ9: *Cross-citations.* We can create the network of citations amongst included papers. For this, we have accessed the DBLP citation database, which contains 2,244,021 papers and 4,354,534 citation requests as of September, 2014[6]. Of the 243 unique papers included in our roadmap, 157 of these papers can be found in the database. Among these 157 papers, there are 61 citation relationships, involving 48 papers. We show the cross-citations in Figure 19.

Given the presence of 157 papers in the database, cross-citations between only 48 of the papers (30.6%) is relatively low. This indicates that work in this area is not frequently referencing or building upon existing work in this area, alluding to the relative isolation of most progress in mapping/transforming goal models. This isolation means that many similar approaches will be proposed, with some variations, as is also reflected by our "how" taxonomy counts in Figures 2 and 3, showing multiple approaches mapping to and from the same targets.

Looking at the content of Figure 19, we see that many papers have cited the original Tropos paper (Bresciani 2004, in the center of the diagram), a prominent goal modeling approach which includes downstream transformation techniques leading towards agent-oriented code. Like in the co-author diagrams, we can see clusters pointing to the same papers, e.g., pointing to Rolland 1996, Liaskos 2005, Amyot 2003, Lamsweerde 2000, and Kazhamiakin 2004, with the Tropos cluster being the largest. It is difficult to determine using this data if such clusters represent extensions, i.e., a deeper use of existing methods, or if they are more indicative of particular schools of approaches, giving a more shallow reference to related papers within that school. Future work should examine this and similar citation networks in more detail to try to determine the nature of the citations.

RQ10: *Interest.* We have asked "Is interest in goal model integration increasing or decreasing?" Figure 20 shows the number of publications per year in our mapping, including an interpolation line. We can see a rise from 2003 to 2007, with peaks in 2007, 09, and 11. The tendency towards publication in this area decreased notably in 2012, but then rose again in 2013. Results for 2014 are only complete up to June, but if these results can be approximately doubled for the rest of the year, the trend of publications on this topic will be maintained.

5. LITERATURE REVIEW

In this section we provide a more detailed review and analysis of the top 50 cited papers, taking into account citations per year. The maximum number of citations/year was recorded at 127.8,

Figure 17. Co-author network: Complete high-level view

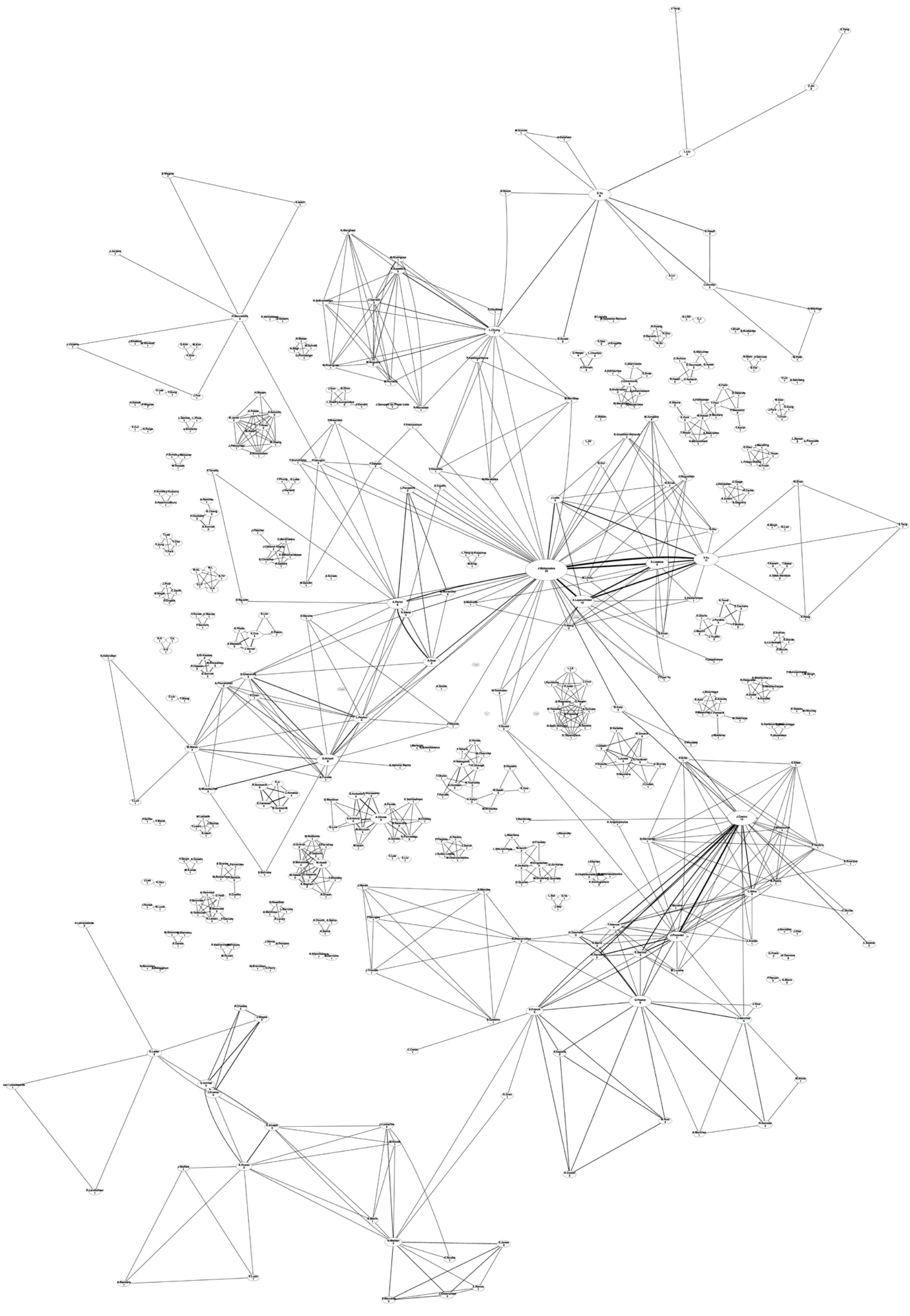
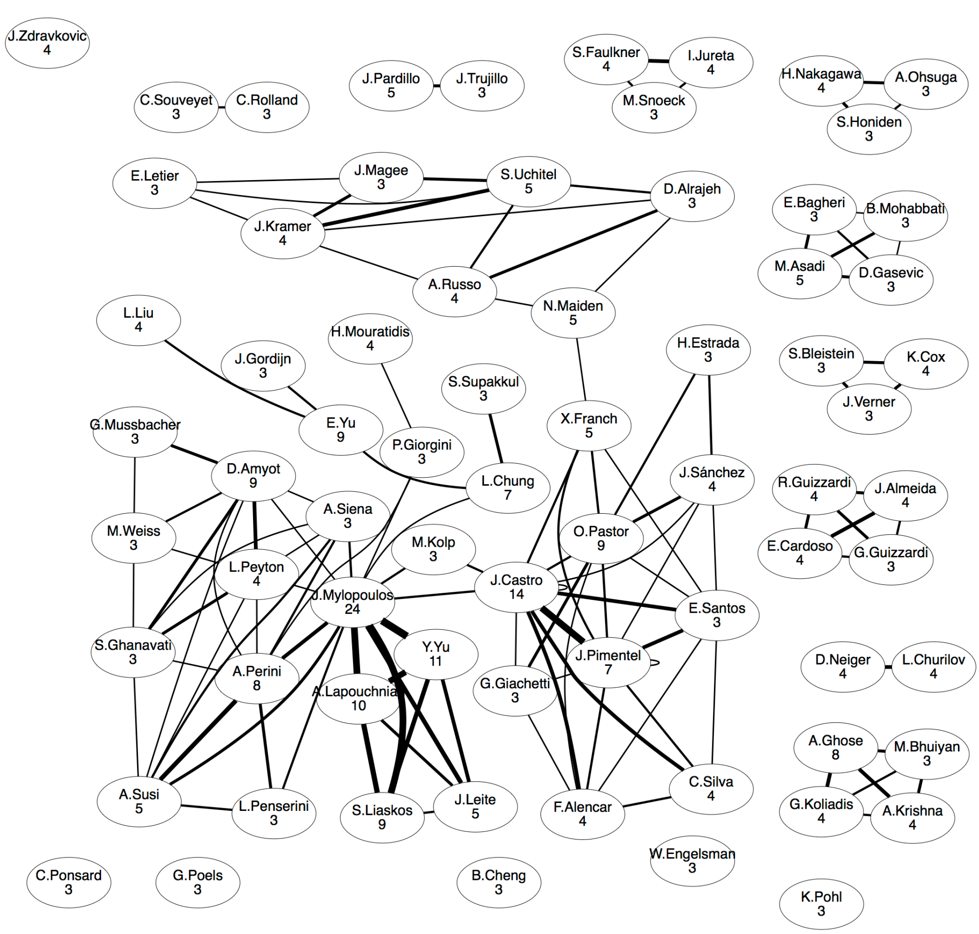


Figure 18. Co-author network: Detailed view of co-authors with more than three included papers



for the Bresciani 2004 Tropos paper (2004) (SNB-48), while the lower boundary of the top 50 citations/year was 5.7. The citation counts have been summarized in Figure 16.

We divide this section into multiple subsections, grouping paper analyses by "why" categories (see Figure 9 for the hierarchy of these categories). We classify papers under their "primary" why category, using our judgment; however, many papers could be classified under multiple headings. More specifically, the top 50 papers include 11 papers classified primarily under Software Quality (six papers classified under Variability, three under Security, one under Testing, one under Reverse Engineering), ten papers under the Paradigm category (including four Agent-Oriented papers, one Aspect-Oriented paper, two Service-Oriented papers, and three MDD papers), ten papers under Change Management (three under Alignment, one under Evolution, and six under Adaptation), one under Decision Making classified as Monitoring, six under Business Analysis classified under Business Process, and twelve

Figure 19. Cross-citations of included papers

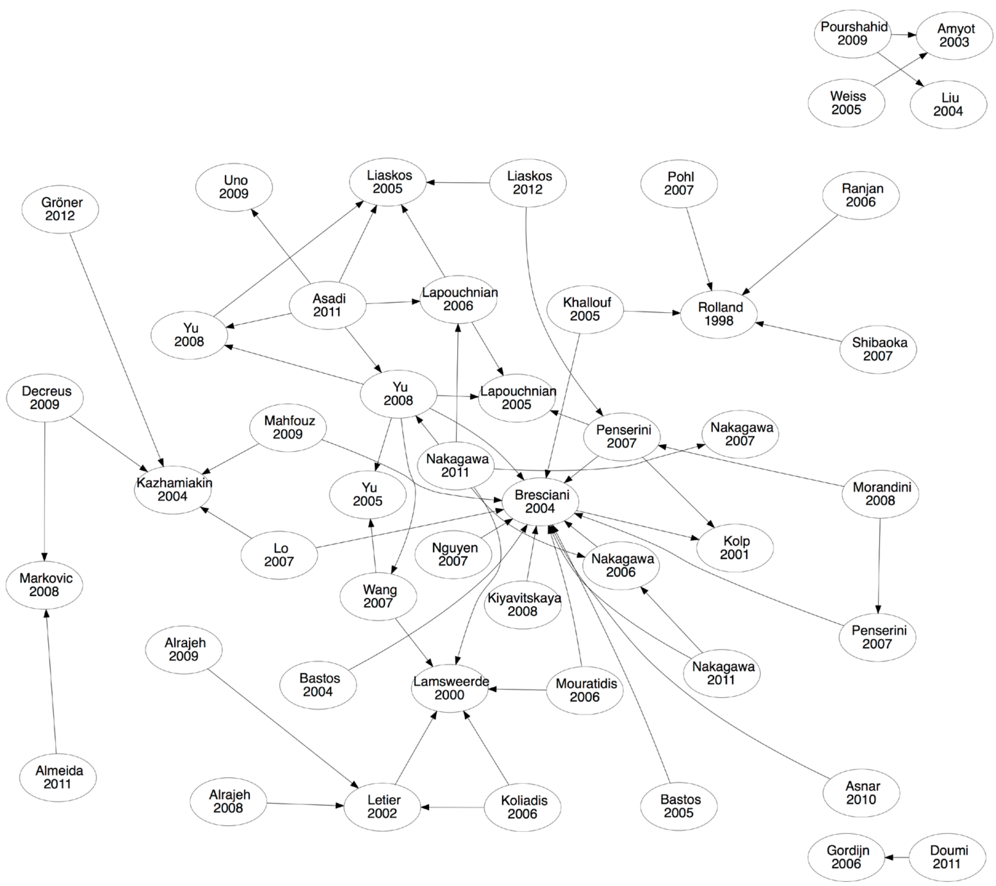
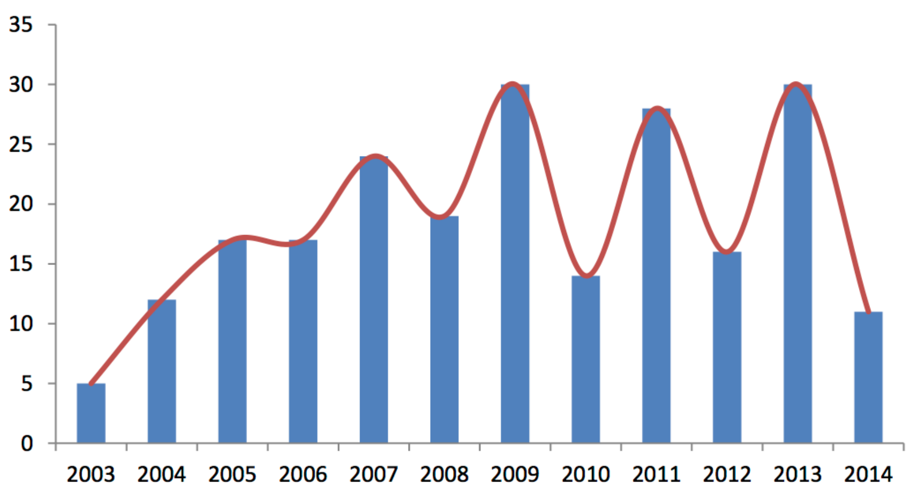


Figure 20. Included papers per year



under Enhanced RE (nine under the Enhanced RE subcategory, one under RE for Data warehouses, one under Law, and one under Policy Refinement). We summarize the high-level breakdown of categories for the top-50 cited papers in Figure 21.

5.1. Software Quality

Overall, 11 of our top 50 papers fall under the category of Software Quality.

Variability. Six of our top 50 papers are classified primarily under the category of Variability.

In one of the first papers focusing on variability using goal models, Hui et al. propose an RE framework which models requirements as goals, actors' skills for performing each function (that achieves a goal), and their preferences with respect to certain goals (Hui, Liaskos, & Mylopoulos, 2003) (identifier SNB-162 in our summary spreadsheet). Design alternatives are identified and mapped to customized software architecture. Variability is accounted for by capturing customization parameters like skills and preferences of the software target users.

Lapouchnian et al. (2005) continue this line of thought from the perspective of autonomic software. Specifically, they outline a possible architecture for autonomic software systems, which can be derived from high-variability requirements goal models (STS-30). A sub-tree of the goal model is assigned to an autonomic element, which uses that piece of goal model as a knowledge base to adapt the system, as part of a feedback loop.

A subset of the same authors later apply similar ideas in an approach which deals with the variability of business process configuration (Lapouchnian, Yu, & Mylopoulos, 2007) (SNB-24). In particular, they connect business processes with high-variability goal models, and configure the business processes according to high-level stakeholder's preferences, captured as NFRs.

Penserini et al. (2007) introduce an approach similar to Lapouchnian et al. (2005) (STS-30), but from the Tropos goal modeling framework perspective, which, by design, maps goal models to later system artifacts such as agent-oriented code, (STS-44). The authors extend Tropos ((Bresciani et al., 2004), SNB-48, summarized later) by explicitly modeling agent capabilities and refining the Tropos design process. The extensions enable high variability system design, when compared to the initial Tropos proposal. This work has been implemented as part of an MDA framework.

Y. Yu et al. (2008) take a different approach, proposing to trace goals to features, and thus, enabling generation of a feature model from a goal model, while preserving the variability identified in the goal model (SNB-31).

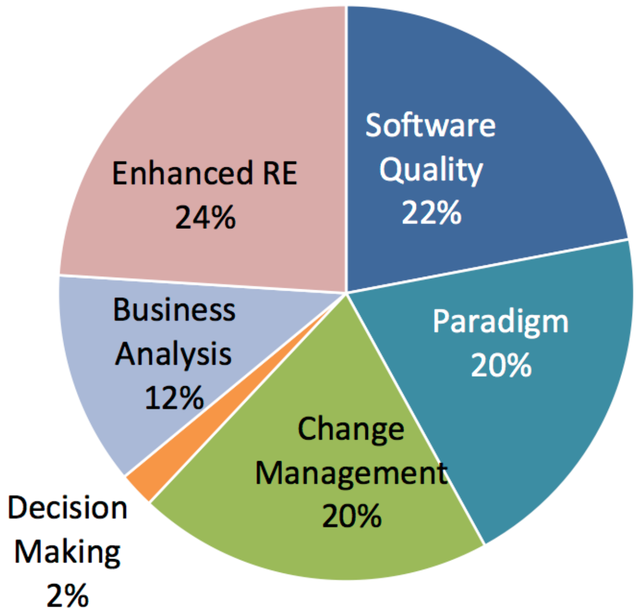
Y. Yu et al. (2008) go further by proposing a systematic process to transform goal models into several complementary design views, including configuration view (feature model), behavioral specifications (state chart), architectures (component diagram) and business processes (BPEL) (SNB-12). As with the previous approaches, the derived models preserve high-variability as specified in the original goal model.

Security. Three of our top-50 papers focus on security. Van Lamsweerde (2000) presents formal techniques using goals and domain properties to generate obstacles, which hinder the satisfaction of requirement goals (SNB-151). Such obstacles cover issues in both safety and security. As the next step, they propose three ways to tackle the identified obstacles, namely obstacle elimination, obstacle reduction, and obstacle tolerance. Some of the above methods can be automated by leveraging formal patterns, while others must be performed manually.

Mouratidis & Giorgini (2007) introduce Secure Tropos, integrating security concerns into the Tropos. As a result, security issues can be considered throughout the entire development process, from early requirements to late requirements, architecture design, and detailed design (SNB-16).

Mouratidis & Jurjens (2010) go further by integrating Secure Tropos (Mouratidis & Giorgini, 2007) (SNB-16) with UMLSec (a

Figure 21. The distribution of the 50 papers with the most citations/year into high-level "why" categories



model-based security engineering approach) in order to provide effective security analysis which covers both social and technical aspects. Goal models, resulting as part of Secure Tropos analysis, are transformed into security enhanced UML diagrams (Class diagram and deployment diagrams) for detailed security analysis.

Testing. In our only top-50 paper focusing on testing, Nguyen et al. (2008) introduce a goal-oriented testing framework that complements the Tropos software development methodology and strengthens the mutual relationship between goal analysis and testing (STS- 106). Their approach provides systematic guidance to derive test cases from Tropos goal models, and introduces a structure to specify them. To support their methodology, they also developed an eCAT (environment for the Continuous Agent Testing) tool, which interoperates with TAOM4E (a tool that supports Tropos) to take the output of goal analysis and generate skeletons of test cases.

Reverse Engineering. One of our top-50 papers focuses specifically on reverse engineering. This work presents a framework for reverse engineering legacy code in order to discover the stakeholder goals the code was intended to fulfill (Y. Yu et al., 2005) (SNB-144). The approach at first refactors source code by using program slicing, and then reconstructs state charts (states and transitions) using hammock graphs, and finally identifies goal models from state charts.

5.2. Paradigm

Eleven of our top-50 cited papers can be classified under one of our included paradigms.

Agent-Oriented. Four of our papers are classified under the agent-oriented paradigm, all of which describe some aspect of the Tropos goal- and agent-oriented methodology. To capture the operational (organizational) environment of software systems, Castro et al. (2002; 2001) propose a software development methodology named Tropos, founded on the i* concepts such as goals, actors, and (actor) dependencies (SNB-136, SNB-105). Tropos is proposed in order to model both early and late requirements, as well as architectural and detailed design. Tropos uses the NFR framework to select architectural styles, and maps i* constructs to BDI agents, using JACK Intelligent Agent platform constructs for implementation.

Kolp et al. (2002) propose a catalogue of architectural styles and agent patterns for designing multi-agent system (MAS) architectures using Tropos (SNB-137). The macro-level architectural styles are modeled using i* concepts such as actors, goals and actor dependencies, and the micro-level patterns define how goals assigned to actors in an organizational architecture will be fulfilled by agents. Moreover, the architectural styles are evaluated with respect to a set of software quality attributes using the NFR framework.

In the same spirit, Bresciani et al. (2004) expand upon the Tropos proposal with a focus on architectural styles and agent patterns (SNB-48). Although the ideas founding Tropos were introduced in earlier papers, this is typically the most-cited Tropos paper, as shown in Figure 19. The authors introduce Tropos as an agent-oriented software development methodology, which spans the software development process from early requirements to implementation. The methodology includes five phases: early requirements analysis (identifying stakeholders and their goals), late requirements analysis (focusing on the system-to-be within its environments), architectural design (defining the system's global architecture in terms of subsystems (actors) interconnected through data and control flows (dependencies)), detailed design (specifying agents' goals, beliefs, and capabilities, as well as communication among agents), and implementation (using JACK).

Aspect-Oriented. In our only top-50 paper focusing on aspect-orientation, Y. Yu et al. propose a systematic process for discovering aspects from relationships between functional and non-functional goals during goal-oriented requirements analysis (2004) (STS-245). The approach treats aspects at a higher level of abstraction (in terms of goals instead of code) and provides a tool-supported process, easing the participation of experienced software engineers. The approach addresses the open question: how does one identify aspects early in the software development process?

Service-Oriented. Two of the top-50 cited papers fall under the category of service-orientation. Noting that e-services are usually provided by a group of companies rather than a single company, and that information system design is often intertwined with business design, Gordijn et al. (2006) discuss how to use i* (covering a range of interests that actors in a constellation can pursue) and e^3value (revealing actors exchanging things of value) in combination to explore a multi-enterprise e-service offering (SNB-68). The approach also provides guidelines for producing e^3value models from i* models and vice versa.

In a different direction, Weiss et al. (2007) use GRL and UCM (summarized later) to model the intentional and behavior aspects of web services, and then analyze the goal models and/or behavior models to find and resolve conflicts (feature interactions) (SNB-113). The paper outlines a classification of web service feature interactions, helping us to understand the scope of the feature interaction problem in the web services domain, and providing a benchmark against which to assess the coverage of solutions to this problem.

MDD. Three of our top-50 cited papers focus on some variation of Model Driven Development. In one of the first publications in this area, Perini & Susi (2006) discuss a practical example of model transformation (using MDD techniques) in AOSE using Tropos (SNB-81). They introduce a technique which automatically transforms a Tropos plan decomposition into a UML 2.0 activity diagram. A CASE tool, based on a modular architecture, is extended to automate the transformations.

From a KAOS perspective, Van Lamsweerde (2003) defines a method to produce software architectural models from KAOS models (SNB-6). First, system goals are elaborated and software requirements can be incrementally derived from them (each step of the requirements derivation process is guided by heuristics and derivation patterns together with specific tactics). Next, software specifications (formulated in terms of objects manipulated by software) are derived from requirements through the usage of a set of rules. Finally, abstract dataflow architectures are derived from functional software specifications (also by the usage of a set of rules). The resulting architecture is recursively refined to meet the various non-functional goals discovered across the requirements engineering process.

An alternative technique starts the MDD process with i* models. Alencar et al. (2009) introduce an approach which describes the transformation of i* models to the conceptual model of the OOMethod using MDD techniques (SNB-1). In this case the OOMethod, introduced by Pastor & Molina (2007), is composed of four different models: Requirements Model, Conceptual Model, Execution Model, and Implementation Model. Parts of the i* model are selected for automated transformation. A set of detailed rules are applied to transform these parts to a conceptual model, resembling a class diagram.

5.3. Change Management

Under the category of change management, three of our top-50 papers focus on alignment, one on evolution, and six on adaptation.

Alignment. In one of the first papers looking at goal-oriented alignment, Liu & Yu (2001) propose a combined use of the goal-oriented language, GRL, and a scenario-oriented architectural notation, UCM, with the aim of aligning early architecture design with requirements (SNB-150). First, GRL models are created, and then business goals and non-functional requirements are refined and operationalized until concrete design decisions are obtained. These design decisions are then further elaborated into UCM scenarios. The use of GRL and UCM together allows for the representation of both functional and non-functional requirements, abstract and concrete requirements, and both intentional strategic rationals and non-intentional temporal details.

Maiden et al. (2004) focus on alignment between multiple types of RE models (SNB-28). The authors describe the application of RESCUE, a process which creates links between i*, human activity modeling (Vicente, 1999), and use cases. Links between models are used to help ensure model completeness and enhance model quality. The process has been validated using an air traffic management case study.

From a more recent Enterprise Architecture (EA) perspective, Engelsman et al. (2011) observe the lack of support for modeling the underlying motivation of EA in terms of goals and requirements. As such, they propose a new language, namely ARMOR, for modeling goals and requirements in EAs (SNB-129). The ARMOR language is based on the existing requirements modeling languages (e.g., i* and KAOS) and is aligned with ArchiMate, the standard enterprise modeling language. The authors use a real-life example in healthcare to demonstrate traceability from stakeholder concerns to the architectural elements.

Evolution. Cleland-Huang et al. (2005) introduce an approach to manage the impact of software system changes on non-functional requirements (STS-168). The authors use a probabilistic network to link a Softgoal Interdependency Graph (SIG), capturing NFRs, to functional elements of the software system. When changes occur to functional elements, their algorithm retrieves a set of potentially impacted SIG elements. These elements will be evaluated by stakeholders, and incorrect elements/links will be removed. Next, the impact of changes to the remaining elements is analyzed based on the SIG. Stakeholders use these results to evaluate the impact of functional changes on system wide non-functional requirements/goals to decide whether the functional changes should be implemented or not.

Adaptation. Six of our top-50 papers focus on adaptation. Lapouchnian et al. (2006) leverage goal models as a foundation for developing an autonomic software system (SNB-134). In particular, goal models help to capture stakeholder's needs, trace the design of autonomic component back to the high-level needs, and explore alternative designs. Each autonomic component corresponds to a part of the goal model. After adding formal details, e.g. pre-/post-conditions, to the goal model, the autonomic components can perform self-configuration, self-optimization, and self-healing by monitoring system behavior and operating environments.

Souza et al. (2011) approach the adaptation problem by introducing awareness requirements: a particular type of requirement capturing the satisfaction of other functional requirements (SNB-186). The paper presents a method for eliciting and formalizing such requirements for monitoring, playing a key role in the feedback loop for supporting self-adaptation. In this approach, awareness requirements are associated to other requirements using a goal model notation.

In further work, Souza et al. (2013) extend their ideas concerning awareness requirements and adaptation (STS-391). Awareness requirements are again evaluated as part of an adaptive feedback loop. The paper goes further by proposing a language for expressing such requirements, as well as techniques for elicitation for and implementation of adaptive systems. Ideas introduced in this work are used again as part of the Zanshin adaptation framework, where requirements are captured using extended goal models (Tallabaci & Souza, 2013). Zanshin is evaluated in a comparative study by Angelopoulos et al. (2013) (STS-343). In this work, Angelopoulos et al. compare Zanshin (Tallabaci & Souza, 2013) and Rainbow (Garlan, et al., 2004), another framework for adaptive software systems. The comparison is interesting, as the two frameworks use different strategies for modeling the design of adaptive systems. The former uses a requirements model to capture variability and support adaptation, while the latter uses architectural models to capture architectural variability and support architectural reconfiguration. The comparison has been conducted by applying both frameworks to the znn.com case study, highlighting discovered weaknesses of both frameworks. The authors propose to combine the frameworks in order to meet all adaptation needs of the target adaptive system.

Baresi et al. (2010) take a different perspective, utilizing KAOS models as part of a requirements-driven approach to the process adaptation problem (STS-52). Specifically, they define adaptive goals in addition to conventional goal models (KAOS). The conventional goals are translated into a functional model (BPEL), and the adaptive goals are translated into a supervision model, which evaluates LTL properties and automatically guides the system adaptation.

Pimentel et al. (2012) focus on adaptation from an architecture perspective, proposing a process for transition between requirements and architectural models for adaptive systems, with the former represented using i*, and the latter represented using Acme (SNB-2). As part of their method, the architectural model of the system is enriched by adding context information to guide the identification of events that may lead to adaptation of the system.

5.4. Decision Making

Our top-50 papers includes one paper falling under the category of decision making, specifically software monitoring.

Monitoring. Wang et al. (2007) propose an adaptive framework to monitor the satisfaction of software requirements (STS-67). The framework is comprised of: (i) an adaptive monitoring framework, which logs the actions of the system; (ii) a SAT-based monitoring framework which analyzes the log generated by the first component and identifies the root problem which denies the achievement of one or more requirements.

5.5. Business Analysis

Business Process. Our roadmap has found six top-cited papers focusing on integrating or linking goals with business processes as part of business analysis. The ARIS method (ARchitecture for integrated Information Systems) consists of an architectural framework used for enterprise description through the usage of several viewpoints, one of which is EPCs, used commonly in business process modeling (Scheer, 2000) (SNB-55). Among the other viewpoints, the Functional (ARIS Functional Viewpoint) and Objective Viewpoints (ARIS Objective Viewpoint) are the most relevant to our study, as they capture the concept of goal (or objective), and its associations with other enterprise elements. In this work, functions are considered as operations applied to objects with the purpose of supporting one or more business goals.

Kazhamiakin et al. (2004) propose an alternative method for linking goals to processes (SNB-168). They propose a framework for representing strategies and goals of an organization (business requirements), how this strategy is operationalized into activities, and, subsequently, how it is implemented by business processes. In this approach, it is also possible to represent the assumptions attached to each interaction between different business applications. The framework allows for the usage of model checking techniques in order to find problems and to identify possible solutions.

Markov & Kowalkiewicz (2008) introduce an approach which integrates business goals and business process models (SNB-73). Specifically, they design a Business Goal Ontology for modeling business goals and propose a modeling pattern for linking goals and process models. While the Business Goal Ontology and the pattern are proposed specifically in this paper, the process ontology is part of a more general effort for designing organizational ontologies to describe the whole organizational environment carried out in the context of the SUPER Project.

Grau et al. (2008) approach the integration of goals and processes from a business process reengineering perspective (SNB-127). This work uses an i*-based the business process reengineering method to drive the generation of information system specifications. They first propose a systematic process to construct the initial i* requirements model by analyzing existing business processes. After identifying new issues introduced to the current process, they systematically reengineer the i* model and evaluate new alternatives. Finally they produce the information system specifications (use case and class diagram) according to the evaluated i* models.

Pourshahid et al. approach the integration of goals and processes from a business intelligence perspective (Pourshahid et al., 2009) (SNB-161). The authors extend the URN framework (GRL + UCM) with Key Performance Indicators (KPIs) and other concepts to measure and align processes and goals. The integrated framework allows one to produce three types of core models, namely, a goal model (modeled using the GRL language, part of URN), a KPI model (extending URN) and a process model (modeled using UCM). Further, the proposal also presents a supporting infrastructure and a methodology for aligning business processes with goals.

Quartel et al. (2009) (SNB-128), in a precursor to the work by Engelsman et al. (2011) (SNB-129), propose a language for modeling the motivations and goals for a given enterprise architecture. This new goal language is integrated in the ArchiMate enterprise architecture modeling framework via the addition of the concept of requirement.

5.6. Enhanced RE

Our roadmap includes several categories under enhanced RE; a general enhanced RE tag, RE for Data warehouses, Law, and Policy Refinement. In total, 12 papers fall under these categories.

Enhanced RE. Several of the top-50 papers within the enhanced RE category focus on the relationship between goals and scenarios, with scenarios often represented as use cases or use case maps. Antón et al. (2001) describe a case study which examines the challenges with scenario use in practice (SNB-82). They address these challenges, in part, by applying the goal-based GBRAM method. In GBRAM, goals are operationalized and refined into requirements pointing to new scenarios, while use cases are used to derive goals, representing the use case context. Heuristics are used to classify goals (e.g., achieve, inform, allow) and to find redundant goals. They point out the benefits of reusing goal classes; distinguishing between system (make) and user (achieve) goals and using goal-orientation to raise the level of abstraction, moving the focus away from system-specific information.

Rolland et al. (1998) point out several problems with goal-oriented methods: users don't often know their own goals, and elicited goals are often too ideal (SNB-83). They argue that existing approaches for goal/scenario coupling focus on structure and documentation, and not discovery and elaboration. Their approach is bidirectional and iterative, tightly coupling goals and scenarios together in Requirement Chunks (RCs), which are then connected in a hierarchy of semi-structured text. They give rules for RC discovery, based on the ideas of refinement, composition, and alternative.

Liu & Yu (2004) introduce the URN (User Requirements Notation) framework (SNB-157), an expansion of work described in 2001 (SNB-150), combining GRL (Goal-oriented Requirements Language), similar to i*, with Use Case Maps (UCM). GRL enables users to capture, non-metricized, informal, and conflicting quality requirements while UCM allows users to illustrate causal relationships among responsibilities. GRL and UCM models are developed concurrently and iteratively. In this work, the mapping between the two models is introduced very informally, with alternatives in GRL mapped to UCMs, and UCM generating ideas for softgoals.

Santander & Castro (2002) (SNB-10), provide guidelines for developing use cases based on the contents of i* models, arguing that organizational requirements captured in i* must be related to functional requirements captured in use cases. The result creates a more systematic way to develop use cases, mapping goals and (goal/task/resource) dependencies found in i* SD diagrams to single use cases. SR diagrams, including softgoals are used to develop the use case descriptions.

One of our top-50 papers takes a different approach to enhancing RE practice, linking NFRs to UML design patterns (Gross & Yu, 2001) (SNB-88). This work uses the NFR framework to capture the motivations and tradeoffs behind design patterns. Softgoals capture considerations helping to choose between patterns. Pattern selection is further supported via an integration of the NFR-pattern models with system goal models, capturing selection justification. "Solution structures" for the system under development are described by mapping functional goal-model elaboration structures to UML class diagrams.

Two approaches classified under enhanced RE focus on the derivation of operational specifications from formally defined goals. Letier & Van Lamsweerde (2002) map goal specifications in real-time temporal logic to software operation specifications consisting of pre-, post- and trigger conditions (SNB-89). The approach supports incremental mapping, building gradually from partial models. A pattern based technique is used for the mapping, simplifying the mapping process and allowing users to take advantage of existing proofs of correctness. The paper includes a taxonomy of frequently used goal operationalization patterns.

Alrajeh et al. (2009) (STS-47), introduce a method which transforms from goals to operational requirements as pre- and post-conditions. The method guarantees the satisfaction of system goals. This paper tackles the same problem as Letier & Van Lamsweerde (2002) (SNB-89), with the same output, but does not use patterns; arguing that the use of patterns is limiting and error-prone. Instead the authors use model checking to check if goals are satisfied by a specification, otherwise produce a counter-example in the form of a scenario. An engineer uses the scenario to give positive or negative scenarios. Learning techniques are used to learn from these scenarios and propose improvements to the operational requirements, which are then selected and added back to the requirements by the engineer. The entire process iterates until the specification meets the goals.

De Landtsheer et al. (2004) take advantage of the complimentary benefits of goal orientation and tabular representation by introducing an approach which transforms KAOS into the SCR tabular language (STS-299). The SCR language allows one to perform consistency checks, general model checking, and simulation, amongst other things. By linking KAOS to SCR they allow for a high-level, richer analysis leading towards such formal checks. The procedure starts with an operational model derived from a KAOS model, using the method described by Letier & Van Lamsweerde (2002) (SNB-89), then performs a number of steps with specific rules to move from such a model to SCR, for example, moving from a multi-agent model to a bi-agent model by mapping KAOS agents to either the machine or environment, and getting rid of non-determinism by choosing a lazy or eager behavior for each operation. The method is evaluated via comparison to an original SCR specification.

Work by Letier et al. (2008) also introduces a method to transform KAOS models, but instead of SCR, transforms to even-based transition systems represented by LTS, enabling use of the LTSA toolset (STS-297). LTS models have been shown to allow for many forms of automated formal analysis, such as checks for incompleteness and inconsistencies, checking goal satisfaction, as well as the ability to animate operations. Transformations occur in several steps: identifying fluents, translating preconditions, and then translating remaining conditions. Compared to the SCR transformation described by De Landtsheer et al. (2004) (STS-299), the authors argue that semantic inconsistencies between KAOS an SCR made the previous transformation problematic, while LTS is a better match, preserving KAOS semantics.

RE for DW. Mazón et al. (2007) argue for the need for RE for Data Warehouse (DW) design, ensuring the data addresses real goals (STS-20). They use the i* framework along with Model Driven Architecture (MDA) techniques in order to capture goal and information requirements for data warehouses. Such models are then transformed into a multidimensional DW model, supporting decision making. They use UML to capture i*, adding constructs for information requirements. The authors apply MDD techniques, moving from an i*-based CIM to PIM to PSM, using the QVT language for the transformations. Only the information requirements are transformed from CIM to PIM, omitting constructs such as softgoals.

Law. Breaux and Antón (2005) (STS-134), introduce tools and techniques to better align policies with system requirements. The "semantic parameterization" process first mines goals from privacy policy documents. The goals are reformed to a form of Restricted Natural Language (RNL). The RNLs are eventually mapped to a machine-readable format, with the mapping captured via semantic models.

Policy Refinement. Bandara et al. (2004) (STS-143), focus on policy refinement, treating policies as constraints over a system, which can be either at a higher-level of abstraction than goals, or as a downstream, implementable artifact. Policies are defined with the intention of constraining behavior while allowing for extensive flexibility. They make use of existing goal refinement techniques defined for the KAOS framework. Abductive reasoning is used to find strategies (sets of operations) which will achieve goals, and subsequently higher-level policies. As such, policies are mapped to concrete system artifacts. Policies, goals, and strategies are captured formally in Event Calculus, with UML statecharts used as intermediate representations.

6. DISCUSSION

Roadmap. The final 243 publications included in our roadmap provide a variety of transformations involving goal models, especially transformations from goal models to other models/artifacts, with a high number of vertical, exogenous approaches. Only a small percentage of publications are widely cited, and most work focuses on new solutions, instead of solution evaluation or application experience.

We observe that many approaches are narrowly focused, with most approaches focusing only on a few stages of the software lifecycle, not often providing an end-to-end solution. Analysis of the network of coauthors shows that authorship in this area is still relatively fragmented, with many small isolated groups. Although the number of publications in this area was decreasing, interest seems to be steady. However, the frequency of solution papers leads us to believe that work in this area is still relatively immature, with a divergent set of approaches.

Literature Review. Reflecting on the deeper review of the top 50 cited papers in Section 5, we can make several observations. We can see that in several of the sections, the papers appear to be split into several "schools" relating to the various goal model frameworks. Many sections have work from the KAOS perspective (e.g., (Alrajeh et al., 2009) (STS-47)), the Tropos perspective (e.g., (Perini & Susi, 2006) (SNB-81)), the i* perspective (e.g., (Alencar et al., 2009) (SNB-1)), and the GRL perspective (e.g., (Pourshahid et al., 2009) (SNB-161)). We can see that the different philosophies underlying these frameworks (e.g., formal specification vs. supported social vagueness) transition to multiple, often similar papers which use each framework for various types of transformations. Despite this fragmentation, we do see a significant number of incremental papers: papers which build on KAOS, i*, Tropos, or GRL, working towards various aims. In this way, the landscape of such work looks like a streamer, or set of tributaries, with many concurrent, similar, threads splitting off from different sources.

Of course not all work fits within this pattern, some "outliers" take a more independent look at goal-orientation(e.g., (Markov & Kowalkiewicz, 2008) (SNB-73), (Antón et al., 2001) (SNB-82)).

One of the most significant challenges in mapping or transforming goal-oriented models is the transformation social or fuzzy aspects, most typically represented via softgoals. Examining the top-50 cited papers, only a few address this challenge. For the KAOS-related papers, this challenge is irrelevant, as all ill-defined requirements are formalized before translations. The challenge becomes more relevant for i*/Tropos/GRL approaches. In these cases, most approaches either chose to selectively transform or map only the "hard" elements (tasks, resources, actors, sometimes hardgoals or dependencies) (e.g., (Lapouchnian et al., 2005) (STS-30), (Alencar et al., 2009) (SNB-1)), while others make use of softgoals in an informal way, requiring much human intervention (e.g., (Santander & Castro, 2002) (SNB-10), (Liu & Yu, 2004) (SNB-157)).

One can argue that if some of the goal model is linked or transformed to downstream artifacts, explicit transformation of softgoals is not necessary. It is possible to make use of the links between models, and then from linked hard elements to soft elements within the goal model for the purposes of reasoning and traceability. For example, in URN links from goal models design options to UCM, and then design options to softgoals, allows for each UCM to be evaluated in terms of goal satisfaction, including the satisfaction of softgoals (Liu & Yu, 2001) (SNB-150). In such cases a direct transformation or mapping of softgoals may be unnecessary.

We can ask what is particular about these top-50 papers that causes them to have a relatively high number of citations. Part of this may be the quality of the papers themselves, although this is difficult to assess without also reviewing the papers with lower citations. One factor may be the co-coherence with goal model "schools" as mentioned above. As each school has a particular community, papers within this school may be more likely to be cited and used as part of future work.

Another factor may be the venue in which these papers appear as compared to the venues for our entire set of 243 papers. Looking through the venues, we see a wide variety of differing venues; with 33 unique venues for 50 papers (SNB-55 is self-published). The most popular venues with their respective counts being the

RE conference (five), SEAMS (four), REJ (three), CAiSE (three), IEEE Trans. SE (two), ICSE (two), and AOSE (two). We can note that most of the top-cited papers do not appear in workshops (only three are workshop papers). Although these venues are prominent and well-known in the areas of RE and SE, there is not a clear link between citation count and a few particular venues.

Another factor which may contribute to the level of citations is the prominence and reputation of the authors. We see a strong correlation between these top-50 papers and the top authors as shown in Figure 18. However, here we have a chicken and egg problem, are the authors well-known because their papers are highly-cited, or are the papers highly-cited because the authors are well known? Answering this question is out of the scope of our analysis.

Industrial Adoption. Although there are many examples of successful industrial applications of goal models (e.g., Van Lamsweerde, 2004; iStarShowcase, 2011), most efforts have a high degree of academic participation - practitioners do not often adopt goal-oriented techniques on their own initiative. The lack of widespread industrial adoption could be attributed to several factors, for example, the presence of many competing goal-oriented languages and tools lacking standardization, or scalability and usability challenges in complex models which are not easily decomposable.

One may hypothesize that a further barrier may be difficulties in integrating goal models with other system artifacts. However, the results of our roadmap have shown that many techniques have been proposed to facilitate this integration. Despite the availability of techniques, the divergence and lack of evaluation for methods in this area are likely to discourage independent adoption.

7. THREATS TO VALIDITY

We can identify several threats to the validity of our study.

Study Completeness. Although we have covered 243 papers through systematic search and snowballing, our study results are likely not complete, threatening *Conclusion Validity*. As of this reporting of our results, we have not yet finished the process of snowballing, finding references through known related work. We have further opted not to snowball over publications found through systematic search. Our experience echoes an issue highlighted by Kitchenham & Brereton (2013), the high amount of effort required to undertake a Systematic Literature Review (SLR).

Our systematic search criteria may also be subject to critique, threatening *Construct Validity*. Initially, we hoped to include goal synonyms in our search string (e.g., intention, motivation); however, as many papers use these words outside the context of goal modeling, our search returned more than 10,000 results.

We had expected to find a higher number of overlapping papers between our systematic and snowballing results. This may indicate that our search string was not an effective representation of our desired scope, or it may be due to our incomplete snowballing process.

Several threats relate to the process of including, summarizing, and tagging papers, i.e. the *Internal Validity* of our results.

Publication Inclusion. The inclusion or exclusion of papers in our survey may be subjective or error prone. We have tried to mitigate this threat by defining and using clear inclusion and exclusion criteria (see Table 2), and by having at least two people read paper titles for relevance, with disagreements checked by reading the abstract and (optional) paper details. We undertook group discussions to decide on the inclusion or exclusion of several papers whose status was uncertain. However, our process may lean towards exclusion, as included papers were checked several times for inclusion, but papers excluded beyond the title stage were not always checked by another reader.

Cross-Citations. We have presented an analysis of the cross citations of papers within our roadmap, answering **RQ9**. However, our source for this citation information, the DBLP citation database, did not cover many of our included publications (35.7%). This may be as the DBLP database focuses mainly on Computer Science related publications, while several of the publications in our roadmap fall more in the area of Management or Business Information Systems. However, the citations for these different areas of papers are not well-linked in available resources. Although the DBLP database is not complete for our data, it is one of the best sources currently available. Other popular citation databases, such as those provided by Google Scholar, Scopus, and Web of Science, may have better coverage than the DBLP database used, but do not well fit our data analysis needs. Although these databases provide an online API to query the citations of a particular paper, they do not make their citation databases available for download. As such, cannot easily check whether there is a citation relationship between included papers.

Publications Summary. The summary information collected for papers could also be subjective. For example, the classification of approaches into horizontal or vertical transformations was particularly difficult, as well the distinction between extension and integration. We addressed these issues through group discussion, identifying clear examples of horizontal vs. vertical transformations and language extension vs. integration.

Tagging. Our tagging classifications may also be subjective, especially when moving away from the leaf tags in the taxonomy. We mitigated this issue in part by re-tagging publications using an agreed-upon taxonomy with clear paths up to more general tags. We have opted not to collect a formal method of reader or tagger agreement (e.g., Kappa measure). Instead of aiming for a high-level of initial agreement between readers and taggers, our strategy was to resolve ambiguous cases via group discussion. Although we noted our initial agreement for tags, especially "why" tags, was quite high (we disagreed on more than half of the papers) all tags eventually converged via pair-wise author discussion.

Author Experience. The authors of this study have significant experience in goal modeling (typically *i**-related languages). This helps to increase our confidence in the size of the initial core set of papers, but may also bias the survey coverage and literature review, i.e. threatening *External Validity*. As shown by the author network analysis, several authors of this paper are authors of papers covered via our roadmap. The chances of us considering our own papers for inclusion, especially through the snowballing process, are high. However, Figures 2 and 3 show that the surveyed publications cover a wide variety of source/target goal modeling languages, while Figures 11 and 12 show the inclusion of many authors beyond the authors of this paper. Such coverage is reflected both in our snowballing and systematic search results.

8. RELATED WORK

Literature Reviews in SE. We have created our roadmap by adopting the methods and approaches prescribed by Petersen et al. (2008), specifically focusing initially on a roadmap of available work, rather than a detailed survey, clearly defining our process of finding and including papers, making our research questions clear. Kitchenham et al. (2002) provide guidelines for empirical studies in software engineering, we apply many of these guidelines where applicable to our systematic mapping study, including clearly specifying a hypothesis (in our case research questions), defining populations (publications from snowballing or systematic search of specific databases), defining a process, providing raw data, and making extensive use of graphics.

Work by Kitchenham et al. (2010) performs a type of systematic "meta" review by reviewing and mapping SLRs in software engineering. This work argues that existing studies often fail to assess the quality of the surveyed publications. As part of the definition of mapping surveys as per (Petersen et al., 2008), in the mapping portion of this study, we have not undergone any explicit assessment of paper quality. Our deeper review has summarized the aims and methods of each paper, but assessing quality in this case is difficult. For this, we would need common evaluation criteria across a broad selection of work.

Further work by some of the same authors evaluates the utility of roadmap (mapping) studies specifically (Kitchenham et al., 2011). Here, the authors emphasize making the mapping results available, specifically the classification of each paper, allowing for study follow-up by other authors. In our case, we make such data available online.

Work by Pham et al. (2011) focuses on a social network analysis of computer science publications, investigating collaboration and citations, applying such analysis to the CAiSE conference series (Jarke, Pham, & Klamma, 2013). Pham et al. rank authors by using the Page Rank algorithm, which mainly considers how the authors' work is cited by others, while the author analysis included in Section 4 ranks authors according to the number of their included publications. Pham et al. also color nodes according to (sub)community, while this information is not yet an output of our roadmap analysis.

Related Literature Reviews. Past papers have made efforts to review and provide an overview of research in Goal-Oriented RE (GORE). Van Lamsweerde has provided a guided tour of the area as per 2001, motivating the use of goal-orientation and summarizing existing methods for modeling, specifying, and reasoning over goals.

Kavakli has provided an overview of existing goal-oriented frameworks as per 2002, classifying approaches in terms of RE Activities: elicitation, specification, and validation. She proposes a meta-model based framework for unifying goal-oriented approaches, bearing similarities with method engineering approaches (Brinkkemper, 1996),

Kavakli & Loucopoulos (2004) further extend their review of existing goal-oriented techniques, arguing, as we do, that research in the area is fragmented. Specifically, they argue that each goal-oriented approach focuses on only a subset of RE issues, without looking at the entire RE process, from elicitation to validation. Furthermore, research at that point had focused on representation and formatting, largely ignoring methodology. They also point out that few goal-oriented methods had focused on production of a specification from resulting models, generally supported by our results, observing that most of the work on GOL transformations appears after 2004. As various goal-oriented methods focused on different RE issues, the authors point out that benefit could be gained by the integration of different methods. However, our results show only a few papers (5) which have focused on transformations from one type of goal model to another, with most approaches focusing on transformations to/from goal models to an altogether different type of language or artifact.

Our roadmapping process found other SLRs which cover topics related to our scope and research questions. As these approaches are literature reviews and not roadmaps, they have a deeper analysis of relevant papers, but have a much narrower focus on fewer papers. Decreus, Snoeck, & Poels (2009) look at six techniques transforming *i** to business process models, while Assar & Souveyet (2012) review the use of the goal concept in eight approaches for web service discovery. Galaster et al. (2006) evaluate current approaches which address the gap between requirements and architecture, including several approaches starting from goals-oriented languages. Galster, Eberlein, & Moussavi (2009) further create a method for assessing and comparing approaches for transitioning from requirements to architecture. They use their method to assess 14 methods, two of which start from goal models. Work by Babar, Wong, & Gill (2011) considers the suitability of five existing goal-oriented frameworks for modeling strategic alignment from a management information systems perspective, using concepts from strategy maps to evaluate the suitability of goal models for this purpose.

Schneider et al. (2013) point out there is currently no a common understanding of the term “goal” within the discipline of Enterprise Architecture Modeling (EAM), although many researchers have confirmed the necessity of modeling goals. In order to tackle this problem, the authors survey 82 books, journals, and conference papers to investigate how literature defines “goal”. Based on their findings, they further sketch a list of properties to specify goals in EAM in order to provide comprehensive information and avoid ambiguity. Salgado, Machado, & Suzana (2013) identify three research areas that deal with the business and IS/IT alignment problem via a literature analysis, namely business model, strategy and goals, enterprise architecture. The authors propose to apply and empirically evaluate innovative approaches from these areas in order to improve the business and IS/IT alignment analysis.

Transformations are strongly related to the topic of requirements traceability. Nair et al. (2013) analyzed traceability in Requirements Engineering from a general viewpoint instead of taking a GORE perspective. They investigated the traceability papers published in the past twenty years in the RE conference. The authors primarily focus on the research trend of traceability in RE, including what kinds of artifacts have been considered. Their findings say that traceability has enjoyed more attention in the last six years (nearly 46% of traceability papers are published during this period), and post-requirements traceability (e.g., high-level to low-level requirements, requirements to design, requirements to code and requirements to testing) plays a key role in traceability research (nearly 50% of the traceability papers are on this track). Other interesting results such as tool features, top institutions and countries have also been presented. Unlike our roadmap, they do not include a deeper analysis of traceability endpoints, e.g., goal models to feature models, goal models to business processes.

Yang et al. (2014) perform a systematic review focusing specifically on requirements modeling and analysis techniques for self-adaptive systems (SAS). The authors examine four aspects: publication (when and where a paper is published), content (what RE modeling methods and quality attributes are studied), quality (which methods are better discussed and evaluated) and topic (generalized topics from studies). Two aspects of their research results are closely related to ours: (1) they classify RE activities for SAS into activities at requirements time (modeling/specifying requirements, adaptation mechanism and verification), activities at design time (map requirements model to architecture model or design decisions), and activities at runtime (runtime adaptation, verification, reconfiguration or evolution); (2) they associate modeling techniques to generalized topics, e.g., KAOS for modeling adaptation mechanism, LTL verifying adaptive programs and transition systems for verifying requirements at runtime. This is somewhat similar to our “why” hierarchy in the adaptation category. They investigated different RE techniques for only adaptation while we analyze GORE techniques for different applications that include, but are not limited to, adaptation, enhanced RE, and business analysis.

Amyot & Mussbacher (2011) perform a SLR of publications, finding 281 using the User Requirements Notation (containing the Goal-oriented Requirement Language (GRL)). The paper classifies surveyed work into 17 categories, several of which (e.g., Web Applications and Web Services, Transformations to Design Models, Feature Interaction Analysis, Aspect-oriented Modeling) fall under the scope of our survey. Our approach differs from this work by focusing more broadly on all goal-oriented languages and more narrowly on transformations, while presenting results as part of a roadmap, instead of a detailed comparison and evaluation. We have examined this closely related survey paper to mine related work included as part of our snowballing papers.

9. CONCLUSION

We have conducted a systematic study creating a roadmap of publications which transform goal modeling language to or from other software artifacts and/or models, summarizing publication details and trends answering **RQ0-11**.

We have performed some social network and citation analysis on authorship in this area. However, a deeper analysis examining how the reference structure between papers indicates what techniques build on other techniques, i.e., not just a reference but utilization, may be of greater interest.

A recent study performed by Bano et al. (2014) reviewed systematic reviews in RE and found a gap in reviews which cover goal-oriented techniques. As part of future work, we plan to apply the process as developed in this study to this broader research topic, examining research papers focusing on goal-orientation in RE. An additional future direction may be to adopt or integrate our approach with the approach of Wang et al. using goal-based conceptual modeling techniques (Know-How Mapping) to support an analysis of available literature (2014).

ACKNOWLEDGMENT

This work was supported in part by ERC advanced grant 267856, titled "Lucretius: Foundations for Software Evolution", http://www.lucretius.eu.

REFERENCES

Alencar, F., Marín, B., Giachetti, G., Pastor, O., Castro, J., & Pimentel, J. (2009). From i* requirements models to conceptual models of a model driven development process. In A. Persson & J. Stirna (Eds.), *The practice of enterprise modeling* (Vol. 39, pp. 99–114). Springer Berlin Heidelberg. Retrieved from; doi:10.1007/978-3-642-05352-8_9

Alencar, F., Marín, B., Giachetti, G., Pastor, O., Castro, J., & Pimentel, J. H. (2009). From i* requirements models to conceptual models of a model driven development process. In *The practice of enterprise modeling* (Vol. 39, pp. 99–114). Retrieved from; doi:10.1007/978-3-642-05352-8_9

Alrajeh, D., Kramer, J., Russo, A., & Uchitel, S. (2009). Learning operational requirements from goal models. In *Proceedings of the 31st international conference on software engineering* (pp. 265–275). IEEE Computer Society. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=1555042

Amyot, D., & Mussbacher, G. (2011). User requirements notation: The first ten years, the next ten years. *Journal of Software*, *6*(5), 747–768. http://ojs.academypublisher.com/index.php/jsw/article/view/0605747768 doi:10.4304/jsw.6.5.747-768

Angelopoulos, K., Souza, V. E. S., & Pimentel, J. (2013). Requirements and architectural approaches to adaptive software systems: A comparative study. In *Proceedings of the 8th international symposium on software engineering for adaptive and self-managing systems* (pp. 23–32). IEEE Press. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=2487343

Antón, A. I., Carter, R. A., Dagnino, A., Dempster, J. H., & Siege, D. F. (2001). Deriving goals from a use-case based requirements specification. *Requirements Engineering, 6*(1), 63–73. Retrieved 2014-09-28, from http://link.springer.com/article/10.1007/PL00010356

Antón, A. I., & Potts, C. (1998). The use of goals to surface requirements for evolving systems. In *Proc. of the 20th international conference on software engineering* (pp. 157–166). doi:10.1109/ICSE.1998.671112

Assar, S., Souveyet, C., & Aljoumaa, K. (2012). A goal-oriented perspective on approaches to web service discovery. In *Proc. of the 36th ieee annual computer software and applications conference workshops (compsacw12)* (p. 347-352). doi:10.1109/COMPSACW.2012.69

Babar, A., Wong, B., & Gill, A. (2011). An evaluation of the goal-oriented approaches for modelling strategic alignment concept. In *Proc. of the 5th international conference on research challenges in information science (rcis11)* (p. 1-8). doi:10.1109/RCIS.2011.6006838

Bandara, A. K., Lupu, E. C., Moffett, J., & Russo, A. (2004). A goal-based approach to policy refinement. In *Policies for distributed systems and networks, 2004. POLICY 2004. proceedings. fifth IEEE international workshop on* (pp. 229–239). IEEE. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1309175

Bano, M., Zowghi, D., & Ikram, N. (2014). Systematic reviews in requirements engineering: A tertiary study. In *Empirical requirements engineering (empire), 2014 ieee fourth international workshop on* (p. 9-16). doi:10.1109/EmpiRE.2014.6890110

Baresi, L., & Pasquale, L. (2010). Live goals for adaptive service compositions. In *Proceedings of the 2010 ICSE workshop on software engineering for adaptive and self-managing systems* (pp. 114–123). ACM. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=1808997

Bastos, L., & Castro, J. (2005). From requirements to multi-agent architecture using organisational concepts. *SIGSOFT Software Engineering Notes, 30*(4), 1–7. Retrieved from http://doi.acm.org/10.1145/1082983.1082980 doi:10.1145/1082983.1082980

Breaux, T. D., & Antón, A. I. (2005). Analyzing goal semantics for rights, permissions, and obligations. In *Requirements engineering, 2005. proceedings. 13th IEEE international conference on* (pp. 177–186). IEEE. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1531039

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, *8*(3), 203–236. doi:10.1023/B:AGNT.0000018806.20944.ef

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and Software Technology, 38*(4), 275 - 280. Retrieved from http://www.sciencedirect.com/science/article/pii/0950584995010599 (Method Engineering and Meta-Modelling) doi:10.1016/0950-5849(95)01059-9

Castro, J., Kolp, M., & Mylopoulos, J. (2001). A requirements-driven development methodology. In *Advanced information systems engineering* (pp. 108–123). Springer. Retrieved 2014-09-28, from http://link.springer.com/chapter/10.1007/3-540-45341-5_8

Castro, J., Kolp, M., & Mylopoulos, J. (2002). Towards requirements-driven information systems engineering: the< i> tropos</i> project. *Information systems, 27*(6), 365–389. Retrieved 2014-09-28, from http://www.sciencedirect.com/science/article/pii/S0306437902000121

Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2000). *Non-functional requirements in software engineering* (Vol. 5). Kluwer Academic Publishers. doi:10.1007/978-1-4615-5269-7

Cleland-Huang, J., Settimi, R., BenKhadra, O., Berezhanskaya, E., & Christina, S. (2005). Goal-centric traceability for managing non-functional requirements. In *Proceedings of the 27th international conference on software engineering* (pp. 362–371). ACM. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=1062525

Czarnecki, K., & Helsen, S. (2003). Classification of model transformation approaches. In Proc. of the 2nd oopsla'03 workshop on generative techniques in the context of model-driven architecture.

Dardenne, A., Van Lamsweerde, A., & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, *20*(1-2), 3–50. doi:10.1016/0167-6423(93)90021-G

De Landtsheer, R., Letier, E., & Van Lamsweerde, A. (2004). Deriving tabular event-based specifications from goal-oriented requirements models. *Requirements Engineering, 9*(2), 104–120. Retrieved 2014-09-28, from http://link.springer.com/article/10.1007/s00766-004-0189-1

Decreus, K., Snoeck, M., & Poels, G. (2009). Practical challenges for methods transforming i* goal models into business process models. In *Proc. of the 17th ieee international requirements engineering conference (re09)* (p. 15-23). doi:10.1109/RE.2009.25

Engelsman, W., Quartel, D., Jonkers, H., & van Sinderen, M. (2011). Extending enterprise architecture modelling with business goals and requirements. *Enterprise Information Systems, 5*(1), 9–36. Retrieved 2014-09-28, from http://www.tandfonline.com/doi/abs/10.1080/17517575.2010.491871

Estrada, H., Martínez, A., & Pastor, O. (2003). Goal-based business modeling oriented towards late requirements generation. In *Proc. of the 22nd international conference on conceptual modeling (er03)* (Vol. 2813, p. 277-290). Retrieved from doi:10.1007/978-3-540-39648-2_23

Galster, M., Eberlein, A., & Moussavi, M. (2006). Transition from requirements to architecture: A review and future perspective. In *Proc. of the 7th acis international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing (snpd06)* (p. 9-16). doi:10.1109/SNPD-SAWN.2006.73

Galster, M., Eberlein, A., & Moussavi, M. (2009). Comparing methodologies for the transition between software requirements and architectures. In Ieee international conference on systems, man and cybernetics (smc09) (p. 2380-2385). doi:10.1109/ICSMC.2009.5346368

Garlan, D., Cheng, S.-W., Huang, A.-C., Schmerl, B., & Steenkiste, P. (2004, October). Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer, 37*(10), 46–54. doi:10.1109/MC.2004.175

Giorgini, P., Massacci, F., Mylopoulos, J., & Zannone, N. (2005). Modeling security requirements through ownership, permission and delegation. In *Proc. of the 13th ieee international conference on requirements engineering*, 2005 (p. 167-176). doi:10.1109/RE.2005.43

Gordijn, J., Yu, E., & van der Raadt, B. (2006). e-service design using i* and e3value modeling. *IEEE software, 23*(3), 26–33. Retrieved 2014-09-28, from ftp://128.100.3.32/pub/eric/eric/IEEEsw06.pdf

Grau, G., Franch, X., & Maiden, N. A. (2008). PRiM: An i*-based process reengineering method for information systems specification. *Information and Software Technology, 50*(1), 76–100. Retrieved 2014-09-28, from http://www.sciencedirect.com/science/article/pii/S0950584907001115

Gross, D., & Yu, E. (2001). From non-functional requirements to design through patterns. *Requirements Engineering, 6*(1), 18–36. Retrieved 2014-09-28, from http://link.springer.com/article/10.1007/s007660170013

Harel, D., & Rumpe, B. (2004, October). Meaningful modeling: What's the semantics of "semantics"? *Computer, 37*(10), 64–72. Retrieved from doi: 10.1109/MC.2004.172

Horkoff, J., Li, T., Li, F.-L., Salnitri, M., Cardoso, E., & Giorgini, P. … Pimentel, J. (2014, May). Taking goal models downstream: A systematic roadmap. In *Research challenges in information science (rcis), 2014 ieee eighth international conference on* (p. 1-12).

Hui, B., Liaskos, S., & Mylopoulos, J. (2003). Requirements analysis for customizable software: A goals-skills-preferences framework. In *Requirements engineering conference, 2003. proceedings. 11th IEEE international* (pp. 117–126). IEEE. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1232743

iStarShowcase. (2011). *istar showcase '11: Exploring the goals of your systems and businesses, practical experiences with i* modeling*. Retrieved from http://www.cs.toronto.edu/km/istar/iStarShowcase_Proceedings.pdf

ITU-T, & Intern. Telecom. Union: Recommendation Z.151 (11/08). (2008). *Recommendation Z.151 (11/08) User Requirements Notation (URN) – Language definition* (Tech. Rep.). ITU-T – International Telecommunications Union. http://www.itu.int/rec/T-REC-Z.151/en

Jarke, M., Pham, M., & Klamma, R. (2013). Evolution of the caise author community: A social network analysis. In *Seminal contributions to information systems engineering* (pp. 15–33). Retrieved from; doi:10.1007/978-3-642-36926-1_2

Jian Wang, D. G. Arnon Sturm, & Yu, E. (2014). Know-how mapping: From i* to me-maps. In *7th international istar workshop, istar14.*

Kavakli, E. (2002). Goal-oriented requirements engineering: A unifying framework. *Requirements Engineering, 6*(4), 237-251. Retrieved from doi: 10.1007/PL00010362

Kavakli, E., & Loucopoulos, P. (2004). Goal modeling in requirements engineering: Analysis and critique. In *Information modeling methods and methodologies: Advanced topics in database research* (pp. 102–124). IGI Global.

Kazhamiakin, R., Pistore, M., & Roveri, M. (2004). A framework for integrating business processes and business requirements. In *Enterprise distributed object computing conference, 2004. EDOC 2004. proceedings. eighth IEEE international* (pp. 9–20). IEEE. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1342501

Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and Software Technology*, *55*(12), 2049–2075http://www.sciencedirect.com/science/article/pii/S0950584913001560. doi:10.1016/j.infsof.2013.07.010

Kitchenham, B., Dyba, T., & Jorgensen, M. (2004). Evidence-based software engineering. In *Proc. of the 26th international conference on software engineering (icse04)* (pp. 273–281). Retrieved from http://dl.acm.org/citation.cfm?id=998675.999432

Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., Emam, K., & Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, *28*(8), 721–734. doi:10.1109/TSE.2002.1027796

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering – a tertiary study. *Information and Software Technology*, *52*(8), 792–805http://www.sciencedirect.com/science/article/pii/S0950584910000467. doi:10.1016/j.infsof.2010.03.006

Kitchenham, B. A., Budgen, D., & Brereton, O. P. (2011). Using mapping studies as the basis for further research – a participant-observer case study. *Information and Software Technology*, *53*(6), 638–651http://www.sciencedirect.com/science/article/pii/S0950584910002272. doi:10.1016/j.infsof.2010.12.011

Kleppe, A., Warmer, J., & Bast, W. (2003). *Mda explained: the model driven architecture: practice and promise*. Addison-Wesley.

Kolp, M., Giorgini, P., & Mylopoulos, J. (2002). A goal-based organizational perspective on multi-agent architectures. In *Intelligent agents VIII* (pp. 128–140). Springer. Retrieved 2014-09-28, from http://link.springer.com/chapter/10.1007/3-540-45448-9_10

Lapouchnian, A., Liaskos, S., Mylopoulos, J., & Yu, Y. (2005). Towards requirements-driven autonomic systems design. In *ACM SIGSOFT software engineering notes* (Vol. 30, pp. 1–7). ACM. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=1083075

Lapouchnian, A., Yu, Y., Liaskos, S., & Mylopoulos, J. (2006). Requirements-driven design of autonomic application software. In *Proceedings of the 2006 conference of the center for advanced studies on collaborative research* (p. 7). IBM Corp. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=1188976

Lapouchnian, A., Yu, Y., & Mylopoulos, J. (2007). Requirements-driven design and configuration management of business processes. In *Business process management* (pp. 246–261). Springer. Retrieved 2014-09-28, from http://link.springer.com/chapter/10.1007/978-3-540-75183-0_18

Letier, E., Kramer, J., Magee, J., & Uchitel, S. (2008). Deriving event-based transition systems from goal-oriented requirements models. *Automated Software Engineering, 15*(2), 175–206. Retrieved 2014-09-28, from http://link.springer.com/article/10.1007/s10515-008-0027-7

Letier, E., & Van Lamsweerde, A. (2002). Deriving operational software specifications from system goals. In *Proceedings of the 10th ACM SIGSOFT symposium on foundations of software engineering* (pp. 119–128). ACM. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=587070

Liu, L., & Jin, Z. (2006). Integrating goals and problem frames in requirements analysis. In *Proc. of the 14th ieee international conference requirements engineering* (p. 349-350). doi:10.1109/RE.2006.34

Liu, L., & Yu, E. (2001). From requirements to architectural design-using goals and scenarios. In *ICSE-2001 workshop: From software requirements to architectures (STRAW 2001) may* (pp. 22–30). Retrieved 2014-09-28, from http://www.cin.ufpe.br/~straw01/epapers/paper11/paper11.html

Liu, L., & Yu, E. (2004). Designing information systems in social context: a goal and scenario modelling approach. *Information systems, 29*(2), 187–203. Retrieved 2014-09-28, from http://www.sciencedirect.com/science/article/pii/S0306437903000528

Maiden, N. A., Jones, S. V., Manning, S., Greenwood, J., & Renou, L. (2004). Model-driven requirements engineering: synchronising models in an air traffic management case study. In *Advanced information systems engineering* (pp. 368–383). Springer. Retrieved 2014-09-28, from http://link.springer.com/chapter/10.1007/978-3-540-25975-6_27

Markov, I., & Kowalkiewicz, M. (2008). Linking business goals to process models in semantic business process modeling. In *Enterprise distributed object computing conference, 2008. EDOC'08. 12th international IEEE* (pp. 332–338). IEEE. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4634785

Mazón, J.-N., Pardillo, J., & Trujillo, J. (2007). A model-driven goal-oriented requirement engineering approach for data warehouses. In *Advances in conceptual modeling–foundations and applications* (pp. 255–264). Springer. Retrieved 2014-09-28, from http://link.springer.com/chapter/10.1007/978-3-540-76292-8_31

Mens, T., & Van Gorp, P. (2006). A taxonomy of model transformation. *Electronic Notes in Theoretical Computer Science, 152*, 125–142. Retrieved from doi: 10.1016/j.entcs.2005.10.021

Mouratidis, H., & Giorgini, P. (2007). Secure tropos: a security-oriented extension of the tropos methodology. *International Journal of Software Engineering and Knowledge Engineering, 17*(02), 285–309. Retrieved 2014-09-28, from http://www.worldscientific.com/doi/abs/10.1142/S0218194007003240

Mouratidis, H., & Jurjens, J. (2010). From goal-driven security requirements engineering to secure design. *International Journal of Intelligent Systems, 25*(8), 813–840. Retrieved 2014-09-28, from http://onlinelibrary.wiley.com/doi/10.1002/int.20432/full

Nair, S., de la Vara, J. L., & Sen, S. (2013). A review of traceability research at the requirements engineering conference. *2013 21st IEEE International Requirements Engineering Conference (RE), 0*, 222-229. doi: http://doi.ieeecomputersociety.org/10.1109/RE.2013.6636722

Nguyen, D. C., Perini, A., & Tonella, P. (2008). A goal-oriented software testing methodology. In *Agent-oriented software engineering VIII* (pp. 58–72). Springer. Retrieved 2014-09-28, from http://link.springer.com/chapter/10.1007/978-3-540-79488-2_5

Pastor, O., & Molina, J. C. (2007). *Model-driven architecture in practice: A software production environment based on conceptual modeling*. Springer.

Penserini, L., Perini, A., Susi, A., & Mylopoulos, J. (2007). High variability design for software agents: Extending tropos. *ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2*(4), 16. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=1293736

Perini, A., & Susi, A. (2006). Automating model transformations in agent-oriented modelling. In *Agent-oriented software engineering VI* (pp. 167–178). Springer. Retrieved 2014-09-28, from http://link.springer.com/chapter/10.1007/11752660_13

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proc. of the 12th international conference on evaluation and assessment in software engineering (ease08)* (pp. 68–77). Retrieved from http://dl.acm.org/citation.cfm?id=2227115.2227123

Pham, M., Klamma, R., & Jarke, M. (2011). Development of computer science disciplines: a social network analysis approach. *Social Network Analysis and Mining, 1*(4), 321-340. Retrieved from doi: 10.1007/s13278-011-0024-x

Pimentel, J., Lucena, M., Castro, J., Silva, C., Santos, E., & Alencar, F. (2012). Deriving software architectural models from requirements models for adaptive systems: the STREAM-a approach. *Requirements Engineering, 17*(4), 259–281. Retrieved 2014-09-28, from http://link.springer.com/article/10.1007/s00766-011-0126-z

Pourshahid, A., Amyot, D., Peyton, L., Ghanavati, S., Chen, P., Weiss, M., & Forster, A. J. (2009). Business process management with the user requirements notation. *Electronic Commerce Research, 9*(4), 269–316. Retrieved 2014-09-28, from http://link.springer.com/article/10.1007/s10660-009-9039-z

Quartel, D., Engelsman, W., Jonkers, H., & Van Sinderen, M. (2009). A goal-oriented requirements modelling language for enterprise architecture. In *Enterprise distributed object computing conference, 2009. EDOC'09. IEEE international* (pp. 3–13). IEEE. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5277742

Rolland, C., Souveyet, C., & Achour, C. B. (1998). Guiding goal modeling using scenarios. *Software Engineering, IEEE Transactions on, 24*(12), 1055–1071. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=738339

Salgado, C., Machado, R. J., & Suzana, R. (2013). Modeling the alignment between business and is/it: A requirements engineering perspective. In Proceedings of the 28th annual acm symposium on applied computing (pp. 1309–1311). New York, NY, USA: ACM; Retrieved from http://doi.acm.org/10.1145/2480362.2480607, doi:10.1145/2480362.2480607

Santander, V. F., & Castro, J. F. (2002). Deriving use cases from organizational modeling. In *Requirements engineering, 2002. proceedings. IEEE joint international conference on* (pp. 32–39). IEEE. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1048503

Scheer, A.-W. (2000). *ARIS — business process modeling: Business process modeling*. Springer Science & Business Media. doi:10.1007/978-3-642-57108-4

Schneider, A., Schulz, C., & Matthes, F. (2013, July). Goals in enterprise architecture management – findings from literature and future research directions. In *Business informatics (cbi), 2013 ieee 15th conference on* (p. 284-291). doi:10.1109/CBI.2013.47

Seaman, C. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, *25*(4), 557–572. doi:10.1109/32.799955

Sebastiani, R., Giorgini, P., & Mylopoulos, J. (2004). Simple and minimum-cost satisfiability for goal models. In Advanced information systems engineering (Vol. 3084, p. 20-35). Springer. Retrieved from doi:10.1007/978-3-540-25975-6_4

Silva Souza, V. E., Lapouchnian, A., Robinson, W. N., & Mylopoulos, J. (2011). Awareness requirements for adaptive systems. In *Proceedings of the 6th international symposium on software engineering for adaptive and self-managing systems* (pp. 60–69). ACM. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=1988018

Souza, V., Lapouchnian, A., Robinson, W., & Mylopoulos, J. (2013). Awareness requirements. In R. de Lemos, H. Giese, H. M_ijller, & M. Shaw (Eds.), Software engineering for self-adaptive systems ii (Vol. 7475, p. 133-161). Springer Berlin Heidelberg. Retrieved from doi:10.1007/978-3-642-35813-5_6

Tallabaci, G., & Souza, V. E. S. (2013). Engineering adaptation with zanshin: an experience report. In *Proceedings of the 8th international symposium on software engineering for adaptive and self-managing systems. SEAMS 2013, san francisco, ca, usa, may 20-21, 2013* (pp. 93–102). Retrieved from http://dl.acm.org/citation.cfm?id=2487353

Van Lamsweerde, A. (2001). Goal-oriented requirements engineering: a guided tour. In *Requirements engineering, 2001. proceedings. fifth ieee international symposium on* (p. 249-262). doi:10.1109/ISRE.2001.948567

Van Lamsweerde, A. (2003). From system goals to software architecture. In *Formal methods for software architectures* (pp. 25–43). Springer. Retrieved 2014-09-28, from http://link.springer.com/chapter/10.1007/978-3-540-39800-4_2

Van Lamsweerde, A. (2004). Goal-oriented requirements engineering: a roundtrip from research to practice. In *Proc. of the 12th ieee international requirements engineering conference.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.6087\&rep=rep1\&type=pdf

Van Lamsweerde, A., & Letier, E. (2000). Handling obstacles in goal-oriented requirements engineering. *Software Engineering, IEEE Transactions on, 26*(10), 978–1005. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=879820

Vicente, K. (1999). *Cognitive work analysis*. Lawrence Erlbaum Associates.

Wang, Y., McIlraith, S. A., Yu, Y., & Mylopoulos, J. (2007). An automated approach to monitoring and diagnosing requirements. In *Proceedings of the twenty-second IEEE/ACM international conference on automated software engineering* (pp. 293–302). ACM. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=1321675

Weiss, M., Esfandiari, B., & Luo, Y. (2007). Towards a classification of web service feature interactions. *Computer Networks, 51*(2), 359–381. Retrieved 2014-09-28, from http://www.sciencedirect.com/science/article/pii/S138912860600212X

Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering, 11*(1), 102-107. Retrieved from doi: 10.1007/s00766-005-0021-6

Yang, Z., Li, Z., Jin, Z., & Chen, Y. (2014). A systematic literature review of requirements modeling and analysis for self-adaptive systems. In *Requirements engineering: Foundation for software quality* (pp. 55–71). Springer. doi:10.1007/978-3-319-05843-6_5

Yu, E. (1997). Towards modelling and reasoning support for early-phase requirements engineering. In *Proc. of the 3rd ieee international symposium on requirements engineering* (Vol. 97, pp. 226–235). doi:10.1109/ISRE.1997.566873

Yu, Y., do Prado Leite, J. C. S., Lapouchnian, A., & Mylopoulos, J. (2008). Configuring features with stakeholder goals. In *Proceedings of the 2008 ACM symposium on applied computing* (pp. 645–649). ACM. Retrieved 2014-09-28, from http://dl.acm.org/citation.cfm?id=1363840

Yu, Y., Lapouchnian, A., Liaskos, S., Mylopoulos, J., & Leite, J. C. (2008). From goals to high-variability software design. In *Foundations of intelligent systems* (pp. 1–16). Springer. Retrieved 2014-09-28, from http://link.springer.com/chapter/10.1007/978-3-540-68123-6_1

Yu, Y., Leite, J. C., & Mylopoulos, J. (2004). From goals to aspects: discovering aspects from requirements goal models. In *Requirements engineering conference, 2004. proceedings. 12th IEEE international* (pp. 38–47). IEEE. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1335662

Yu, Y., Wang, Y., Mylopoulos, J., Liaskos, S., Lapouchnian, A., & Sampaio do Prado Leite, J. C. (2005). Reverse engineering goal models from legacy code. In *Requirements engineering, 2005. proceedings. 13th IEEE international conference on* (pp. 363–372). IEEE. Retrieved 2014-09-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1531056

ENDNOTES

1. The type of study we perform is often called a systematic literature map. However, as the subject of our map includes mappings between models, we prefer the term roadmap.
2. http://goo.gl/tW2LlO
3. http://goo.gl/hUCMKG
4. http://goo.gl/tW2LlO
5. http://goo.gl/fC72c0
6. http://arnetminer.org/citation

Jennifer Horkoff is the holder of a 2-year Marie Sklodowska Curie Intra-European Fellowships for career development (IEF) at the Centre for HCID. She is also a holder of a Natural Sciences and Engineering Research Council of Canada Postdoctoral Fellowship. Jennifer received her Ph.D. in Computer Science from the University of Toronto. Her dissertation, under the supervision of Dr. Eric Yu, focused on interactive analysis for agent-goal models. She has been an author or coauthor of more than 20 papers in peer-reviewed journals, conferences, or workshops. Her research interests lie in enhancing the use of conceptual modeling for requirements engineering, including the development of interactive analysis techniques, model uncertainty, scalability issues, and the application of conceptual modeling to business intelligence. She is on the program committee of several international conferences in her field, including RE, and CAiSE, has been on the organizing committee of RE, and has been a co-organizer organizer of several workshops, including iStar, RIGiM, and MReBA.

Tong Li is a PhD candidate at the Department of Information Engineering and Computer Science, University of Trento. He works as a research member of the ERC-Lucretius project under the supervision of Prof. John Mylopoulos, and is co-supervised by Prof. Fabio Massacci. His research interests are in the areas of Security Requirements Engineering and Conceptual Modeling, concentrating on analyzing security requirements of socio-technical systems by using modeling and reasoning techniques.

Feng-Lin Li is currently a Ph.D. Student at the Department of Information Engineering and Computer Science, University of Trento, Italy. His research interest is mainly on Requirements Engineering, Conceptual Modeling and Software Services.

Mattia Salnitri is a PhD candidate at department of Information Engineering and Computer Science of University of Trento, Italy. He received his bachelor's degree in computer science from University of Trento in 2009 and his master's degree in computer science from University of Trento in 2011. His research interests include software engineering, requirement engineering, security requirement enforcement and alignment, business processes.

Evellin C. S. Cardoso is a Ph.D. candidate at the Department of Information Engineering and Computer Science (DISI) at the University of Trento (Italy). She holds a M.Sc. degree in Computer Science from the Federal University of Espírito Santo (Brazil). Her main research interests include goal and business process modeling, service-oriented architecture, enterprise architecture and agent-oriented methodologies.

Paolo Giorgini is associate professor and head of the Software Engineering, Formal Methods and Security group at the Department of Engineering and Computer Science of University of Trento. He received his Ph.D. degree from the Computer Science Institute of University of Ancona - Italy - (1998) and then he joined the University of Trento as assistant professor. He has worked on the development of requirements and design languages, and the application of agent and goal-oriented techniques to (security) software analysis. He co-funder of the International Journal of Agent-Oriented Software Engineering (IJAOSE) and his publication list includes more than 180 refereed journal and conference proceedings papers and twelve edited books. He has worked and coordinated a number of provincial, national and international research projects and he has contributed to the organization of international conferences as general chair, program chair and program committee member.

John Mylopoulos holds a distinguished professor position (chiara fama) at the University of Trento, and a professor emeritus position at the University of Toronto. He earned a PhD degree from Princeton University in 1970 and joined the Department of Computer Science at the University of Toronto that year. His research interests include conceptual modelling, requirements engineering, data semantics and knowledge management. Mylopoulos is a fellow of the Association for the Advancement of Artificial Intelligence (AAAI) and the Royal Society of Canada (Academy of Sciences). He has served as programme/general chair of international conferences in Artificial Intelligence, Databases and Software Engineering, including IJCAI (1991), Requirements Engineering (1997), and VLDB (2004). Mylopoulos was recently awarded an advanced grant from the European Research Council for a project titled "Lucretius: Foundations for Software Evolution".