Advancing Repeated Research in Requirements Engineering: A Theoretical Replication of Viewpoint Merging

Nan Niu*, Amy Koshoffer†, Linda Newman†, Charu Khatwani*, Chatura Samarasinghe*, and Juha Savolainen‡
* Department of Electrical Engineering and Computing Systems, University of Cincinnati, USA
† University of Cincinnati Libraries, University of Cincinnati, USA
‡ Head of Software Architecture, Roche Diagnostics, Switzerland
{nan.niu, amy.koshoffer, linda.newman}@uc.edu, {khatwacu, samaraca}@mail.uc.edu, juha.savolainen@roche.com

***Abstract*—Compared to building a single requirements view, modeling stakeholder viewpoints and then merging them is shown to improve the understanding of the problem domain, but also very time consuming. How has the situation changed? This paper reports our replication of a case study, where we take theoretical replication's advantage to mitigate the original study design's threat and to embrace an important evolving factor, namely automated tool support for producing i^* models. Our replicate case study verifies the rich domain understanding gained through viewpoint-based modeling, and updates the prior results by showing the time saving enabled by the tool. Our work offers operational insights into independent, theoretical replications. These insights, we believe, can advance requirements engineering research toward an empirically backed body of knowledge.**

***Index Terms*—Replication, theoretical replication, case study, Scholar@UC, viewpoints, model merging, i^*, tech transfer.**

I. Introduction

In no science or engineering discipline should one accept knowledge on the basis of the effects and observations reported in a single study. Being able to repeat experiments is a hallmark of the scientific method, used to confirm or refute hypotheses and previously obtained results. In software engineering, replications allow us to build knowledge about which results or observations hold under which conditions [1].

The need for repeated research is especially high for requirements engineering (RE) due to the strong influence posed by the stakeholders and the contextual factors [2]. Effects about an RE approach observed in a single study may be caused by factors that were not measured or controlled, such as variability in human behavior, difficulty of isolating confounding variables, and researcher bias [3, 4]. The aim of replication is to examine the extent to which a published study's results are valid, reliable, and useful in RE practice.

To scale empirical RE research to practice, Wieringa [5] suggested two generalization dimensions: statistical inferencing from samples to populations, and case-based reasoning that tests the underlying theory under increasingly realistic conditions. An example of the former is replicating the original Siemens case study [6] with two other partners (BMW and Cassidian) [2] so as to verify and broaden the benefits of artifact-based RE across different domains (populations).

The sample-based inference is often made by a *literal replication* [3] whose objective is to execute close enough to the original experiment so that the results can be directly compared. In contrast, case-based reasoning corresponds to *theoretical replication* [7] which seeks to investigate the theory's scope of applicability and to update the assumptions that have evolved greatly since the initial studies. For example, to test the theory concerning a linguistic tool's superior performance over a baseline method in supporting the requirements consolidation task [8], Wnuk *et al.* [9] performed a replication by changing the baseline method from the research prototype's simple keyword searching to the advanced searching and filtering capabilities offered in DOORS, a state-of-the-practice requirements management tool. In this way, the theory was tested in a more realistic setting, rather than by sticking rigidly to the original experimental setup.

Theoretical replications, therefore, play a key role in technology transfer by assessing whether the predictably (dis)similar results hold when conditions are systematically altered [3, 5]. However, beyond [9], there are very few theoretical replications published in RE. The survey by Sjøberg *et al.* [10] showed that only 20 of their 103 (18%) reviewed software engineering experiments were replications, and only one replication was RE related. This leaves many questions about theoretical replication unanswered. For example, which conditions should be altered, what evolving factors ought to be incorporated, and with these changes, how much adaptation is needed to test the original hypothesis?

In an attempt to answer such questions, and more importantly, to gain operational insights into theoretical replications in RE, we performed one ourselves. We selected an exploratory case study reported in [11] where Easterbrook and his colleagues tested a key tenet of the viewpoints theory, namely, modeling stakeholder viewpoints separately and then explicitly merging them leads to a richer domain understanding than constructing a single coherent requirements model. The largely positive results, though obtained qualitatively and subjectively, were having some long-lasting impacts, influencing work in RE [12, 13] as well as in the broader areas of information systems and enterprise modeling [14, 15]. Easterbrook

et al.'s study of viewpoint merging has even begun to shape new and challenging RE domains like health care [16]. For these reasons, we believe repeating the test of the underlying viewpoint theory and doing so in a more quantitative and objective manner is a worthwhile endeavor.

This paper makes two main contributions: We integrate a state-of-the-art tool into the goal modeling process, and we develop a new way to evaluate the goal modeling product. Our study not only updates the previously obtained results [11] in the face of potentially practical support, but also illuminates that theoretical replication is fruitful in advancing RE research toward an empirically backed body of knowledge. In what follows, we present background information in Section II, detail our replication design in Section III, analyze the results in Section IV, and draw some concluding remarks in Section V.

II. Background

A. Repeated Research in RE

The idea behind establishing software engineering's empirical foundations is to separate "what is actually true" from "what is only believed to be true", and in doing so to build knowledge [17]. Clearly no single study has the independent power to produce definitive answers for separating truth from belief. Therefore, replication of previously published empirical studies is frequently advocated [1, 17]. Repeated research, as it turns out, takes many forms.

Probably the most well-known distinction in software engineering is between *internal* and *external* replications, as defined in [17]. Internal replication is undertaken by the original researchers themselves or the team involving them, whereas external replication is performed by independent researchers. Brooks *et al.* [17] pointed out that, without the confirming power of external replication, many principles and guidelines in software engineering should be treated with caution.

In mature scientific disciplines, external replication is a must. A recent remarkable discovery in physics exemplifies this: Even though the gravitational waves were detected in September 2015, the news was kept secret until February 2016 after the results were independently verified [18]. Unfortunately, external replication is still rare in RE. Although the number of software engineering replications was updated from 20 in Sjøberg *et al.*'s survey [10] to 133 in da Silva *et al.*'s study [19], 31 of the 32 RE replications (97%) were internal ones. Our repeated research, carried out as an external replication, addresses the critical need by checking whether the published RE evidence is able to stand scrutiny.

Who replicates the experiment is only one of the permissible changes in repeated research. Others include what and how to measure, whether to use the same materials, and so forth [20]. Mendonça *et al.* [21] advocated careful control over the variabilities and suggested to abort a replication if its planning deviates too much from the original experiment. Contrariwise, Juristo and Vegas [22] proposed a "run-and-see" approach by encouraging a replication's actual execution and post-treatment analysis rather than abandoning an otherwise useful study with context-induced changes.

These opposing views can be explained by the difficulty in replicating human-subject studies in software engineering [3]. The context of each study can easily cover tens and hundreds of variables [22]. For example, programmer productivity has been linked to more than 250 contributing factors [23], and in RE, an independent review uncovered 8 potential confounding variables of Wnuk *et al.*'s replication mentioned earlier [9]. Performing an identical, exact, literal, strict, or even close replication in the same way as the natural sciences like physics is neither practically attainable nor methodologically advantageous for software engineering [3, 17, 22, 23].

In contrast to following the original experimental procedures as closely as possible, theoretical replication takes advantage of the opportunities to improve the study design. Moreover, the improvement is made to advance the body of knowledge in a systematic way, e.g., by addressing a serious threat, updating a response variable's measuring, or embracing a key change in the context of the phenomena under investigation. Such an advancement is illustrated by the aforementioned change of the baseline requirements consolidation tool to DOORS in Wnuk *et al.*'s replication [9]. Referring to Juristo and Vegas's "run-and-see" motto [22], we believe theoretical replication can achieve a "run-and-see-big" effect by selecting the critical factors to re-examine the underlying theory.

In summary, replications are essential to constructing and evolving knowledge in RE. Although the number of published replications has grown in the last few years, there is a pressing need to conduct *external* RE replications [19]. Theoretical replication, compared with literal replication[1], can potentially improve the repeated study's quality because the researchers can pursue a less contrived design and execution.

B. Replication Base

A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident [7]. Case studies have gained considerable acceptance in software engineering, and so have their replications. In da Silva *et al.*'s surveyed 133 replications, for example, 15 (11%) are replicate case studies, growing steadily from 1 in 1999 to 4 in 2009 [19]. In RE, the distinctive need for case studies arises out of the desire to understand the complex *environment* in which the requirements are located [24, 25].

This environment is constantly changing. To structure the evolving requirements, *viewpoints* are proposed to partition a large information space into loosely coupled yet overlapping chunks ("viewpoints") [26]. Although viewpoints are believed to produce better requirements models, one of the first empirical tests is [11] which serves as our replication base.

The original case study explored the underlying theory of viewpoints: "When approaching a conceptual modeling problem, it is better to build many fragmentary models representing different perspectives than to attempt to construct

[1]We use "theoretical" and "literal" replications for the rest of this paper in the same way as [3]. Please refer to [20] for a review of various replication types in experimental disciplines.

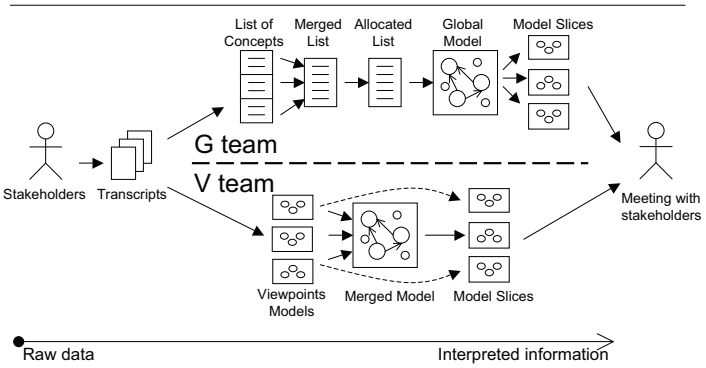


Fig. 1. Original study's design (*adopted from [11]*).

a single coherent model" [11]. "Better" was translated to "a richer domain understanding" and further operationalized by 3 response variables: "hidden assumptions", "disagreements between stakeholders", and "new requirements". To test the predicted differences, one team followed a global (G) approach, whereas the other team adopted viewpoints (V), to build i^* models [27] for the Kids Help Phone (KHP) organization. The fundamental distinction was model *merging* that was explicit for the V team but nonexistent for the G team. Fig. 1 shows the original study's design. Table I helps explain the key components of this design. We summarize the original study's main findings as follows.

- $\mathbf{R_1}$: Viewpoints led to a richer domain understanding. While the benefits of viewpoints were observed, there lacked detailed and quantitative analyses (especially those of the 3 response variables) in [11].
- $\mathbf{R_2}$: Viewpoints-based modeling was slower. In fact, it was so time-consuming that the V team was *not* able to produce their merged i^* model. In Fig. 1, only the slices, rather than the integrated whole, from both teams were compared and presented to the KHP stakeholders.
- $\mathbf{R_3}$: Process was more important than product. This could be seen as a combination of $\mathbf{R_1}$ and $\mathbf{R_2}$. On one hand, the *process* of merging stakeholder viewpoints did improve the understanding of the problem domain [11]. On the other hand, the merged *product* never existed, due to the lack of modeling tool support for handling i^* syntax [11].

In summary, not only were the viewpoints theory and hypothesis stated clearly in [11], but the results were thought-provoking, including such startling claims that promoted a requirements modeling process even though no end result was produced. This stands in stark contrast to artifact-based RE which values the requirements tangibles rather than the way of creating them [6]. Nevertheless, Easterbrook *et al.*'s study design was straightforward and sound. Their work [11] also appeared to be influential, especially in meeting some emerging RE challenges [12]–[16]. For these reasons, we believe Easterbrook *et al.*'s work [11] is a study worth replicating.

III. Replication Design and Execution

Our theoretical replication investigates the same central hypothesis as the original study: "Modeling stakeholder viewpoints separately and then combining them leads to a richer understanding of the domain" [11]. Furthermore, we take theoretical replication's advantage to improve the study procedure in three aspects.

TABLE I
Original study's design explained

Study Context	Kids Help Phone (KHP), a non-profit social organization that provides counseling to kids and their parents across Canada
Study Period	around 2004
Organizational Need Related to the Study	KHP wanted to analyze the strategic technology change of developing new internet-based services
Modeling Input	transcripts from interviewing 14 KHP stakeholders (approx. 140 pages in total)
V Team	viewpoint modeling team consisting of 3 graduate students
G Team	global modeling team consisting of 2 graduate students
Modeling Output	team-based i^* models

- **Mitigate a threat.** The original study collected purely qualitative data, and relied on the subjective opinions of the modelers to measure "a richer domain understanding". In contrast, we examine 3 finer measures — "hidden assumptions", "stakeholder disagreements", and "new requirements" — which were laid out but not analyzed in [11]. In our replication, these 3 response variables are assessed by the domain experts rather than by the modelers themselves, reducing the experimenter bias.
- **Take into account an evolving factor**. Among the many things changed from the original study, we intentionally incorporate the i^* tool support in our replication. In [11], both the G and V teams used Microsoft Visio for the modeling. While the V team failed to build the merge, both teams encountered difficulty with Visio in managing large, evolving models. In the past decade, i^* tooling has greatly increased. The community wiki, for example, lists over 20 tools, many of which are open-source [28]. We choose OpenOME [29] to update the study design and describe this tool in more detail in Section III-B.
- **Devise a new mechanism to evaluate i^* models**. Unlike the original study's focus on the *internal* qualities of the models, such as size and readability [11], we resort to the domain expert by eliciting a set of questions from the expert and then assessing how well the resulting i^* models are capable of answering those questions. We refer to such an approach as an *external* way of evaluating i^* models. Horkoff and Yu [30] recently presented an external framework for interactive i^* model analysis, which we discuss further, along with other goal model evaluation approaches, in Section III-C.

A. Replication Context

We adopt case study [7] as the basis for our replication design. The contemporary phenomenon of our investigation is the Scholar@UC project [31]. Scholar@UC is a digital repository that enables the University of Cincinnati (UC) community to share its research and scholarly work with a worldwide audience. Its mission includes preserving the

Submission 21 – Type of Work # Early Adopter

As a: repository submitter

I want: to be able to upload a video

So that: my content will be viewable

Done looks like: a format option in the input form that includes video

Fig. 2. Example user story of Scholar@UC (*adopted from [33]*).

permanent intellectual output of UC (e.g., publications, presentations, datasets, etc.) and enhancing discoverability of these resources. UC faculty and students, for instance, can use Scholar@UC to store, organize, and distribute their scholarly creations in a durable and citable manner.

The development of Scholar@UC evolved from a couple of legacy Web systems and began over a year ago in partnership by the UC Libraries and UC Information Technologies. The principal technological platforms are the Fedora Commons repository architecture, Apache Solr server, Ruby on Rails engine, and Blacklight interface [31]. Scholar@UC is made open source on GitHub [32], and its project team follows agile development, employing such practices as sprint iterations (each cycle typically covers 2 weeks) and scrum stand-ups (roughly 3 meetings per week).

The requirements of Scholar@UC over the past 2 years were shaped by the *early adopters*, a group of more than 30 enthusiastic users (faculty, graduate students, and library staff) who provided in-depth feedback on the functionalities of the software and helped set priorities for the development at various points. As a result of engaging these user representatives in the agile process, user stories were developed and released in GitHub [33]. A sample user story is shown in Fig. 2.

We collaborated with the Scholar@UC team to share the expertise in software engineering research and practice. In particular, the Scholar@UC artifacts provide a valuable real-world dataset for research. Meanwhile, the research findings can feed back into the project practice, leading to greater awareness of the state-of-the-art and more informed decision-making. In the fall of 2015, we designed our theoretical replication of viewpoint merging with the specific aim of deepening the understanding of the problem domain. This fit Scholar@UC's plan as the project was transitioning from early adoption toward institution-wide self-submissions[2].

As illustrated in Fig. 2, over a hundred user stories were elicited and documented in an agile way [34]: *who* it is for, *what* it expects from the intended software, *why* it is important, and optionally, *how* it is delivered. The user stories were organized into 11 categories [33]: data management, digital archives, publishing, etc. Linking these user stories could help consolidate the stakeholder roles, identify their intentional dependencies, and uncover possible inconsistencies and incompleteness. This made i^* an appropriate modeling framework due to its built-in constructs emphasizing strategic relationships among organizational actors [27].

[2]"Scholar@UC Open for Self-Submissions" was officially announced on February 3, 2016. Please see http://www.uc.edu/News/NR.aspx?id=22818 .

TABLE II
DESIGN OF OUR REPLICATE CASE STUDY

Study Context	Scholar@UC, an institution-wide Web application that supports preservation and access for digital scholarly works
Study Period	late 2015 – early 2016
Organizational Need Related to the Study	Scholar@UC supported research on consolidating existing user stories and wanted to use research findings to help deepen the domain understanding
Modeling Input	user stories, some elicited from early adopters (134 user stories & 49 pages in total)
V1 Team	viewpoint modeling team1 consisting of 2 undergraduate & 1 graduate students
V2 Team	viewpoint modeling team2 consisting of 3 undergraduate students
G1 Team	global modeling team1 consisting of 3 graduate students
G2 Team	global modeling team2 consisting of 4 undergraduate students
Modeling Output	team-based *i** models

We recruited 13 UC students from a split-level RE class to participate in our study. The students did not know Scholar@UC before the class. All of them were familiar with i^* syntax based on the class's earlier readings [27, 35], but none had learned OpenOME or any other automated i^* modeling tools. As their i^* experiences were similar, we randomly assigned the student modelers into 4 groups and further divided the groups into 2 G (global modeling) teams and 2 V (viewpoint modeling) teams. Table II presents our study design, which is to be contrasted with Table I. Note that Scholar@UC keeps evolving its artifacts including the user stories. The version that served as our modeling input, together with all other study materials, is made publicly accessible in [36], facilitating future replications.

B. OpenOME: Tech Transfer from a Research Prototype

We required all 4 teams to use OpenOME to produce their i^* models, updating an important factor from the original study. OpenOME supports modeling of the social and intentional aspects of a system, allowing users to capture the motivations behind system development in a graphical form [30]. OpenOME extends the Organizational Modeling Environment (OME) which is part of the Tropos project [37]. To enlarge the user base, OME was made open source in the spring of 2004 and hence renamed to OpenOME. Since then, many researchers and students have contributed to its development.

The latest version of OpenOME operates on the Eclipse platform. The main features exploited by the modelers in our study are editing-related and shown in Fig. 3. By simply dragging and dropping items from the palette, for example, one can generate and edit an i^* model within the canvas. In addition, our modelers benefited from OpenOME's interoperability, downloading and successfully running the tool on Windows, Linux, and Mac computers.

In our opinion, OpenOME has grown from a research prototype to a community asset. Not only does the tool maintain a record of sustained downloads[3], but its users are able to

[3]In the first two months of 2016, for example, OpenOME received 38, 7, 28, 18, 40, 81, 15, and 55 weekly downloads [29].

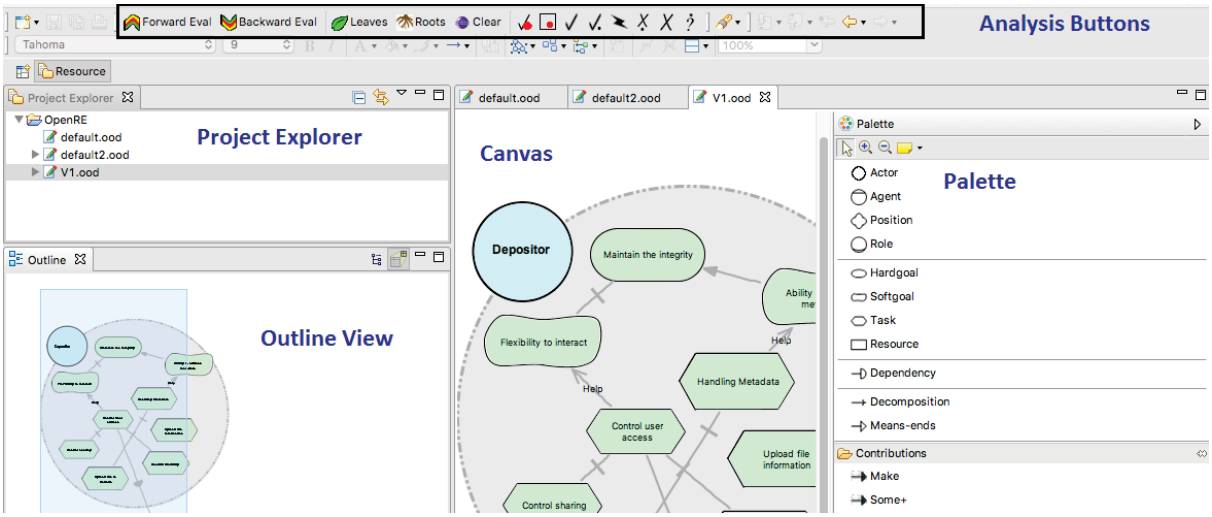


Fig. 3. Screenshot of the OpenOME tool highlighting the editing-related features.

build enterprise architectures, create ontological visualizations, monitor early aspects, and tackle other problems [38, 39, 40]. The diverse and independent usages clearly signal a trajectory of successful technology transfer [5] for OpenOME.

C. Evaluating i^* Models

To evaluate i^* models, we must understand what i^* goal-oriented modeling is trying to achieve. Broadly speaking, i^* models are intended to facilitate requirements exploration with an emphasis on social aspects by providing a graphical depiction of system actors including their intentions, dependencies, and alternatives [27, 30]. Five evaluation categories exist: analyzing goal satisfaction or denial, computing model metrics, planning action sequences, simulating model behavior, and model checking formal properties [41]. The evaluation of our replication base falls mostly into the metrics computing category, assessing measures such as the i^* model sizes in terms of the number of nodes and the number of edges [11].

Recently, Horkoff and Yu [30] introduced two procedures for analyzing i^* models: *forward* analysis addressing "what if?" types of questions so that the alternatives can be compared, and *backward* analysis answering "are certain goals achievable?" questions. These procedures are implemented in OpenOME, shown by the "Analysis Buttons" in Fig. 3. The domain experts are encouraged to interact with the OpenOME analysis features to iteratively improve the i^* models, e.g., by uncovering ambiguity and incompleteness.

We propose in our work a similar approach by engaging experts in identifying the questions that are important for domain understanding. Different from [30], our approach is non-interactive. The questions are defined by domain experts without being constrained by the content and layout of any specific model. The questions are then answered by analysts or researchers who are familiar with i^* syntax and semantics. We believe this can provide the best of both worlds, allowing stakeholders and modelers to do what they do best. The questions resulted from our approach can be used to carry out what Horkoff and Yu [30] described as "sanity check" to test if the produced i^* models are sensible or not, before interactive and/or formal analyses are performed. This question-asking and question-answering divide stems from our view that, during the early stages of requirements exploration, the i^* models are a means to an end — to gain a richer understanding about the problem domain — rather than the end itself.

D. Replication Execution

Our replicate case study aims to answer 3 research questions: (1) Can we confirm the prior results: $\mathbf{R_1}$, $\mathbf{R_2}$, and $\mathbf{R_3}$ (cf. Section II-B)? (2) How does the OpenOME tool affect the modeling process? and (3) How well can the resulting i^* models answer the stakeholder questions?

We introduced the i^* modeling task to the 4 teams in November 2015. The introduction was made separately to each modeling team without any other team's presence. As a result, the modelers were not exposed with the G-V process difference, the viewpoint theory, or the study hypothesis. Every team was instructed to use [33] as the only source for their modeling, and to use OpenOME to construct their i^* models throughout their work. For the G1 and G2 teams, all members were asked to work together from day one. For the V1 and V2 teams, the modelers were required to divide existing Scholar@UC requirements artifacts [33] as a group, use divided input to build viewpoint models individually, and merge the viewpoints collectively.

All the 4 teams were given 3 weeks to complete the modeling. After that, a meeting with Scholar@UC stakeholders was held, during which the final i^* models of all 4 teams were presented in foam boards, and the domain experts, modelers, and researchers exchanged feedback in an open format.

In addition to the i^* models, all 4 teams were instructed to submit 3 lists: "hidden assumptions", "stakeholder disagreements", and "new requirements", as well as detailed data tracking their modeling efforts. These instructions can be found in our study packet [36]. Note that, compared to Fig. 1, our study execution had two main differences: our G and V teams had exactly the same modeling input (namely [33]), and it was the final integrated model from each team (instead of model slices) that was presented in the stakeholder meeting.

TABLE III
NUMBER OF RAW AND RATED DOMAIN-UNDERSTANDING ITEMS

Team		G1	G2	V1	V2
hidden assumptions	raw #	5	8	9	4
	rated #	3	3	9	3
stakehoder disagreements	raw #	2	6	5	4
	rated #	2	4	2	2
new requirements	raw #	3	5	7	7
	rated #	3	4	5	5

IV. RESULTS

A. Problem Domain Understanding: Richer or Not?

For each of the 3 response variables used to operationalize "a richer domain understanding", we collected the modeling team's data directly from their submissions. By relating to the submitted i^* models from the teams, two researchers then jointly processed the raw data to filter out the items that were insensible or of low quality. Sample removed and preserved items are listed below (more information is available in [36]).

- The hidden assumption "Devs[4] know things about stuff" is clearly too general to sensibly help domain understanding, which we filtered out. Another submission from the same team, "Digital archivist is the moderator of every repository", makes an assumption about the responsibility and rights of a stakeholder role, which we kept.
- The reported disagreement "It is unclear what the approval process should be for collections" looks more like under-specification than lack of consensus to us, so we removed it. In contrast, we felt that the tension between the "proxy service desired by archivist" and "repository user's usability" reflected a sensible stakeholder disagreement, so we kept it.
- "Create a glossary of terms so that there is less confusion for requirements documenting" may be needed internally to the project team, but would not count as a new requirement for the Scholar@UC system itself. "Download multiple works at a time", to us, would count. We therefore discarded the former and kept the latter.

The preserved items were presented to Scholar@UC domain experts and assessed in two different ways: interview and survey[5]. Because hidden assumptions and stakeholder disagreements were contextually rich, we conducted an interview with one expert (a science informationist) to obtain qualitative ratings and justifications[6]. Because new requirements were relatively self-contained, we designed an online survey to collect ratings from a broader and more diverse group of project members. Table III lists the number of raw and rated items. No team, according to Table III, seemed to outperform the others in terms of domain understandings' quantities. We next compare their qualities.

[4]Here "Devs" mean "development engineers" as used in DevOps. DevOps is the practice of operations and developers participating together in the entire service lifecycle, from design through development to production support.

[5]The rating items were completely anonymized (i.e., containing no modeling-team information) in both the interview and the survey.

[6]The interview lasted about 1 hour involving the expert and one researcher.

Hidden Assumptions. For hidden assumptions, in addition to being valid and non-obvious, we wanted them to assert indicative environmental properties, as defined by Jackson [24]. Such problem-domain conditions, events, and states are critical to the operation of the intended software. As shown in Table IV, what the V teams produced were more about environmental assertions. These included "time frame is not necessary for assigning permission from consumer to depositor" (V1) and "depositor can achieve same level of integrity in downloading small chunks as the large ones" (V2). Neither assumption touched upon implementation details, and both were deemed very hidden. In general, the domain assumptions resulted from the V teams received higher ratings in terms of the environmental indicativeness and hiddenness, shown in Table IV.

The V teams' domain assumptions, however, were less valid compared to the G teams'. Referring to the above records, V2's assumption about the downloading integrity was valid whereas the time-oblivious permission assertion made by V1 was not. The G teams, overall, made more sound assumptions about Scholar@UC. For instance, all the G2's rated assumptions — "uploaded data is readable", "system is secure", and "system has enough permission rights" — were assessed as correct by the domain expert, though their hiddenness was virtually nonexistent in that their average rating is 1.33 in Table IV where 1 indicates "completely obvious".

We conclude that the V teams outperformed the G teams in generating hidden assumptions. While what the V teams found might not always be factually correct, their assumptions were both more about the intrinsic properties of the problem domain and more concealed. Thus, we believe that at the stage of requirements exploration it is crucial to surface the less than perfect environmental assertions that otherwise would be kept out of stakeholders' sight.

Stakeholder Disagreements. Disagreements between Scholar@UC stakeholders could occur at different levels: syntactic, semantic, and pragmatic. Although we do not claim that one level is a prerequisite for another, they are clearly not disjoint. Table IV lists all these levels, together with the severity and validity of the reported disagreement, as perceived by the domain expert that we interviewed.

A syntactic disagreement indicates that some well-formedness rule is broken when Scholar@UC requirements are stated. G1's two rated disagreements: "Who should nominate the URL for a work (Depositor or Repository User)?" and "Are Metadata Specialist and Digital Archivist the same in assuring work attribute quality?" identified the overlapping and potentially conflicting information presented in Scholar@UC's user stories [33]. Consequently, G1's results received the 3 out of 3 rating on 'Syntactic' in Table IV, which is better than all the other three teams.

Semantic disagreements go beyond the syntax and signal inconsistencies relating to meaning. The aforementioned disagreement: "proxy service desired by archivist" versus "repository user's usability" submitted by V2 is an instance of semantic disagreements, as well as an instance of pragmatic

TABLE IV
ASSESSING HIDDEN ASSUMPTIONS AND STAKEHOLDER DISAGREEMENTS (ALL RATINGS ARE DONE QUALITATIVELY ON A 3-POINT LIKERT SCALE WHERE 3 INDICATES THE POSITIVE END, 2 INDICATES NEUTRAL, AND 1 INDICATES THE NEGATIVE END)

Team	Average rating of hidden assumptions			Average rating of stakeholder disagreements				
	Environmental	Hidden	Valid	Syntactic	Semantic	Pragmatic	Severe	Valid
G1	1.67	1.67	2.33	3.00	2.00	1.50	1.00	1.50
G2	2.33	1.33	3.00	2.00	2.25	2.00	1.50	1.50
V1	2.78	2.89	1.87	2.50	3.00	2.50	2.50	3.00
V2	3.00	3.00	2.00	2.00	3.00	2.50	3.00	2.50

disagreements reflecting practical considerations rather than theoretical ones (e.g., well-formed-ness). By comparison, the syntactic disagreement by G1 concerning URL nomination received low rating on 'Pragmatic' because, in reality, Depositor and Repository User are both given the right to do so.

Compared to the 'Syntactic', 'Semantic', and 'Pragmatic' ratings, the differences of 'Severe' and 'Valid' between V teams' findings and those from the G teams are clearly visible in Table IV. While 'Valid' can be seen as an aggregate of the three levels of disagreements, 'Severe' shows the negative impact of the reported disagreements on Scholar@UC if they are not resolved. We therefore conclude that the V teams did a better job at finding stakeholder disagreements than the G teams, both in terms of the pragmatic meanings and the practical values.

New Requirements. Unlike hidden assumptions and stakeholder disagreements, the new requirements appear to have some very similar records across multiple teams. We held a meeting with three Scholar@UC experts (a project lead, an informationist, and a developer) and shared with them the 17 new requirements without disclosing the modeling team's information. This one-hour meeting helped us better design a survey via Google Docs with 14 distinct requirements, which we e-mailed the entire Scholar@UC project team, asking them to respond in a two-week window.

For each surveyed new requirement, we designed 5 multiple-choice options shown in the left column of Table V. Our original design focused only on value. The meeting with the 3 Scholar@UC team members helped us refine this focus by adding priority. A survey respondent may feel a requirement has value, but if the value is not immediately needed, then that requirement may be nice to have rather than important to have. The meeting also made us realize that some "new" requirements were already implemented in the system: "having anti-virus scan before a work is uploaded" is such an example. The reason for our modeling teams not realizing these already existing requirements is that Scholar@UC evolved from a couple of legacy Web systems at UC. While certain features like anti-virus scan were inherited from the legacy system, they were not documented in the project's current GitHub repository [33]. We therefore added an "already exists" option in our survey.

TABLE V
RATINGS USED TO ASSESS AND ANALYZE NEW REQUIREMENTS

Surveying Scholar@UC team (choosing one and only one)	**Analyzing and reporting** (e.g., the starplots in Fig. 4)
○ Valuable and of high priority	3
○ Neutral	1
○ Not valuable or of low priority	0
○ Already exists	2
○ Do not understand	1

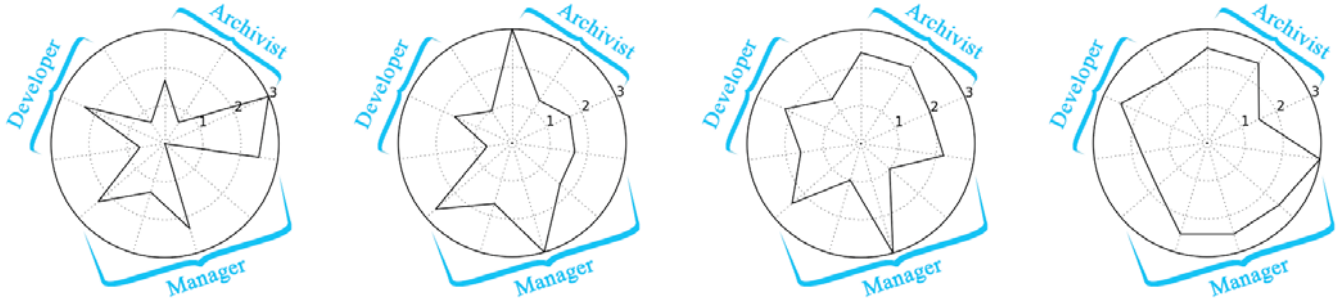
We received 11 survey responses in two weeks, showing Scholar@UC's strong support to our case study. We grouped the responses in 3 categories based on the respondents' roles in the project: 3 were clustered as 'Archivist' including informationist and metadata librarian, 5 played the 'Manager' role consisting of a project lead along with 4 task force members, and 3 software developers ('Developer').

Fig. 4 presents the starplot for each of the 4 modeling teams. In each plot, there are in total 11 axes denoting the 11 Scholar@UC team members who responded to our survey. Each axis is scaled according to the numeric values defined in the right column of Table V. Zero shows an explicit "not valuable or of low priority" response, and one represents either a neutral or an uncertain opinion. On the positive end, "valuable and of high priority" is clearly the most desirable choice, but in our view, "already exists" also signifies a value proposition. Thus, the more area a modeling team's scores cover the starplot, the more valuable the team's new requirements were perceived by the members from the Scholar@UC team. The two V teams, according to Fig. 4, outperformed their G team counterparts. The superior performance is also in line with the top-5 ranked new requirements listed in Table VI. The discrepancy is apparent here: Only one G2's finding made it to Table VI and all others were contributed by the V teams.

B. Modeling Process with OpenOME

Table VII presents self-reported modeling effort of each team. Compared to our replication base where the V team did not produce their final i^* model [11], all the 4 teams in our study successfully completed their integrated models by spending a comparable amount of total time.

OpenOME played a significant role according to the modeling teams' own reflections. All the modelers agreed that OpenOME was easy to learn and to use. The V1 team, however, pointed out two problems: merging individual models and saving the final merge in a format suitable for large prints. We share their former experience here. In V1's first model merging meeting, they were successful in loading the three i^* viewpoints into OpenOME. After choosing one base file (strategic rationale model), they encountered great difficulty in copying and pasting other diagrams to the base. The i^* actors would collapse (rather than staying expanded) and all of the elements inside an actor were piled onto one location. Fig. 5 illustrates this issue. Although the problem may seem to relate only to the user interface, we believe addressing the subtle



G1: 3 new requirements assessed **G2**: 4 new requirements assessed **V1**: 5 new requirements assessed **V2**: 5 new requirements assessed

Fig. 4. Starplots summarizing eleven Scholar@UC team members' ratings on the new requirements (cf. Table V for the mapping between the survey options and the Likert-scale numeric values).

TABLE VI
TOP-5 RATED NEW REQUIREMENTS AND THEIR CONTRIBUTING TEAMS (CLUSTERED BY SCHOLAR@UC SURVEY RESPONDENTS' ROLES)

New Requirements (partial list)	**Archivist**	**Manager**	**Developer**
NR1: add approval/mediation mechanism (V1, G2)	NR6 (V1, V2)	NR6 (V1, V2)	NR6 (V1, V2)
NR5: enforce data quality validation (V1)	NR1 (V1, G2)	NR11 (V2)	NR8 (V1)
NR6: report work usage statistics (V1, V2)	NR10 (V2)	NR10 (V2)	NR11 (V2)
NR8: allow new content to be monitored (V1)	NR11 (V2)	NR8 (V1)	NR10 (V2)
NR10: drag & drop new works (V2)	NR8 (V1)	NR5 (V1)	NR5 (V1)
NR11: view works inside the browser (V2)			

issues like this will improve not only OpenOME's usability but also its support for viewpoint merging and collaborative modeling in general. The copy-and-paste issue, along with several other concrete suggestions, is shared in [36] with the intention to make OpenOME an even more valuable community asset.

C. Modeling Products' Sanity Check

Our interview with the informationist also engaged this Scholar@UC expert in teasing out a set of questions important for domain understanding. As stated in Section III-C, we did not present the informationist during the interview any of the i^* models resulted from the modeling teams. The main reason was to avoid causing the domain expert to be bogged down by the i^* syntax or to be biased by any specific mode construct.

Table VIII lists seven questions elicited from the domain expert. Relating to the forward ("what if" questions to compare alternatives) and backward ("goal satisfaction" questions) analyses defined in [30], Q5 and Q7 of Table VIII exhibit a backward nature whereas Q1, Q3, and Q4 fit more into the forward reasoning. Q2 and Q6 seem to evoke AI (artificial intelligence) planning that concerns the realization of strategies or action sequences executed by agents. While automated forward and backward goal model analysis procedures have already been built in OpenOME (cf. Fig. 3 "Analysis Buttons"), some planning solution is also proposed for requirements goal models [13].

TABLE VII
SELF-REPORTED MODELING EFFORT (TIME IN HOURS)

Team	# of meetings	total meeting time	individual effort	**Σ**
G1	5	unknown	unknown	13
G2	3	7	3+5+2.5+2.5	20
V1	4	3	3.5+4.5+4.5	15.5
V2*	n/a	n/a	4+4+2	10

*A V2 member had a 2-week travel during the 3-week modeling period, which was not foreseen. While this helped viewpoints-based modeling, V2's group-wide communication and coordination were largely done via e-mails. Thus, the meeting measures were not applicable (n/a) for V2.

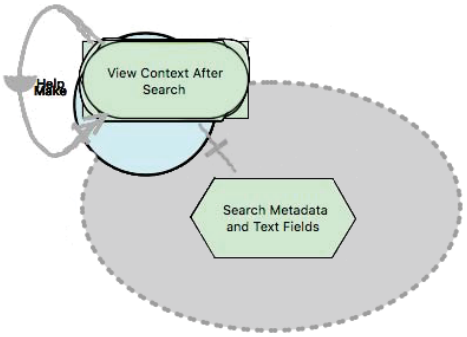


Fig. 5. Copy-and-paste issue in OpenOME, hurting model merging.

In our analysis, the focus is not automation but a sanity check of the produced i^* models [30]. To do so, we took two steps: checking whether the model contained the relevant elements (e.g., for Q2, testing if "orphaned works" appeared in the i^* model) and gaining a sense of how easy for the model to answer the question. The two steps are sequential: If the model elements do not exist in the first place, then it is not sensible to perform the relevant analysis on the model. One researcher carried out the two steps manually.

Our analysis results are shown in Table IX. V1's i^* model was the most comprehensive in terms of containing the necessary elements of all the seven questions. The two G models missed more elements. For the second step, no actual answer was attempted though obtaining one would be "easy" on the capable models. Although the V teams' models passed the

TABLE VIII
QUESTIONS ELICITED FROM A SCHOLAR@UC DOMAIN EXPERT WITHOUT REFERRING TO ANY OF THE i^* MODELS PRODUCED IN OUR STUDY

Q1	What sequence of actions must be taken to assure data quality?
Q2	What is the best plan of actions to manage the orphaned works?
Q3	What are the acceptable branding guidelines?
Q4	How to achieve the versioning of records?
Q5	Can anti-virus check and fast responsiveness be satisfied simultaneously?
Q6	How involved must archivist be to approve collection?
Q7	What is the effect of deciding on URL acceptance by archivist?

TABLE IX
SANITY CHECK ON TEAM-BASED i^* MODELS

	Do question elements appear? (Yes/No)				**How capable of answering? (Easy/Hard)**			
	G1	**G2**	**V1**	**V2**	**G1**	**G2**	**V1**	**V2**
Q1	Yes	Yes	Yes	Yes	Hard	Easy	Easy	Easy
Q2	No	No	Yes	No			Easy	
Q3	No	No	Yes	Yes			Hard	Easy
Q4	No	No	Yes	No			Hard	
Q5	Yes	Yes	Yes	Yes	Easy	Hard	Easy	Easy
Q6	Yes	No	Yes	Yes	Hard		Easy	Easy
Q7	Yes	No	Yes	Yes	Easy		Easy	Easy

sanity check better than the G teams' models, it is important to point out two key observations from Table IX. First, none of the 4 i^* models seemed to be fully capable of answering all the 7 questions, which were elicited from only one domain expert. Second, none of the 7 questions was addressed adequately by any of the models. Both these points stress the importance of interactive and incremental i^* model analysis [30].

V. CONCLUDING REMARKS

A. Summary and Limitations

Our study updates the replication base's results (cf. Section II-B) as follows:

- **R_1'**: Viewpoints *did* lead to a richer domain understanding because it helped generate better hidden assumptions, stakeholder disagreements, and new requirements.
- **R_2'**: With proper tool support like OpenOME, viewpoints-based modeling was *no longer* slower and was successfully in producing the merged i^* model, though certain features of OpenOME could be improved to better support collaborative requirements modeling.
- **R_3'**: Process was *still* important, but with the appropriate support, the better process (e.g., viewpoints) would lead to better product (e.g., merged i^* model).

Some important factors must be taken into account when interpreting our results. Our covering of 3 response variables of "a richer domain understanding" can affect the construct validity [7]. "Stakeholder disagreement", for instance, is a domain-dependent construct and its manifestations in legal and regulatory requirements are well studied (e.g., [42]). One internal validity [7] threat relates to the modelers' self-reported effort data. Confounding variables include the modelers' potentially differing levels in mastering OpenOME, as well as our filtering of the insensible raw domain-understanding items (cf. Table III). To mitigate the latter, we have shared our entire study packet in [36]. Regarding the external validity [7], although our study doubles the number of V and G teams from [11], it is not the statistical generalization, but the theoretical generalization (i.e., testing and updating the viewpoints theory [11]) that our replicate case study is intended to achieve.

B. Replication Insights and Future Work

Replication has been at the heart of science for as long as the scientific method has existed. By independently carrying out a theoretical replication, we highlight the lessons learned.

- **Enable external replications via open repository.** While it is not possible to acquire all the information from the study packet [17], we believe an open repository approach like Scholar@UC will greatly facilitate *external* replications in RE. We are feeding our own study materials back to Scholar@UC [36] and cordially invite researchers to extend our work to further the relevant RE knowledge.
- **Replicate in increasingly realistic settings.** There is little doubt that replication helps evolve an empirically backed body of knowledge, but which aspect(s) to evolve? Due to the fast pace of technological advances in RE, we believe factors key to tech transfer (e.g., OpenOME) are worth updating. The "Ready-Set-Transfer" track held in recent RE conferences offers promising candidates; meanwhile, replications involving the techniques and tools will yield a better understanding about their usefulness and range of applicability.
- **Advance case study research in RE.** Admittedly, our study is an exploratory case study just like our replication base [11]. We plan to investigate two ways for methodological advancement. One is to move from *exploratory* to *explanatory* case study [7], i.e., to embark on an explanation-building process to stipulate the causal links for the phenomenon. The other is to move from literal or theoretical *replication* to *triangulation* where the same phenomenon (e.g., viewpoint merging) is examined with different empirical research methods: controlled experiments, case studies, ethnographies, etc. [43, 44, 45].

Our collaboration with Scholar@UC continues. The research team was recently invited to participate in the project's "train the trainer" program where we realized more viewpoints could be built and merged. More importantly, the project team fully embraces the i^* models produced in our work and deems these models valuable and complementary to [33]. Quoting Brooks *et al.* [17] here: "The work of the replicator should be seen as glamorous not gruesome". As RE replicators, our work with Scholar@UC has certainly made us feel so.

ACKNOWLEDGEMENT

We thank all the management and staff at Scholar@UC for allowing us to conduct this case study, and especially to Ted Baldwin, Eira Tansey, Thomas Scherz, Glen Horton, Sean Crowe, James Van Mil, Carolyn Hansen, Arlene Johnson, and Elizabeth Meyer for providing valuable feedback in the stakeholder meeting and via the online new requirements survey. We also thank Wentao Wang for

assisting with data analysis. The work is funded in part by the U.S. National Science Foundation (Award CCF 1350487) and the National Natural Science Foundation of China (Fund No. 61375053).

REFERENCES

[1] F. Shull, J. C. Carver, S. Vegas, and N. Juristo, "The role of replications in empirical software engineering," *Empirical Software Engineering*, vol. 13, no. 2, pp. 211–218, April 2008.

[2] B. Penzenstadler, J. Eckhardt, and D. M. Fernández, "Two replication studies for evaluating artefact models in RE: results and lessons learnt," in *RESER*, 2013, pp. 66–75.

[3] J. Lung, J. Aranda, S. Easterbrook, and G. Wilson, "On the difficulty of replicating human subjects studies in software engineering," in *ICSE*, 2008, pp. 191–200.

[4] D. Callele, K. Wnuk, and M. Borg, "Confounding factors when conducting industrial replications in requirements engineering," in *CESI*, 2013, pp. 55–58.

[5] R. Wieringa, "Empirical research methods for technology validation: scaling up to practice," *Journal of Systems and Software*, vol. 95, pp. 19–31, September 2014.

[6] D. M. Fernández, K. Lochmann, B. Penzenstadler, and S. Wagner, "A case study on the application of an artefact-based requirements engineering approach," in *EASE*, 2011, pp. 104–113.

[7] R. K. Yin, *Case Study Research: Design and Methods*. Sage, 2003.

[8] J. Natt och Dag, T. Thelin, and B. Regnell, "An experiment on linguistic tool support for consolidation of requirements from multiple sources in market-driven product development," *Empirical Software Engineering*, vol. 11, no. 2, pp. 303–329, June 2006.

[9] K. Wnuk, M. Höst, and B. Regnell, "Replication of an experiment on linguistic tool support for consolidation of requirements from multiple sources," *Empirical Software Engineering*, vol. 17, no. 3, pp. 305–344, June 2012.

[10] D. I. K. Sjøberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanović, N.-K. Liborg, and A. C. Rekdal, "A survey of controlled experiments in software engineering," *IEEE Transactions on Software Engineering*, vol. 31, no. 9, pp. 733–753, September 2005.

[11] S. Easterbrook, E. Yu, J. Aranda, Y. Fan, J. Horkoff, M. Leica, and R. A. Qadir, "Do viewpoints lead to better conceptual models? An exploratory case study," in *RE*, 2005, pp. 199–208.

[12] N. Niu, J. Savolainen, and Y. Yu, "Variability modeling for product line viewpoints integration," in *COMPSAC*, 2010, pp. 337–346.

[13] N. Ernst, A. Borgida, and I. Jureta, "Finding incremental solutions for evolving requirements," in *RE*, 2011, pp. 15–24.

[14] G. Valença, C. Alves, V. Alves, and N. Niu, "A systematic mapping study on business process variability," *International Journal of Computer Science & Information Technology*, vol. 5, no. 1, pp. 1–21, February 2013.

[15] N. Niu, J. Savolainen, Z. Niu, M. Jin, and J.-R. C. Cheng, "A systems approach to product line requirements reuse," *IEEE Systems Journal*, vol. 8, no. 3, pp. 827–836, September 2014.

[16] S.-F. Chang, P.-J. Hsieh, and H.-F. Chen, "Key success factors for clinical knowledge management systems: Comparing physician and hospital manager viewpoints," *Technology and Health Care*, vol. 24, no. s1, pp. 297–306, December 2015.

[17] A. Brooks, M. Roper, M. Wood, J. Daly, and J. Miller, "Replication's role in software engineering," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. Springer, 2008, pp. 365–379.

[18] National Public Radio, "Physicist reacts to discovery of gravitational waves," http://www.npr.org/2016/02/11/466458500/physicist-reacts-to-discovery-of-gravitational-waves, February 11, 2016.

[19] F. Q. B. da Silva, M. Suassuna, A. C. C. França, A. M. Grubb, T. B. Gouveia, C. V. F. Monteiro, and I. E. dos Santos, "Replication of empirical studies in software engineering research: a systematic mapping study," *Empirical Software Engineering*, vol. 19, no. 3, pp. 501–557, June 2014.

[20] O. S. Gómez, N. Juristo, and S. Vegas, "Replications types in experimental disciplines," in *ESEM*, 2010, Article 3.

[21] M. G. Mendonça, J. C. Maldonado, M. C. F. de Oliveira, J. Carver, S. C. P. F. Fabbri, F. Shull, G. H. Travassos, E. N. Höhn, and V. R. Basili, "A framework for software engineering experimental replications," in *ICECCS*, 2008, pp. 203–212.

[22] N. Juristo and S. Vegas, "The role of non-exact replications in software engineering experiments," *Empirical Software Engineering*, vol. 16, no. 3, pp. 295–324, June 2011.

[23] J. L. Krein and C. D. Knutson, "A case for replication: synthesizing research methodologies in software engineering," in *RESER*, 2010.

[24] M. Jackson, "The meaning of requirements," *Annals of Software Engineering*, vol. 3, no. 1, pp. 5–21, January 1997.

[25] X. Chen and Z. Jin, "Capturing requirements from expected interactions between software and its interactive environment: an ontology based approach," *International Journal of Software Engineering and Knowledge Engineering*, vol. 26, no. 1, pp. 15–40, February 2016.

[26] B. Nuseibeh, J. Kramer, and A. Finkelstein, "A framework for expressing the relationships between multiple views in requirements specification," *IEEE Transactions on Software Engineering*, vol. 20, no. 10, pp. 760–773, October 1994.

[27] E. Yu, "Towards modeling and reasoning support for early-phase requirements engineering," in *RE*, 1997, pp. 226–235.

[28] i^* Wiki | Available i^* Tools, http://istar.rwth-aachen.de/tiki-index.php?page=i*+Tools, Last accessed: July 2016.

[29] OpenOME: An Open-Source RE Tool, https://se.cs.toronto.edu/trac/ome/wiki/WikiStart, Last accessed: July 2016.

[30] J. Horkoff and E. Yu, "Interactive goal model analysis for early requirements engineering," *Requirements Engineering*, vol. 21, no. 1, pp. 29–61, March 2016.

[31] Scholar@UC, https://scholar.uc.edu, Last accessed: July 2016.

[32] Scholar@UC on GitHub, https://github.com/uclibs/scholar_uc, Last accessed: July 2016.

[33] Scholar@UC User Stories, https://github.com/uclibs/scholar_use_cases, Last accessed: July 2016.

[34] M. Cohn, *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional, 2004.

[35] D. L. Moody, P. Heymans, and R. Matulevicius, "Improving the effectiveness of visual representations in requirements engineering: An evaluation of i^* visual syntax," in *RE*, 2009, pp. 171–180.

[36] N. Niu and C. Khatwani, http://dx.doi.org/doi:10.7945/C25K5P, Hosted on Scholar@UC: https://scholar.uc.edu/show/05741s72s, Last accessed: July 2016.

[37] J. Mylopoulos, J. Castro, and M. Kolp, "The evolution of Tropos," in *Seminal Contributions to Information Systems Engineering*, J. Bubenko *et al.*, Ed. Springer, 2013, pp. 281–287.

[38] S. Sunkle and H. Rathod, "Visual and ontological modeling support for extended enterprise models," in *CAiSE Forum*, 2014, pp. 193–200.

[39] N. Niu and S. Easterbrook, "Analysis of early aspects in requirements goal models: a concept-driven approach," *Transactions on Aspect-Oriented Software Development*, vol. III, pp. 40–72, 2007.

[40] C. Almeida, M. Goulão, and J. Araújo, "A systematic comparison of i^* modelling tools based on syntactic and well-formedness rules," in *iStar*, 2013, pp. 43–48.

[41] J. Horkoff and E. Yu, "Analyzing goal models: different approaches and how to choose among them," in *SAC*, 2011, pp. 675–682.

[42] A. K. Massey, R. L. Rutledge, A. I. Antón, and P. P. Swire, "Identifying and classifying ambiguity for regulatory requirements," in *RE*, 2014, pp. 83–92.

[43] S. Reddivari, A. Asaithambi, N. Niu, W. Wang, L. D. Xu, and J.-R. C. Cheng, "Ethnographic field work in requirements engineering," *Enterprise Information Systems (*accepted*)*.

[44] N. Niu, A. Y. Lopez, and J.-R. C. Cheng, "Using soft systems methodology to improve requirements practices: an exploratory case study," *IET Software*, vol. 5, no. 6, pp. 487–495, December 2011.

[45] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. Springer, 2008, pp. 285–311.