Requirements Eng (2017) 22:317–338
DOI 10.1007/s00766-017-0271-0

RE 2016

Advancing viewpoint merging in requirements engineering: a theoretical replication and explanatory study

Charu Khatwani[1] · Xiaoyu Jin[1] · Nan Niu[1] · Amy Koshoffer[1] · Linda Newman[1] · Juha Savolainen[2]

Received: 24 November 2016 / Accepted: 6 May 2017 / Published online: 19 May 2017

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 advantage of theoretical replication to mitigate one of the original study design's threats and to embrace an important evolving factor, namely automated tool support for producing $i*$ models. Our replicate study updates the prior results by showing the time saving enabled by the tool and verifies the rich domain understanding gained through viewpoint-based modeling. In an attempt to explain why viewpoints lead to richer domain understanding, we examine in a posteriori way the role that traceability plays in building individual and team-wide requirements models. Our post hoc analysis results suggest that better traceability from the sources makes team-level requirements modeling more focused, whereas the lack of traceability makes it less fruitful. Our work not only shifts the case study from an exploratory to an explanatory nature, but also proposes the integration of conflict-centric views into viewpoint merging to further improve the understanding about stakeholder requirements' trade-offs.

Keywords Replication · Theoretical replication · Viewpoints · Model merging · Comparative study · Scholar@UC · $i*$ · Traceability

✉ Nan Niu
nan.niu@uc.edu

Charu Khatwani
khatwacu@mail.uc.edu

Xiaoyu Jin
jinxu@mail.uc.edu

Amy Koshoffer
amy.koshoffer@uc.edu

Linda Newman
linda.newman@uc.edu

Juha Savolainen
juhaerik.savolainen@danfoss.com

[1] University of Cincinnati, Cincinnati, OH, USA

[2] Danfoss Power Electronics A/S, Gråsten, Denmark

1 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].

While the general aim of replication is to examine the extent to which a published study's results are valid and reliable, there are different kinds of replication that can be carried out in requirements engineering (RE). Following Lung et al. [2], we distinguish two replication kinds: literal and theoretical. In a *literal replication*, the goal is to come close enough to the original experiment so that the results can be directly compared. In contrast, a *theoretical replication* [3] seeks to investigate the scope of the underlying theory, for example, for redesigning the study for a different target population, or by testing a variant of the original hypothesis.

An example of literal replication is on testing the benefits of artifact-based RE. Fernández and his colleagues [4] conducted an initial study by collaborating with a street

traffic management business unit at Siemens to demonstrate the usefulness of establishing a company-wide reference model by putting the focus on the RE artifacts and their dependencies rather than dictating a strict process with interconnected methods. In a literal replication, two other industrial partners (BMW and Cassidian) collaborated in the same research thread where the researchers used the same instrumentations (e.g., Likert-scale questionnaires) to assess the benefits of artifact-based RE [5].

Theoretical replication [3] seeks to investigate the underlying 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 [6], Wnuk et al. [7] 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 [2]. However, beyond [7], there are very few theoretical replications published in RE. The mapping by da Silva et al. [8] reported 32 RE replications, showing an increase from the 4 RE experiments surveyed over a decade ago [9]. To gain operational insights into theoretical replication in RE, we performed one ourselves [10]. We selected an exploratory case study [11] to replicate. Our main rationale was that the underlying theory was clearly stated and tested in [11], namely modeling stakeholder viewpoints separately and then explicitly merging them leads to a richer domain understanding than constructing a single coherent requirements model.

Our goals in [10] were twofold: to reproduce the results observed in the base study [11] (i.e., the richer domain understanding resulted from viewpoint-based modeling), and to take into an important evolving factor (i.e., the requirements modeling tool support) when testing the underlying theory. The results from our theoretical replication confirmed the richer domain understanding achieved by viewpoint-based modeling, and also revealed the time saving enabled by the modeling tool support [10].

Like the replication base [11], our empirical study [10] was also *exploratory*, meaning that the focus was on comparing the requirements modeling done in a viewpoint-based way and that done in a globally coherent way. After the benefits of viewpoint-based requirements modeling reported in repeated studies, we extend the work by trying to *explain* why viewpoints lead to better requirements modeling. To that end, we concentrate on one key factor outlined in [11]: *traceability* from the information sources to the requirements models. The main rationale behind our investigation of traceability lies in one of the basic tenets of viewpoint theory on "stakeholder buy-in" [11], i.e., viewpoint-based modeling promotes more accurate traceability and thus better coverage and preservation of different stakeholders' requirements. Our findings show a significantly higher degree of traceability in viewpoint-based models, indicating that reduced traceability could result in limited domain understanding. However, the high traceability degree in our case study maps to domain understanding in a rather isolated way. Such isolation makes it difficult to analyze the stakeholder disagreements. In pursuit of a coherent and central representation of the disagreements, and more importantly, in order to reason about the trade-off of various resolutions to the disagreement, we integrate conflict-centric views [12] into viewpoint merging.

Overall, the main contributions of our work can be characterized by the two thrusts of advancing viewpoint merging in RE: (i) the theoretical replication tackling important evolving contextual factors like goal-modeling tool support; and (ii) the shift of case study research from exploratory toward explanatory, making the research findings more apt to influence RE practice. The remainder of the paper is organized as follows. Section 2 presents background information underlying our theoretical replication. We then follow Carver's guidelines [13] to describe the original study [11] in Sect. 3 and detail our replication results in Sect. 4. Note that most of Sects. 2, 3, and 4 appear in [10]. The new material that extends [10] is centralized in Sect. 5 where we examine traceability as a factor impacting requirements goal modeling. Finally, Sect. 6 discusses our study limitations, draws some concluding remarks, and outlines the future work.

2 Background of replications

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 [14]. Clearly no single study has the independent ability to produce definitive answers for separating truth from belief. Therefore, replication of previously published empirical studies is frequently advocated [1, 14, 15]. While this serves to increase or decrease confidence in the obtained results and to probe the conditions under which the hypotheses hold [15], repeated research takes many forms.

Probably the most well-known distinction in software engineering is between *internal* and *external* replications,

as defined in [15]. 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. [15] 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 [16]. 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 [9] to 133 in da Silva et al.'s study [8], 31 of the 32 RE replications (97%) were internal ones.

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, where the replication takes place, and so forth [17]. Mendonça et al. [18] 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 [19] 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 [2]. Such studies have become increasingly important to evaluate the merits of RE approaches, as well as to improve our understanding of the social and cognitive aspects involved in RE. The context of each study can easily cover tens and hundreds of variables [19]. For example, programmer productivity has been linked to more than 250 contributing factors [20], and in RE, an independent review [21] uncovered 8 potential confounding variables of Wnuk et al.'s replication mentioned earlier [7]. Human behavior and research bias are intrinsic sources of variability when RE replication is concerned.

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 [7]. Referring to Juristo and Vegas's "run-and-see" motto [19], we believe theoretical replication can achieve a "run-and-see-big" effect by selecting the critical factors to re-examine the underlying theory. Connecting with the distinction made earlier between internal and external replications, our study on viewpoint merging is both external and theoretical. Due to the theoretical character, the impact of new variables (e.g., specialized goal-modeling tool support) can be assessed and incorporated into the base 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 [8]. Following [2], we distinguish two kinds of replication in this paper: theoretical and literal; though Gómez et al. [17] reviewed various replication types in experimental disciplines. Theoretical replication, compared with literal replication, can potentially improve the repeated study's quality because the researchers can pursue a less contrived design and execution. However, theoretical and literal replications are not independent. For instance, Penzenstadler et al.'s literal replications [5, 22] helped formulate and refine an initial theory of artifact-based RE's expected effectiveness, which in turn enables better theoretical replications.

3 Original study

We follow Carver's guidelines [13] to describe in this section the necessary information about the original study [11] serving as the basis for our replication.

3.1 Research questions

The use of viewpoints has long been proposed as a technique to structure evolving requirements models [23, 24]. Viewpoints in RE help to partition a large and dynamic information space into loosely coupled yet overlapping chunks ("viewpoints"). One of the first empirical tests of viewpoint-based modeling was carried out by Easterbrook and his colleagues [11]. Their central research question was to test the hypothesis: "When approaching a conceptual modeling problem, it is better to build many fragmentary models representing different perspectives than to attempt to construct 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." In the original study, the measures of the response variables were based exclusively on the subjective opinions of the participants.

3.2 Participants

The subject system of Easterbrook et al.'s study was Kids Help Phone (KHP), a nonprofit social service organization that provides counseling to kids and parents across Canada through the phone. The study was conducted around 2004 when KHP wanted to analyze the strategic technology change of developing new internet-based services.

The participants of the original study were 5 graduate students who majored in computer science. The participants were researchers themselves, and according to the comparative study design, 3 students formed a viewpoint-based modeling team (called "V team") and the other 2 students were grouped in the "G team" to perform requirements modeling in a global and coherent manner.

3.3 Design

Figure 1 shows the original study's design. The study was of a comparative nature in that the two teams (G team and V team) had the same starting point of their modeling practices. The control, as shown in Fig. 1, was the process that the requirements modeling was performed: the G team tried to build a single $i*$ model [25] for the KHP organization, whereas the V team was instructed to construct several viewpoint $i*$ models before merging them together. The key distinction was model *merging* that was explicit for the V team but nonexistent for the G team.

3.4 Artifacts

The inputs to both the G and V teams were the interview transcripts that the research team conducted with 14 KHP stakeholders, including CEO, senior management, counselors, operational managers, information technology specialists, human resource management, and fundraising [11]. The size of the interview transcripts was approximately 140 pages in total.

The outputs were $i*$ models produced by the G team and the V team. The size information about the output models was listed in [11]. For example, $i*$ models produced by the V team had an average of 10 actors, whereas those produced by the G team had an average of 13 actors.

3.5 Context variables

The context variables are those that affected the design of the study or the interpretation of the results [13]. In the original study, an important control variable was the tool used for $i*$ modeling. The participants of both the G team and the V team used Microsoft Visio for their modeling. Visio is a general graphical modeling tool, lacking sufficient $i*$ syntactic support. Another likely confounding variable of the original study was the difference between the participants assigned to each team. The researchers did notice problem arising from different levels of familiarity with $i*$, but these showed up as differences within each team, rather than differences between the teams [11].

3.6 Summary of results

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].
- $\mathbf{R_2}$: Viewpoint-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 model slices, rather than the integrated whole, from

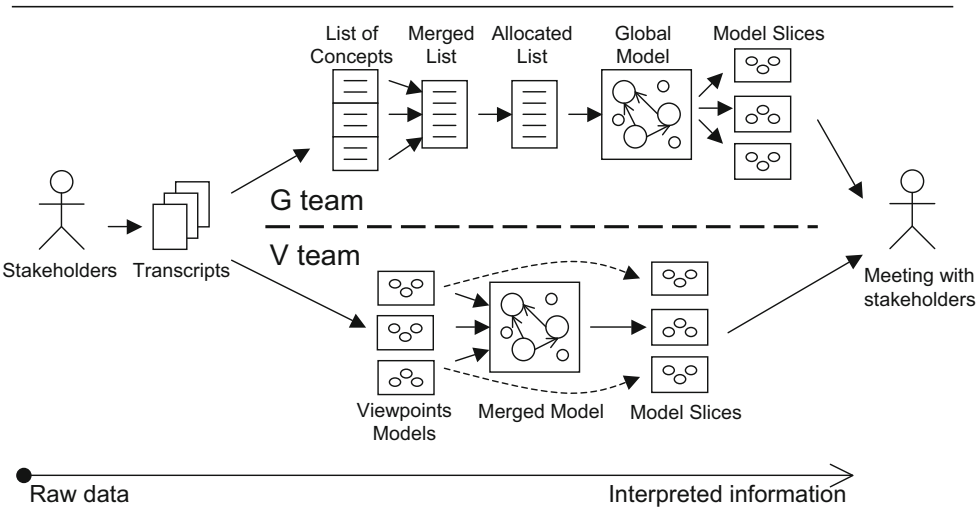

Fig. 1 Original study's design. (*adopted from* [11])

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 the 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].

4 Replication study

4.1 Motivation for conducting the replication

The original study clearly stated the hypothesis under testing, and its design was also straightforward [11]. The work by Easterbrook et al. appeared to be influential, especially in meeting some emerging RE challenges [26]: global and decentralized development, continuous integration and delivery, internet of things, smart cities, healthcare, and the like [27–31]. Despite the influences, the original study was not without weaknesses. We next describe a couple of changes made to improve the study design. Given these changes, our main motivation is to examine the extent to which the results observed from the base study can be reproduced.

4.2 Changes to the original experiment

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 advantage of theoretical replication to improve the study procedure in two aspects. These aspects correspond to the goals of our replication stated in Sect. 1.

– *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]. One reason for the lack of detailed assessments in the original study was the inability to merge the viewpoints [11]. In other words, it would be difficult to achieve measurable effects without an integrated and consolidated model. In our replication, these 3 response variables are assessed by the domain experts rather than by the modelers themselves, reducing the experimenter bias. 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.
– *Take into account an evolving factor* Among the many things changed from the original study, we intentionally incorporate $i*$ 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 improved. The community wiki, for example, lists over 20 tools, many of which are released under open-source licenses [32]. We choose OpenOME [33] to update the study design and describe this tool in more detail in Sect. 4.3. Note that our tool intervention differs from [7] in that Wnuk et al. equipped only the control group (as opposed to the experimental group) with DOORS as a standard treatment, whereas in our replication, OpenOME acts as a contextual variable being applied to both control (G) and experimental (V) groups and being held constant throughout the investigation.

4.3 Replication context

We collaborated with the Scholar@UC project [34] for our replication. 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. Scholar@UC is made open source on GitHub [35], 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 are documented as user stories and released in GitHub [36]. A sample user story is shown in Fig. 2.

The inputs of our replication were 134 user stories similar to the one shown in Fig. 2. These user stories were organized into 11 categories [36]: data management, digital archives, display download, metadata, organizing

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* [36])

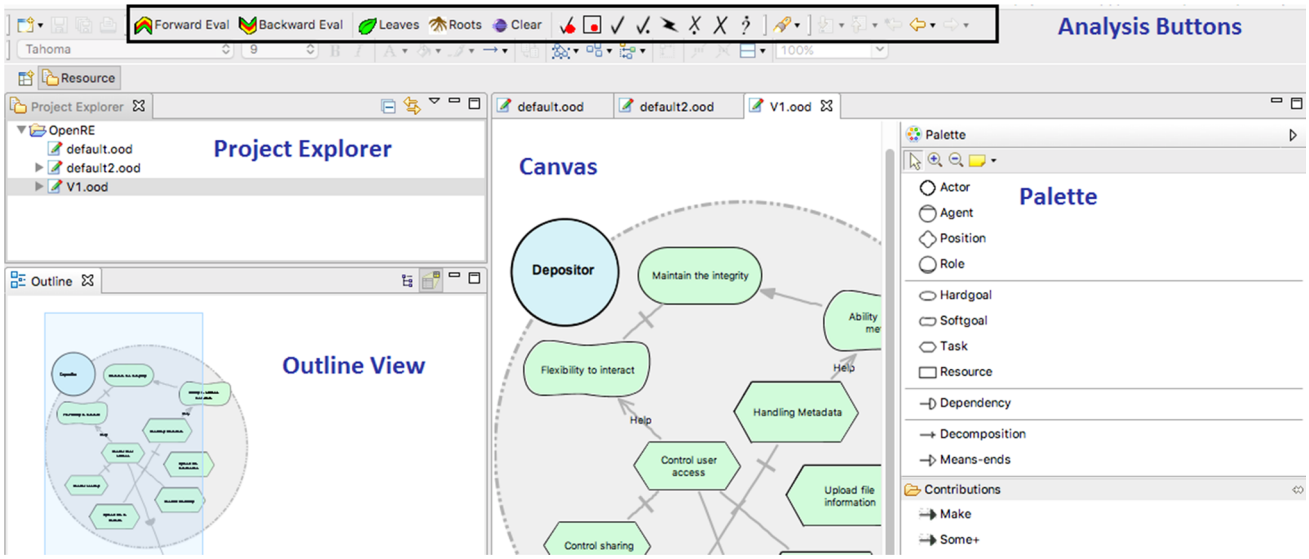

Fig. 3 Screenshot of the OpenOME tool highlighting the editing-related features. (*adapted from* [39])

content, preservation, publishing, search retrieval, submission, TOU (terms of use) rights, and value added. 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 [25].

We recruited 13 UC students from a split-level RE class (fall 2015) to participate in our study. The participation of students was approved by the University's Institutional Research Board (study 2014-355). All students were familiar with $i*$ syntax based on the class's earlier readings [25, 37], 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. Note that Scholar@UC kept evolving its artifacts including the user stories. The version that served as our modeling inputs, together with all other study materials, is made publicly accessible in [38], facilitating future replications.[1]

As our replication goals were to mitigate the threat of measuring the "richer domain understanding" and to take into account OpenOME as an $i*$ model tool, we describe next those two factors by first focusing on OpenOME.

We required all the four teams in our study (V1, V2, G1, and G2) 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 [39]. OpenOME extends the Organizational Modeling Environment (OME) which is part of the Tropos project [40]. 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, making use of the Eclipse and Graphical Modeling Frameworks (EMF and GMF). 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.

We adopt two strategies to mitigate the threat of response variables' measuring: (1) collecting lists of "hidden assumptions," "stakeholder disagreements," and "new requirements" directly from the two G teams and the two V teams, and (2) devising a new way of evaluating the quality of the generated $i*$ models by our participants. 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 [25, 39]. 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

[1] Our study package [38] was updated to include the new data and analyses that we performed since the publication of our conference paper [10].

mostly into the metrics computing category, assessing measures such as $i*$ model sizes in terms of the number of nodes (actors, goals, softgoals, tasks, and resources) and the number of edges (goal contribution links, means-ends links, decomposition links, and dependency links) [11].

Recently, Horkoff and Yu [39] 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 $i*$ 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 [39], 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 [39] 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, $i*$ models are a means to an end—to gain a richer understanding about the problem domain—rather than the end itself.

4.4 Research questions of our replication

Our replicate study aims to answer three research questions: (1) Can we confirm the prior results: $\mathbf{R_1}$, $\mathbf{R_2}$, and $\mathbf{R_3}$ (cf. Section 3.6)? (2) How does the OpenOME tool affect the modeling process? and (3) How well can the resulting $i*$ models answer the stakeholder questions? The first research question follows Carver's guidelines with respect to the comparison of replication results to original results [13]. The second research question corresponds to our goal of testing the important evolving factor of tool support. The third research question is part of our effort of threat mitigation.

4.5 Replication execution

We introduced $i*$ 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. This helped ensure process conformance. Every team was instructed to use [36] 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 [36] 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 $i*$ 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. The main rationale was to collect the measures of the 3 response variables directly from the modeling teams. For the two V teams (V1 and V2), each team member's individual $i*$ model before viewpoint merging was also instructed to be submitted. These instructions can be found in our study package [38]. 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 [36]), and it was the final integrated model from each team (instead of model slices) that was presented in the stakeholder meeting. Four Scholar@UC members participated in the stakeholder meeting: a project lead, a digital archivist, an informationist, and a developer.

4.6 Replication results

4.6.1 Problem-domain understanding: richer or not?

We list the size information about $i*$ models built by our participants in Table 1. All the four teams were able to identify between 4 and 6 actors; however, other model elements were of various sizes across the modeling teams. 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. These lists are available in [38]. Our naming convention is that team name (V1, V2, G1, or G2) followed by HA (hidden assumption), SD (stakeholder disagreement), and NR (new requirement) number; for example, V1-HA1 is the first hidden assumption submitted by the V1 team and G2-NR7 is the seventh new requirement submitted by the G2 team. The complete descriptions of all the teams' HA, SD, and NR submissions are accessible at [38]. By relating to the submitted $i*$ models from the teams, two researchers then

Table 1 Size information about *i*∗ models in our replication study ("Vx-TMy" means the individual model built by team member y of the Vx team)

*i** element	V teams								G teams	
	V1	V1-TM1	V1-TM2	V1-TM3	V2	V2-TM1	V2-TM2	V2-TM3	G1	G2
Actors	5	1	3	1	4	1	1	2	4	6
Goals	56	15	21	13	18	5	5	0	5	37
Softgoals	23	13	8	4	11	3	2	8	9	14
Tasks	65	13	36	14	8	4	2	10	26	28
Resources	25	6	15	1	10	3	2	4	4	7
Dependency links	32	6	22	4	7	1	0	4	14	14
Decomposition links	44	6	31	5	25	8	6	11	19	7
Means-end links	0	0	0	0	0	0	0	4	4	62
Softgoal contributions	30	34	25	21	23	6	5	8	15	5

jointly processed the raw data with the intention to have the domain expert of Scholar@UC evaluate only high-quality items. Sample removed and preserved items are listed below.

- The hidden assumption "Devs know things about stuff" (G2-HA7) is clearly too general to help domain understanding and was filtered out. Another submission from the same team, "Server can recover from data corruption and outages" (G2-HA1), makes an assumption about the fault tolerance capabilities of the data storage, which we kept.
- The reported disagreement "It is unclear what the approval process should be for collections" (V1-SD4) 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" (V2-SD1) reflected a sensible stakeholder disagreement, so we kept it.
- "Create a glossary of terms so that there is less confusion for requirements documenting" (G2-NR5) 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" (V2-NR4), 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.[2] 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.[3] 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 2 lists the number of raw and rated items. In [38], the rated items receive lower identifiers than the unrated ones, e.g., G1-HA1, G1-HA2, and G1-HA3 are the rated hidden assumptions of G1, whereas G1-HA4 and G1-HA5 are filtered out by us and therefore receive no rating from the Scholar@UC domain expert. No team, according to Table 2, seemed to outperform the others in terms of domain-understanding quantities. G1, G2, V1, and V2 had 8, 11, 16, and 10 rated items, respectively. We next compare their qualities.

Table 2 Number of raw and rated domain-understanding items

Team		G1	G2	V1	V2
Hidden	Raw #	5	8	9	4
Assumptions	Rated #	3	3	9	3
Stakehoder	Raw #	2	6	5	4
Disagreements	Rated #	2	4	2	2
New	Raw #	3	5	7	7
Requirements	Rated #	3	4	5	5

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 [42]. Such problem-domain conditions, events, and states are critical to the operation of the intended software. As shown in Table 3, what the V teams produced was more about environmental assertions. These included "time frame is not necessary for assigning permission from consumer to depositor" (V1-HA4) and "depositor can achieve same level of integrity in downloading small chunks as the large ones" (V2-HA1). Neither assumption touched upon implementation details, but both assumptions were deemed very hidden by the Scholar@UC expert. In general, the domain assumptions resulted from the V teams

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

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

Table 3 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		
	Environmental	Hidden	Valid
G1	1.67	1.67	2.33
G2	2.33	1.33	3.00
V1	2.78	2.89	1.87
V2	3.00	3.00	2.00

Team	Average rating of stakeholder disagreements				
	Syntactic	Semantic	Pragmatic	Severe	Valid
G1	3.00	2.00	1.50	1.00	1.50
G2	2.00	2.25	2.00	1.50	1.50
V1	2.50	3.00	2.50	2.50	3.00
V2	2.00	3.00	2.50	3.00	2.50

received higher ratings in terms of the environmental indicativeness and hiddenness, shown in Table 3.

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 G1's rated assumptions—"uploaded data are readable" (G1-HA1), "system is secure" (G1-HA2), and "system has enough permission rights" (G1-HA3)—were assessed as correct by the domain expert, though their hiddenness was virtually nonexistent in that their average rating is 1.33 in Table 3 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 3 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)?" (G1-SD1) and "Are Metadata Specialist and Digital Archivist the same in assuring work attribute quality?" (G1-SD2) identified the overlapping and potentially conflicting information presented in Scholar@UC's user stories [36]. Consequently, G1's results received the 3 out of 3 rating on "Syntactic" in Table 3, which is better than all the other three teams.

Semantic disagreements go beyond the syntax and signal inconsistencies relating to meaning. The aforementioned disagreement (V2-SD1): "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 disagreements reflecting practical considerations rather than theoretical ones (e.g., well-formedness). 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 3. 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 4. Our original design focused only on value. The meeting with the 3 Scholar@UC team members helped us

Table 4 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

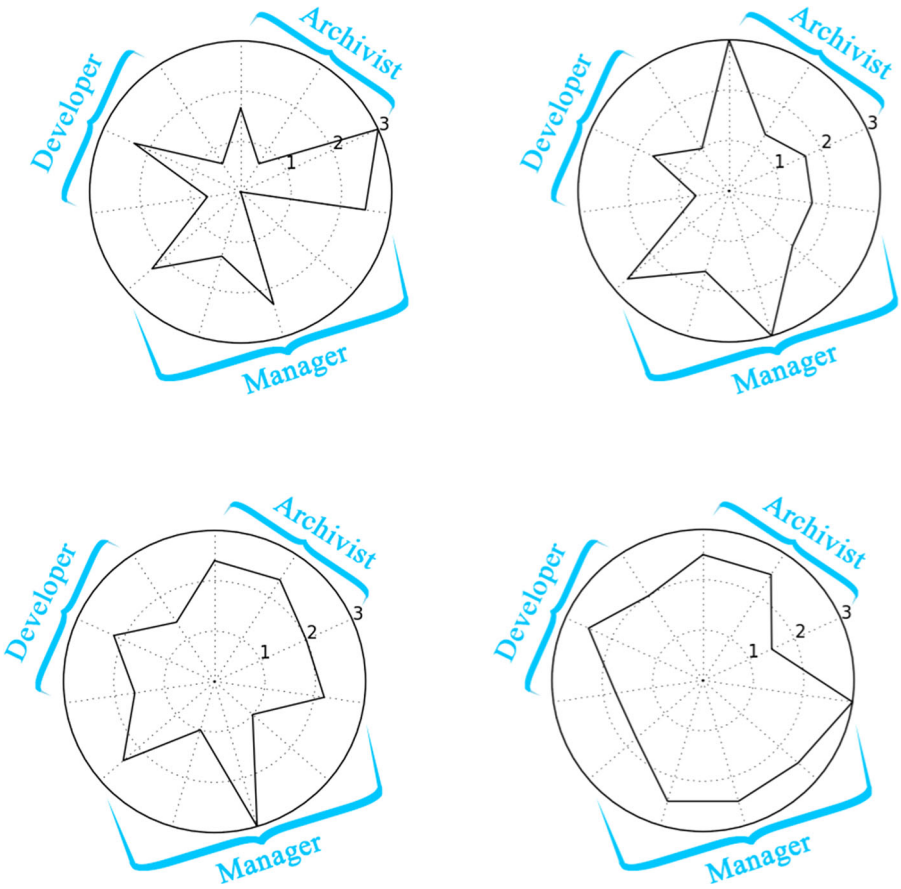

Fig. 4 Starplots summarizing eleven Scholar@UC team members' ratings on the new requirements (cf. Table 4 for the mapping between the survey options and the Likert-scale numeric values). **G1** 3 new requirements assessed (*top-left*), **G2** 4 new requirements assessed (*top-right*) , **V1** 5 new requirements assessed (*bottom-left*), **V2** 5 new requirements assessed (*bottom-right*)

Table 5 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 and drop new works (V2)	NR8 (V1)	NR5 (V1)	NR5 (V1)
NR11: view works inside the browser (V2)			

Table 6 Self-reported modeling effort (time in h)

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 viewpoint-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

Table 7 Stakeholder names and constituents

Stakeholder	All other names used
Consumer	Repository user, User of the Repository
Depositor	Delegate, depositor, member of a group, repository contributor, repository submitter, repository user/journal editor, submitter, submitter to the repository, depositor with multiple related files, some of which may be large, depositor with large files, repository developer, public user of the repository, depositor to the repository, depositor with large discrete files
Digital archivist	archivist, digital archivist, archivist or collection manager
Manager	Collection manager, group manager, Repository manager, repository manager, university records manager
Metadata specialist	Developer, metadata professional, metadata specialist, visual resources librarian, visual resources librarian/user/archivist

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" (G2-NR4) 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 [36]. We therefore added an "already exists" option in our survey.

The five options listed in Table 4 have the numerical range from 0 to 3. We regard "valuable and of high priority" as the best rating because the new requirement is deemed useful and immediately needed for the project. The next highest rating, in our opinion, is "already exists" because the requirement has already been implemented in Scholar@UC though our participants serving as requirements engineers mistakenly labeled the requirement to be "new." We rank two options: "neutral" and "do not understand" with the second lowest rating because neither the value nor the priority was endorsed. Finally, if a survey respondent explicitly chooses "not valuable or of low priority," then we rate the new requirement the lowest in Table 4.

We received 11 survey responses in two weeks, showing Scholar@UC's strong support to our case study. We grouped the responses in three 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').

Figure 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 4. 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 5. The discrepancy is apparent here: Only one G2's finding made it to Table 5, and all others were contributed by the V teams.

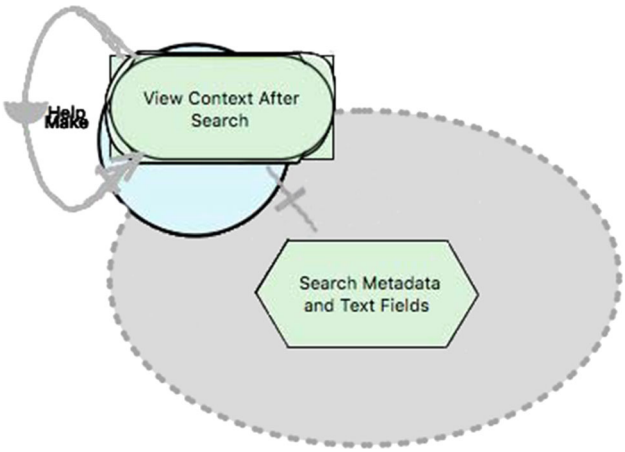

Fig. 5 Copy-and-paste issue in OpenOME, hurting model merging

4.6.2 *Modeling process with OpenOME*

Table 6 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. While the specific areas such as the number of meeting and individual effort are not complete in Table 6 due to the self-reporting nature, the total effort of G1, G2, V1, and V2 is 13, 20,

Table 8 Questions elicited from a Scholar@UC domain expert without referring to any of $i*$ 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 9 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

15.5, and 10 h respectively. Note that the total effort should be interpreted in the light of the 3 weeks given for each team to complete their modeling.

The participants' reporting paid much attention to the 3 response variables. As a result, we did not collect detailed data about the processes of V teams' viewpoint merging, and for that matter, G teams' model generation. Based on our interactions with the modeling teams, their processes tended to be *ad hoc* rather than systematic in terms of following some upfront strategies or tactics. However, we did observe the practice of the V1 team trying to resolve semantic conflicts before viewpoint merging. They explicitly mapped the terminology of Scholar@UC stakeholders as shown in Table 7. In this way, the team members would have a shared vocabulary which potentially reduced their struggles on terminological interferences during viewpoint merging. For all the four teams in our study, the data related to consistency checking of the resulting models were not collected. The string of recent work on model merging [43–45] and consistency checking [46–49] could further improve the modeling processes and products.

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. $i*$ actors would collapse (rather than staying expanded), and all of the elements inside an actor were piled onto one location. Figure 5 illustrates this issue. Although the problem may seem to relate only to the user interface, we believe addressing the subtle 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 [38] with the intention to make OpenOME an even more valuable community asset.

4.6.3 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. The interview was conducted by one researcher and lasted about one hour. The elicitation of domain questions was carried out in a collaborative manner, with the researcher's preparation of a dozen or so seed questions. The informationist modified certain questions as necessary, removed the ones she regarded as unimportant, and added a few that the researcher did not prepare beforehand. As stated in Sect. 4.3, we did not present the informationist during the interview any of $i*$ models resulted from the modeling teams; rather the interview was focused directly on the set of domain questions important for Scholar@UC. The main reason was to avoid causing the domain expert to be bogged down by $i*$ syntax or to be biased by any specific model construct.

Table 8 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 [39], Q5 and Q7 of Table 8 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 [28].

In our analysis, the focus is not automation but a sanity check of the produced $i*$ models [39]. We therefore believe the questions listed in Table 8 are solid starting points; testing the adequacy of those 7 questions requires future work. To perform the sanity check, we took two steps: checking whether the model contained the relevant elements (e.g., for Q2, testing if "orphaned works" appeared in $i*$ 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 9. 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 sanity check better than the G teams' models, it is important to point out two key observations from Table 9. 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 all the models. Both these points stress the importance of interactive and incremental $i*$ model analysis [39].

Summary of Replication Results Our study updates the replication base's results (cf. Sect. 3.6) as follows:

- $\mathbf{R_1}$': Viewpoints *did* lead to a richer domain understanding because it helped to generate better hidden assumptions, stakeholder disagreements, and new requirements.
- $\mathbf{R_2}$': With proper tool support like OpenOME, viewpoint-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.
- $\mathbf{R_3}$': 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).

5 Explanatory analysis of traceability in viewpoint merging

Our theoretical replication allows for testing and updating the viewpoints theory [11]; however, like our replication case [11], the work reported so far is of an exploratory nature. After viewpoints' better performance in requirements goal modeling has been clearly described and repeatedly observed [10, 11], we want to advance the empirical body of knowledge by explaining the "why" through a post hoc data analysis.

The factor that we investigate in this section is *traceability*, or more accurately, backwards traceability that links stakeholder viewpoints back to their source [11]. In our Scholar@UC case study, the source refers to the 134 user stories [38] and the traceability helps rationalize how $i*$ models are derived. It was reported in Easterbrook et al.'s study [11] that due to the explicit merging of viewpoints, the stakeholder contributions were easily traceable. This better traceability enhanced the V team's ability to discover important requirements and domain understanding by comparing viewpoints. However, reduced traceability also had advantages, e.g., helped the G team to choose initial decompositions of the modeling problem [11]. It was therefore not clear based on [11] whether traceability plays a significant role in viewpoints' better requirements modeling performance. Examining such a role in our Scholar@UC case study is precisely the objective of this section.

For backwards traceability, our interest is in tracking $i*$ model elements in the 134 Scholar@UC user stories. $i*$ model elements include those appeared in the four team-wide models as well as the six individual models produced by each viewpoint-based team member (V1 and V2).[4] While automated traceability support exists, we draw our experience in studying requirements analysts' tracing behaviors [50–52] and adopt a manual traceability link identification approach for the purpose of achieving high accuracy. Our manual approach was based primarily on keyword search, i.e., searching every $i*$ model element (goal, softgoal, task, and resource) to decide its traceability.

The tracing data were generated in three phases. The first phase was to establish a tracing protocol. To do so, three researchers traced 30 randomly selected elements collaboratively. We decided to distinguish traceability into three degrees: directly traceable (e.g., V1's "batch submission" goal maps to Scholar@UC's user story with the same title), partially traceable (e.g., V1's "view context after search" matches in part with Scholar@UC's "keyword in context (KWIC) view of full-text search results"), and not traceable. Our second phase involved two

[4] The six individual $i*$ models are shared in our study package [38].

Table 10 Team-wide traceability information ("DT" means "directly traceable," "PT" means "partially traceable," and "NT" means "not traceable")

	# of links (Raw #, percentage)			DT link's frequency of occurrence (Raw #, percentage)		
	DT	PT	NT	1	2	3 or 4★
V1						
Goals	(29, 56%)	(20, 38%)	(3, 6%)	(23, 79%)	(4, 14%)	(2, 6%)★
Softgoals	(12, 50%)	(10, 42%)	(2, 8%)	(12, 100%)	(0, 0%)	(0, 0%)
Tasks	(42, 65%)	(22, 34%)	(1, 2%)	(38, 90%)	(2, 5%)	(2, 5%)★
Resources	(9, 36%)	(9, 36%)	(7, 28%)	(9, 100%)	(0, 0%)	(0, 0%)
∑	(92, –)	(61, –)	(13, –)	(82, –)	(6, –)	(4, –)
V2						
Goals	(9, 50%)	(5, 28%)	(4, 22%)	(8, 89%)	(1, 11%)	(0, 0%)
Softgoals	(0, 0%)	(7, 70%)	(3, 30%)	(0, –)	(0, –)	(0, –)
Tasks	(5, 63%)	(2, 25%)	(1, 13%)	(5, 100%)	(0, 0%)	(0, 0%)
Resources	(2, 22%)	(4, 44%)	(3, 33%)	(2, 100%)	(0, 0%)	(0, 0%)
∑	(16, –)	(18, –)	(11, –)	(15, –)	(1, –)	(0, –)
G1						
Goals	(1, 20%)	(0, 0%)	(4, 80%)	(1, 100%)	(0, 0%)	(0, 0%)
Softgoals	(1, 11%)	(4, 44%)	(4, 44%)	(1, 100%)	(0, 0%)	(0, 0%)
Tasks	(7, 27%)	(15, 58%)	(4, 15%)	(6, 86%)	(0, 0%)	(1, 14%)
Resources	(0, 0%)	(4, 100%)	(0, 0%)	(0, –)	(0, –)	(0, –)
∑	(9, –)	(23, –)	(12, –)	(8, –)	(0, –)	(1, –)
G2						
Goals	(11, 30%)	(16, 43%)	(10, 27%)	(9, 82%)	(2, 18%)	(0, 0%)
Softgoals	(3, 21%)	(4, 29%)	(7, 50%)	(3, 100%)	(0, 0%)	(0, 0%)
Tasks	(5, 18%)	(13, 46%)	(10, 36%)	(5, 100%)	(0, 0%)	(0, 0%)
Resources	(0, 0%)	(5, 71%)	(2, 29%)	(0, –)	(0, –)	(0, –)
∑	(19, –)	(38, –)	(29, –)	(17, –)	(2, –)	(0, –)

★ One model element with 4 directly traceable links appeared only twice in our dataset

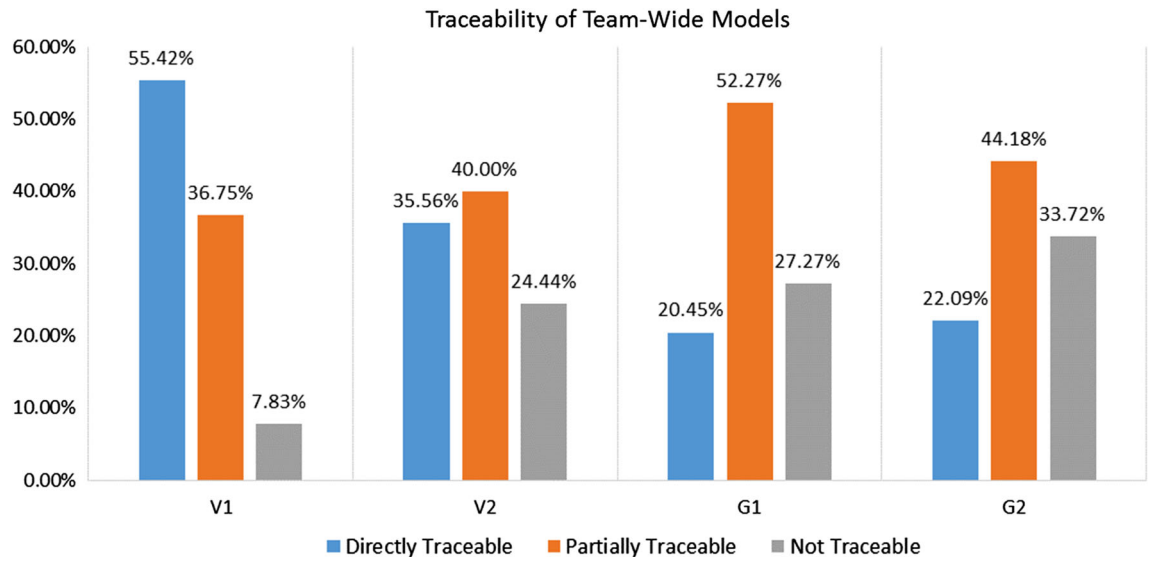

Fig. 6 Comparing team-wide *i*∗ models' traceability where the total number of V1, V2, G1, and G2 elements is 166, 45, 44, and 86, respectively

Table 11 Team-wide χ^2 test results ("DT" means "directly traceable," "PT" means "partially traceable," and "NT" means "not traceable")

Team	Data name	DT	PT	NT	SUM
V1	Raw	**92 (+)**	61	**13 (−)**	166
	Residuals	**5.71**	−1.53	**−5.14**	
V2	Raw	16	18	11	45
	Residuals	−0.64	−0.15	−0.99	
G1	Raw	**9 (−)**	23	12	44
	Residuals	**−2.82**	1.62	−1.49	
G2	Raw	**19 (−)**	33	**29 (+)**	86
	Residuals	**−3.90**	0.68	**4.00**	

Table 12 Team-wide pairwise χ^2 test results ("DT" means "directly traceable," and "NT" means "not traceable")

Team	DT	NT	Team	DT	NT
VI	92 (+)	13 (−)	V2	16 (+)	11 (−)
V2	16 (−)	11 (+)	Gl	9 (−)	12 (+)
G1	9 (−)	12 (+)	G2	19 (–)	29 (+)
G2	19 (−)	29 (+)	G1	9	12
			G2	19	29

researchers performing independent tracing of another randomly chosen 50 model elements. The comparison of their tracing results revealed an almost perfect inter-rater agreement (Cohen's $\kappa = 0.82$) [53]. This showed the reliability of our tracing protocol derived after phase one. Resolving the inter-rater differences further enhanced the protocol's robustness as well as the raters' consensus. The third and final phase was to split the remaining model elements evenly for the two raters (researchers) to trace. In total, 546 elements from the ten $i*$ models were traced and the detailed traceability data can be found in our study package [38]. Next we analyze the team-wide model traceability followed by individual model traceability of the viewpoint teams.

Team-Wide Model Traceability Table 10 provides the statistics of the model traceability at the team level. The two viewpoint teams (V1 and V2) had more directly traceable elements in their models than the G teams (G1 and G2). Among the directly traceable model elements, most appeared in the user stories once, but V1 chose to model a proportion of the elements that appeared twice or more. The traceability trends of team-wide models are shown in Fig. 6 where each team's bar charts are normalized by the total number of model elements. Except for V1, the greatest proportion of model elements is partially traceable to the user stories. This is not surprising as requirements modeling is not (and should not be) a straightforward transcribing–translating process. We therefore pay much more attention to "directly traceable" elements and "not traceable" ones in the following analyses.

In Fig. 6, V1 and V2 have more "directly traceable" model elements than "not traceable" ones, whereas G1 and G2 exhibit an opposite trend. This indicates that the viewpoint teams were careful about naming the constructs

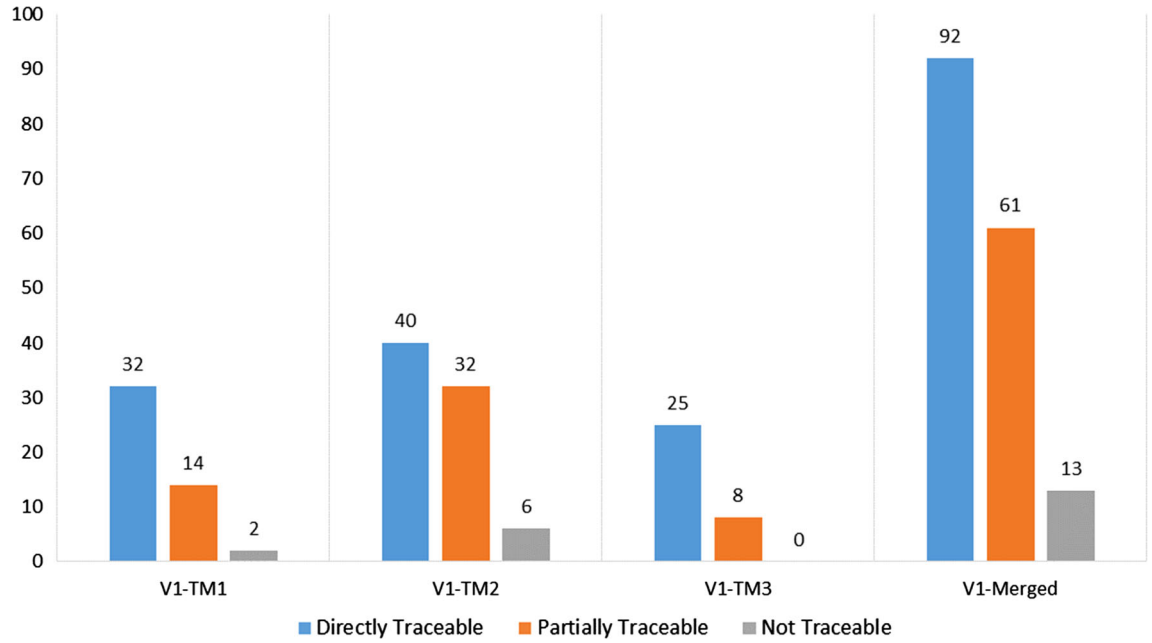

Fig. 7 Traceability of V1's team member's models compared with that of V1's team-wide model ("V1-TM1," "V1-TM2," and "V1-TM3" refer to the three team members of V1)

in their $i*$ models by sticking more to the vocabulary of the requirements source (i.e., Scholar@UC user stories). In contrast, "not traceable" elements can be seen as the constructs coined, invented, or otherwise created by the team while doing the requirements modeling. In this sense, the global teams (G1 and G2), as shown in Fig. 6, were less constrained to choose from the requirements source but devised more words and phrases to label their model elements.

In order to assess whether the team-wide model traceability is significantly different, we use Pearson's chi-squared test (χ^2). χ^2 test fits our purposes because it is a statistical test applied to sets of categorical data to evaluate how likely it is that any observed difference between the sets arose by chance and the test is suitable for unpaired data samples [54]. Table 11 shows χ^2 analysis we performed among the four teams.

The results of team-wide model traceability analysis are $\chi^2 = 46.49$, df = 6, $p < 0.001$. Setting the significance level at $\alpha = 0.05$, the results imply a statistically significant relationship between teams and the traceability measures. However, the results do not reveal much since according to Sharpe [55], the source of a statistically significant result is unclear when a χ^2 test result is associated with more than one degree of freedom (i.e., larger than a 2×2 contingency table). In our case, the degree of freedom (df) of Table 11 is 6, so that the follow-up tests are essential. Following [55], we performed residual calculation and partitioning for further analysis.

A residual is the difference between the observed and expected values for a cell [55]. The expected values are calculated based on the data under the null hypothesis of no association from χ^2 test. The larger the residual, the greater the contribution of the cell to the magnitude of the resulting χ^2 obtained value [55]. Table 11 also shows the results of standardized Pearson's residuals.

The positive or negative value of the residual in Table 11 means the positive or negative association between a team and a traceability measure. However, the association is significant only when the residual value exceeds some threshold. According to [55], a standardized residual having absolute value that exceeds about 2 when there are few cells or about 3 when there are many cells indicates the significance of the association. Since our analysis involved few cells, we used the absolute value of 2 as the threshold. The bolded values in Table 11 then indicate significance. We further specified the positive ("+") or negative ("−") association for the values that were significant. We can see that the V1 team has a highly positive association with the DT (directly traceable) column, while both G teams have strong negative associations with the DT measure.

We further employed partitioning [55] to perform pairwise analysis of the association between any two teams. Our partitioning was performed on DT and NT measures only because otherwise PT would skew the distribution. Considering that PT dominates the distribution in our case, it may impact the significance of associations between DT

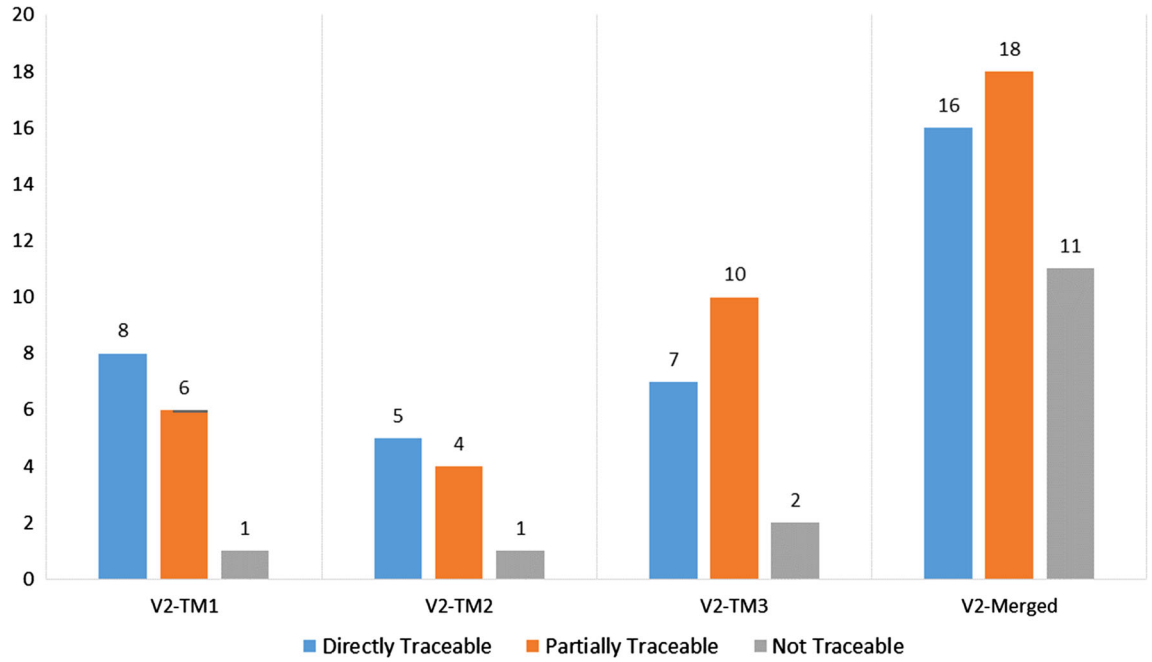

Fig. 8 Traceability of V2's team member's models compared with that of V2's team-wide model ("V2-TM1," "V2-TM2," and "V2-TM3" refer to the three team members of V2)

and NT. We performed χ^2 test between each pair of two teams, resulting in six pairs among the four teams. We then summarized all the pairwise comparisons in Table 12 based on which the following conclusions can be made: V1's model elements are significantly more traceable than the other teams', V2's elements are more traceable than both G teams', and G1 and G2 have no significant difference in terms of their model elements' traceability.

Individual Model Traceability We applied the same tracing protocol that we used for the team-wide $i*$ models to trace the model elements submitted by the team members of the two viewpoint teams. The number of traceability links of V1's and V2's team members is shown in Figs. 7 and 8, respectively. It is interesting to note that, for V1, the distributions of the "directly traceable," "partially traceable," and "not traceable" elements in the individual models are in line with those in the team-wide model. One common trend of Figs. 7 and 8 is that the "directly traceable" elements outnumber the "not traceable" ones. This trend, together with the conclusions drawn from our team-wide analyses, suggests that traceability is practiced very differently between the V teams and the G teams.

In light of the V team observations made in Easterbrook et al.'s study [11], our results indicate that when individual modelers develop viewpoints, more attentions can be paid to the details to their specific viewpoint and not to others' viewpoints. This level of detailing leads to richer domain understanding already for the individual modelers, as reflected in their individual models' traceability in our case study (cf. Figs. 7 and 8). When these more traceable, richer-domain-understanding-bearing viewpoints are explicitly compared and merged at the team level, the resulting requirements models become significantly more traceable. The question that we want to answer next is: Does the more traceable viewpoint requirements modeling process also lead to the richer domain understanding?

To answer this question, we map the model elements to the domain-understanding items in Fig. 9. For the V teams, the model elements in Fig. 9 are the ones that appeared in the individual team members' viewpoints but not in the merged team-wide $i*$ model. In other words, 8 and 6 elements were "lost" during V1's and V2's viewpoint merging, respectively. For the G teams, Fig. 9 shows the "not traceable" elements, i.e., those model elements that

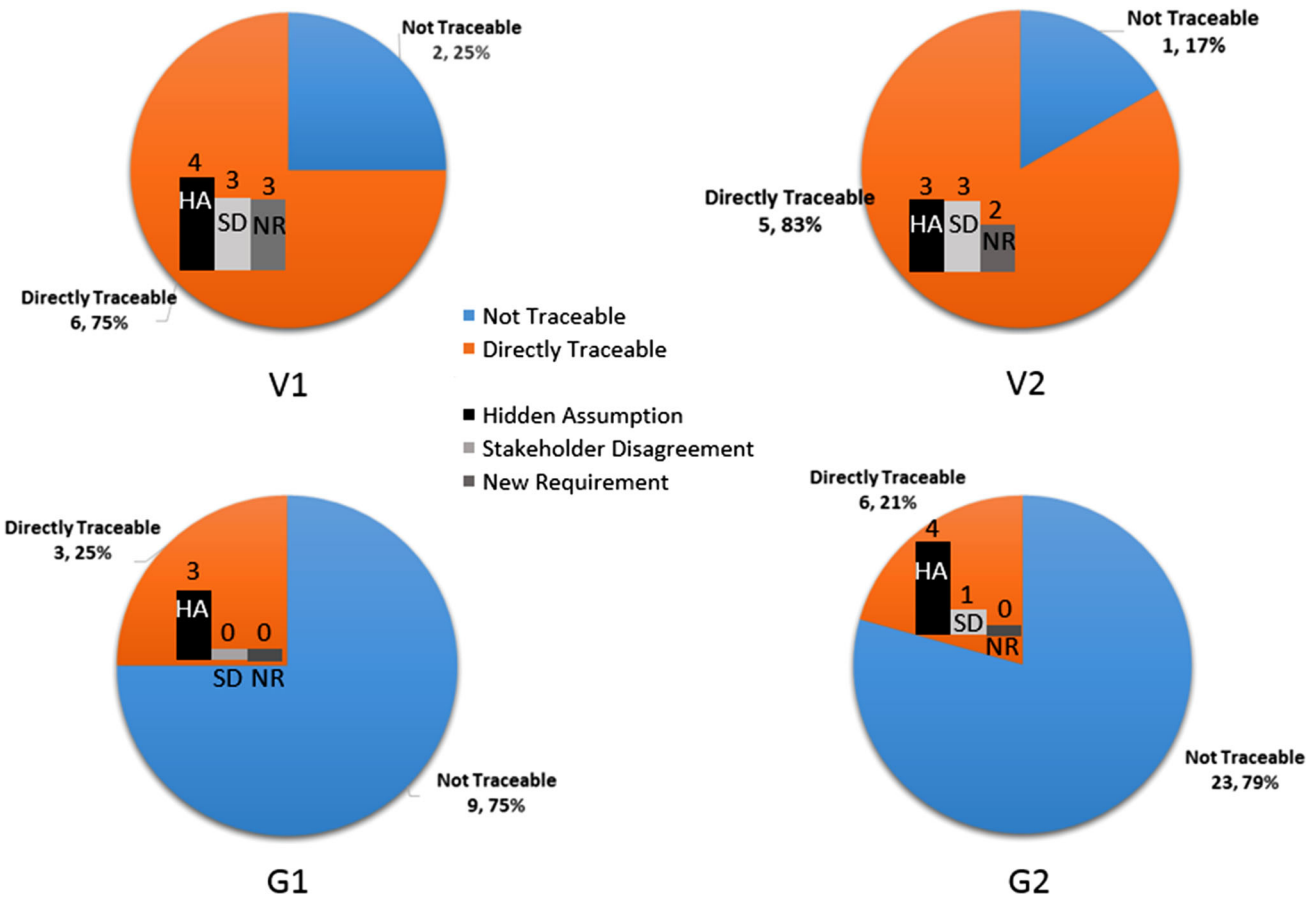

Fig. 9 Mapping model elements to domain-understanding items

Table 13 Applying conflict-centric approach to one of V1's stakeholder disagreement (top) and a newly discovered stakeholder disagreement of V2 (bottom)

ID	Conflict	Resolutions	Implications
V1-SD2	The depositor's "Set permission" task is presumed to define what users can view. However, it is being connected to manager's "Give submission authorization" task, which indicates a mismatch for allocating permission to files	SD2.R1: Link between the collection management goal and approve submission task is missing	Adding this link assures that the collections are approved by the manager
		SD2.R2: Metadata specialist can add metadata fields only after record submission by depositor	It ensures that the files are uploaded and the metadata is maintained in order
		SD2.R3: Depositor can only submit the record	It ensures that the depositor does not approve submissions
		SD2.R4: Set Permission task moved from depositor to manager	It ensures that the permission is granted only by the manager
V2-SD1'	Who approves the URL links? Archivist or manager?	SD1'.R1: URL approval is done by manager not by archivist	It ensures that the URL approval process is aligned well
		SD1'.R2: Identify broken URL task added to collection manager actor	Helps in the identification of broken URL
		SD1'.R3: Add Metadata specialist actor, and add enter parameters for URL task	Helps to find a way to enter the parameters to the URL
		SD1'.R4: Add Metadata specialist actor, and add Response to URL links goal	The URL response/feedback procedure is not well defined

could not be traced back to the requirements source. There are 12 and 29 elements that were coined by G1 and G2, respectively. For all the four teams, the domain-understanding items shown in Fig. 9 are the HAs, SDs, and NRs submitted by the modeling teams, i.e., the number of these items corresponds to the "raw #" instead of the "rated #" in Table 2.

It is clear from Fig. 9 that the seemingly lost elements during viewpoint merging were mapped to domain-understanding items in a majority way (75% for V1 and 83% for V2). Although certain viewpoints' elements did not appear in the final merge, the traceability that these elements had helped generate the richer domain understanding. On the contrary, the new elements created during the G teams' requirements modeling were not only at the cost of reduced traceability, but also had only moderate linkage to the domain-understanding items (25% for G1 and 21% for G2). This suggests that lack of traceability could result in shallower domain understanding. While our results clearly show that traceability plays a significant role in viewpoint-based requirements modeling, further improvements can be made. Next we discuss our effort of integrating conflict-centric views [12] into viewpoint merging.

Conflict-Centric i∗ Views The main reason of our incorporation of conflict-centric views is to overcome the isolated and independent ways that domain-understanding items were generated by the modelers in our study. These challenges were noticed from interviewing the modelers as well as reflected in the relatively flat mapping bar charts of V1 and V2 in Fig. 9. Rather than generating three lists (HA, SD, and NR), conflict-centric views use conflict as a central theme which connects all other domain understanding around that central theme. For example, hidden assumptions should be surfaced to better understand the conflict, and new requirements can be seen as resolutions to the conflict. We believe that, by integrating conflict-centric views into viewpoint merging, the association between more traceable requirements models and richer domain understanding will be essential rather than accidental.

We adapt conflict-centric views from architectural documentation [12] to viewpoint merging and demonstrate how this technique can be applied via a couple of worked examples. Table 13 details these examples. Take V1-SD2 as an example, since this SD was already valid, the conflict-centric approach supported a systematic way of developing 4 resolutions. The approach further allowed us to reason explicitly about the implications of each resolution. Finally, we were able to generate new conflicts (e.g., V2-SD1' shown in Table 13) which enriched Scholar@UC's domain understanding. Our conflict generation was performed manually by adapting the steps depicted in [12]: We first teased out important softgoals (similar to architecturally significant requirements in [12]) from the interested *i∗* model, and then used *i∗* model to give operational definitions of the softgoals. While refining the meanings of the softgoals, potential conflicts could be identified [12].

Since the conflict-centric views are graphical representations [12], we are interested in having an *i∗* version of

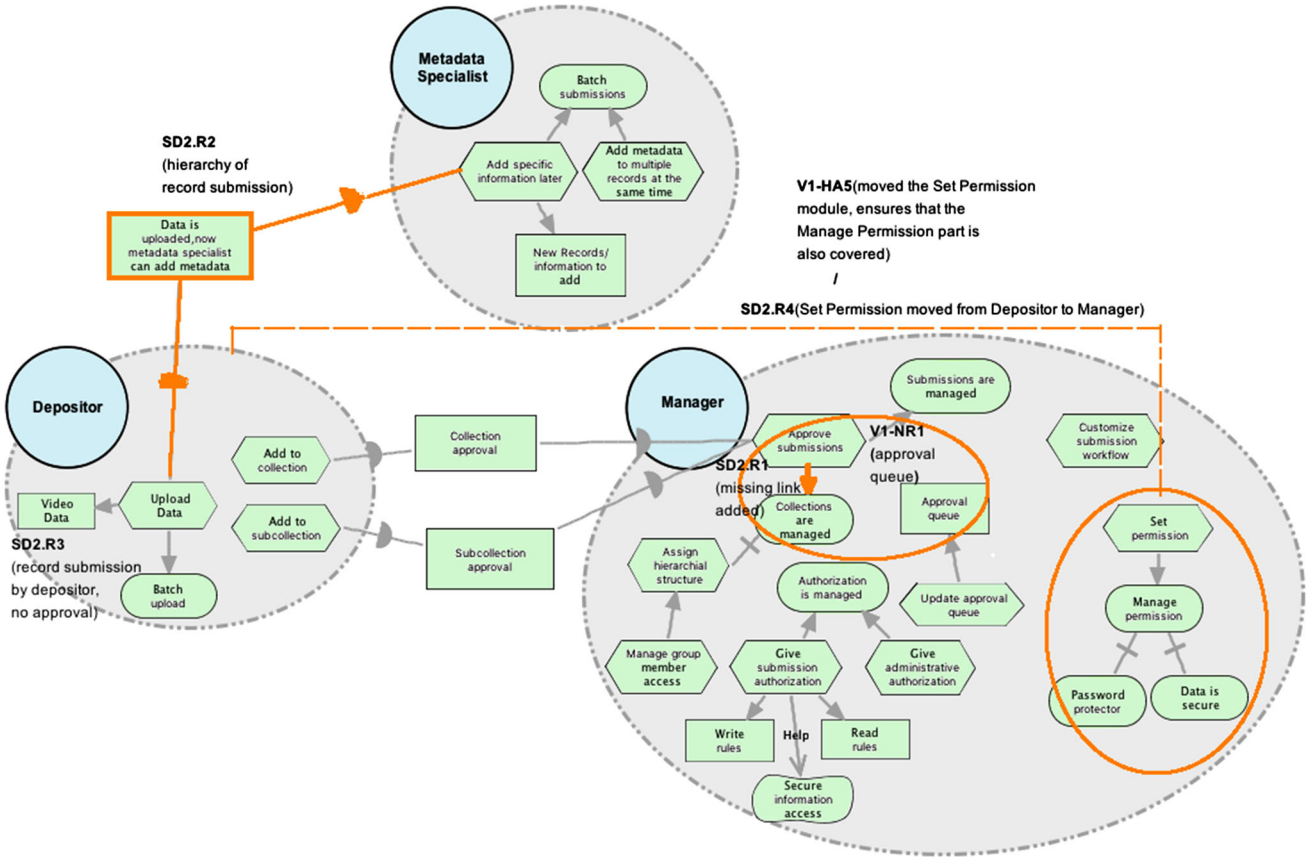

Fig. 10 An example of conflict-centric $i*$ view

such views. Figure 10 provides an illustration by modeling V1-SD2, its resolutions, and their implications in a single view. Note that only two items from V1's submissions were relevant to V1-SD2 (V1-HA5 and V1-NR1), both of which were covered by our resolution options in Table 13 and Fig. 10. This shows conflict-centric views are not only focused but also comprehensive. While specific tools (e.g., TReMer+ [47]) and capabilities (e.g., consistency checking [48, 49]) have been proposed in the literature, following the same spirit of Fig. 5, we offer three OpenOME features based on our struggles while producing Fig. 10.

– *Semantic query* One of our needs was to "find all the model elements mentioned in any given resolution of V1-SD2," which is not currently supported by OpenOME.
– *Subgraph selection* Although OpenOME allows for selection of a set of nodes, consistency checking should be performed when the selected subgraph is moved, copied, or pasted. The "metadata specialist" of Fig. 10 contains a dangling softgoal ("save metadata entry time"). The reason why this softgoal was isolated was because our selection of a subgraph from V1's merged model did not include any element which the softgoal connected with. Subgraph selection what can perform consistency checking and/or warn the users about the inconsistencies would be a valuable feature for OpenOME to address.
– *Free-form annotation* To produce Fig. 10, we had to use a different tool to add a circle to annotate SD2.R1, a dotted arrow to annotate SD2.R4, a free text box to annotate V1-HA5, among others. Having the free-form annotation capabilities or providing settings for the modelers to create their own visual notations, in our opinion, would help engineer conflict-centric views or other $i*$ extensions and variants.

While making the above feature suggestions, we realize the parallel to model merging, namely model projection or slicing [43]. Not only is integrating fragmented models important, but selecting the relevant parts for specific stakeholders to perform model comprehension is equally important. Recent work by da Silva and colleagues [56] empowered the stakeholders with the big picture, the syntax-based, and the concern-based views for better comprehension of (part of) $i*$ models. The different $i*$ modeling features and tool capabilities are surveyed in [57].

6 Conclusions

6.1 Limitations

Some important factors must be taken into account when interpreting our replication results presented in Sect. 4. Our covering of 3 response variables of "a richer domain understanding" can affect the construct validity [3]. "Stakeholder disagreement," for instance, is a domain-dependent construct, and its manifestations in legal and regulatory requirements are well studied (e.g., [58]). Automated ways of measuring, rather than relying on domain expert surveys, can therefore reduce the construct validity. Another threat to construct validity relates to our design of the options used to analyze the new requirements (cf. Table 4). While our intention was to collect one and only one opinion from each survey respondent, the options combined several independent concerns (e.g., value, priority, existence, etc.) and the five options shown in Table 4 might not be the only combinations. This limitation also impacts the starplots of Fig. 4 due to the numeric values defined in Table 4.

One internal validity [3] threat relates to the modelers' self-reported effort data. Our results also suffer from the threat of having a low number of domain experts involved in rating the lists of domain understanding items and in devising the sanity check questions. Confounding variables include the modelers' potentially differing levels in mastering OpenOME, as well as our filtering of the low-quality raw domain-understanding items (cf. Table 2). To mitigate the latter, we have shared our entire study package in [38]. Another threat relates to our preprocessing that filtered out insensible domain-understanding items. Although our intention was to present only sensible items for the domain expert to rate, the filtering that we applied unavoidably introduced some researcher bias. Due to this preprocessing, there were a limited number of domain-understanding items rated (cf. Table 2); rating all the raw items may alter the observations made in Table 3. When the sanity checks were performed, no actual answering was attempted. This could threaten the quality of the resulting $i*$ models.

Regarding the external validity of our replication results, the size and complexity of Scholar@UC may not be directly comparable to those of KHP project studied in [11]; however, the 134 Scholar@UC user stories do present a nontrivial case for requirements engineers. Although our study doubles the number of V and G teams from [11], it is not the statistical generalization, but the theoretical generalization that our replicate study is intended to achieve.

A limitation of our traceability analysis was the post hoc nature of examining $i*$ model elements and their sources relating to the Scholar@UC user stories [36]. Our intention was to test traceability as a factor influencing the modeling quality, e.g., coverage and preservation of different stakeholders' requirements. While our results suggest the significant influence of traceability, the conclusion is valid a posteriori and we do not know how the degree of traceability may vary if this factor is handled at the same time as $i*$ model is created. Another limitation was our directional way of analyzing traceability, i.e., tracing from $i*$ model elements to the 134 Scholar@UC user stories, but not the other way around. As a result, the traceability of all the user stories was not assessed.

6.2 Concluding remarks

Replication is considered a cornerstone of building and evolving scientific knowledge and has been at the heart of science for as long as the scientific method has existed. We independently carried out a theoretical replication of viewpoint merging in RE. Our study confirmed the rich domain understanding observed in the base study. By updating key evolving variables like tool support, we were able to conduct our replication in increasingly realistic settings. Such updates also led to updates of the previously published results [11], most notably, the cost associated with viewpoint-based modeling is now significantly reduced.

Having done repeated comparative explorations, we shifted the study toward an explanatory nature and showed in this paper how such explanations could be made. Although our analysis of traceability was a posteriori and based on correlation, from a practical perspective, we argue that it is relatively less important to distinguish whether traceability causes or correlates to better requirements modeling. Rather, recognizing that traceability makes a difference would allow the modelers and tool builders to take advantage of such a factor in creating better requirements and gaining a richer set of domain understanding. Our explanatory study results confirmed the significant role that traceability plays in requirements goal modeling. However, it is important to realize that traceability is not the only explanatory variable underpinning viewpoint merging. Other factors include modeling style [11], choice of (viewpoint) notations and work plans [23], and even for traceability, the model elements' relationships rather than model elements themselves.

Establishing causality is difficult from a statistical standpoint. However, case study research's objective is not to generalize findings to a population but to probe theory [59]. The viewpoint theory, hinged on the explicit

comparison and merging of stakeholder contributions and their requirements, needs more studies to explore the unknowns, to explain the whys, and to guide the practices. To continue the test and refinement of the viewpoint theory, we invite others to verify our results [38] and to advance RE research toward an empirically backed body of knowledge.

Acknowledgements 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 and Chatura Samarasinghe 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. Shull F, Carver JC, Vegas S, Juristo N (2008) The role of replications in empirical software engineering. Empir Softw Eng 13(2):211–218
2. Lung J, Aranda J, Easterbrook S, Wilson G (2008) On the difficulty of replicating human subjects studies in software engineering. In: International conference on software engineering (ICSE), Leipzig, Germany, pp 191–200
3. Yin RK (2003) Case study research: design and methods. Sage, Beverley Hills
4. Fernández DM , Lochmann K, Penzenstadler B, Wagner S (2011) A case study on the application of an artefact-based requirements engineering approach. In: International conference on evaluation and assessment in software engineering (EASE), Durham, UK, pp 104–113
5. Penzenstadler B, Eckhardt J, Fernández DM (2013) Two replication studies for evaluating artefact models in RE: results and lessons learnt. In: International workshop on replication in empirical software engineering research (RESER), Baltimore, MD, USA, pp 66–75
6. och Dag JN, Thelin T, Regnell B (2006) An experiment on linguistic tool support for consolidation of requirements from multiple sources in market-driven product development. Empir Softw Eng 11(2):303–329
7. Wnuk K, Höst M, Regnell B (2012) Replication of an experiment on linguistic tool support for consolidation of requirements from multiple sources. Empir Softw Eng 17(3):305–344
8. da Silva FQB, Suassuna M, França ACC, Grubb AM, Gouveia TB, Monteiro CVF, dos Santos IE (2014) Replication of empirical studies in software engineering research: a systematic mapping study. Empir Softw Eng 19(3):501–557
9. Sjøberg DIK, Hannay JE, Hansen O, Kampenes VB, Karahasanović A, Liborg N-K, Rekdal AC (2005) A survey of controlled experiments in software engineering. IEEE Trans Softw Eng 31(9):733–753
10. Niu N, Koshoffer A, Newman L, Khatwani C, Samarasinghe C, Savolainen J (2016) Advancing repeated research in requirements engineering: a theoretical replication of viewpoint merging. In: International requirements engineering conference (RE), Beijing, China, pp 186–195
11. Easterbrook S, Yu E, Aranda J, Fan Y, Horkoff J, Leica M, Qadir RA (2005) Do viewpoints lead to better conceptual models? An exploratory case study. In: International requirements engineering conference (RE), Paris, France, pp 199–208
12. Savolainen J, Männistö T (2010) Conflict-centric software architectural views: exposing trade-offs in quality requirements. IEEE Softw 27(6):33–37
13. Carver J (2017) Proposed replication guidelines. http://carver.cs.ua.edu/ReplicationGuidelines.htm. Last accessed: February 2017
14. Basili VR, Shull F, Lanubile F (1999) Building knowledge through families of experiments. IEEE Trans Softw Eng 25(4):456–473
15. Brooks A, Roper M, Wood M, Daly J, Miller J (2008) Replication's role in software engineering. In: Shull F, Singer J, Sjøberg DIK (eds) Guide to advanced empirical software engineering. Springer, Berlin, pp 365–379
16. National Public Radio (2016) Physicist reacts to discovery of gravitational waves. http://www.npr.org/2016/02/11/466458500/physicist-reacts-to-discovery-of-gravitational-waves
17. Gómez OS, Juristo N, Vegas S (2010) Replications types in experimental disciplines. In: International symposium on empirical software engineering and measurement (ESEM), Article 3, Bolzano-Bozen, Italy
18. Mendonça MG, Maldonado JC, de Oliveira MCF, Carver J, Fabbri SCPF, Shull F, Travassos GH , Höhn EN, Basili VR (2008) A framework for software engineering experimental replications. In: International conference on engineering of complex computer systems (ICECCS), Belfast, Northern Ireland, pp 203–212
19. Juristo N, Vegas S (2011) The role of non-exact replications in software engineering experiments. Empir Softw Eng 16(3):295–324
20. Krein JL, Knutson CD (2010) A case for replication: synthesizing research methodologies in software engineering. In: International workshop on replication in empirical software engineering research (RESER), Cape Town, South Africa
21. Callele D, Wnuk K, Borg M (2013) Confounding factors when conducting industrial replications in requirements engineering. In: International workshop on conducting empirical studies in industry (CESI), San Francisco, CA, USA, pp 55–58
22. Penzenstadler B, Fernández DM, Eckhardt J (2013) Understanding the impact of artefact-based RE—design of a replication study. In: International symposium on empirical software engineering and measurement (ESEM), Baltimore, MD, USA, pp 267–270
23. Nuseibeh B, Kramer J, Finkelstein A (1994) A framework for expressing the relationships between multiple views in requirements specification. IEEE Trans Softw Eng 20(10):760–773
24. Easterbrook S, Nuseibeh B (1995) Managing inconsistencies in an evolving specification. In: International symposium on requirements engineering (RE), York, UK, pp 48–55
25. Yu E (1997) Towards modeling and reasoning support for early-phase requirements engineering. In: International symposium on requirements engineering (RE), Annapolis, MD, USA, pp 226–235
26. Daneva M, Damian D, Marchetto A, Pastor O (2014) Empirical research methodologies and studies in requirements engineering: how far did we come? J Syst Softw 95:1–9
27. Strohmaier M et al (2008) Can patterns improve i^* modeling? Two exploratory studies. In: REFSQ, pp 153–167
28. Ernst N, Borgida A, Jureta I (2011) Finding incremental solutions for evolving requirements. In: International requirements engineering conference (RE), Trento, Italy, pp 15–24
29. Krumeich J, Werth D, Loos P (2013) Towards a viewpoint-based modeling method to foster collaborative modeling—conceptual design and implementation. In: PACIS, Paper 249
30. Babar A, Wong B, Abedin B (2014) Investigating the role of business analysts competencies into strategic business requirements gathering. In: PACIS Paper 18

31. Chang S-F, Hsieh P-J, Chen H-F (2015) Key success factors for clinical knowledge management systems: comparing physician and hospital manager viewpoints. Technol Health Care 24(s1):297–306
32. *i*∗ Wiki | Available *i*∗ Tools. http://istar.rwth-aachen.de/tiki-index.php?page=i*+Tools. Last accessed: February 2017
33. OpenOME: An Open-Source RE Tool. https://se.cs.toron-to.edu/trac/ome/wiki/WikiStart. Last accessed: February 2017
34. Scholar@UC. https://scholar.uc.edu. Last accessed: February 2017
35. Scholar@UC on GitHub. https://github.com/uclibs/scholar_uc. Last accessed: February 2017
36. Scholar@UC User Stories. https://github.com/uclibs/scholar_use_cases. Last accessed: February 2017
37. Moody DL, Heymans P, Matulevicius R (2009) Improving the effectiveness of visual representations in requirements engineering: an evaluation of *i*∗ visual syntax. In: International requirements engineering conference (RE), Atlanta, GA, USA, pp 171–180
38. Khatwani C, Jin X, Niu N. doi:10.7945/C25K5P, Hosted on Scholar@UC: https://scholar.uc.edu/show/05741s72s. Last accessed: February 2017
39. Horkoff J, Eric Y (2016) Interactive goal model analysis for early requirements engineering. Requir Eng 21(1):29–61
40. John M, Jaelson C, Manuel K (2013) The evolution of Tropos. In: Bubenko J, Krogstie J, Pastor O, Pernici B, Rolland C, Sølvberg A (eds) Seminal contributions to information systems engineering. Springer, Berlin, pp 281–287
41. Horkoff J, Yu E (2011) Analyzing goal models: different approaches and how to choose among them. In: ACM symposium on applied computing (SAC), TaiChung, Taiwan, pp 675–682
42. Jackson M (1997) The meaning of requirements. Ann Softw Eng 3(1):5–21
43. Brunet G, Chechik M, Easterbrook S, Nejati S, Niu N, Sabetzadeh M (2006) A manifesto for model merging. In: International workshop on global integrated model management (GaMMa), Shanghai, China, pp 5–11
44. Sabetzadeh M, Easterbrook S (2006) View merging in the presence of incompleteness and inconsistency. Requir Eng 11(3):174–193
45. Niu N, Savolainen J, Yu Y (2010) Variability modeling for product line viewpoints integration. In: Annual international computer software and applications conference (COMPSAC), Seoul, South Korea, pp 337-346
46. Sabetzadeh M, Nejati S, Liaskos S, Easterbrook S, Chechik M (2007) Consistency checking of conceptual models via model merging. In: International requirements engineering conference (RE), New Delhi, India, pp 221–230
47. Sabetzadeh M, Nejati S, Easterbrook S, Chechik M (2008) Global consistency checking of distributed models with TReMer+. In: International conference on software engineering (ICSE), Leipzig, Germany, pp 815–818
48. Dam HK, Reder A, Egyed A (2014) Inconsistency resolution in merging versions of architectural models. In: International conference on software architecture (WICSA), Sydney, Australia, pp 153–162
49. Egyed A, Winikoff M, Reder A, Lopez-Herrejon RE (2016) Consistent merging of model versions. J Syst Softw 112:137–155
50. Niu N, Mahmoud A , Chen Z, Bradshaw G (2013) Departures from optimality: understanding human analyst's information foraging in assisted requirements tracing. In: International conference on software engineering (ICSE), San Francisco, CA, USA, pp 572–581
51. Wang W, Niu N, Liu H, Wu Y (2015) Tagging in assisted tracing. In: International symposium on software and systems traceability (SST), Florence, Italy, pp 8–14
52. Niu N, Wang W, Gupta A (2016) Gray links in the use of requirements traceability. In: International symposium on foundations of software engineering (FSE), Seattle, WA, USA, pp 384–395
53. Cohen J (1968) Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. Psychol Bull 70(4):213–220
54. Gosall NK, Singh G (2012) The doctor's guide to critical appraisal. PasTest Ltd, Knutsford
55. Sharpe D (2015) Your chi-square test is statistically significant: now what? Pract Assess Res Eval 20(8):1–10
56. da Silva LF, Moreira A, Araújo J , Gralha C, Goulão M, Amaral V (2016) Exploring views for goal-oriented requirements comprehension. In: International conference on conceptual modeling (ER), Gifu, Japan, pp 149–163
57. Almeida C, Goulão M, Araújo J (2013) A systematic comparison of *i*∗ modelling tools based on syntactic and well-formedness rules. In: International *i*∗ workshop (iStar), Valencia, Spain, pp 43–48
58. Massey AK, Rutledge RL, Antón AI, Swire PP (2014) Identifying and classifying ambiguity for regulatory requirements. In: International requirements engineering conference (RE), Karlskrona, Sweden, pp 83–92
59. Woodside AG, Wilson EJ (2003) Case study research methods for theory building. J Bus Ind Mark 18(6/7):493–508