Supporting Creative RE with *i**

Jennifer Horkoff, Neil Maiden

City University London, UK – horkoff,n.a.m.maiden@city.ac.uk

Abstract. Successful software must be both useful and innovative. Techniques for Requirements Engineering (RE) have mainly focused on utility, with a prominent body of work using goal modeling and analysis to ensure that systems meet user goals. However, these techniques are not designed to foster creativity, meaning that resulting systems may be functionally useful but not sufficiently innovative. Further work has focused on applying creativity techniques for RE through workshops. However, the free-form representation of creative workshop outputs (text and informal diagrams), although flexible, is not grounded in user goals, or able to take advantage of goal model analysis, e.g., trade-off analysis. Furthermore, successfully conducting a creative RE workshop requires much experience and soft-skills, as well as a significant economic commitment. In this work, we summarize initial progress aiming to combine goal modeling and creativity techniques for enhanced RE. We focus on methods and tools for introducing creative ideas to goal modeling, and grounding creative outputs in goal-oriented models. Our focus on tooling and methods help to alleviate the need for expert-lead, costly workshops. We outline and illustrate proposed methods.

Keywords: creativity, istar, goal modeling, method, tool support

1 Introduction

Existing work in Requirements Engineering (RE) has focused primarily on software utility, introducing systematic methods such as goal-oriented modeling and analysis to ensure that requirements meet user needs. Although these techniques have been well studied, little emphasis has been placed on goal-oriented creativity: making sure that goal models capture creative ideas and creative design alternatives.

The past decade has seen the application of creativity techniques to software Requirements Engineering (RE), typically in the form of multi-day workshops (e.g., [7,8]). These workshops gather domain experts and, with the help of experienced facilitators, apply a number of creative activities (e.g., Round Robin, Creativity Triggers, Assumption Busting) in order to elicit creative ideas concerning new software.

Although these workshops have been successful in generating creative requirements, feeding into the design and construction of innovative systems, challenges exist. Workshop output is captured in a free and open format (text, use cases). Although this freedom enables capture of creative output, it makes it difficult to transfer these outputs to a format with is more precise and unambiguous, more amenable for downstream development and for transformation into design and code. The free-form nature of the creative output makes it difficult to perform any sort of systematic analysis of alternative ideas, with rationale for the rejection or acceptance of ideas often lost. Furthermore, creativity workshops are costly and require much guidance by experienced facilitators.

In this paper we outline ideas and progress for a multi-year project exploiting synergies between creativity techniques and goal modeling for RE. We aim to effectively use conceptual models in creative RE activities as part of a creativity methodology guided by tool-support. As such, we reduce reliance on creativity experts and expensive workshops, capturing creative ideas in a more structured form, taking advantage of existing RE modeling and analysis techniques, such as those offered by i* ([11,6]).

In addition to the creative RE workshops, individual methods have been introduced to support creativity in RE (e.g., [9,1]). Although these methods may be useful, they are somewhat fragmented and not joined together as part of one, model and tool-supported process. Our aim is to create a tool-supported framework which would allow for the integration of these and future techniques.

In the rest of the paper, we provide an overview of the proposed creativity method (Sec. 2), illustrate part of the method via an example (Sec. 3), then provide conclusions and future work (Sec. 4). Parts of the proposed method have been illustrated in previous short papers, with ([4]) focusing on an initial proposal for a tool-supported method and an exploratory experiment and ([5]) focusing on illustrating the combination of creativity and goal modeling with an air traffic control example. In this paper, we illustrate different parts of the proposed method with a further example.

2 Method Overview

Existing work (e.g., [10]) has identified many techniques which foster creativity. Creativity can be transformational, changing boundary rules to consider transformative ideas, possibly in another paradigm [2], exploratory, exploring a space of possibilities, or combinatorial, combining together creative output. Creativity techniques can be classified along these dimensions (see BeCreative for example classifications `becreative.city.ac.uk`).

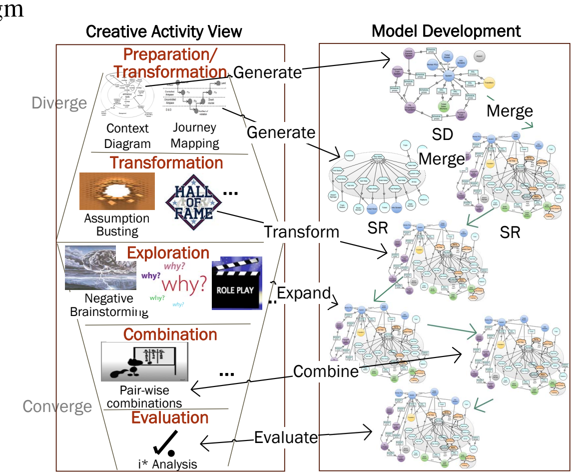


Fig. 1: Overview of Creativity and Goal Modeling Method and Tooling

We propose a method which guides users through a series of creativity activities, with goal models as the output and/or input of such activities. The activity order roughly follows the ordering of activities in previous RE creativity workshops, working through preparatory activities, then activities supporting transformational, exploratory, and combinatorial creativity. See Fig. 1 for an overview. On the left hand side we show the creativity activity view. Here, icons are used to represent various creativity activities, such as assumption busting and role play. Activities will link to either guided instructions, or to external or integrated creativity tools such as Bright-Sparks (`brightsparks.city.ac.uk`). We use models such as context diagrams and journey maps to complement our goal-oriented creative process. We believe that the simplicity and different foci of these models, as compared to i*, leads to different types of creative thinking. Starting with transformations from these simpler models can help users to overcome difficulties in creating initial i* models.

On the right, we show the envisioned evolution of a resulting goal model after progressing through activities. In this paper, we illustrate some of these steps, showing the development of an SD then SR model. Further examples have been presented in [4,5]. In the first stages of development, modeling will be done using external RE tools such as OpenOME or online modeling tools such as draw.io; however, we aim to ultimately include modeling support within the tool.

After an iterative process of creative activities and modeling, the resulting goal model can be processed automatically to derive candidate textual requirements or structured Requirement Specifications, feeding into downstream development.

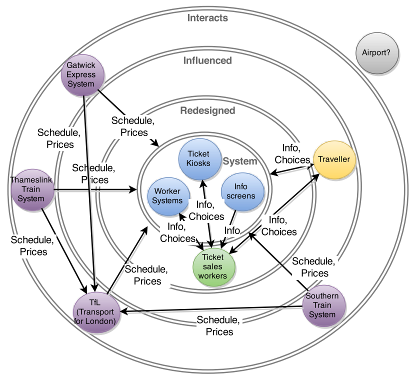
3 Illustrative Example: London Airport Trains

We illustrate parts of our method using a running example of train transport from London airports (inspired by transport from Gatwick Airport), specifically, the purchase of tickets, which offers the possibility of many different train services at different prices and routes, and can be confusing to visitors. Our recent findings as part of an exploratory experiment have shown that new users have trouble beginning to draw an i* model from scratch [3]. As such, in this paper we focus on the initial stages of the creative modeling methodology, which guides users through the creation of a starting, incomplete i* model via the creation of more simpler, intuitive, creativity-inducing models such as context diagrams and customer journey maps.

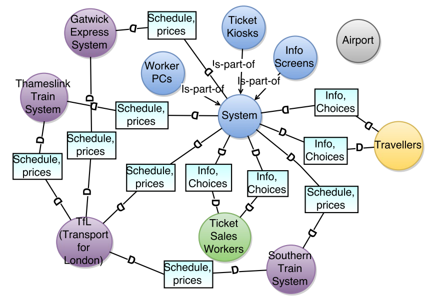
Context to SD. The process starts by urging users to draw a context diagram for the system. In this diagram, actors are drawn in a series of concentric circles, as actors move farther away from the center, they are less impacted by the new system. We show an example Fig. 2a. At the center are the core system actors, then the ticket sales workers, then the travelers, the system for each train and tube company, then, on the outer layer lies the airport, as the relation to the system is not currently well-understood. Actors are connected by arrows showing flows of information.

The creation of a context diagram is not necessarily creative – the model could capture the as-is situation without changes. However, the simple structure of this model well supports transformational creativity when the positions of the actors are challenged, potentially making transformative changes. For example, what is the role of the Airport, could it be moved closer to the core system, could there be a flow of information? Perhaps travelers could be provided information as they depart their flight? Perhaps they could book transport tickets with their flight? For simplicity, we continue the example without applying these ideas.

The information contained in a context diagram can form the beginnings of an i* SD diagram. Context actors are SD actors, while information flows are potential dependencies. We show the result of a proposed automatic conversion to an initial SD model in Fig. 2b. User are encouraged to rearrange, modify or add to this auto-generated figure.



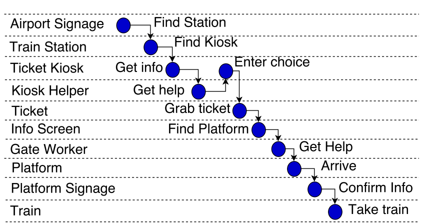
(a) Context Diagram for Airport Train System



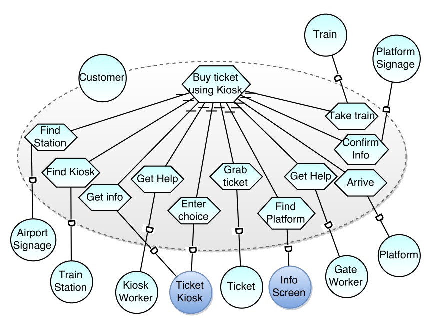
(b) Initial SD Diagram for Airport Train System - Transformed from Context Diagram

Fig. 2: Transformation from Context to SD

Journey Map to SR. A customer journey map is a simple way for users to explore one path through the system, capturing system actors or interaction points, and "touchpoints". We can explore the current path of purchasing a ticket from an airport train kiosk, illustrated in Fig. 3a. As with context diagrams, as-is journey maps are not necessarily creative. However, their structure forces users to explore and question the boundaries of a system, thus they are often used as part of exploratory and transformative creativity. In this case, when drawing the diagram we discovered that the first step in buying a ticket was finding the train station from within the airport, relying on the airport to provide appropriate direction and signage. In this case, we have expanded the boundary of our system as compared to the context diagram, when we did not know exactly what role the airport would play in our journey. Similarly, the ticket buying process ends when the travelers board the train. How can the train be included as part of our system design? Again, for simplicity, we continue the example without deeply exploring these creative prompts.



(a) Customer Journey Map for Buying a Ticket Using Kiosk



(b) Initial SR Diagram for Airport Train System - Transformed from Journey Map

Fig. 3: Transformation from Customer Journey Map to SR (colors match Fig.2)

As with context diagrams, we can make a simple transformation from journey maps to i*, specifically SR diagrams. In this case the columns are potential actors, while touchpoints are potential tasks. As we can see in the transformed SR model in Fig. 3b some of the transformations are imperfect, for example, a ticket is likely a resource, not an actor. In this case we present the SR model to the user as a starting point and ask for their input in changing and rearranging the elements, again with the theory that it is easier to update and fix an i* diagram than to build one from scratch.

We can see that there are differences in the SD diagram produced via the context diagram and the SR diagram produced via the journey map, for example, Customer vs. Traveler. We can offer users a partial merge of the two diagrams, producing a new SR. Here, elements and actors with matching names are merged, while those with similar names can be clustered. We leave it to the user to finish the merge using their judgment. The end result will be a partial SR model which is roughly consistent with the other diagrams. We show the results of such a human-aided merge on the left of Fig. 4.

Exploratory Creativity applied to SR. We can make a simple mapping between several exploratory creativity techniques and the SR diagram, with the purpose of helping users to fill out the SR detail. For example, one of the existing creativity techniques is called "Why why why?", urging users to constantly question the motivation for each element of a system design. This, obviously, can help the user to move up the i* model, adding higher level intentions until the why question is no longer sensible. Similarly, we can consider "how how how?" and "who who who?", helping to elicit operationalizations and dependencies. Another useful technique, "negative brainstorming", urges user to think of what can go wrong, helping to elicit negative contribution links and other related elements. Conversely, we could apply "positive brainstorming" to help find positive contributions. We show some results of these activities in Fig. 4 in orange/bold.

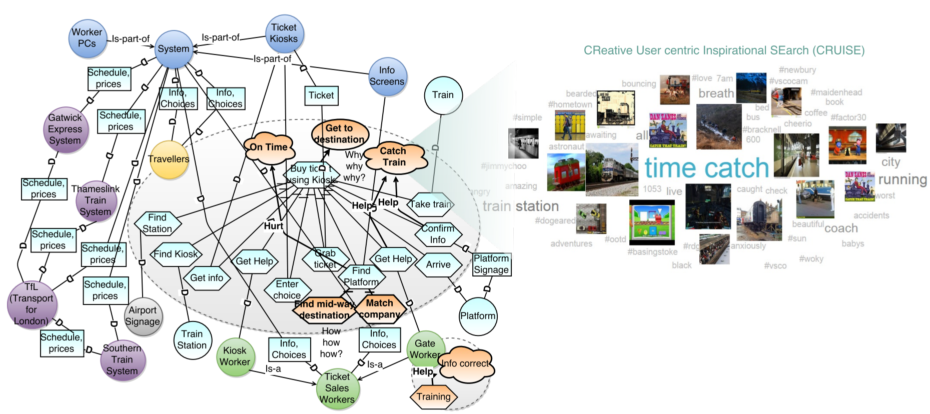


Fig. 4: SR Diagram Resulting from Context and Journey Map with Suggestions for Catch Train from CRUISE (Orange/Bold show Additional Creative Results)

Using i* Structure for Creativity. Thus far we have focused on techniques which help users to build and fill in the details of an i* model. Other techniques can take the i* model as an explicit input. Element names, such as Catch Train, could be used as input for creative search tools such as cruise.imuresearch.eu (shown in Fig. 4), with generated ideas (e.g., warnings for full trains, identifying which trains have snack cars) incorporated back into the goal model. Combinatorial creativity techniques such as pairwise comparison, can use the structure of the model to automatically compute different combinations of actors, suggesting potential new dependencies. Techniques, such as the one described in [1], could take as input i* element labels and suggest unfamiliar combinations of elements. Finally, i* analysis, such as in [6] can be used to evaluate and select the outputs of creative activities as captured in the model. We intend for our method and tooling to support such activities via incremental development, with successive releases including support for more and more creativity activities.

4 Conclusions and Future Work

We have outlined a tool-supported methodology for combining creativity and i* modeling. Our proposal incorporates creative ideas into the goal-oriented RE process and grounds creative ideas in user goals, allowing for systematic analysis and (partial) transformation to requirements specifications.

We are currently working on the design and flow of the proposed tooling. We intend to pilot the method internally, then to test successive versions with willing industry contacts, eventually leading to larger-scale, in-situ deployments and case studies.

Acknowledgments. This research is supported by an ERC Marie Skodowska-Curie Intra European Fellow-ship (PIEF-GA-2013-627489) and a Natural Sciences and Engineering Research Council of Canada Postdoctoral Fellowship (Sept. 2014 - Aug. 2016).

References

1. T. Bhowmik, Nan Niu, A. Mahmoud, and J. Savolainen. Automated support for combinational creativity in requirements engineering. In *IEEE 22nd International Requirements Engineering Conference (RE14)*, pages 243–252, 2014.
2. M. A. Boden. *The Creative Mind*. London: Abacus, 1990.
3. J. Horkoff. Observational studies of new i* users: Challenges and recommendations. In *First iStar Teaching Workshop (iStarT15)*, 2015.
4. J. Horkoff and N. Maiden. Creativity and conceptual modeling for requirements engineering. In *CreaRE: Fifth International Workshop on Creativity in Requirements Engineering*, 2015.
5. J. Horkoff, N. Maiden, and J. Lockerbie. Creativity and goal modeling for software requirements engineering. In *Creativity & Cognition (poster)*, 2015.
6. J. Horkoff and E. Yu. Interactive goal model analysis for early requirements engineering. *Requirements Engineering*, pages 1–33, 2014.
7. N. Maiden, S. Jones, K. Karlsen, R. Neill, K. Zachos, and A. Milne. Requirements engineering as creative problem solving: A research agenda for idea finding. In *18th IEEE International Requirements Engineering Conference (RE10)*, pages 57–66, 2010.
8. N. Maiden, C. Ncube, and S. Robertson. Can requirements be creative? experiences with an enhanced air space management system. In *Software Engineering, 2007. ICSE 2007. 29th International Conference on*, pages 632–641, May 2007.
9. L. Mich, C. Anesi, and D. M. Berry. Requirements engineering and creativity: An innovative approach based on a model of the pragmatics of communication. In *Requirements Engineering: Foundation of Software Quality (REFSQ04)*, 2004.
10. M. Michalko. *Thinkertoys: A Handbook of Creative-Thinking Techniques*. Potter/TenSpeed/Harmony, 2010.
11. E. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *3rd IEEE International Symp. on Requirements Engineering*, pages 226–235. IEEE, 1997.