Information and Software Technology 59 (2015) 109–135

Contents lists available at ScienceDirect

Information and Software Technology

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

Synergy between Activity Theory and goal/scenario modeling for requirements elicitation, analysis, and evolution

Geri Georg [a,*], Gunter Mussbacher [b], Daniel Amyot [c], Dorina Petriu [d], Lucy Troup [e], Saul Lozano-Fuentes [f], Robert France [a]

[a] Computer Science Department, Colorado State University, Fort Collins, CO, USA
[b] Department of Electrical and Computer Engineering, McGill University, Montreal, Canada
[c] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
[d] Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada
[e] Department of Psychology, Colorado State University, Fort Collins, CO, USA
[f] Department of Microbiology, Immunology and Pathology, Colorado State University, Fort Collins, CO, USA

A R T I C L E   I N F O

Article history:
Received 17 December 2013
Received in revised form 11 October 2014
Accepted 6 November 2014
Available online 13 November 2014

Keywords:
Requirements engineering
Activity Theory
User Requirements Notation
Goal modeling
Scenario modeling

A B S T R A C T

Context: It is challenging to develop comprehensive, consistent, analyzable requirements models for evolving requirements. This is particularly critical for certain highly interactive types of socio-technical systems that involve a wide range of stakeholders with disparate backgrounds; system success is often dependent on how well local social constraints are addressed in system design.

Objective: This paper describes feasibility research, combining a holistic social system perspective provided by Activity Theory (AT), a psychological paradigm, with existing system development methodologies and tools, specifically goal and scenario modeling.

Method: AT is used to understand the relationships between a system, its stakeholders, and the system's evolving context. The User Requirements Notation (URN) is used to produce rigorous, analyzable specifications combining goal and scenario models. First, an AT language was developed constraining the framework for automation, second consistency heuristics were developed for constructing and analyzing combined AT/URN models, third a combined AT/URN methodology was developed, and consequently applied to a proof-of-concept system.

Results: An AT language with limited tool support was developed, as was a combined AT/URN methodology. This methodology was applied to an evolving disease management system to demonstrate the feasibility of adapting AT for use in system development with existing methodologies and tools. Bi-directional transformations between the languages allow proposed changes in system design to be propagated to AT models for use in stakeholder discussions regarding system evolution.

Conclusions: The AT framework can be constrained for use in requirements elicitation and combined with URN tools to provide system designs that include social system perspectives. The developed AT/URN methodology can help engineers to track the impact on system design due to requirement changes triggered by changes in the system's social context. The methodology also allows engineers to assess the impact of proposed system design changes on the social elements of the system context.

1. Introduction

A prime goal of investment in requirements engineering (RE) is to produce a comprehensive, consistent set of system requirements covering various aspects of the system, e.g., operational environment constraints, general functionality requirements, and so-called non-functional requirements such as performance and security. Since systems and their requirements evolve over time, system designers also benefit from requirements models that can be easily evolved and that provide rationale for design decisions, where rationale includes the source of a requirement such that its continued relevance can be tested over time as the system evolves. The psychological framework of Activity Theory (AT) [14] may be used to help identify the societal constraints that need to be addressed

* Corresponding author.
E-mail addresses: georg@cs.colostate.edu (G. Georg), Gunter.Mussbacher@mcgill.ca (G. Mussbacher), damyot@eecs.uottawa.ca (D. Amyot), petriu@sce.carleton.ca (D. Petriu), Lucy.Troup@colostate.edu (L. Troup), slozano@colostate.edu (S. Lozano-Fuentes), france@cs.colostate.edu (R. France).

http://dx.doi.org/10.1016/j.infsof.2014.11.003
0950-5849/

by a system in order to help ensure its acceptance and success. AT systematically leads the designer through a process that can identify previously unknown stakeholders and their implicit social constraints as part of requirements elicitation. This is particularly critical for certain types of socio-technical systems that employ citizens to collect the data used by officials to, for instance, make policy. These systems can be highly interactive and often involve a wide range of stakeholders with very different backgrounds. Multiple studies indicate that the success of these types of systems often depends on how well local social constraints have been taken into account and addressed in the system design, as discussed below.

An example of a socio-technical system is one that Dr. Lozano-Fuentes has helped develop in collaboration with universities and public health officials in Mexico [13]. This system is designed to support collecting vector surveillance (in this case mosquito breeding site) data, which is used to provide evidence for health officials deciding vector control policy. Currently public health workers collect data regarding vector populations, but with the advent of mobile devices, the general public can also collect this data. In many parts of the world, vector surveillance is critical to the control of diseases such as Dengue, a mosquito-borne disease. Spiegel et al., van den Berg et al., and Vanlerberghe et al. report studies of different surveillance systems (both automated and manual) that have been deployed to help track mosquito vectors [46,49,50]. These authors report that the few successful programs addressed local community social issues, and that the local community drove and participated in data collection in these successful programs. In these socio-technical systems, the quantity of timely data, in addition to its accuracy, is critical to support vector control decisions. We also note that the general population from whom the data is gathered is affected by the system results in the form of the vector control decisions (e.g., when and where to spray for mosquitos).

Another example of a socio-technical system is that of home automation. Again, the general public interacts with the system: perhaps controlling all or portions of it, acting as sources of data for it through their actions, and being affected by it as the system makes changes in the home. Studies of home automation discuss some of the challenges in the adoption of these systems. For example, Brush et al. note the large diversity of target users, and cite barriers to adoption that include a tradeoff between convenience for users and security functions [8]. The study by Mennicken and Huang identified that it can be difficult for users to imagine and plan for the changes in their daily routines that come with home automation. Users sometimes chose to perform tasks without automation in response to these changes [34]. Takayama et al. identified social issues such as friction between household members regarding automation, anxiety about the system, and cognitive issues related to getting the system to work properly [47]. Sadri, in a survey of intelligent automation in several areas including home automation, cites concerns over security and social/ethical implications of these systems and the data they collect [44].

Our research is aimed toward the application of social sciences theories to the problem of determining the social requirements of systems such as the above, where data is collected directly or indirectly from the general public and is used to control some noticeable aspect of their environment. Our goal is to apply social sciences theories to identify these requirements in ways that allow them to be taken into account in system design, with the expectation that addressing them will help improve the successful deployment and use of these types of systems. We have identified several important requirements for such a social theory: (1) it must have prior successful use in the social sciences in identifying a system's social constraints and explaining how addressing them or not affected the system, (2) it should be used to at least some extent in the field of software development, (3) it must be adaptable for use in software development by persons who are not social scientists, (4) the adapted theory must produce repeatable results that are supported by automation as much as possible, (5) methodologies must be identified that allow the adapted theory results to be used in conjunction with existing system development methodologies and tools, and (6) the adapted theory results must identify social constraint requirements that might not have been found otherwise.

The second requirement may be relaxed to include social theory that has not previously been applied to the field of software development. However, if the social theory has been used before in such a context, it is more likely that it is usable by non-social scientists as there is at least some evidence to this effect. The sixth requirement is one of validation against other comprehensive system development methodologies, and has not yet been completed.

In this context, the main research question of this paper is: *can Activity Theory (a particular social science theory) be adapted, formalized, and combined with an existing system development methodology to support the repeatable discovery of non-obvious social requirements in socio-technical systems with local constraints*? The satisfaction of the above six requirements will result in a positive answer to this research question.

The contribution of this paper is to show that AT fits the first five requirements. We briefly discuss how AT meets the first two requirements, using examples from the social sciences and limited use of AT concepts in systems development, particularly in Human-Computing Interaction (HCI) development. We then focus on the technical aspects of the above requirements, specifically how we can constrain AT to be used by non-social scientists, how much we can automate its use to produce repeatable results, and how we can integrate it into an existing system design methodology and tools. To address this last issue, we have developed an RE methodology that leverages synergistic opportunities provided by the human activity formulations embodied in AT and the formal definitions of goal and scenario modeling of the User Requirements Notation (URN) [2,23]. The URN tool jUCMNav, developed at the University of Ottawa, is the tool we have used for the work described in this paper [24].

Our combined methodology can be used during initial system development and also, through tracing capabilities, it provides a way to monitor the effects of potential changes in the system. For example, changes might occur in the stakeholders' social constraints that can affect the system. These changes can provide a basis for planning system evolutions. Our method provides a mechanism to explore changes across basic objectives, stakeholders, and social constraints that occur over time, including their effect on the overall system and individual user needs. The combination of AT and URN allows the social constraints that are identified through AT to be traced back and forth to their respective places in goal models and in system design. This bidirectional tracing can be used to validate that a system design (or an evolved design) preserves social constraints that are still relevant for the system or to identify new social constraints that must be addressed in evolving designs.

We present a short introduction to AT and URN as well as the rationale for using them in Section 2. Section 3 describes how we rigorously defined an AT language for use in requirements engineering and how it is formally mapped to URN. Section 4 describes one possible methodology using AT and URN that we demonstrate in the paper. Section 5 describes the example system based on the Mexico Vector Surveillance system that we have used as a proof-of-concept for our approach and how both AT and URN are applied to it. Section 6 demonstrates how our approach can be used in the context of system evolution to create a possible extension to the example system. We discuss the results of our approach in the context of the example system, including the complementary nature of AT and URN, as well as our experiences, lessons learned, and threats to validity in Section 7. In Section 8, we contrast our