A Holistic Socio-Technical Approach to Systems Analysis: Trace-Linking Activity Theory to UML Activity Diagrams

Emergent Research Forum (ERF)

Fabian Wiser
TU Braunschweig
winfo-radar@tu-braunschweig.de

Carolin Durst
HS Ansbach
carolin.durst@hs-ansbach.de

Abstract

The complexity of socio-technical interaction is increasingly representing a challenge to systems analysists and designers. A good practice, therefore, is to analyze socio-technical systems extensively to discover demands, needs and requirements for a better design solution both on a macro- and micro-level perspective. In order to combine both perspectives, we developed an Activity Theory meta-model and connected it with UML Activity Diagrams by means of a trace-linking approach. The resulting trace-link model features both holistic and granular analysis, in which Activity Theory is used to describe the broad activity context and the UML Activity Diagram is used to describe the process of actions undertaken by the users.

Keywords

Activity Theory, UML Activity Diagrams, Trace-Linking, Socio-technical Systems

Introduction

Technology is evolving faster than ever in today's world and, associated therewith, the complexity of socio-technical interaction is increasingly representing a challenge to systems analysists and designers. A good practice, therefore, is to analyze socio-technical systems extensively to discover demands, needs and requirements for a better design solution both on a macro- and micro-level perspective (Georg et al., 2015). In order to do so, requirement engineers and system designers need proper tools and methods. Such a method, as we propose, is Activity Theory (AT) in combination with UML Activity Diagrams (UML AD). AT is frequently used in research to investigate human-computer interaction (Kaptelinin and Nardi, 2012) and information systems (Allen et al., 2011), and was often suggested as an explorative lens for requirements engineering (Fuentes-Fernandez et al., 2009) as well as system design (Collins et al., 2002).

Activity Theory

AT is a conceptual framework to analyze human activity within its context (Engeström, 1987). Although the theory was developed far before the invention of the personal computer or collaborative information systems, it sheds light on the most important facets of socio-technical systems, as can be seen by its illustrative frame for activities, namely the Activity System (AS) (see Figure 1).

The elements of the AS are briefly described in Table 1:

AS Element	Description
Basic Components	*The basic components Subject, Community, Object and Outcome describe the setting and conditions of the activity.*
Subject	Subjects are the prime actors and perform the activity (collectively).

Community	An activity is mostly always performed within the context of stakeholders who are often also responsible for the definition of the objective.
Object(-ive)	The subjects are guided by objectives, which motivate the activity.
Outcome	By finalizing the activity, an object is transformed into an outcome.
Mediators	*Mediators are artifacts which are shaped by cultural and historical evolution and assist the subject and community to perform the activity.*
Tool	The subjects transform the object by means of physical, social and psychological tools, into an outcome (Subject-Object Interaction).
Rules	Rules are defined by the community and have to be followed by the subject (Subject-Community Interaction).
Division of Labor	The community defines the responsibilities and the distribution of work to reach the object (Community-Object Interaction).

Table 1. Description of Activity System Elements (Engeström, 1987; Kaptelinin and Nardi, 2012)

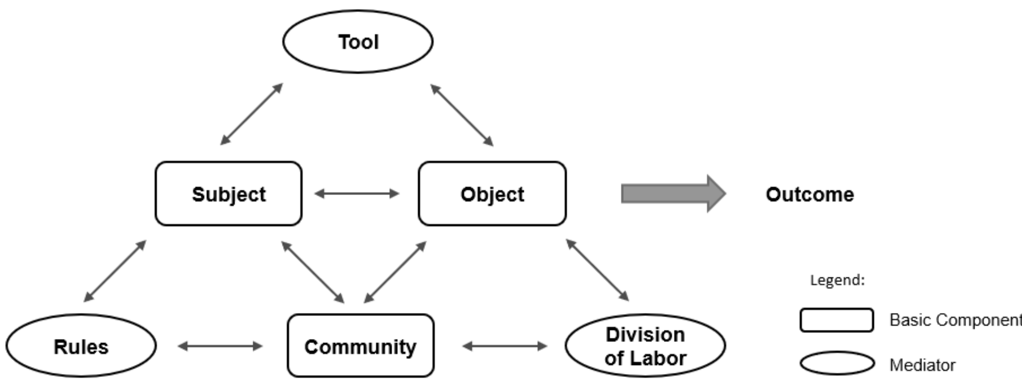

Figure 1. The Activity System (Engeström, 1987)

<u>*Contradictions:*</u> Contradictions are the main drivers of change and development within an activity (Kaptelinin and Nardi, 2012). There exist various interpretations of the contradiction concept, however, for the sake of systems analysis, we suggest to construe them as problems, limitations and tensions that system designers want to solve. Contradictions arise on four different levels: Within an AS element (e.g. a software has a bug) [1st Level]; between two AS elements (e.g. the user does not know how to operate the system) [2nd level]; between the current activity system and a more evolved version of it (e.g. users do not adapt to a newly implemented system) [3rd level]; and between two different ASs that share resources (e.g. limited access to the system for users who perform different activities) [4th level] (Engeström, 1987).

<u>*Hierarchy:*</u> In AT, an activity can be divided into three levels: The macro-level activity (e.g. a user designs new business processes), represented by the AS (see Figure 1); the micro-level procedural actions (e.g. a user writes a process description into the editor field) which are directed towards goals and altogether lead to the completion of the activity; and the nano-level subconscious operations (e.g. the user's keystrokes while writing) performed by a subject during an action (Engeström, 1987).

Objectives and Approach

Although AT has broad analysis capabilities for socio-technical systems, prior research identified some limitations (cf. Clemmensen et al., 2016, Wiser et al., 2019) regarding the theory, such as the absence of a clear and standardized method to describe workflows, that is the actions and operations in AT. AT to date rather serves as a high level-analysis tool and practitioners often ignore the hierarchical micro-level action (and nano-level operation levels) beneath the activity system representation.

In this paper, we develop in two steps a trace-link model by conceptually mapping AT (activity perspective) to a process modeling language (action perspective) in order to overcome the gap between macro-level activity and micro-level actions (note: the nano-level operations are preliminary not considered in this paper): In the first step, we define an AT meta-model like other researchers already did (cf. Fuentes-Fernandez et al., 2009; Georg et al., 2015; Martins, 2007). In the second step, we trace-link the AT meta-model to a process modeling language following a similar approach to Georg et al. (2015) who trace-linked their AT meta-model to the Goal-Oriented Requirement Language (GRL), a method which especially supports the capturing of non-functional requirements.

Other to previous research, we highly emphasize on the hierarchical concept of AT, develop a AT meta-model especially suited for Systems Analysis and Design and connect it with a simpler and more frequently used modeling language than GRL, namely the UML AD, to represent the action level of AT.

Results

The AT meta-model of Martins (2007) was considered as a reasonable starting point for our trace-linking approach because of the model's plain and comprehensive composition. Resultant, we added classes and changed the model (see Figure 2) according to the critique of Georg et al. (2015) and other important facets of AT that were neglected before:

1. The structure of the meta-model was designed according to the composition of the traditional AS. Only the rules and community element are swapped to avoid confusing line crossings.
2. Tools were exempted from its role as mediators within the diagram. We believe that mediation and interaction are much better illustrated within the UML AD to be linked.
3. A reflexive association was added from and to the activity class, to highlight the evolution of the activity system. Evolution illustrates how and why an activity emerged.
4. Contradictions were added to the class diagram because they are insightful for systems analysis and design and should be considered for requirements elicitation.
5. An activity is not only embedded in a setting described by the nodes of the AS but can be part of an activity conglomerate (Engeström, 2008). Due to this fact, the activity network and overall objective were linked to the activity class.

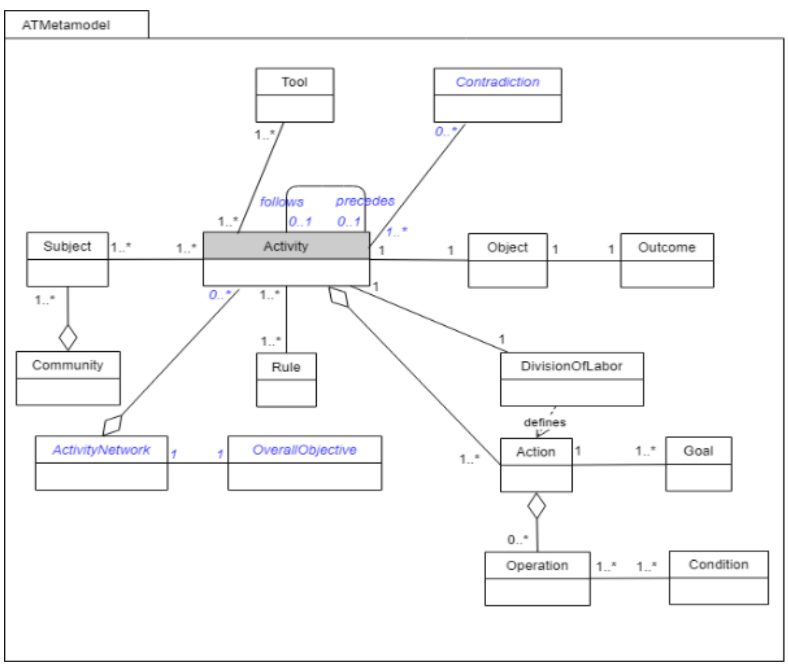

Figure 2. Activity Theory Meta-Model (own representation based on Martins, 2007; added elements in blue italics)

After the design of a more complete AT meta-model for systems analysis, we explore its possible mapping to the UML AD meta-model. Therefore, the concept of trace-linking is applied, as trace-links are defining relationships between models by conceptually connecting the different source and target models with one another (Paige et al., 2011). Trace-links have the advantages that both meta-models do not need to be drastically modified and pertain in their original state, a 'bi-directional' information exchange between components of both models can take place and, that the models' information can be verified against one another (Georg et al., 2015).

In Figure 3 the AT meta-model and the UML AD meta-model are mapped via the *ATUMLADTraceLinking* package. The underlying rationale and description of the defined connections are described below:

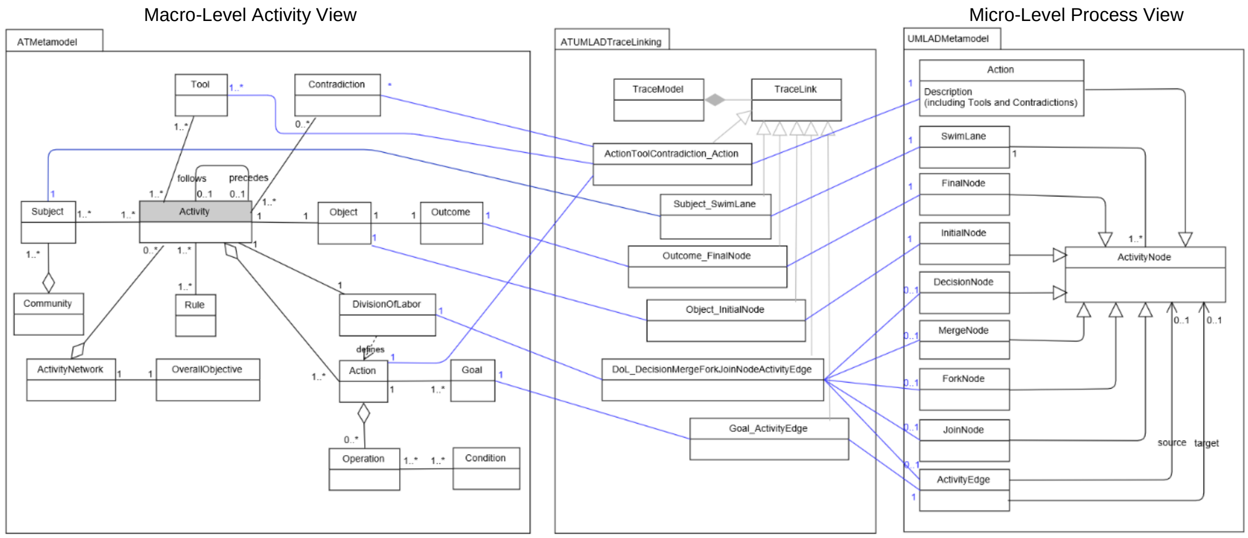

Figure 3. Trace-Linking Activity Theory and UML Activity Diagrams (Own representation based on Bisztray and Heckel, 2014; Martins, 2007; Paige et al., 2011)

Subject_Swimlane: We mapped subjects to swim lanes. Thus, every active community member is assigned to a swim lane, and it's clear to the reader of the UML AD diagram who performs the action.

ActionToolContradiction_Action: We added tools and contradictions to the UML AD action's description, which are accordingly traced from the AT meta-model. Consequently, every action performed can precisely be described and provides insights about the tools used. On occasion, an action can be affected by a 1st- or 2nd-level contradiction. By analyzing these contradictions on the micro-level, solutions can be defined more adequately than by only looking at the macro-level AS.

Outcome_FinalNode: When the outcome is reached, all necessary actions have been performed. Consequently, outcome and final node are linked.

Object_InitialNode: Although the object is something subjects are striving for, it is also the motivator and starting point for activity. Thus, object and initial node are linked.

DoL_DecisionMergeForkJointNodeActivityEdge: Division of labor defines how tasks within an activity have to be performed. Therefore, the division of labor constitutes the UML AD counterparts decision-, merge-, fork- and joint node, as well as the sequence of actions, that is, the activity edges. It is important to note that other relevant DoL aspects like power hierarchy or authority cannot be mapped to the UML AD.

Goal_ActivityEdge: Whenever a goal is reached by an action, the activity is guided by the activity edge towards the next action to be performed.

Overall, the trace-link model features both a high-level analysis of the activity as well as a deep-dive on the process level, respecting the hierarchical notion and the action level of AT. On account of that, we considered not mapping the community (passive stakeholders, social environment), rules and the activity network to UML AD, as these are components that have to be analyzed within the overall activity's context. Further, we excluded operations and conditions from the mapping because they are rather important for human-computer interaction analysis, which is out of scope in the presented trace-linking model.

Discussion

In this paper, we developed a conceptual mapping by means of trace-links to connect AT with UML ADs. Our results extend prior research as we added important activity theoretical concepts to the existing meta-model of Martins (2007) and trace-linked it to a more common modeling language than GRL, namely UML ADs (Georg et al., 2015). Further, we defined a clear difference between the elements' relevance on the macro- or on the micro-level, by trace-linking AT actions with UML AD actions and, therefore, emphasizing the hierarchical concept of AT. Also, in contrast to other studies, we respected in our model the occurrence of contradictions on both levels of analysis. The micro-level process view especially helps to determine where and when in the performed actions 1st- or 2nd-level contradiction between subjects and tools appear. All other resulting contradictions, then, can still be holistically analyzed within the macro-level AS. As a whole, the developed trace-link model with more elaborate contradiction analysis capabilities can assist in discovering demands and requirements for systems design.

However, the presented results also have limitations: AT includes many more complex and wide-ranging concepts, which cannot be mapped all together into one single meta-model. As this paper presents work which is still in progress, we have not yet tested the designed model in a practical setting. Future steps should, therefore, include follow-up work on the evaluation by means of a qualitative case study.

Acknowledgements

This research and development project is / was funded by the German Federal Ministry of Education and Research (BMBF) within the Program "Innovations for Tomorrow's Production, Services, and Work" (02K16C190) and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication.

References

Allen, D., S. Karanasios and M. Slavova (2011). "Working with activity theory. Context, technology, and information behavior" *Journal of the American Society for Information Science and Technology* 62 (4), 776–788.

Bisztray, D. and R. Heckel (2014). "Combining termination proofs in model transformation systems" *Mathematical Structures in Computer Science* 24 (4).

Clemmensen, T., V. Kaptelinin and B. Nardi (2016). "Making HCI theory work. An analysis of the use of activity theory in HCI research" *Behaviour & Information Technology* 35 (8), 608–627.

Collins, P., S. Shukla and D. Redmiles (2002). "Activity Theory and System Design: A View from the Trenches" *Computer Supported Cooperative Work* 11, 55–80.

Engeström, Y. (1987). *Learning by Expanding. An Activity-Theoretical Approach To Developmental Research*.

Engeström, Y. (2008). "Enriching activity theory without shortcuts" *Interacting with Computers* 20 (2), 256–259.

Fuentes-Fernandez, R., J. J. Gomez-Sanz and J. Pavon (2009). "Requirements Elicitation and Analysis of Multiagent Systems Using Activity Theory" *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 39 (2), 282–298.

Georg, G., G. Mussbacher, D. Amyot, D. Petriu, L. Troup, S. Lozano-Fuentes and R. France (2015). "Synergy between Activity Theory and goal/scenario modeling for requirements elicitation, analysis, and evolution" *Information and Software Technology* 59, 109–135.

Kaptelinin, V. and B. Nardi (2012). *Activity Theory in HCI. Fundamentals and Reflections:* Morgan & Claypool Publishers.

Martins, L. E. G. (2007). "Activity Theory as a feasible model for requirements elicitation processes" *Scientia Interdisciplinary Studies in Computer Science* 18 (1), 33–40.

Paige, R. F., N. Drivalos, D. S. Kolovos, K. J. Fernandes, C. Power, G. K. Olsen and S. Zschaler (2011). "Rigorous identification and encoding of trace-links in model-driven engineering" *Software & Systems Modeling* 10 (4), 469–487.

Wiser, F., C. Durst and N. Wickramasinghe (2019). "Using Activity Theory Successfully in Healthcare. A Systematic Review of the Theory's Key Challenges to Date". In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*.