A Model for the Design of Interactive Systems based on Activity Theory

Sebastian Döweling
SAP Research Darmstadt
Bleichstr. 8, 64283 Darmstadt, Germany
sebastian.doeweling@sap.com

Benedikt Schmidt
SAP Research Darmstadt
Bleichstr. 8, 64283 Darmstadt, Germany
benedikt.schmidt@sap.com

Andreas Göb
SAP Research Darmstadt
Bleichstr. 8, 64283 Darmstadt, Germany
andreas.goeb@sap.com

ABSTRACT
Activity theory has gained increasing popularity as a tool for the design of interactive systems. Its central idea is that user actions can only be fully understood when analyzed within the context of motive-driven activities. However, this context is currently only partially explicit in existing models; e.g. the well-known model by Engeström promotes social aspects of context, but does not make the physical and technical environment – relevant aspects for interactive systems design – explicit.

We present a model that incorporates the social, physical and technical context of activities. This model supports designers and developers in understanding users, their activities and the respective contexts, and allows deriving requirements for the design process directly from the model. We believe this will prove especially helpful when designing for novel devices and multi-user scenarios. We illustrate the practical use of our model with examples from two domains: composition of IT services and co-located collaborative modeling.

Author Keywords
Activity Theory; Systems Design; Context

ACM Classification Keywords
H.5.2 User Interfaces: Theory and methods

General Terms
Design; Human Factors; Theory

INTRODUCTION
In the last decade, the focus in personal computing has significantly shifted from desktop computers in office situations to new devices with different form factors, such as tablet computers or large interactive displays, which are used in a variety of ways, with different motives and in different contexts (e.g. the same mobile phone model could be used by a student to kill time with mobile gaming in spare hours, but also by a business consultant to check the latest version of a presentation before entering a meeting at a customer's site).

Yet, most contextual aspects are absent in traditional HCI models like the human-information processor [4]. Motivated by the criticism of the lack of context in these models (e.g. [2]), activity theory (AT) [23, 36] has been proposed as a framework for new methods, which overcome this limitation (e.g. [20, 22]). The use of AT is supported by the results of studies, which showed that people organize their work in terms of activities that are executed to meet some overall objective, often in collaboration with others (cf. [3]); moreover, AT has already been adopted as a design approach by a number of researchers (e.g. [2], [12] or [3]).

However, AT is often used rather as a set of guiding principles than as a model from which problems or requirements can be directly derived. Moreover, while AT "requires that the scope of analysis be extended from tasks to a meaningful context of a subject's interaction with the world" [20], this is currently only partially explicit in existing models.

First steps into the direction of making context explicit have been taken by Engeström [11], whose model promotes social aspects of context. This model, however, does not make the physical and technical environment explicit. While this may be acceptable when the focus is on the redesign of work processes on a rather high level of abstraction (where hard- and software can be treated as a unit subsumed under the tool concept of AT), it does not allow to make new challenges visible in the model that occur when novel hardware is used or integration with existing IT infrastructures is necessary.

Further on, while it was noted that activities shape the context for actions and operations [27], the way this actually happens has currently not been specified in an integrated model. This would however be necessary to integrate the findings of an AT-based analysis directly into commonly used design artifacts, e.g. wireframe models or conceptual sketches for new applications or interaction techniques.

In this paper, we propose a model that takes both the physical/technical and the human/social context explicitly into account and provides an integrated perspective on the different levels of analysis in AT. This model will allow designers and developers to identify potential conflicts between the individual elements and derive the respective requirements directly from the model.

The rest of this paper is structured as follows: First, we give a brief overview of AT. Then, we review related work, with a focus on context-of-use analysis and AT-based approaches. Subsequently, we present our approach, and illustrate how our model can be applied with two examples. We conclude with a discussion on design decisions and limitations of our approach, a summary of our contributions and a brief outlook on future work.

ACTIVITY THEORY

In this section, we briefly introduce the core concepts of activity theory (AT) and the basic model of tool-mediated interaction between subject and object, on which we build in the rest of the paper. A more comprehensive treatment can be found e.g. in the original works by Leontiev and Vygotsky [23, 36], as well as in newer introductory texts like [22] or [20]. Shorter introductions are also available in numerous publications on activity theory-informed designs, e.g. [12]. We also briefly touch the activity system model by Engeström, which introduced the idea of conflict analysis in the AT-model [11].

In AT the subject is seen as placed on a continuum between individual and collective, and is defined by possessing agency, which is understood as "the ability and the need to act" [20] – a need that focuses on objects, and is based on a *motive* [23]. An important feature of activity theory is, that the manipulations a subject carries out on an object are not seen as direct, but rather as mediated through *tools*, or more general *instruments*, and that thus "the object is seen and manipulated not 'as such' but within the limitations set by the instrument" [22]. Together with the aforementioned concepts, this is subsumed in Figure 1.

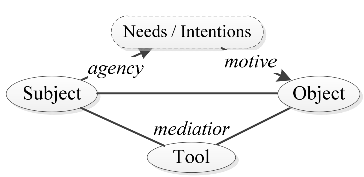

Figure 1: Standard model of activity theory (agency added).

The hierarchical structure of activity theory

Activities are not the only level of analysis foreseen by AT, but are rather seen as being composed of actions and operations. While activities are driven by motives, actions are conscious goal-directed processes executed to transform an object into the desired state. Given favorable circumstances, actions may become operations, which are intuitive interactions with the world that do not require much cognitive effort. The hierarchical structure of AT materializes some of these developmental aspects. Transitions between actions and operations are possible based on individual adaptation and contextual changes. Actions and operations for similar goals may exist in parallel and are selected based on the context and individual preferences. Notably, that implies that actions do not suddenly become operations when they are decomposed, but rather that they are equivalent solution procedures linked by processes of in- and externalization [20].

Engeström's activity system model

Among the various works based on AT, especially Engeström's model of AT has found wide adoption. In his work on learning activities [11], he extends the classical subject-tool-object triangle with additional elements derived from an integration of community/population perspective. He thus arrives at a combination of three "mediated" triangles: *rules* describe both implicit and explicit regulations of the interaction between a subject and the relevant community for an activity, "division of labor" describes how the community organizes work with and access to the relevant object (cf. elements in conflict analysis, Figure 2).

Among others, Engeström used this model for the re-design of work processes in hospitals, where he used it for a contradiction analysis between the respective elements (cf. Figure 2). Notably, the use of the model was not restricted to the analysts; it was a public artifact used to spawn discussion among the affected personnel of the hospital.

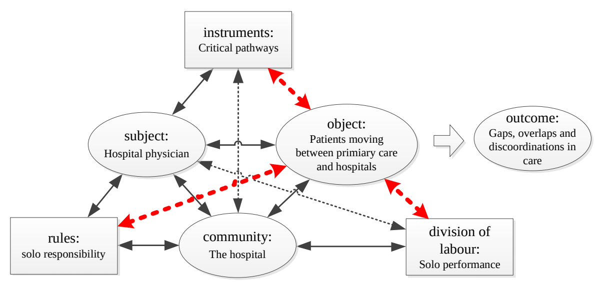

Figure 2: Example of Engeström's contradiction analysis (affected relations in bold red). Figure reproduced from [10].

Engeström's view of activity as object-oriented, collective, and culturally mediated [10] fits the requirements of analyses of organizations. There are, however, some subtle difficulties (discussed in more detail in section after the next) that prevent the immediate use of Engeström's model for other types of analysis, e.g. the dichotomy of subject and community, and the integration of collectives and the AT hierarchy.

RELATED WORK

While a large number of methods guide and support the design of interactive systems, for the review of related approaches we concentrate on two issues: first, existing work in the area of context-of-use analysis in general (orienting towards the human-centered design process as outlined by ISO 9241-210 [18]); and second, AT-based approaches that can help to understand the context-of-use.

Context-of-use analysis

The user-centered design process (ISO 9241-210) describes design cycles based on the transformation of 1) context-of-use details to 2) user and organizational requirements that are 3) transformed into system designs. The system's design is then 4) evaluated against the requirements, resulting in a new cycle or the successful finalization of the cycle.

The context-of-use is the foundation of the human-centered design process, with an important role in the requirement identification and the design evaluation context. The context-of-use consists of information about the user group, tasks, the technical environment, the physical environment and the organizational environment [24].

Existing methods focus on a structured collection of the required context-of-use information. Methods like discussions, interviews, surveys, diaries and field studies are used. Often context-of-use is captured by using artifacts like lists or tables. The methods collect single pieces of context information that are treated like atomic units without describing the relations that exist between them. Such a systemic relation is necessary. Context is not a static construct that consists of passive elements but a dynamic set of relations. As Schultze puts it, analysis should not only list what users do, but also needs to identify "what the doing does" [33].

The context-of-uses analysis for systems design in terms of activity theory describes the situatedness of an activity. One approach is given by [13], considering activities on a micro level (personal, independent activities), a meso (collective activities) and a macro (activities in large social contexts) level for past, present and future. For systems design, it is necessary to identify those context fragments that influence the system with respect to the introduction of new designs.

Approaches for context-of-use analysis based on AT

AT, with its understanding of context-embedded activities, provides a framework to describe systemic relations. Following this insight, different AT-based methods have been proposed that support the systems design process with respect to a context-of-use analysis.

Simple methods focus on the collection of data that conforms to AT. An example is the activity checklist to support ethnographic studies [19]. The list is used to support interview processes that are intended to identify relevant context elements.

Beyond the AT-conform collection of data, different methods use Engeström's model to create systemic relations. Neto et al. [28] combine the I* framework for organizational modeling and AT to address context-of-use analysis. Martins and Daltrini [25] use Engeström and decompose identified activities to actions and operations. Based on the creation of the systemic relations and the hierarchical decomposition of activities requirements are elicited. Korpela [21] uses concepts of activity theory to analyze information systems development as complex work practice based on activity networks. For Korpela the organizational perspective is in the focus: different involved stakeholders with different areas of expertise across organizational boundaries.

Other methods use Engeström's model and explicitly integrate the creation of conflicts in the designed model. Collins et al. [6] show the applicability of AT to capture data collected in interviews. They make a hierarchical use of Engeström's model to organize the findings of the interview and identify tensions and conflicts to derive design requirements. The Activity-Oriented Design Method (AODM) based on Engström's model [26] intends to support requirement elicitation based on a context-of-use analysis. The situation is modeled in Engeström's activity system and then decomposed to different sub-triangles that get analyzed.

Bardram describes activity analysis, a method of activity theory application that applies ethnographic field studies and an analysis of observations [3]. Activities within a setting are identified and, in a second step, patterns of these activities are analyzed.

Method assessment

The excerpt of design methods based on AT as well as a comparative study of AT-based design tools [29] show the trend to the application of Engeström's model. Especially the use of contradictions is an important mechanism that is used to identify requirements. E.g. AODM is a valuable extension of Engeström's model. The permutation of different sub-triangles of the activity system supports the identification of conflicts and contradictions.

Despite of the benefits of Engeström's model, there are different drawbacks. The roots of Engeström's model are in the analysis of organizational change [20]. Consequences are a focus on activity models with organizational mediators ("rules", "division of labor") that involve a community. Especially for the design of new systems other context factors are highly relevant. Examples are the physical and the technical environment.

Another limit of the reviewed systems is a lack of integration with respect to the hierarchical decomposition of activities. Only few methods make use of the hierarchical decomposition and if they use it, the decomposition is an additional method step that returns results that are not integrated in the activity system constructed with Engström's model.

Overall, we see a need for AT-based models for systems design that have a broad perspective on the context-of-use, explicitly integrating the technical and the physical environment. Such methods should reflect activities as hierarchical constructs and should, in particular, provide means to refine the context-of-use for an analysis of actions and operations during the process of decomposition.

A MODEL TO SUPPORT THE DESIGN OF INTERACTIVE SYSTEMS

We perceive the design of interactive systems as an exploration and analysis of the design space of possible solutions for a problem in a given context. Each design for a new solution introduced into this context, be it a tool or a new work process, can (and usually will) have consequences for other entities in the context, with effects differing, depending on the level of detail on which analysis and design of the new solution happen. This entails interactions between the newly introduced solution and other entities in the domain as well as interactions between different levels of analysis.

Support for a structured and integrated analysis of the effects of design decisions is currently not well supported by the reviewed AT-based systems design tools. Together with the initially outlined need to design for novel device contexts (e.g. smartphones, tablet computers, large interactive displays), the model we describe in this section has three major design goals:

- the generation of requirements for the design of interactive systems directly from the model,
- the explicit inclusion of the physical/technical and the human/social context into the model, and an
- integrated perspective on all levels of analysis in AT.

Beyond the general ideas of AT – in particular that a subject's actions can only be fully understood in context – our work is mainly inspired by the works of Engeström, especially his contradiction-based approach to the analysis of work with AT [10], and the work of Mwanza [26], who employed a systematic iteration through the subgraphs of Engeström's model to identify research questions.

Model overview

Each model instance is a fully connected hexagon (see Figure 3), in which we distinguish between mediating (rectangles) and non-mediating entities (ellipses). This distinction is largely identical to that of Engeström's model (we discuss the ways in which we deviate from his model separately in the next sub-section). A separate element is foreseen to represent the (intended) outcome of an activity (or action or operation respectively). Summing up, the seven elements in our model are ("(m)" = mediating entity):

- **Subject:** Individual or collective entities from the domain of interest that have agency
- **Object(s):** the material component of the entities towards which a subject directs its activities (, actions, or operations)
- **Outcome:** the state of entities or the information, which a subject seeks to produce through its activities (, actions, or operations)
- **Activity (, action, or operation) context**: material and social aspects of entities in the background that frame an activity (, action, or operation)
- **Rules (m)**: technical or social constraints imposed on the subject when pursuing an activity (, action, or operation)
- **Instruments (m)**: artifacts through which the subject perceives and/or modifies an object; reverse effects may include learning triggered by the application of an instrument to an object
- **Workflows (m)**: temporal patterns governing interaction between a subject and instruments, objects or entities from the activity's (, action's or operation's) context

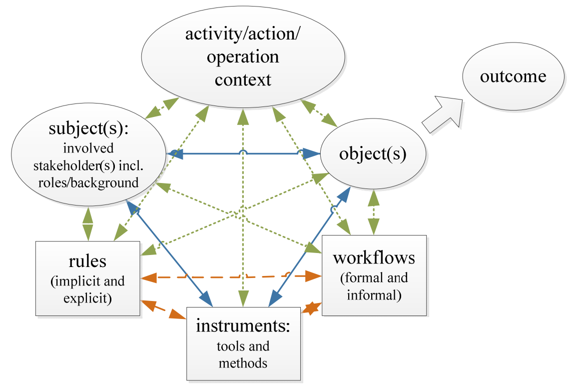

Figure 3: Our model is comprised of seven elements; the main difference to Engeström's model is the introduction of an explicit element representing the context of the activity/ action/ operation; relations have been derived from the original activity theory model (solid blue lines); Engeström's model (dotted green lines); or newly added to reflect changes to the model elements (dashed orange lines).

Differences to Engeström's activity system model

The elements in our model differ in three respects from those in that of Engeström's model (and those approaches that directly base on his model):

- We re-phrased *division of labor* as *workflows* to emphasize the inclusion of temporal aspects (e.g. access frequency to a system), even when the subject is actually only a single person.
- We introduced a new element called *activity context* (or action or operation context respectively), which describes the situation and the environment a user is in and acts in, i.e., for a start, the (physical or virtual) acts of other users in the immediate environment, but also material aspects of that environment (e.g. the layout of a meeting room or the type of device an application is installed on). With a view to the original model of activity theory, the *context*-element can be seen as an explicit representation of the idea that "the activity itself is the context" [27].
- We split up the "community" element in Engeström's model into those members actively participating in the analyzed subject-object-interaction and those involved in other interactions. The former have been merged into the subject-concept, which now explicitly incorporates collective subjects. This allows us to express situations in which a group of subjects shares a common goal, e.g.

software developers assembling in front of a whiteboard. Implicitly collective subjects are already present in existing works on activity theory, e.g. in [20] and [26].

The last aspect has further consequences for the semantics of the relations between elements in our model: first, in Engeström's model the "rules" element mediates the interaction between community and subject, yet our model sees it as a mediator between the subject and the activity context, essentially broadening its scope. We also perceive rules to be a mediator for interaction between subject and object, e.g. via user access rights; second, the "workflows" element can now describe both process coordination in the given context, but also the temporal patterns in which the subject accesses or manipulates the object of the activity.

While we also made the relations between rules, workflows and instruments explicit in our model, this does actually not mark a change to Engeström's model as "an activity is actually a systemic whole in the sense that all elements have a relationship to other elements, but all those connections have not been presented in the picture because of sake of clarity" [22].

Empirical grounding and conflict analysis

To use the described model for a concrete problem in a specific domain, it is necessary to specify the respective elements for each model instance. Similar to many AT researchers, we deem first-hand empirical research to be the optimal source for this specification, we also consider on literature reports of empirical research, (predictive) theories and practical experience/design knowledge as valid grounding. In addition, creative solutions may emerge from each of these methods of grounding.

Based on the element connections in the model, an analysis of the relations and interactions is possible, and conflicts and contradictions can be systematically identified. This analysis is not restricted to an "as-is" analysis of the status. One or more nodes can be filled with concepts for new solutions, e.g. tool designs, allowing a systematic analysis of the possible consequences of the change. Consequently a set of related models may be generated on each level of the activity, action and operation hierarchy that observe the presumed effects of design decisions or changes to other elements of the model. As such it is a powerful tool for the analysis of the effects of design decisions as well, especially based on an initial "as-is" analysis of the environment.

Multiple designs & design perspectives

Within the same domain, a set of model instances may exist that are horizontally or vertically connected. Horizontal connections stand for design space explorations on the same level of detail. This may be necessary because there are a number of designs (or design variants) to be evaluated. In a similar way, the system designer may decide to analyze with different design perspectives.

We foresee three of these perspectives: an instrument perspective (designing new tools or sign), a workflow perspective (proposing new ways of organizing actions) and a rule perspective (formulate new rules that govern interaction).

Subsequently each design can be explored in more detail by following the vertical connections. Vertical connections are the result of decomposing activities. Based on the possibility of our model to integrate the analysis on different levels of detail (see below), this decomposition can be repeated multiple times until the desired level of detail is reached.

Support for design on different levels of detail

Designing interactive systems requires specifications on multiple levels of detail, e.g. the design of landing pages, the layout of forms for data entry and the type of menu for a webpage. AT provides the perspective of activities and related actions or operations (e.g. for a web application on traditional desktop computers activities might be mapped to overview or landing pages, actions and operations may exist on the level of forms, menus and buttons).

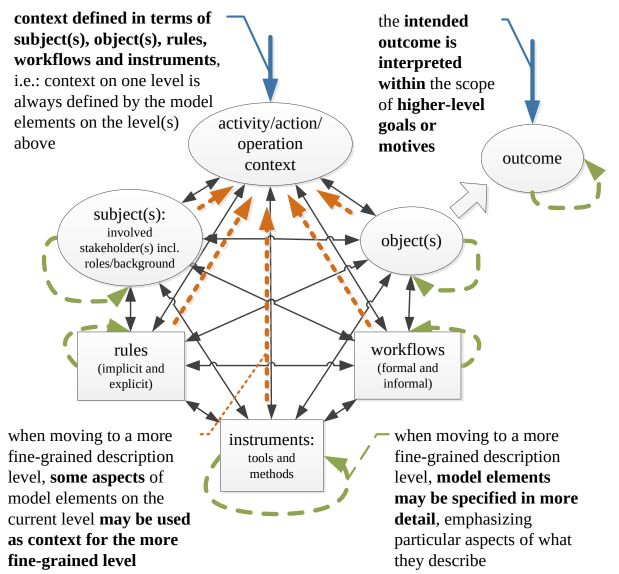

Figure 4: Integration of the different levels of analysis in AT is done by following a three steps approach when proceeding to a more fine-grained level of analysis: element refinement (green dashed arrows), context enrichment (orange dotted arrows) and context integration (blue solid arrows).

To support a holistic approach to the design of interactive systems, our model integrates the different levels of analysis via the newly introduced "context"-element. In particular, we propose a three-step algorithm when proceeding to a more fine-grained level of analysis (subsumed in Figure 4):

1. **Element refinement:** the individual elements are specified in more detail to reflect the appropriate

granularity for the analysis on the next level of analysis.

2. **Context enrichment**: the aspects/parts of the individual elements not directly involved in the interaction on the more fine-grained level are treated as background information (and background actions). They are thus carried over to the more fine-grained level for integration into the context element (this is particularly important to accommodate for the dual nature of computers and software, i.e. they may work as tools on a higher-level of abstraction, but also as a technical context/platform on a more fine-grained level).

3. **Context integration:** Context information carried over from the previous level of analysis is used to define the situation/background for the analysis on the current level. Also the objective/intended outcome is carried over to provide interpretation guidance to the intended outcome on the current level.

It should be noted that these steps will usually be applied multiple times for a transition between two levels, as higher-level interactions are commonly composed of multiple lower-level interactions (e.g. activities are composed of a larger number of actions or operations).

Using the model to identify requirements

To derive requirements directly from our model, we employ a method that builds upon the idea of using the AT model to detect conflicts between individual elements by Engeström [10] and that of systematic iteration through subgraphs of the model as proposed by Mwanza [26].

For a systematic identification of requirements for a new component that is part of a newly created (or re-designed) interactive system, empirical grounding and the decomposition of activities should be performed as described above. When the desired level of detail is reached, only two steps are necessary to generate requirements:

1. **Analyze possible permutations for conflicts**: treat four elements as fixed: subject, context, object, and a fourth depending on the initial choice of the design perspective (i.e. instrument, workflow or rule). Now iterate through the possible subgraphs (the graph itself is not considered) of the respective model and analyze for conflicts (as computer support would require extensive domain modeling, we assume this step to rely on the designers' assessment).

2. **Generate requirements from involved elements**: if a conflict occurred in the previous step, combine the elements to generate a requirement for a newly designed solution (e.g. a new software tool or functionality).

EXEMPLARY APPLICATIONS OF OUR MODEL

To illustrate how our model can be applied for the design of interactive systems in various contexts, we present two examples: composition of IT services and collaborative graphical modeling at a large touchscreen. While we did include some comments regarding the empirical grounding, the main purpose of this section is to provide more detail on the practical aspects of our contribution.

Composition of IT Services

Service-Based applications are a constituent part of many modern IT systems. One of the core principles of service-oriented architecture (SOA) is the composition of existing services to yield new functionality. Reuse as such is a well-established concept in software engineering and has been applied for many years, however merely for reasons of convenience and to increase development efficiency. In SOA, reuse via service composition is essential, especially in the context of cross-enterprise services [35].

Scenario

In order to facilitate efficient composition, services have to expose certain properties [14]. During service development, quality measures have to be taken in order to ensure the fulfillment of these properties. Of course, composition efficiency is only one out of many quality goals that are important when designing and implementing IT services. Most of the currently available quality models for software (e.g. the SQuaRE model [17]) share the approach of structuring quality along characteristics like maintainability, reliability or performance. However, this classification approach often lacks clear semantics regarding the relationship between super and sub-attributes. To overcome this, activity-based quality models were introduced [8]. These models describe software quality along activities that are conducted on or with a software system, leading to clear and intuitive decomposition semantics, which corresponds to the hierarchical nature of our model.

Design Perspective

From a systems design point of view, the investigation of possible problems can be used to improve tools involved in the customization process. We therefore take the instrument perspective when applying our model.

Empirical Grounding

The data that is necessary to fill the model elements is obtained in two ways. First, a literature review on SOA characteristics and their impact on product quality [35] provides the basic framework for the model. Second, industry experts are involved in applying the model, using their practical experience in order to provide information on specific parts, in particular the rules and workflow elements.

Model Application

In order to identify problems and derive quality requirements, first the activity to be analyzed has to be described in terms of our model. Second, several combinations of model elements are selected and considered conjointly. Subject, object and context are always selected; combinations of rules, tools and workflows are added optionally.

Figure 5 shows a problem identified using our approach. When creating a compound service, typical rules for the service integrator could be legal (there have to be license agreements in order to use the desired service), financial (the service must be available at acceptable charge in order to create a profitable compound service) or technical ones (the service interface has to match the available parameters and data formats). Using our model, we were able to deduce that the latter rule can be violated by services whose operations are not defined at a reasonable granularity. This observation further leads to the requirement that services should have adequately granular operations.

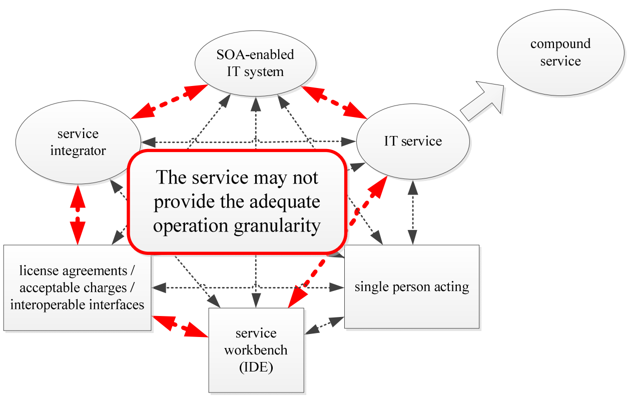

Figure 5: Example of a problem in the composition of IT services detected with our model

In a similar manner, a number of additional potential problems can be identified: A Service whose license is unclear, particularly restricted or incompatible with other licenses is hard to use for composition. Furthermore, the pricing model could be unfavorable for compositional use.

Generating Software Quality Models

Likewise, this procedure can be repeated for other common activities occurring during development, deployment, usage or maintenance of the SOA-based IT system, covering its whole life cycle. The resulting set of potential problems and derived quality requirements can be used to develop or enhance a corresponding software quality model [14]. Hence, our approach provides a theoretically well-founded and structured way to create activity-based quality models.

Collaborative graphical modeling

Scenario

Conceptual models, often in the form of diagrams, have many applications in modern enterprises. Among the most common types are arguably UML or UML-like diagrams used by software engineers and workflow or business processes diagrams used by system analysts. Building the respective models however is a complex task that requires both a thorough understanding of the domain as well as knowledge and practice in process design and visualization through formal modeling. This knowledge is commonly socially distributed, i.e. the input of multiple stakeholders with different backgrounds is required [31].

Current tools, however, fail to facilitate stakeholder involvement in the actual modeling process – as Rittgen recently noted, “the majority of the currently existing modeling tools are single-user tools. Strangely, this is even the fact for the ones that explicitly address group modeling” [31]. Moreover, even more advanced tools (e.g. COMA [30]) still rely on standard desktop computers – despite their limitations in displaying large data sets (cf. e.g. [1]) and their limited effectiveness for collaborative work; a few notable use electronic whiteboards (e.g. [5]) but focus rather on producing documentation quality diagrams than actual support of group interaction.

Recently, however, researchers have proposed the use of wall-size touchscreen displays for collaborative modeling [9]. Using these devices for collaborative model building will, however, require new tools that need to be tailored to the affordances and challenges of the devices.

Design Perspective

Using the example of interaction design for these new tools, the following section will illustrate the application of the analysis model described in this paper. Consequently the design perspective we adopt for the analysis is the instrument perspective. To illustrate, how analysis on more fine-grained levels of detail happens, the analysis will go through multiple steps of decomposition until a level suitable for support through interaction techniques for very large touchscreens is reached.

Empirical Grounding

For the empirical grounding of the analysis, we will largely orient towards existing literature on collaborative modeling, in particular the works of Rittgen [30], Rouwette [32] and those of Herrmann [15]. We can, however, also confirm the modeling actions found in their works from informal observation in a number of internal modeling sessions.

Model Application

The starting point, i.e. the activity under analysis, is the actual construction of a group model (cf. Figure 6a). While different motives for collaborative model building are possible, we assume the driving force is the need to understand the current situation of a company or government organization and the possible implications of strategies for the future (cf. e.g. [32]). This translates into the objective of building a shared mental model of the involved stakeholders, made explicit via a graphical representation.

Because more complex processes will likely require the input of multiple stakeholders, we assume that subgroups will be formed to work on specific aspects of the model (cf. e.g. [32]) and that these groups of users will interact in parallel. Furthermore, we assume that users will show respectful behavior towards their peers.

Pursuing this activity requires a number of actions to be executed, among them the use of divergent thinking techniques, e.g. brainstorming, to create an initial collection of model elements (cf. e.g. [15]), the discussion, or negation, of modification proposals for the model (cf. e.g. [31]) and, of course, actually performing modifications on a local part of the model. Each of these actions may happen multiple times and in parallel, when there are multiple sub-groups in the modeling session.

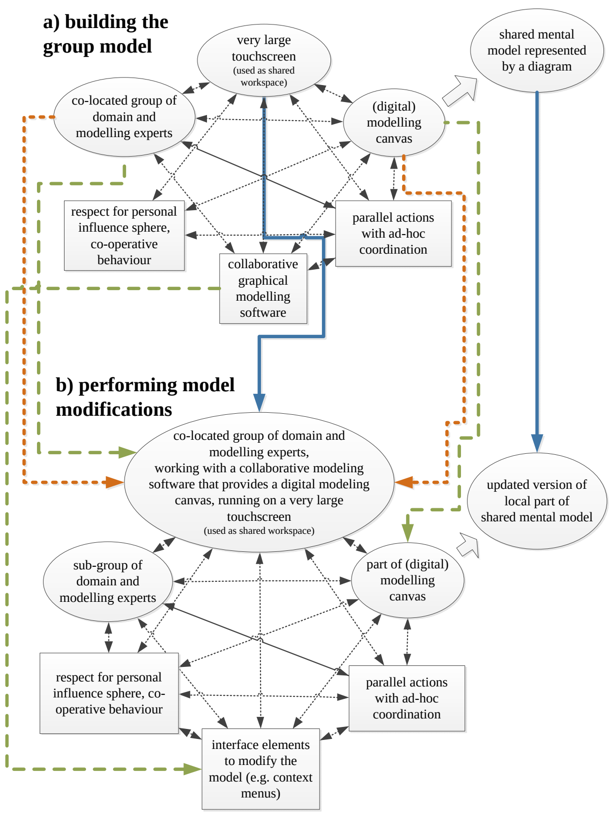

Figure 6: One problem detected by the analysis of the "access to nodes" operation as part of a "link nodes" action in order to update a local model in a collaborative modeling session.

For each of the identified actions, the three step procedure described in the model section has to be applied. Using the example of the "performing model modifications"-action, this means (cf. Figure 6b):

1. **Element refinement:** the "co-located group of domain and modeling experts" (subject element) is refined to the respective "sub-group of domain and modeling experts", the "collaborative graphical modeling software" (instrument element) to the "interface elements to modify the model (e.g. context menus)", and the "(digital) modeling canvas" (object element) to the respective "part of (digital) modeling canvas"

2. **Context integration:** the "very large touchscreen (used as shared workspace)" (context element) is carried over to the next level, the same applies to the "shared mental model represented by a diagram" (outcome element), which is however, specified in more detail, i.e. as an "updated version of local part of shared mental model"

3. **Context enrichment**: on the more fine-grained level of analysis, aspects of the subject element ("co-located group of domain and modeling experts"), the object element ("(digital) modeling canvas") and the instrument element ("collaborative graphical modeling software") become background to the action under analysis. Thus, the new context element integrates these aspects and now contains the "co-located group of domain and modeling experts, working with collaborative modeling software that provides a digital modeling canvas, running on a very large touchscreen (used as shared workspace)".

This procedure has to be repeated for the next two levels of detail to decompose the respective actions on the current level into more fine-grained ones; in the case of performing modifications on the model, these include creating, labeling, deleting or linking nodes. Assuming that linking happens by drawing an arrow from a source to a target node, this requires that users have access to both the source and the target node.

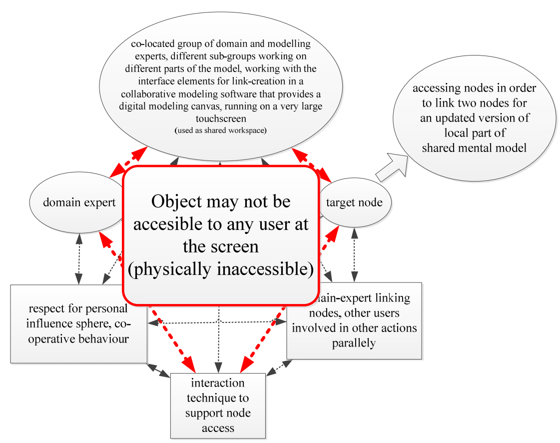

Figure 7: Conflict analysis for "accessing nodes" (example)

At this point, the conflict analysis outlined in the previous section can be used to identify potential conflicts that arise when users perform the linking action. The iteration through all subgraphs that contain subject, object, context and instrument (instrument perspective) reveals three potential conflicts (an example graph used for conflict detection is shown in Figure 7):

1. the display is very large and touch interaction is used (as indicated by the context element), thus nodes may simply be placed so distant on the screen hat they required physical movement to reach them (c.f. e.g. [7]).

2. multiple users occupy different areas on the screen (context element), thus conflicts with social protocols (rules element) may arise when one of the nodes of interest is located in the screen area right in front of another user (c.f. e.g. [34])
3. multiple users on the screen are interacting parallel and with different objectives (context element + workflow element), thus physical interference can occur when multiple users try to access nodes located near to each other on the touchscreen (cf. e.g. [16]),

To ensure efficient collaborative modeling, these problems need to be addressed with suitable interaction techniques.

DISCUSSION

The model is result of an iterative prototyping process that involved frequent tests of the applicability of the model to different types of system designs.

Design decisions

Although, our model is based on Engeström's activity system model, a number of significant changes have been made – among them the use of a collective subject. As a result, the community element from Engeström's model is split into subject and context, and rules are seen as mediators between subject and workspace rather than subject and community. This does however enable modeling of activities of groups consisting of multiple, equal actors, involved in ad hoc collaboration. Among others, this reflects the integration of semi-autonomous groups that are difficult to express in Engeström's organization-oriented model.

Open issues

AT with its perspective on human activity execution tackles a complex domain that is not accessible by formal methods. Terms like context, subject and tool are inherently fuzzy, not providing a precise one-for-all definition but require an understanding of the meaning for a domain. The proposed model demands the explicit identification of entities that realize the fuzzy terms.

As a result, entities for context, subject and instruments are required. Although the identification of the entities is a complex problem if a definition for the generic entity classes is required, the actual identification is simpler: an analysis of an explicit activity system generates entities that can be easily distributed among the different classes based on the systemic relations between the entities.

Another problem of AT is the representation of distributed systems that mediate between different subjects. Such activity systems require an extension of the model, e.g. the creation of two model instances that are interconnected.

CONCLUSION & OUTLOOK

In this paper, we have presented a model based on activity theory (AT), which supports the design of interactive systems. Contrary to existing AT-informed approaches, this model incorporates both the social, physical and technical context of activities, allowing system designers to handle both the technical possibilities and limitations of novel devices and new challenges that arise with ad-hoc collaborative interaction. Furthermore, the model integrates the different levels of activity theory analysis to allow design on all desired levels of detail.

Using a conflict-detection approach similar to the work of Engeström [11] and systematic iteration through possible subgraphs of the model (cf. [26]), this model allows designers to derive requirements directly from the model.

Preliminary results from interviews we conducted with requirements engineers and user interface designers show positive feedback on the context integration via our model. We also identified a need for integration with existing tools, e.g. personas or scenario descriptions. For future work, we will thus concentrate on a better understanding of these requirements for the use of our model by practitioners.

REFERENCES

1. Andersen, D.F. and Richardson, G.P. Scripts for group model building. *System Dynamics Review*, *13*. 107-129.
2. Bannon, L.J. and Bodker, S. Beyond the Interface: Encountering Artifacts in Use. in Carroll, J.M. ed. *Designing Interaction: Psychology at the Human-Computer Interface*, Cambridge University Press, 1991, 227-253.
3. Bardram, J.E. and Doryab, A., Activity analysis: applying activity theory to analyze complex work in hospitals. in *CSCW '11: Proceedings of the ACM conference on Computer supported cooperative work*, (2011), ACM, 455-464.
4. Card, S.K., Moran, T.P. and Newell, A. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, 1983.
5. Chen, Q., Grundy, J. and Hosking, J. SUMLOW: early design-stage sketching of UML diagrams on an E-whiteboard. *Software: Practice and Experience*, *38*. 961-994.
6. Collins, P., Shukla, S. and Redmiles, D. Activity theory and system design: A view from the trenches. *Computer Supported Cooperative Work (CSCW)*, *11*. 55-80.
7. Czerwinski, M., Robertson, G., Meyers, B., Smith, G., Robbins, D. and Tan, D., Large display research overview. in *CHI '06: Proceedings of the SIGCHI conference on Human factors in computing systems*, (2006), 69.
8. Deissenboeck, F., Wagner, S., Pizka, M., Teuchert, S. and Girard, J.-F., An Activity-Based Quality Model for Maintainability. in *ICSM '07: Proceeding of the IEEE International Conference on Software Maintenance*, (2007), 184-193.
9. Döweling, S. and Nolte, A., Closer to the Model – Collaborative Modeling with Wall-Size Interactive

Displays. in *MENSCH &COMPUTER '11: Workshop on "Anforderungen und Lösungen für die Nutzung interaktiver Displays im Kontext kollaborativer Arbeit"*, (2011).

10. Engeström, Y. Activity theory as a framework for analyzing and redesigning work. *Ergonomics*, *43*. 960-974.
11. Engeström, Y. *Learning by Expanding: An Activity Theoretical Approach to Developmental Research*. Orienta-Konsultit Oy, 1987.
12. Fjeld, M., Lauche, K., Bichsel, M., Voorhorst, F., Krueger, H. and Rauterberg, M. Physical and virtual tools: Activity theory applied to the design of groupware. *Computer Supported Cooperative Work (CSCW)*, *11*. 153-180.
13. Gay, G. and Hembrooke, H. *Activity-centered design: An ecological approach to designing smart tools and usable systems*. MIT Press, 2004.
14. Goeb, A. and Lochmann, K., A Software Quality Model for SOA. in *WoSQ '11: International Workshop on Software Quality*, (2011).
15. Herrmann, T. and Nolte, A., The Integration of Collaborative Process Modeling and Electronic Brainstorming in Co-located Meetings. in *CRIWG '10: Proceedings of the Conference on Collaboration and Technology*, (2010), Springer, 145-160.
16. Hornecker, E., Marshall, P., Dalton, N.S. and Rogers, Y., Collaboration and interference: awareness with mice or touch input. in *CSCW '08: Proceedings of the ACM conference on Computer supported cooperative work*, (2008), ACM, 167-176.
17. ISO, J.S. ISO/IEC 25010:2011, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, International Organization for Standardization, 2011.
18. ISO, T.S. ISO 9241-210, Ergonomics of System Interaction - Part 210: Human-centered design for interactive systems, International Organization for Standardization, 2010.
19. Kaptelinin, V. The Activity Checklist : A Tool for Representing the "Space" of Context. *Interactions*, *6*. 27-39.
20. Kaptelinin, V. and Nardi, B.A. *Acting with technology - Activity Theory and Interaction Design*. MIT Press, 2006.
21. Korpela, M., Mursu, A. and Soriyan, H.A. Information Systems Development as an Activity. *Computer Supported Cooperative Work (CSCW)*, *11*. 111-128.
22. Kuutti, K. Activity theory as a potential framework for human-computer interaction research. in *Context and consciousness*, 1996, 17-44.
23. Leontev, A.N. *Activity, Consciousness, and Personality*. Prentice-Hall, 1978.
24. Maguire, M. Methods to support human-centred design. *International Journal of Human-Computer Studies*, *55*. 587-634.
25. Martins, L.E.G. and Daltrini, B.M., Activity Theory: a Framework to Software Requirements Elicitation. in *WER '99: Workshop em Engenharia de Requisitos*, (1999), 30-44.
26. Mwanza, D., Where theory meets practice: A case for an Activity Theory based methodology to guide computer system design. in *INTERACT '01: Proceedings of the IFIP TC13 Conference on Human-Computer Interaction*, (2001).
27. Nardi, B.A. Studying context: A comparison of activity theory, situated action models, and distributed cognition. in Nardi, B.A. ed. *Context and consciousness*, MIT Press, 1996, 69-102.
28. Neto, G.C., Gomes, A.S., Castro, J. and Sampaio, S., Integrating activity theory and organizational modeling for context of use analysis. in *CLIHC '05: Proceedings of the Latin American conference on Human-computer interaction*, (2005), ACM, 301-306.
29. Quek, A. and Shah, H., A comparative survey of activity-based methods for information systems development. in *ICEIS '04: Proceedings of the International Conference on Enterprise Information Systems*, (2004), 221-232.
30. Rittgen, P., COMA: A tool for collaborative modeling. in *CEUR Workshop Proceedings No 344*, (2008), CEUR-WS.org, 61-64.
31. Rittgen, P. IT support in collaborative modelling of business processes - a comparative experiment. *International Journal of Organisational Design and Engineering*, *1*. 98-108.
32. Rouwette, E.A.J.A. and Vennix, J.A.M. Group model building: A decision room approach. *Simulation & Gaming*, *31*. 359-379.
33. Schultze, U. and Borland, R.J.J. Knowledge management technology and the reproduction of knowledge work practices. *The Journal of Strategic Information Systems*, *9*. 193-212.
34. Scott, S.D., Carpendale, S. and Inkpen, K.M., Territoriality in collaborative tabletop workspaces. in *CSCW '04: Proceedings of the ACM conference on Computer supported cooperative work*, (2004), ACM, 294-303.
35. Voelz, D. and Goeb, A., What is Different in Quality Management for SOA? in *EDOC '10: Proceedings of the International IEEE EDOC Conference*, (2010).
36. Vygotsky, L.S. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1978.