Tangible Context Modelling for Rapid Adaptive System Testing

Frank Honold, Felix Schüssel, Michael Munding, Michael Weber
Ulm University
Institute of Media Informatics
89069 Ulm, Germany
{frank.honold, felix.schuessel, michael.munding, michael.weber}@uni-ulm.de

***Abstract*—With the rise of adaptive interactive systems came a growing demand for tools to test the envisioned adaptivity. This article presents a tactile interface allowing direct manipulation of a context model for adaptive behavior in intelligent environments. The presented tool starts with an abstract context model, which is transformed to an easy-to-use interface supporting tangible interaction. Different context parameters can directly be manipulated. Diverse context-relevant items can be integrated using tangible objects to emulate and use their real world location parameters. The article gives an insight in the technical implementation and the usage as substitute for a sensor topology from an existing intelligent environment to support evaluations and tests.**

I. Introduction

In 1991 Mark Weiser shaped the term of *Ubiquitous Computing* with his successful article "The Computer for the 21st Century" [1]. He named ubiquitous computing a computer technology, which can be used by any person without any further notice or effort. Based on this, the coming age of computer technology was announced as "the [...] age of calm technology" [2]. The development of adaptive systems is a further step in this direction. If a system adapts to a user due to gathered information from the context of use, the user, in turn, must adjust himself to the system much less.

Adaptive components of intelligent environments do not only adapt to the user, but also to the environment. The information about the user and environment are part of the context model, in which all of the context of an application can be described. This knowledge is the key to adaptivity.

A context model describes a situation in the real world. The context information contained therein can be used to allow applications to adapt to the situation. They can therefore change their behavior without direct (active) user intervention and thus allow an implicit form of interaction. In this case implicit means that a user does not have to trigger an action by himself, but that the action is triggered by contextual analysis and the identification of a certain circumstance. In return, the adaptive behavior can even change the context by itself (e. g. switching the lights on causes the effect that it is not dark anymore).

Situations are very dynamic and in permanent change. Users enter and leave rooms, they express different emotions, perform a variety of tasks, and are motivated by different and changing intentions and goals. Devices may have different user interfaces, their positions may change, and their availability for interaction may also be quite unpredictable.

A. Adaptivity in Intelligent Environments

A system or application is always executed in a well defined situation. If the application is aware of the situation, the individual user can benefit from a system's context adaption as described by Shackel [3]. For example, systems can adapt their processing behavior, the individual dialogue strategy or the user interface to support individual users in their tasks. Any useful information items that characterize a certain situation as a certain context of use are collectively referred to as context.

One way to support a user by context information is the identification and application of implicit actions. In this case implicitly means that the user has not to trigger a certain action obviously, but the system triggers a suitable action by identifying the user's needs based on certain context knowledge items itself. As an example, a smartphone would therefore trigger an implicit action and enable the silent mode, if the context information can be interpreted in a way that the user is in a library (pure location based) or in a meeting situation (location: meeting room plus other mobile phones in the closer environment), or the light switches on when entering a dark room. Another way of adaption focuses on the adaption of the communicable content and the dialogue strategy. Depending on the user's knowledge a more basic or high level vocabulary can be used via interaction with the user, or additional information can be offered on the fly if a user seemed to be irritated.

Particularly in mobile and ubiquitous scenarios contextual information is highly dynamic. The users' locations and thus the reachable objects in their surroundings often vary. To respect these different situations adaptive systems and services need to access this contextual information at runtime. Even more important: to develop and test an adaptive behavior system developers must be able to emulate each possible context of use.

B. Problems when Evaluating Adaptive Behavior

Context information usually is based on sensory data, which might be affected with uncertainty. In general, these data streams can be recorded and re-played for evaluation. If possible, the sensors' data gets interpreted and recorded in a transcribed form on a symbolic level as well. To test variants of the recorded context of use (or even transition between them) the recordings have to be manipulated in advance to test an intended adaption hypothesis. This method for testing and evaluating an envisioned adaption behavior can be time-consuming and costly.

C. Contribution

Facing this problem, our article presents a flexible and extensible approach to realize a tangible and easy-to-use context editor. We give a theoretical introduction to context knowledge and name essential requirements for supporting rapid testing adaptivity in intelligent environments. Based on that, we describe our approach and present the realized system of a flexible model driven multi-user context editor. The presented editor supports tangible interaction and direct manipulation to manipulate the linked context model in a quick and easy way.

II. What is Context?

Since there are different domains of context-adaptive intelligent environments there are various definitions of the term context. Day and Abowd investigated various definition approaches to form their own definition of context [4]. They go along with the idea from Schilit et al. [5] to address the main aspects of context like "where you are, who you are with, and what resources are nearby", as well as to summarize these items as "the constantly changing execution environment". By "environment" they subsume "computing environment, user environment and physical environment".

In the remainder of this article we refer to context as defined by Dey and Abowd [4]:

> Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

One interesting statement in this definition is the fact, that entities can temporarily be associated as context, depending on the current interaction. So the context model itself may vary regarding the number of contextual knowledge items.

III. Rapid Testing Based on Tangible Interaction

As stated in section I-A, it is a costly endeavor to provide and edit a complex context model in order to test a system's adaptive behavior. Data bases or text files as manipulation interfaces are hard to maintain, and – concerning the necessary interaction concepts – often lack a good usability.

A. Tangible Interaction

Tangible interaction can be the key to a better usability. Durell Bishop's Marble-Answering-Machine from 1992 is often named to mark the beginning of tangible interaction. Ishii and Ullmer came up with the idea of "Tangible Bits" to bridge the gap between cyberspace and the physical environment [6]. Their idea of interactive surfaces and the use of graspable objects for interaction led to new and intuitive forms of interaction and direct manipulation. One of the early representatives of tangible interaction is Underkoffler's and Ishii's tangible workbench for urban planning (Urp) [7]. Current systems like the Reactable hardware[1] allow the use of fiducials as interactive objects. One of the best known commercial products for surface interaction is the Microsoft PixelSense Table (formerly known as Microsoft Surface). This product is able to detect visual markers. Objects can be equipped with so called Byte Tags[2] to realize interactive tangible objects on the screen. The Byte Tags can be placed on the screen and any linked event like the occurrence, moving, rotation, or disappearance of each tag can easily be processed by event handling.

B. Requirements for Rapid Testing Support

Stemming from an expert consultation we conducted amongst research colleagues, different requirements for a context manipulation tool for rapid testing of adaptive components were identified. First of all, such a tool must support the *separation of context model and application*. This means that it shall be able to run independently from the application it is intended for. It must be *easy to add or remove items* of the context model (e. g. persons, devices). It should be possible to *model and manipulate uncertain data*. The underlying *context model should be extensible and easy to change* (e. g. specific data types and attributes). Furthermore, the tool should have a *high usability* and support *easy supervision of the current state of the model*. Keeping these requirements in mind, we designed and implemented our tool as described in the following section.

IV. Realization

The realization is implemented on a Microsoft Surface Tabletop (first generation of the PixelSense table). This hardware is able to detect the visual markers, called Byte Tags.

To meet the aforementioned requirements a model driven approach is applied to realize the interactive fragments of the editor. As an example we want to model some persons, different devices and the environment for a fictive scenario. Since each context attribute of interest shall be configurable, and the context model shall be extensible and flexible, we decided to realize a model driven approach. Our editor's basic concept is to provide a UI where different users can be emulated in a fictive surroundings. The users and environment's attributes are defined using an XML configuration file (cf. listing 1) based on well-defined XML-Schema description. The model's knowledge attributes are inspired by the later application domain as well as by studies (e. g. by the findings described in [8]).

The editor interprets the XML file and realizes an editable UI. An exemplary scene is shown in figure 1. A test person's context menu is shown and ready to be edited. The tactile context items on the screen can be re-arranged to form new scenarios or settings. Due to the hardware's ability to detect Byte Tags each tactile item's context menu is always rendered besides the item, even if the item is moved on the screen. Apart from this the implementation supports concurrent multi-user interaction.

Based on the work of Strang and Linnhoff-Popien [9] we decided to use an XML representation as context knowledge interchange format with other components, as we believe that

[1] Reactable – http://www.reactable.com/ [online: 2013-02-15]

[2] Microsoft Byte Tags – http://msdn.microsoft.com/en-us/library/ee804885%28v=surface.10%29.aspx [online: 2013-02-15]

```
<?xml version="1.0" encoding="UTF-8"?>
<ui xmlns="http://sfb-trr-62.de/b3/uiConfig.xsd">
  <person>
    <!-- ... more items here -->
    <container contextParameter="name" label="name">
      <uiElement type="string_var" value="TestPerson"
                 probability="1"/>
    </container>
    <container contextParameter="gender" label="gender">
      <uiElement type="string_fix" value="male"
                 probability="0.1"/>
      <uiElement type="string_fix value="female"
                 probability="0.9"/>
    </container>
    <container contextParameter="age" label="age">
      <uiElement type="integer" value="35" probability="1"
                 resolution="5" min_value="0" max_value="120"/>
    </container>
    <!-- ... more items here -->
  </person>
  <!-- ... more items here -->
</ui>
```

Listing 1. Excerpt from an exemplary configuration file. Changes to this file automatically result in a changed final UI for each person item, which is placed on the surface (see figure 1).

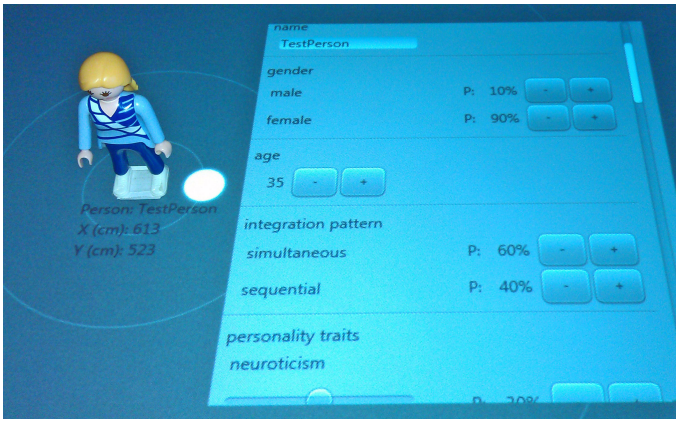

Fig. 1. The automatic UI generation process realizes an editable widget based on the given model description from listing 1. The different containers are separated by horizontal rows. The user model can easily be manipulated by touch interactions from multiple users. The location can be changed by re-arranging the figure like in real world.

this is the best trade-off between expressiveness, universal readability and deployment.

Our toolkit is designed to link re-usable tangible objects (e. g. toy figures) with additional context data. To support the possible occurrence of sensor- and inference-based ambiguous context data, it is possible to explicitly model ambiguity. As an example compare the *gender* knowledge item as defined in listing 1 and its resulting UI representation in figure 1. In this case the *gender* container represents a probability distribution for the two attributes *male* and *female*. Increasing the probability at runtime of one certain attribute will decrease the other value's probability within the same container. The overall probability will always be 1 (or 100 % as displayed in the UI).

The tool supports the use of five different data types to model diverse context data. To model a probability distribution for nominal data, like gender, we use the `string_fix` type. To realize probability distributions for string input fields the `string_var` type can be used. The use of the `integer` type results in a spinner widget. To manipulate decimal data via a slider we can assign the `double` type. If there is only one UI element within a container, the data cannot be ambiguous. Therefor the model driven concept does not offer the possibility to manipulate the probability for the single item. The use of the `boolean` type results in a checkbox with a probability spinner. This type behaves different, since each single UI element of a boolean value represents a probability distribution by itself. The probability for the checked state represents the inverse probability for the unchecked state and vice versa. That is why different boolean UI elements in one container do not affect each other, while other types within a container do.

V. Tactile Context Manipulation In Use

From the way of specifying data using the XML configuration file over data modeling and manipulation by tangible interaction to context data deployment, the presented approach supports developers in any point, and provides great usability. Even children are able to model context situations with the presented system.

Changes to any of the context model's items (user models, surroundings model, device- and component models, as well as distance relations) are communicated to dedicated subscriber modules via a message oriented middleware. We use the *Semaine API* as presented by Schröder [10] to realize a very flexible middleware concept. Semaine allows us to send the context data as text, XML, or even as EMMA messages. Based on change events from the surface, the tool's XML output is a set of probability distributions. Each container (cf. listing 1 and figure 1) gets represented by its own probability distribution. Again the `boolean` type forces a different handling. Not the container, but the nested UI elements are represented as probability distributions in XML. Using XSLT in a post-processing step, the context data can be transformed into almost any subscriber-specific text format.

We successfully used the tactile context editor in our collaborative research center. It turned out that this kind of context manipulation has interesting additional benefits. By the ability of the hardware to support concurrent tactile and tangible operations by several users, it is possible to change several context values simultaneously. This allows us to simulate complex real-world scenarios where different changes to the context model may occur at the same time. Another benefit arises from the reusable tangible context items. Since there is a binding coupling a tangible's byte tag and its context model data, the tangibles can be collect, stored in a repository, and can be reactivated whenever they are needed. This allows us to create reusable context items. A little weakness is owed to the fact that the XML-Schema is not as restrictive as it should be. The initial configuration file must be well-formed and valid but this still cannot prevent all modelling mistakes.

The presented tool was used to provide the context model in order to evaluate a system for context adaptive multimodal fission as described in [12]. As an example, one of the adaptive aspects of the fission module was to reason about the proper output device and output encoding of arbitrary information. In this case the decision depends

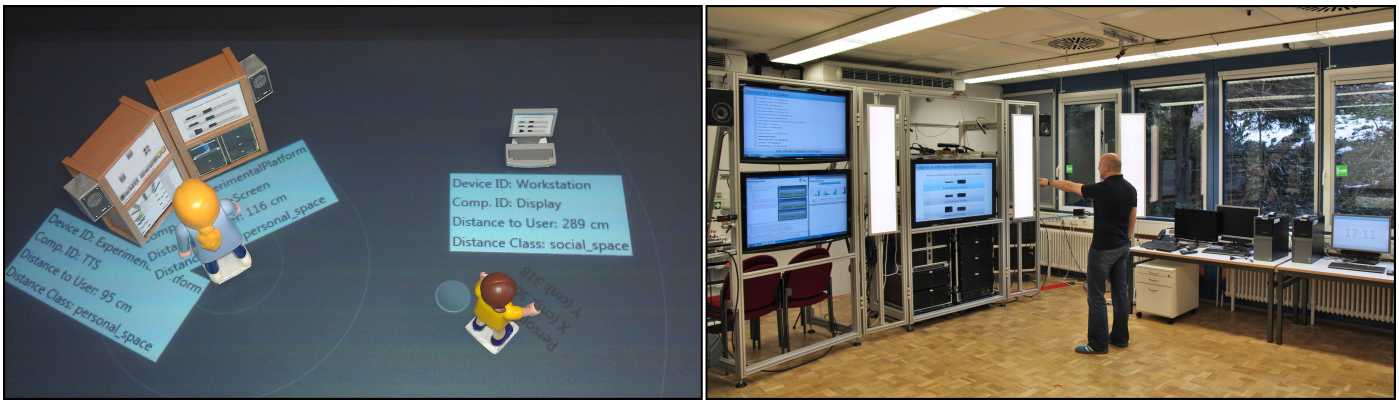

Fig. 2. On the left: the scenario as modeled and tested on the surface. On the right: A comparable real-world situation. The realized and tested system gathers its data from diverse sensors. The user localization is realized by laser scanners as described in [11]. This context knowledge is used to influence the UI's realization on different devices as described in [12].

on the user's preferences and abilities, the distances to the different devices, and other context information [12]. As depicted in figure 2 (left) the fission's reasoning was tested using the presented toolkit. More specific: by playing through different scenarios, different evaluation functions, which form the basis of the adaptive system, have been adjusted until the final fission module was successfully integrated in the consortium's multi-sensor platform (cf. figure 2 (right)). Using the presented tool allowed us to test and evaluate distributed context adaptive systems in advance without the need to use massive and costly sensory equipment. The fully tested fission module was integrated in the final system. From that point on, the user localization and the inference of distance data are realized using the intelligent environment's sensors as described by Geier et al. in [11]. A movie, which shows our realized tool in use is accessible here: http://companion.informatik.uni-ulm.de/ie2013/Tangible_Context_Modelling_for_Rapid_Adaptive_System_Testing.mp4.

VI. Conclusion

According to the needs when testing adaptive behavior, we presented a flexible and extensible way of how to model and manipulate context information. We gave a theoretical introduction to context knowledge and motivated a flexible approach for an easy-to-edit context model. The context model was realized with an implementation supporting tactile interaction on a multi-user platform. The presented prototype allows to alter context information in a very quick and easy way and meets the mentioned requirements from section III-B. As mentioned, this approach forms the basis of an evaluation and testing of our envisioned adaptive intelligent environment. The use of reusable tangible context objects proved to be very useful. The described concept on the tabletop is able to substitute diverse items from an intelligent environment. Due to the fact that the system's user interface is realized with a model-driven approach it can be tailored and used to test any context-adaptive component in a fast and easy way.

Acknowledgements

This work is originated in the Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG).

References

[1] M. Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, 1999, this article first appeared in Scientific America, Vol. 265, No. 3 (September 1991), pp 94-104.

[2] M. Weiser and J. S. Brown, "Beyond calculation," P. J. Denning and R. M. Metcalfe, Eds. New York, NY, USA: Copernicus, 1997, ch. The coming age of calm technolgy, pp. 75–85.

[3] B. Shackel, "Human-computer interaction," J. Preece, Ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 1990, ch. Human factors and usability, pp. 27–41.

[4] A. Dey and G. Abowd, "Towards a better understanding of context and context-awareness," in *CHI 2000 workshop on the what, who, where, when, and how of context-awareness*, vol. 4. Citeseer, 2000, pp. 1–6.

[5] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*. IEEE, 1994, pp. 85–90.

[6] H. Ishii and B. Ullmer, "Tangible bits: towards seamless interfaces between people, bits and atoms," in *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, ser. CHI '97. New York, NY, USA: ACM, 1997, pp. 234–241.

[7] J. Underkoffler and H. Ishii, "Urp: a luminous-tangible workbench for urban planning and design," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ser. CHI '99. New York, NY, USA: ACM, 1999, pp. 386–393.

[8] F. Schüssel, F. Honold, and M. Weber, "Influencing factors on multimodal interaction during selection tasks," *Journal on Multimodal User Interfaces*, pp. 1–12, 2012.

[9] T. Strang and C. Linnhoff-Popien, "A context modeling survey," in *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 – The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004, p. 8.

[10] M. Schröder, "The semaine api towards a standards-based framework for building emotion-oriented systems," *Adv. in Hum.-Comp. Int.*, vol. 2010, p. 21, January 2010.

[11] T. Geier, S. Reuter, K. Dietmayer, and S. Biundo, "Goal-based person tracking using a first-order probabilistic model," in *Proceedings of the Ninth UAI Bayesian Modeling Applications Workshop (UAI-AW 2012)*, 8 2012.

[12] F. Honold, F. Schüssel, and M. Weber, "Adaptive probabilistic fission for multimodal systems," in *OzCHI'12 – Proceedings of the ACM OzCHI 2012*. Melbourne, Australia: ACM, November, 26–30 2012.