Demo: The *acoowee*-Framework

Gerhard Fuchs, Lorenz Büttner, Christoph Damm, Mirko Hansen,
Florian Heisig, Tim Seyschab, and Reinhard German
Computer Science, Chair for Computer Networks and Communication Systems - Universität Erlangen-Nürnberg
Martensstr. 3, 91058 Erlangen, Germany
Email: {gerhard.fuchs, german}@informatik.uni-erlangen.de
lorenz.buettner@ce.stud.uni-erlangen.de / christoph.damm@googlemail.com
{mirko.hansen, florian.heisig, tim.seyschab}@informatik.stud.uni-erlangen.de

***Summary*—Within the framework of the *acoowee*-project we want to improve the programming of Wireless Sensor Networks (WSNs) by using and adapting the expressiveness of UML2 Activity Diagrams (UADs) to specify the behavior of the WSN. We expect that our proposal can simplify the way of programming the functional behavior of a WSN and so bring the programming closer to domain experts.**

We develop the *acoowee*-framework for Sun SPOTs (spots). The user of the *acoowee*-framework visually programs UADs and saves them. Afterwards, these UADs are converted into scripts and deployed to the spots e.g. over the air. Finally, the interpreter that runs on the spots executes the scripts.

In this demo we present the usage of the *acoowee*-framework by means of a network consisting of 5 spots and a host with a connected base station. We present all steps listed above. Additionally, we show a rule based manipulation of already deployed scripts and give the attendees the possibilities to try this kind of programming by themselves.

I. Introduction

For the usage of WSNs a simple and intuitive way of programming is important, as the behavior of a large number of SNs must be specified. As the saying goes, "a picture says a thousand words," we are developing a visual programming language for WSNs. Within the framework of the *acoowee*-project ('ac'tivity 'o'riented pr'o'gramming of 'w'ireless s'e'nsor n'e'tworks) [1] we use and adapt the graphical expressiveness of UML Activities (UAs) for this purpose. We see UAs as scripts that can be interpreted by the SNs. Programming means to compose actions like building blocks by specifying the workflow, the dataflow and the executing spots (fig. 1). We expect that our proposal will offer a powerful manner to specify the functional behavior ("**WHAT**" must be done? "**WHICH**" SN must do it?) of the WSN. Our idea is that the interpreter conceals the rough conditions of the WSN by dynamically caring about the necessary fault tolerance. The expected impact is that not only WSN experts, but also domain experts (potential user of WSNs) can program the functional behavior of the WSN. Another approach for visual programming of WSNs is *TinyInventor* [2].

II. Basics

The *acoowee*-framework is a prototype for Sun SPOTs [4] that realizes our concepts. It consists of `IDE`, `CORE`, `ACCESS` and `C-RULE`. We use Papyrus UML [3] that supports XMI2 (XML Metadata Interchange 2 [5]) as `IDE`, the rest is realized by us. Fig. 2.1 shows its usage: First of all the user of the *acoowee*-framework visually `programs UA[VIS]` using `IDE`. The user `saves UA[VIS]` as a XMI2-file (`UA[IDE]`), and converts `UA[IDE]` to a script (`UA[UAD]`) using `C-RULE`. `UA[UAD]` is being installed on the spot initially or via ACCESS during runtime, over the air. `CORE instantiates UA[UAD]` and parses it. Now `CORE` can `interpret` the internal representation of the UML Activity `UA[INT]`.

If a UA is instantiated by `CORE` as `UA[INT]`, `CORE` can `modify` it using a modification rule (`M-RULE`) (fig. 2.2). The resulting UA (`UA'[INT]`) can be serialized and saved on the spot as `UA'[UAD]`.

III. Exhibit

We demonstrate the current state the *acoowee*-framework and give the attendees the possibility to use it. For this we use the network shown in fig. 3. To explain the concepts and features of our framework we have prepared UAs e.g. the experiment presented in [1]. In detail an attendee can:

- see the programming concepts that are offered by our framework. He can see how to specify workflows, dataflows, synchronization, concurrency, allocation rules (static and dynamic), group allocation, decisions and explicit data aggregation;
- use the shell which is provided by `ACCESS` to control the WSN over the air. He can query the activities that are installed on the SNs, install UAs over the air during runtime, start the execution of our example activities, and modify already deployed activities;
- write and use scripts that use the commands of the shell to automate `ACCESS`;
- program its own `UA`s using `IDE`;
- convert, deploy and start the execution of its UA;
- modify its UAs during runtime.

For our demonstration we need a dedicated 802.15.4 frequency channel.

IV. Conclusions and further Work

Our experiences in developing and using the *acoowee*-framework indicates us that adapted UAs can be used to specify the functional behavior of WSNs in a powerful manner. Further improvements of `CORE` are necessary to gain fault tolerant behavior that is necessary for robust applications.

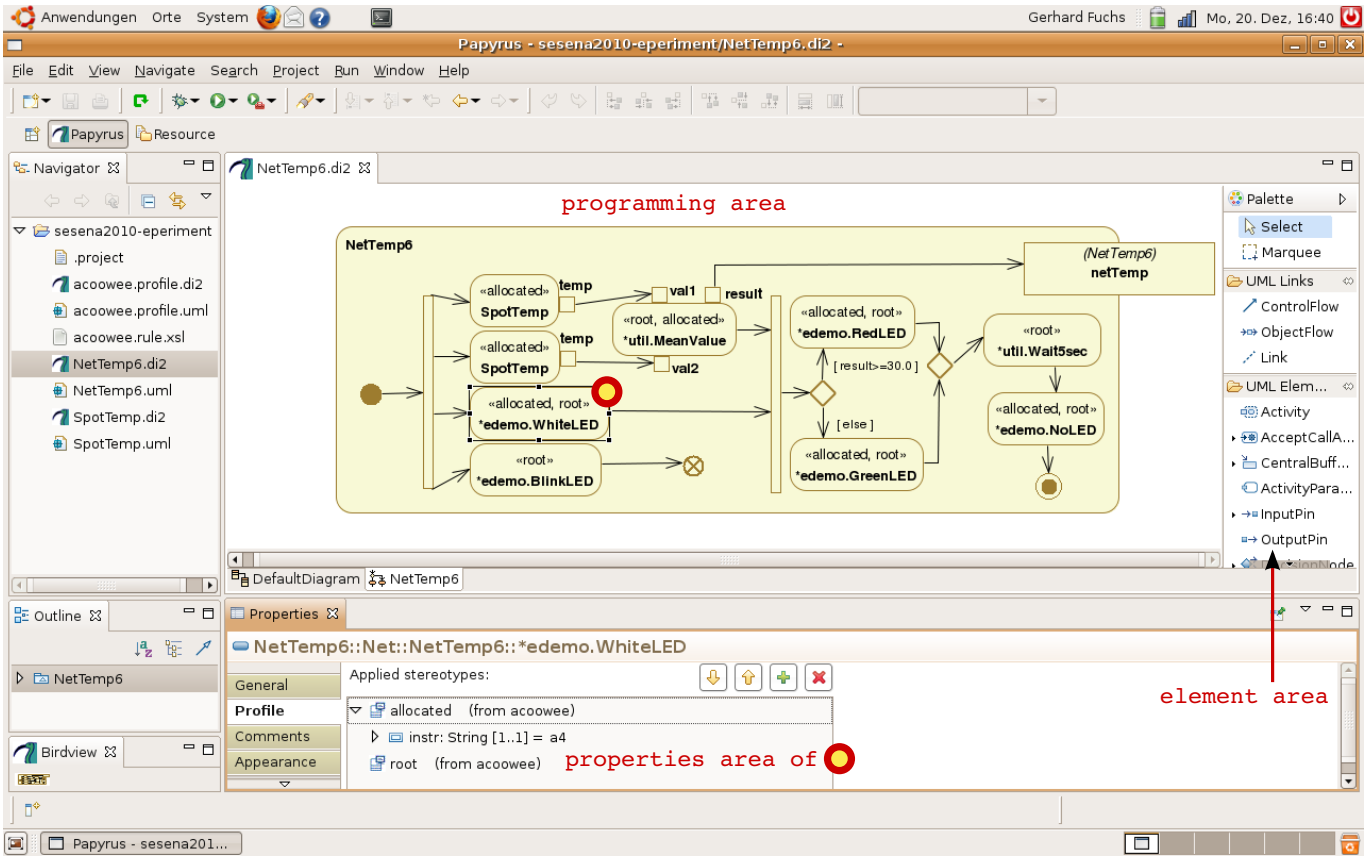

Fig. 1. Modified screenshot of the programming of a UA using Papyrus UML [3]. The *acoowee*-programmer composes the UA **NetTemp6** in the **programming area** by using the elements from the **element area**. In the **programming area** the workflow and the dataflow is specified. In the **properties area** a rule specifies how to select the spot that executes the labeled action. E.g **a4** means: "statically select SN a4". Amongst others, we demonstrate this example. Further details can be found at [1].

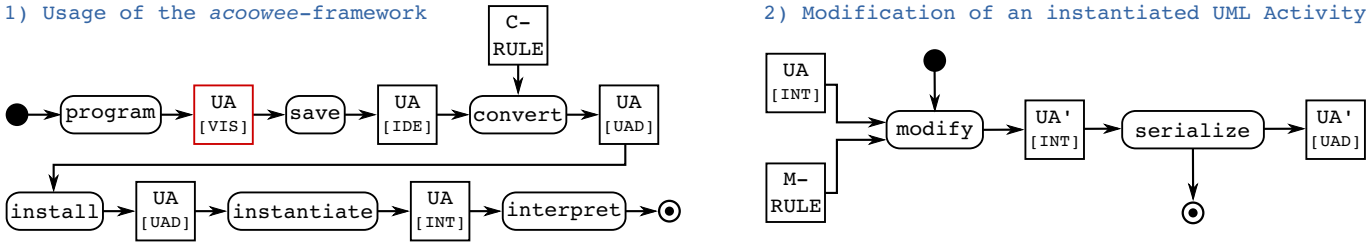

Fig. 2. Principal workflows during the usage of the *acoowee*-framework.

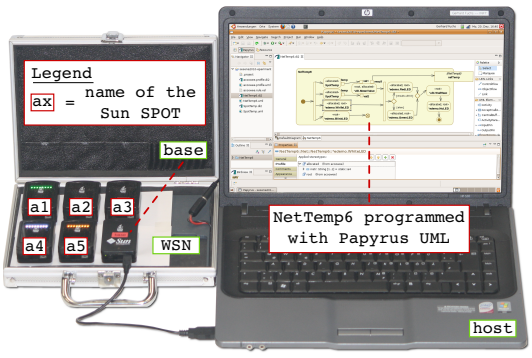

Fig. 3. Network consisting of 5 spots and a host with a connected base.

REFERENCES

[1] G. Fuchs and R. German, "UML2 Activity Diagram based Programming of Wireless Sensor Networks," in *Proc. of the ACM/IEEE Int. Conf. on Software Engineering (ICSE): Workshop on Software Engineering for Sensor Network Applications*. ACM, 2010, pp. 8–13, (SESENA: Cape Town, ZA-WC; May 2010).

[2] M. T. Hansen and B. Kusy, "TinyInventor: A Holistic Approach to Sensor Network Application Development," in *Extending the Internet to Low Power and Lossy Networks (to appear)*, 2011, (IP+SN: Chicago, US-IL; Apr. 2011).

[3] S. Gérard, "Papyrus UML," [web: 2011/04/04] http://www.papyrusuml.org.

[4] Sun Microsystems, "Sun SPOT," [web: 2011/04/04] http://www.sunspotworld.com/.

[5] OMG Object Management Group, "MOF 2.0/XMI Mapping, Version 2.1.1," OMG Available Specification without Change Bars formal/2007-12-01, Dec. 2007.