Poster-Abstract: Programming Wireless Sensor Networks using UML2 Activity Diagrams

Gerhard Fuchs, Christoph Damm, Reinhard German

Department of Computer Science 7, University of Erlangen-Nuremberg

Martensstr. 3, 91058 Erlangen, Germany

{gerhard.fuchs,german}@informatik.uni-erlangen.de

christoph.damm@googlemail.com

I. Summary

Wireless Sensor Networks (WSNs) consist of a large amount of sensor nodes (spots). We address the problem of how theses spots can be programmed so that they collaborate to fulfil a common task. The novelty of our work is that we assay how UML2 Activity Diagrams (UADs) can be used and adapted for this purpose. To gain insights, we develop a framework for Sun SPOTs. We design activities with Papyrus UML and transform the resulting XMI output to a data format that can be interpreted by the execution environment we have programmed for our Sun SPOTs. We have extended the expressiveness of UADs by an <<allocated>>-stereotype. This allows us to describe to which Sun SPOT an activity should be allocated. So we can not only program the behaviour of one Sun SPOT locally, but also the behaviour of the whole network. With this poster we introduce our attempt, the framework, a detailed experiment showing all supported UAD-elements and our preliminary results. Additionally, we give a brief insight in our further work.

II. Motivation

After intensive research in the field of WSNs [1] in the past, there is a trend towards studying how to program and use WSNs for real applications. Many unreliable spots (to avoid confusion, we use “spots” for sensor nodes and “nodes” in UADs) shall interact and fulfil a common task. How can a programming model cope with this problem? Sugihara and Gupta have written a detailed survey about programming models for WSNs [2]. None of the listed models uses UADs. Guerrero et al. have written a position paper [3] discussing some theoretical aspects in the field of workflow support for WSNs. To our knowledge a concrete implementation is not available. Unlike to our proposal they describe the workflows using state charts. UML2 Activity Diagrams (UADs) [4,5] enable the user to model workflows in a graphical, structured and hierarchical manner. To adapt the syntax and semantic of the diagrams, stereotypes can be specified.

Our framework consists of a tool for programming the UADs (IDE), an execution environment for UADs that runs on the Sun SPOTs (CORE), a transformation rule (RULE) and an access software to the network for the user (ACCESS). We use Papyrus UML 1.11.0 [6] as IDE, the rest is realized by us.

III. Preliminary Results and Further Work

Fig. 1 shows a simple example workflow that can be realized with our framework. The activity *example* is programmed with the IDE in this graphical syntax. After a transformation via RULE, the output can be transferred to CORE and afterwards started via ACCESS. CORE starts the execution at the Initial Node and executes *actionA* locally. It selects a Sun SPOT just in time, via an allocation process that is specified using the <<allocated>>-stereotype and delegates the execution of *actionB* RPC-like to its choice. It sends the output of *actionA* to the input at *actionB*, as specified in the diagram. CORE stops the execution when it reaches the Activity Final Node. This example gives us a hint that the dataflow in WSNs can be clearly specified via UADs.

We are currently investigating action allocation mechanisms, methodologies for reprogramming and are about to increase the number of Sun SPOTs of our network. Additionally, we are building robots that are controlled by Sun SPOTs to gain heterogeneity. To see all consequences of this attempt and to draw conclusions about WSNs in general, further research is necessary.

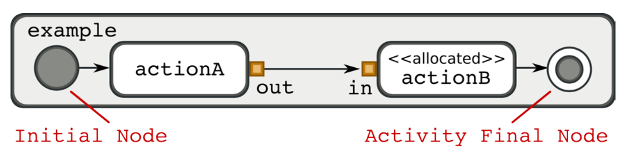

Fig.1. Sample activity showing an object-flow between *actionA* and *actionB*, symbolized by the arrow and the two squares *out* and *in*. actionB is labelled with the <<*allocated*>>-stereotype. This means that there is further information for the allocation process added (not shown in the diagram).

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E Cayirci., “Wireless sensor networks: a survey,” Elsevier Computer Network, vol. 38, pp. 393-422, 2002.

[2] R. Sugihara, and R. Gupta, “Programming models for sensor networks: A survey,” ACM Trans. Sen. Netw., vol. 4, no. 2, pp.1-29, 2008.

[3] Sun SPOT: https://www.sunspotworld.com [web.: 2009/09/27]

[4] OMG Object Management Group, “OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2,” OMG Available Specification without Change Bars formal/2007-11-04, 2007.

[5] OMG Object Management Group, “OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2,” OMG Available Specification without Change Bars formal/2007-11-02, 2007.

[6] S. Gérard, “Papyrus UML,” http://www.papyrusuml.org/ [web: 2009/05/18].