HELENA@Work: Modeling the Science Cloud Platform*

Annabelle Klarl, Philip Mayer, and Rolf Hennicker
Ludwig-Maximilians-Universität München, Germany

Abstract. Exploiting global interconnectedness in distributed systems, we want autonomic components to form teams to collaborate for some global goal. These teams have to cope with heterogeneity of participants, dynamic composition, and adaptation. HELENA advocates a modeling approach centered around the notion of roles which components can adopt to take part in task-oriented teams called ensembles. By playing roles, the components dynamically change their behavior according to their responsibilities in the task. In this paper, we report on the experiences of using HELENA in modeling and developing a voluntary peer-2-peer cloud computing platform. We found that the design with roles and ensembles provides a reasonable abstraction of our case study. The model is well-structured, easy to understand and helps to identify and eliminate collaboration mismatches early in the development.

1 Introduction

The development of distributed software systems, i.e. systems in which individual parts run on different machines connected via some sort of communication network, has always been a challenge for software engineers. Special care has to be taken to the unique requirements concerning concurrency and sharing of responsibilities. In this area, difficult issues arise particularly in those systems in which the individual distributed software components have a certain degree of autonomy and interact in a non-centralized and non-trivial manner.

Such systems are investigated in the EU project ASCENS [1], where the individual distributed artifacts are components which provide the basic capabilities for collaborating teams. These components dynamically form ensembles to perform collective tasks which are directed towards certain goals. We believe that the execution and interaction of entities in such ensembles is best described by what we call roles. They are an abstraction of the part an individual component plays in a collaboration. We claim that separating the behavior of components into individual roles leads to an easier understanding, modeling, and programming of ensemble-based systems. Our modeling approach HELENA [9,12] thus extends existing component-based software engineering methods by modeling roles. Each role (more precisely role type) adds particular capabilities to the basic functionalities of a component which are only relevant when performing the

* This work has been partially sponsored by the EU project ASCENS, 257414.
role. Exploiting these role-specific capabilities, we specify role behaviors which the component dynamically adopts when taking over a role. For the specification of role behaviors we extend [9] by introducing a process language which allows to describe dynamic creation of role instances on selected component instances. The structural characteristics of collaborations are defined in ensemble structures capturing the contributing role types and potential interactions.

In this paper, we report on the experiences of using HELENA in modeling and developing a larger software system. As our case study we have selected the Science Cloud Platform (SCP) [14] which is one of the three case studies used in the ASCENS project. The SCP is, in a nutshell, a platform of distributed, voluntarily provided computing nodes. The nodes interact in a peer-to-peer manner to execute, keep alive, and allow use of user-defined software applications. The goal of applying HELENA to the SCP is to find a reasonable abstraction that serves as clear documentation, analysis model, and guideline for the implementation. We experienced that the HELENA model helps to rigorously describe the concepts of the SCP. During analysis of the models, collaboration mismatches can be eliminated at early stages. As we shall discuss, the implementation also benefits from the encapsulation in roles. However, during implementation some additional effort is required to provide an infrastructure which supports the role concept on top of the component-based system. Lastly, special care has to be taken to make the system robust against communication failures and to provide communication facilities between ensembles and the outside world which is not yet tackled in HELENA.

In the following sections, we first describe the case study in Sec. 2. Afterwards, we summarize the HELENA modeling approach in Sec. 3 and apply it to the case study in Sec. 4. Sec. 5 describes the realization of the HELENA model on the infrastructure of the SCP and Sec. 6 discusses some related work. Lastly, we report on experiences and give an outlook in Sec. 7.

2 Case Study

One of the three case studies in the ASCENS project is the Science Cloud Platform (SCP) [14]. The SCP employs a network of distributed, voluntarily provided computing nodes, in which users can deploy user-defined software applications. To achieve this functionality, the SCP reuses ideas from three usually separate computing paradigms: cloud computing, voluntary computing, and peer-to-peer computing. In a nutshell, the SCP implements a platform-as-a-service in which individual, voluntarily provided computing nodes interact using a peer-to-peer protocol to deploy, execute, and allow usage of user-defined applications. The SCP takes care to satisfy the requirements of the applications, keeps them running even if nodes leave the system, and provides access to the deployed applications. For a full description of the SCP, we refer to [14]. In the following, we only discuss those parts relevant for this paper.

The SCP is formed by a network of computers which are connected via the Internet, and on which the SCP software is installed (we call these nodes).