Model-driven Development of Self-organizing Control Applications

Hans-Ulrich Heiß, Gero Mühl, Helge Parzyjegla, Jan Schönherr
Berlin Institute of Technology

Torben Weis, Arno Wacker, Sebastian Schuster
University Duisburg Essen

We are developing a modeling language that describes the behavior of an AS-Net application at a high level of abstraction. Developers using this language do not have to care about heterogeneity, network structures, self-stabilization, or even self-organization. A model transformer splits the model in separate roles and determines how these roles cooperate, i.e. which ones are strongly coupled and which cooperate rarely. The extraction of this meta-information is the key reason for using a model-driven approach. Based on this meta-information, the system can self-organize at runtime. Furthermore, the model transformer can generate an executable implementation from the model which is at the same time guaranteed to be self-stabilizing.

Roles

Roles are extracted from the application model and realize a certain part of the application’s workflow. Each role presumes a set of capabilities that describe the minimum requirements for a device to be able to execute that application part. Additionally a role specification may be enriched with information about the desired quality of service, the induced load, or the expected bandwidth usage. This additional knowledge can be kept as meta-information during the rest of the transformation process and exploited at runtime to guide the self-organization process. Since roles do only address other roles and not a concrete node, they decouple the distributed application from the actual device it runs on and, thus, facilitate the development of flexible and adaptive applications.

Self-Organization

One interesting feature of self-stabilizing algorithms is that they do not need any initialization and, hence, are perfectly suited for systems that must organize themselves. We employ a self-stabilizing role assignment algorithm which takes care of assigning all roles that are needed to nodes which are capable of serving them. Additionally, the assignment must be adapted to network changes and faults afterwards without external intervention. The more details of the enriched role model are available at runtime, the more adequately the role assignment can be carried out. If role requirements are additionally equipped with meta-information about the desired quality of service, devices are able to advertise how good they are in performing a particular role. Using these advertisements it is possible to determine the most convenient assignment for the user in the present context. However, the assignment currently does not take network limitations into account. It can happen that two heavily communicating roles are assigned to different nodes that only share a small-bandwidth link. Therefore, we use meta-information to optimize the role placement at runtime. We exploit knowledge derived from the application model as well as from the fragmentation process of an application into roles to mark these roles that are presumed to have a high communication demand.

Conclusions

The MODOC project aims at encapsulating necessary expert knowledge in the model transformation process in order to free application developers from having to care about distribution, heterogeneity, deployment, and self-organization. In order to facilitate applications to organize themselves, we build on a flexible role concept and the usage of self-stabilizing algorithms that do not require initialization. Knowledge about behavioral aspects of the application as specified in the model and embedded in the role specification is a very valuable resource that can be exploited in the self-organization process at runtime. As an example, we use meta-information derived from the application model and the role model to find optimized role assignments regarding the quality of service and communication costs.

Optimized dynamic role assignment.

Web: http://kbs.cs.tu-berlin.de/projects/modoc
E-mail: modoc@kbs.cs.tu-berlin.de