awareness magazine
SELF-AWARENESS IN AUTONOMIC SYSTEMS
10.2417/3201111.003884

Engineering autonomic ensembles

Martin Wirsing, Matthias Hölzl, Annabelle Klarl, and Nora Koch

New techniques aim to cope with the complexities of designing and developing ensembles: self-aware swarms of autonomous components that adapt their collective behavior to different situations.

The European Union ASCENS (autonomic service-component ensembles) project[1] aims to research and invent new approaches for designing and developing ensembles. A very good example of an ensemble is a swarm of robots autonomously performing complex tasks, similar to the way some animals do in nature. For example, several ants together can carry objects that are much too heavy for a single ant. In swarm robotics, the designer of the system sets up simple local rules that lead to cooperation among the robots. This approach is called distributed control, to distinguish it from centralized control systems. In the rescue scenario shown in Figure 1, the robots rely only on distributed control to move a child out of a simulated danger zone. The video recorded by our partner,[2] the École Polytechnique Fédérale de Lausanne, shows the rescue even more clearly. This example illustrates how distributed control has the potential to provide flexible, robust systems.

To date, swarm robotic systems tend to be designed in an ad hoc fashion, based largely on the intuition of the system designer. It is very hard to predict in advance what such a system will do, or to provide any formal guarantees about its behaviour. Swarm robotics is one of three case studies of the ASCENS project. Similar engineering challenges arise in the other two case studies, namely cloud computing and collaborating e-vehicles (intelligent vehicles that communicate with each other and with the road infrastructure). In ASCENS we are developing novel solutions to these ensemble engineering problems, based on the approach sketched in Figure 2.

Ensembles in general are software-intensive systems with massive numbers of nodes or complex interactions between nodes. They operate in open and non-deterministic environments in which they must interact elaborately with humans or other software-intensive systems to achieve the overall system's goal. Ensembles have to adapt dynamically to new requirements, technologies or environmental conditions without undergoing redeployment or interrupting the system's functionality,

Figure 1. *Rescue scenario in which a swarm of autonomous robots cooperates to move a child. Please view a video, available online, to see the scenario in action.[2] (Photo courtesy of Francesco Mondada of the École Polytechnique Fédérale de Lausanne.)*

thereby blurring the distinction between design time and runtime. For example, robots in the rescue scenario have to adapt to overcome the problem that each of them is unable to move the person by itself.

The inherent complexity of ensembles is a huge challenge. One of the most powerful tools for taming complexity is compositionality: the aim is to structure a system into well-understood building blocks, reducing the innumerable interactions between low-level components to a manageable number of interactions between these building blocks. The result is a so-called hierarchical ensemble, built from service components, simpler ensembles and knowledge units connected via a highly dynamic infrastructure.

Ensembles exhibit four main characteristics: adaptation, self-awareness, knowledge and emergence. Research on ensembles would benefit from a better shared understanding and a precise mathematical semantics of these fundamental notions. We are therefore in the process of defining a formal denotational system model for ensembles.[3] At the most fundamental and ab-

Continued on next page

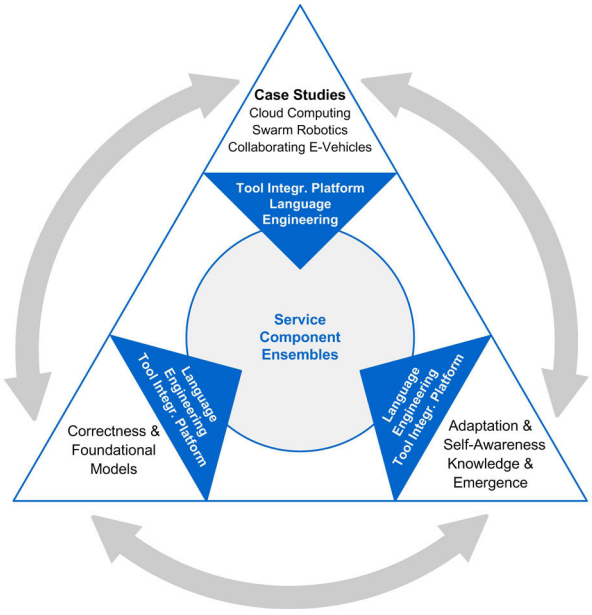

Figure 2. *The elements that make up the ASCENS approach to engineering ensembles.*

stract level, our system model represents an ensemble as a composition of input/output relations. It allows us to investigate the fundamental properties of ensembles independently of any operational mechanism. At a more pragmatic level—complementing these formal models and languages—we are providing an open-source software tool integration platform,[4] originally developed within the scope of the SENSORIA project[5] (see Figure 2). This platform will be further improved to support ensemble engineering in the ASCENS project.

Engineering ensembles requires a deep understanding and exhaustive modelling of the application domain and the goal of the whole system. The resulting implications for each ensemble and component must be thoroughly analysed. This high degree of complexity in development, deployment and management means that a very flexible engineering process for ensembles is advisable.

We need a sophisticated inspection phase where the ensembles are evaluated against the requirements, but are also carefully investigated for any unforeseen problems or difficulties, such as how a swarm of robots overcomes an obstacle (see Figure 3). The engineering process must also provide traceability mechanisms (for instance, to keep track of which scenarios lead to which requirements on components) and a way to propagate findings back to the modelling phase to improve the current model of the ensemble system (and of its components).

Figure 3. *Robot swarm passing a step. Engineers must carefully evaluate how ensembles behave in the face of unforeseen difficulties such as overcoming an obstacle. (Photo courtesy of Francesco Mondada of the École Polytechnique Fédérale de Lausanne.)*

The ensemble development process envisaged by the ASCENS project combines this flexibility with an explicit development feedback procedure. The process starts with an initial model of the individual components (e.g., a robot), the description of the collective goal or behaviour of the ensembles and their possible interactions as well as a specification of the environment and any constraints. To validate these initial models, we propose three different patterns: verification, simulation and prototypic experiments. In each case, the results of the validation provide feedback on the initial models, which is used to improve them. The models will be refined and validated iteratively until they satisfy the given collective goal.

The ASCENS project focuses on the many challenges of engineering self-organized, self-aware and autonomous ensembles, including their collective behaviour and interactions with the environment. We are developing sound techniques for formal reasoning and verification to support the design and development of these ensembles, as well as the analysis of their properties at runtime.

Continued on next page

Author Information

Martin Wirsing, Matthias Hölzl, Annabelle Klarl, and Nora Koch
Ludwig Maximilians University Munich
Munich, Germany
E-mail: wirsing@pst.ifi.lmu.de

Martin Wirsing is a full professor and head of the Programming and Software Engineering Unit in the Institute of Informatics. His areas of expertise include software development using formal methods and algebraic specifications. He is the coordinator of the European Union ASCENS project.

Matthias Hölzl is a senior researcher in the Programming and Software Engineering Unit and executive manager of ASCENS. His current research interests are high-level programming languages, and model-based architectures for intelligent reasoning and adaptation in open-ended environments.

Annabelle Klarl is a PhD candidate in the Programming and Software Engineering Unit. Her research focuses mainly on software engineering for ensembles, especially robot swarms.

Nora Koch is a senior researcher who leads the Web Engineering Group in the Programming and Software Engineering Unit. Her research interests focus on adaptive systems, modelling and model-driven engineering.

References

1. http://www.ascens-ist.eu ASCENS (Autonomic Service-Component Ensembles), EU project 257414. Accessed 26 November 2011.
2. http://www.youtube.com/watch?v=CJOubyiITsE Swarm-bots pull a child. *Credit: Francesco Mondada, École Polytechnique Fédérale de Lausanne.* Accessed 26 November 2011.
3. M. Hölzl and M. Wirsing, *Towards a system model for ensembles*, in G. Agha, O. Danvy, and J. Meseguer (eds.), **Formal Modeling: Actors, Open Systems, Biological Systems**, **Lecture Notes in Computer Science 7000**, pp. 241–261, Springer, 2011. http://www.pst.ifi.lmu.de/~hoelzl/papers/talcott.pdf
4. http://sde.pst.ifi.lmu.de/trac/sde/ The Service Development Environment (SDE) integrates various tools for developing service-oriented software. Accessed 1 September 2011.
5. http://www.sensoria-ist.eu SENSORIA (Software Engineering for Service-Oriented Overlay Computers), EU project 16004. Accessed 26 November 2011.