awareness magazine
SELF-AWARENESS IN AUTONOMIC SYSTEMS
10.2417/3201401.005231

Self-aware and self-expressive systems

Tatiana Djaba Nya, Stephan Stilkerich, Peter Lewis, and Xin Yao

A reference architectural framework and a modelling and simulation environment for self-aware and self-expressive systems are tested on an avionic subsystem.

Engineering self-awareness and self-expression in computing systems is an emerging trend in design and operation of modern computing systems, which have increasingly complex, heterogeneous structures and growing requirements. Embedded avionics systems could make particularly good use of self-awareness and self-expression. These concepts offer flexibility to avionic subsystems, allowing them to adapt and deal with the lack of knowledge, changes and unpredictability of their deployment environments, and also to work around or even overcome their functional and performance limitations due to their strict weight, power, size and density constraints.

Many efforts have been made previously to integrate self-awareness in computing systems. Some studies, such as that by Cox, consider self-awareness as a part of metacognition.[1] Other studies, such as that of Agarwal, focus on self-awareness in computing systems as an ability of a system to explicitly consider knowledge about itself.[2] As for the concept of self-expression, it has barely been considered in the engineering and computer science context.

Given the disparate and occasionally overly rigid use of the term ‘self-awareness’ in the literature and the lack of previous treatment of the idea of self-expression in the computing and engineering literature, we developed a more general working definition for self-aware and self-expressive computing systems inspired by biology. Based upon that working definition and with the aim of guiding the design of self-aware and self-expressive systems, we developed a reference architectural framewok which structures their requirements.[3] Then we implemented a modelling and simulation environment which enables the development of virtual simulators to systematically model, simulate and study the behaviour and performance of particular self-aware and self-expressive systems.[4] This environment is is new and there is no such environment in previous work. We intend for it to fill the gap between research at the conceptional level and implemention of the first proof-of-concept demonstrations.

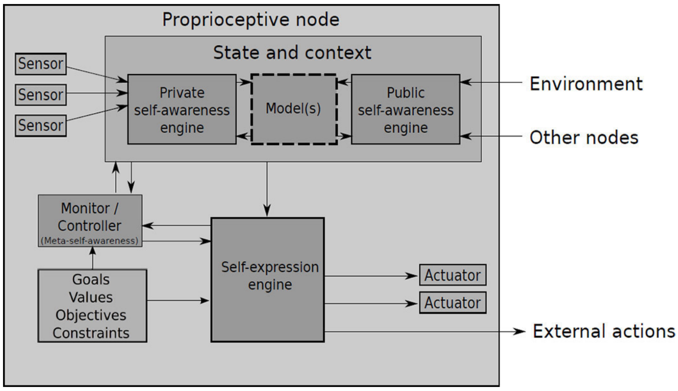

Figure 1. *Reference architecture for a single self-aware and self-expressive node.*

A self-aware and self-expressive system in our work is defined as a collection of nodes with both self-aware and self-expressive capabilities. The reference architectural framework of such a node is shown in Figure 1 and defines the node’s high-level structure, its required conceptual components and the information flow among them. Such a node does not need to be a specific physical system, but instead provides a conceptual container for the system being considered: the element in that context which is being referred to as *self*. A node could therefore be, for example, an autonomous agent, a running thread, a physical machine or a collective of these. Importantly, the node represents a level of abstraction at which knowledge acquisition, representation and behavioural processes occur.

A node is self-aware if it is in possession of information about its internal state, and has sufficient knowledge of its environment (including other potential nodes) to determine how it is perceived by the other parts of the wider system.[5,6] To enable self-awareness in a node, the defined reference architecture comprises (private) sensors which gather information about the node’s internal state and public sensors, illustrated here by environment, and other sensors which gather information about the node’s environment. Such information is collected inside the

Continued on next page

corresponding self-aware engines and processed within them with the help of the optional component model(s), if available. Depending on the processing results, the self-aware node will initiate self-expression by sending adequate information to the self-expressive engine.

To exhibit self-expression, a node should be able to assert its behaviour upon either itself or other nodes of the system, and this behaviour should be based on the node's goals, values, objectives and constraints.[5,6] The self-expressive engine is responsible for taking decisions about the actions that should be executed by the node to assert its behaviour, and chooses the corresponding actuator to perform the desired actions. For actions with effects on the node's internal state, the component termed 'actuators' in the reference architecture will be selected. For actions with effects on the outside world, the self-expressive engine will select external actuators, here termed 'external actions'.

Finally, the reference architecture comprises a monitor/ controller which has a high-level view of the node's behaviour and, when necessary, can adapt the node's own level of self-awareness and self-expression. This represents the meta-self-awareness in the node.

Due to the unpredictability associated with both deployment environments and the dynamics within them, one key challenge in realizing self-awareness and self-expression in computing systems is the appropriate use of effective online learning schemes. Online learning is applied in two contexts within a self-aware, self-expressive system: at the adaptation level and at the meta level.[3] In the reference architectural framework, online learning algorithms occur at the adaptation level in the self-awareness engines and the self-expression engine, and additionally at the meta level in the meta-self-awareness component.

Based on the defined reference architecture, we implemented a modelling and simulation environment using transaction-level modelling in SystemC, a system description language (see Figure 2). It is a virtual prototype of the reference architecture in which the communication among the components of a node is implemented and well separated from the implementation of the functional units, which depend on the system to be investigated. The processes used to initiate this communication are done so that the environment offers not only an accurate and reliable execution chronology of transactions (see Figure 3), which in turn ensures the functionality and information flow described in the reference architecture, but also gives fine timing resolution. Furthermore, the environment enables the instantiation of all nodes, it comprises elements enabling asynchronous communication of each single node with the outside world including other nodes, and thus enables us to simulate any topology of self-aware and self-expressive nodes.

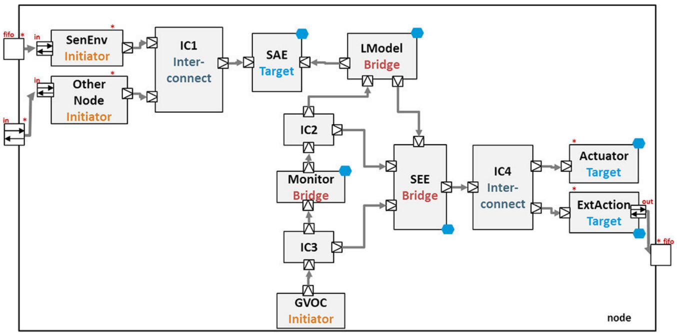

Figure 2. *Structure of the simulation environment. GVOC, SenEnv, LModel and SEE are components of transaction-level modelling.*

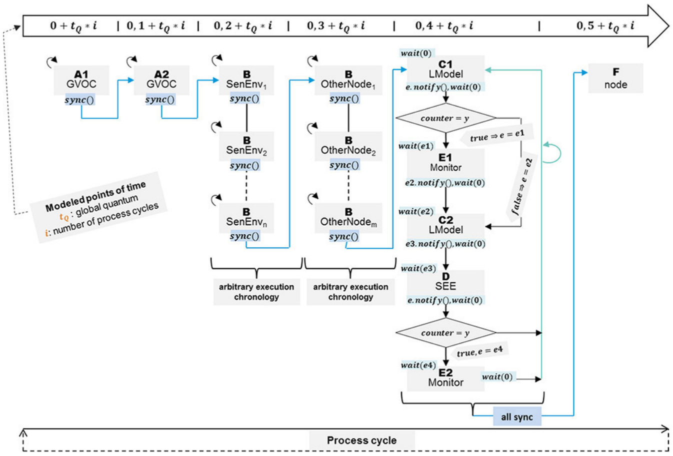

Figure 3. *Chronological execution order within a proprioceptive node. As in Figure 2, GVOC, SenEnv, LModel and SEE are components of transaction-level modelling. A, B, C, D, E and F are processes enabling communication between transaction-level modelling components.*

As a first use case, the environment is currently employed for a self-aware and self-expressive system that realizes novel concepts for fault tolerance to overcome one drawback of today's over-designed avionic systems. This concrete system is composed of one single self-aware and self-expressive node representing an avionic sub-system with safety critical functionality and showing a novel approach for software fault tolerance in avionic systems using self-awareness and self-expression.

In our ongoing work we continue to conduct research into the conception of self-aware and self-expressive systems, and further optimize the modelling and simulation environment

Continued on next page

which allows us to systematically model, simulate and study the behaviour and performance of self-aware and self-expressive systems.[7]

The research leading to these results was conducted through the EPiCS project and received funding from the European Union Seventh Framework Programme under grant agreement 257906.

Author Information

Tatiana Djaba Nya and Stephan Stilkerich
EADS Innovation Works
Munich, Germany

Peter Lewis and Xin Yao
CERCIA
School of Computer Science
University of Birmingham
Birmingham, United Kingdom

References

1. M. T. Cox, *Metacognition in computation: a selected research review*, **Artif. Intell. 169** (2), pp. 104–141, 2005. doi:10.1016/j.artint.2005.10.009
2. A. Agarwal, J. Miller, J. Eastep, D. Wentziaff, and H. Kasture, *Self-aware computing*, Tech. Rep. AFRL-RI-RS-TR-2009-161, Massachusetts Institute of Technology and DARPA, 2009.
3. T. Becker, A. Agne, P. R. Lewis, R. Bahsoon, F. Faniyi, L. Esterle, A. Keller, A. Chandra, A. R. Jensenius, and S. C. Stilkerich, *EPiCS: engineering proprioception in computing systems*, **Proc. IEEE CSE**, pp. 353–360, 2012.
4. T. D. Nya, S. C. Stilkerich, and P. R. Lewis, *A modeling and simulation environment for self-aware and self-expressive systems*, **Proc. IEEE SASO**, 2013.
5. P. R. Lewis, A. Chandra, S. Parsons, E. Robinson, K. Glette, R. Bahsoon, J. Torresen, and X. Yao, *A survey of self-awareness and its application in computing systems*, **Proc. IEEE SASOW**, p. 102107, 2011.
6. S. Parsons, R. Bahsoon, P. R. Lewis, and X. Yao, *Towards a better understanding of self-awareness and self-expression within software systems*, Tech. Rep. CSR-11-03, University of Birmingham, School of Computer Science, 2011.
7. P. Lewis, M. Platzner, and X. Yao, *An outlook for self-awareness in computing systems*, **Awareness**, 2012. doi:10.2417/3201203.004093