awareness magazine
SELF-AWARENESS IN AUTONOMIC SYSTEMS
10.2417/3201310.005094

Reflective systems are a good step towards Aware systems

Kirstie Bellman

Reflective systems that explicitly reason about themselves are effective and responsive to changes in the environment and to users.

Computational reflection is an engineered system's ability to reason about its own resources, capabilities and limitations in the context of its current operational environment.[1–3] Reflection, which can include many types of self-modelling, analysis and decision processes, allows systems to better understand how to map or fit their current capabilities—whatever they are—to the challenges of dynamic environments and human needs. Reflection capabilities range from straightforward processes that adjust the parameters or behaviours of other computer programs (e.g., altering the step size of a numerical process or applying rules that select models used for an application) to sophisticated analyses of the system's own reasoning processes (e.g., switching to a new strategy due to a failing plan). There are many types of reflective processes at different levels of a system.

The 'Wrappings' approach we have developed is one way to implement computational reflection and self-modelling systems.[4] Wrappings uses both explicit meta-knowledge and recursively applied algorithms that act on that meta-knowledge to recruit and configure resources dynamically to 'problems posed' to the system by users, external systems or the system's own internal processing. The Wrappings problem-posing style has many benefits, including separating problems from solution methods and keeping an explicit, analysable trace of what problems were used to evoke and configure resources.

The Wrappings meta-knowledge—machine-interpretable qualitative and quantitative descriptions for all of the system's resources—is not just about how to use the resources but also whether, when and why they should or can be used, often with additional qualitative information on best practices for using the resource. By limiting each Wrapping to the use of a resource for a specific problem and context, one can incrementally accumulate meta-knowledge for complex resources. The Problem Managers (PMs) algorithms use the Wrappings to select and configure resources to apply to problems. The PMs choreograph seven major functions: *discover, select, assemble, integrate, adapt, explain* and *evaluate.* Discover refers to what new resources can be inserted into the system for this problem. Where can they be found? Selection asks which resource(s) should be applied to this problem in this context. Assembly denotes syntactic integration and helps set up selected resources so that they can pass information or share services. Integration refers to semantic integration, including constraints on when and why resources should be assembled. Adaptation helps to adjust or set up a resource for different operational conditions. Explanation is more than a simple event history because it provides information on what was not selected and why. Evaluate includes the impact or effectiveness of the particular use of this resource.

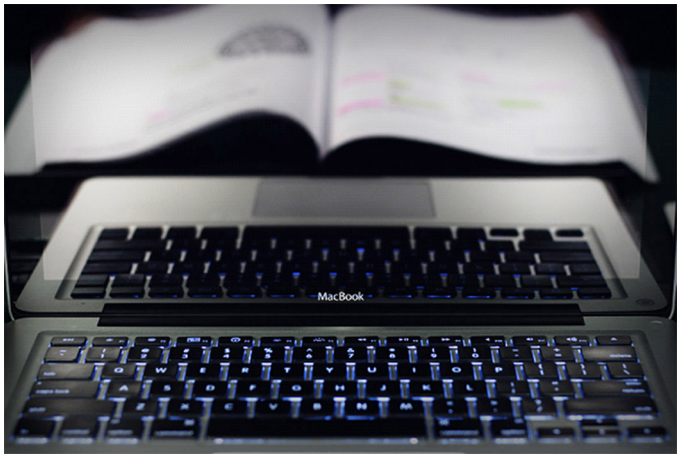

Figure 1. *The validity of computational reflection depends on the appropriate use of data, instrumentation, reasoning processes and analyses. (Photocredit: CC BY-NC-ND 2.0 Aaronth.)*

The Wrappings approach has no privileged resources: everything is 'wrapped'. This includes analyses, tools, databases, planners, platforms, devices and all Wrappings infrastructure and reflection programs. Hence, everything is available to be reasoned about and potentially dynamically reparameterized or swapped out for other resources. The Wrappings resources and their interactions allow, in essence, a simulation of the system. With appropriate reflection processes the system can use this information to replan and alter the current use of resources. This allows sophisticated adaptive processing. Furthermore,

Continued on next page

unlike biological systems, Wrappings allows access to any self-monitoring and self-decision processes. Our current research testbed has diverse robotic cars with different sensors adaptively playing collaborative and competitive games.

We need systems that not only adhere to our designs but also act in accordance with our intentions and values. Computational reflection gives systems some of the information that could be helpfully communicated to us—developers and users—on their state, perceptions and goals. However, the validity of the system's self-modelling and reflective processes depends on the appropriate use of data, instrumentation, reasoning processes and analyses, which like all complex systems can become overwhelmed with large data sets and with unexpected interactions and side-effects. We need new methods of proving that these reflective processes work as desired. One small step in Wrappings is to have reflective processes subject to exactly the same meta-knowledge (with rules of use, limits and assumptions) and the same evaluative checks and instrumentation as any other resource in the system. This can help ensure that resources are used appropriately. However, this is not enough. Consequently, we are researching new methods and logics. The formidable task of proving correctness in reflective systems should act as a significant harbinger of what the Awareness community faces eventually in devising more interesting capabilities.

The work on robotic cars is being carried out in collaboration with California State Polytechnic University, Pomona, directed by P. Nelson.

Author Information

Kirstie Bellman
The Aerospace Corporation
Los Angeles, CA

References

1. K. Bellman, *Self-conscious modeling*, **IT 47** (4), pp. 188–194, 2005.
2. C. Landauer and K. Bellman, *New architectures for constructed complex systems*, **Appl. Math. Comput. 120**, pp. 149–163, 2001.
3. C. Landauer and K. Bellman, *Knowledge-based integration infrastructure for complex systems*, **Intell. Control Syst. 1** (1), pp. 133–153, 1996.
4. C. Landauer, K. Bellman, and P. Nelson, *Modeling spaces for real-time embedded systems*, **Proc. 4th IEEE Wrkshp. Self-Org. Real Time Syst.**, 2013.