Comments on "An Autopoietic View of the Concept Information Systems" by E.-S. Abou-Zeid

Motoshi Saeki

Tokyo Institute of Technology, Department of Computer Science, Ookyama 2-12-1, Meguro-ku, Tokyo 152-8552, Japan, email:saeki@cs.titech.ac.jp

Key words: Autopoiesis, Information system modelling, Reflection

Abstract: The modelling framework that the author proposed has the capability of adapting the dynamic changes of information systems by themselves. In this paper, I analyse the computational aspect of the author's "An Autopoietic View of the Concept 'Information Systems'" and compare it with reflective computation technique.

Information systems frequently change their characteristics such as their structures and behaviour even during their actual operation according to the changes of external environments. It is very important to adapt the systems without stopping their operation if the changes occur. We should model and design the systems so that they can have the flexibility to adapt the changes and to evolve themselves. One of the key techniques to get the flexible and evolvable model of an information system is to make the model have a "self-referential" and "self-updating" mechanism. The model having this mechanism can access its current state and update it by itself. It can change even its interpretation of the model representation.

El-Sayed Abou-Zeid's paper discussed a meta model with "self-referential" mechanism for modelling the dynamics of information systems. The author employed the concept "Thing" to model the structure of information systems and it expresses "Object" that are structural constituents. Furthermore the author introduced the notion of "Negotiated Meaning (shortly NM)" for interpreting the constituents of information systems, and it has various kind of information, e.g. data that the constituent has, the

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35500-9_30

E. D. Falkenberg et al. (eds.), *Information System Concepts: An Integrated Discipline Emerging*

business rules related to the constituent, actions that changes the state of the constituent and so on.

The NM encapsulates the data, the operations (actions) and the constraints (business rules) on a constituent. To generate a new NM and to update the existing NMs, the modelling framework has event-driven rules. The rules generate a new NM or update the information the NM has when an event that causes dynamic change of the information system occurs. This mechanism is similar to computational reflection of programming languages [Maes87]. Programming languages with reflection have syntactic constructs for accessing to their interpreters, for updating computational states of the interpreters and even for changing computation rules of the interpreters. From comparative view with reflection, I listed up the discussion points and the comments to the author's modelling framework as follows.

1. Separating or identifying clearly event-driven rules that create and update NMs from Things. When developers model actual systems with two level architecture; meta level as event-driven rules and object level as Thing, one of the most crucial problems is the methods to separate them.
2. Maintaining reflective consistency between meta level and object level. Suppose that we have two event-driven rules, both of them update the same NM. As the external environments to a system changes, these rules try to change the same NM simultaneously. This simultaneous updating causes semantic inconsistency of the Thing of the NM.
3. Methods or methodologies to model information systems following the author's framework. This topic is closely related the item 1. The methods provide some guidelines, e.g. how to identify Things, for developers to model an information system. The existing methods, e.g. James Martin's Information Engineering, can be extended into the methods that are adapted to two-level architecture (NMs and Things).
4. Possibility of defining all or almost of event-driven rules that can express all dynamics of information systems during modelling a system following the framework. The changes of external environments and the system's reactions to them are unpredictable.
5. Reflective tower problem in programming language community. When we consider dynamic change of event-driven rules on NMs, we should construct meta-meta level where some operational rules that update the event-driven rules of the NMs can be defined. Continuing this consideration, we get an infinite tower. The point is where we should stop the construction.
6. More examples, especially event-driven rules can show more practical benefits of the author's technique.

References

[Maes87] Maes,P., *Concepts and Experiments in Computation Reflection*, In: Proc. of Object-Oriented Programming Systems, Languages and Application (OOPSLA), pp.147-155, 1987