Modelling of a Safety Instrumented System by a Biologically Inspired Modular Construct

Luca Pazzi

1 Abstract

We present an ongoing research aimed at investigating aspects of a modelling paradigm [1] where system behaviour is modelled by biologically inspired [2] concurrent and autonomous modules through a state based formalism. Such modules are named *holons* after the work of Arthur Koestler, since they are designed in order to host both the features of parts and wholes. Current modelling paradigms tend at emphasising the parts, but miss the notion of whole. A whole models the associative behaviour observed in the domain of interest, while the parts model the behaviour of a specific entity. Holons are aimed at filling the gap. Holons can act as *parts* by exhibiting the interface of the state behaviour. At the same time holons can act as *wholes*, by having the state machine behaviour annotated with actions and triggers which allow them to communicate with other holons, coordinating them and therefore modelling the related associative behaviour. In the paradigm, the two roles are tied together, the associative behaviour becoming recursively the behaviour of a single entity which can be composed into further wholes.

The basic idea in the the original work of Koestler [2] is that an entity "is not a simple structural aggregation of elementary parts nor a functional chain of elementary units of behaviour", rather it is a multi-level hierarchy of units, named holons, arranged into hierarchies called holarchies. Holons are self-regulating autonomous systems which display both the independent properties of wholes and the dependent properties of parts. The concept of holon by Koestler is intended to go beyond the reductionist approach, mainly since it provides a recursive model of composition, mainly by its whole-related nature.

In the proposed paradigm, the state machine hosted within the holon plays both the role of part and whole at the same time: by implementing the associative behaviour among its components, and by making such a behaviour available as if it was a single entity. Consider for example the modelling of a safety critical device, composed by a laser L and by a protective cover C. A device implementing a safety function according to IEC-61508, that is a Safety Instrumented System, can be modelled by an holon S whose behaviour

consists in turning the laser off when the cover is raised and not allowing to turn it on until it is lowered. S can therefore be seen as the whole which models the associative behaviour among the cover and the laser. Figure 1 shows the holon S implementing the joint behaviour among the cover C and the laser L. The behaviour of the cover is modelled by a state machine whose transitions may happen at any time autonomously emitting events close and open. The behaviour of the laser is instead given by a state machine whose transitions may be controlled by receiving events on and off. Transitions which happen autonomously, like t_3 and t_4 in the cover, are distinguished by a starting white dot in the arrow. Underlined events may trigger transitions, like t_2 and t_3 in the laser. In other words the cover may be seen as sensor, that is a device whose behaviour may be sensed by the holon H having it as component according to the events it emits. Conversely the laser is an actuator, whose behaviour may be prescribed by events sent to it by holon H having it as component.

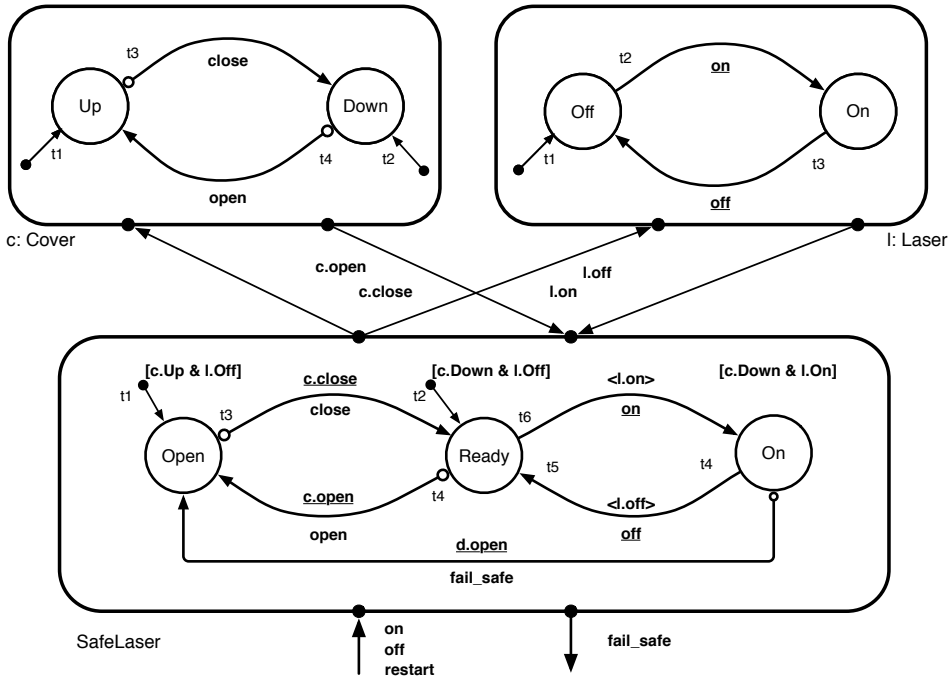

Figure 1: The holon SafeLaser which implements the associative behaviour of a cover and a laser.

Holons are arranged in tree-like hierarchies, called *holarchies*. An holarchy represents a whole working system which may become part of more complex systems. Holons are the nodes of the tree. A node h_i having a set of N parent nodes $C = \{h_{i1}, h_{i2}, h_{iN}\}$ models the associative knowledge among them. The dynamical aspects of such association are

modelled by a state machine hosted within h_i, which communicates with state machines hosted within component holons in set C.

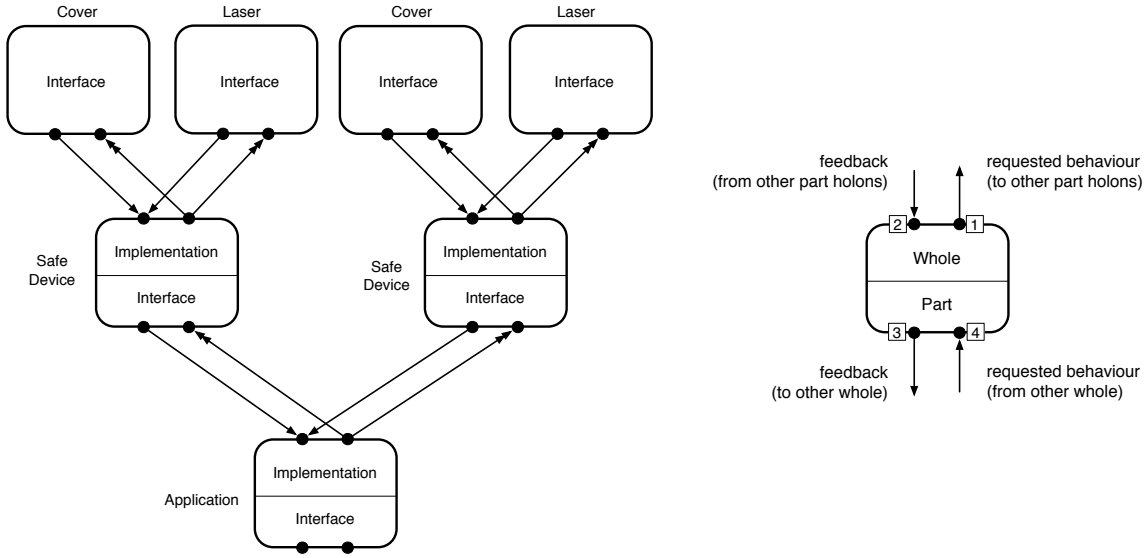

Figure 2: The holarchy of an application built from two instances of the safe device of Figure 1 and the communication ports of a single holon.

The modular part-whole structure allows to assign a *state semantics* to each state in the behaviour. This is feasible by adopting Part-Whole Statecharts [3] for implementing state behaviour. Each module can be therefore checked against safety axioms and reused without having to recheck it once composed in different contexts.

In order to show the scalability of the approach we show how different devices S may work together in order to obtain further coordinated behaviour. Suppose for example that two lasers have to work together in a plant. A control system implements some sort of redundancy ensuring at the same time mutual exclusion by activating a second safe laser if the first goes to fail safe state by the cover being opened accidentally. The modules are arranged as shown in the holarchy of Figure 1. Figure 3 shows a portion of the behaviour of the application which coordinates the behaviour of the two safe lasers. Observe that the safe lasers s_1 and s_2 encapsulate the basic safety function of the cover. In this way we separate basic application control by safety concerns, reducing the overall complexity.

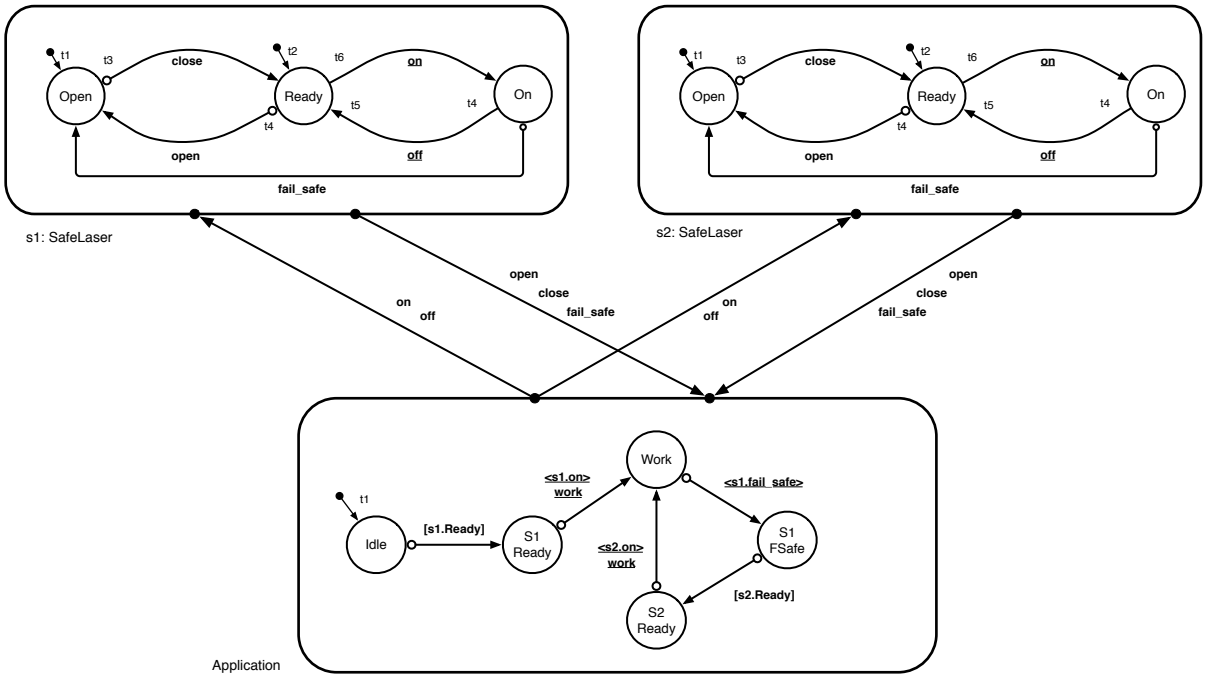

Figure 3: An application built from the two safe device. Observe that we employ the interface and not the implementation of the SafeLaser of Figure 1.

References

[1] Pazzi, L.: Modeling systemic behavior by state-based holonic modular units. In Dingel, J., Schulte, W., Ramos, I., Abrahão, S., Insfran, E., eds.: Model-Driven Engineering Languages and Systems, Cham, Springer International Publishing (2014) 99–115

[2] Koestler, A.: Some general properties of self-regulating open hierarchic order. In Koestler, Smythies, eds.: Beyond Reductionism: New Perspectives in the Life Sciences. Hutchinson, London (1969) 210–216

[3] Pazzi, L., Pradelli, M.: Modularity and part-whole compositionality for computing the state semantics of statecharts. In: Application of Concurrency to System Design (ACSD), 2012 12th International Conference on. (june 2012) 193 –203