Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P616-cd

Towards a Graphical Specification of Operational Rules in RiskSpectrum ModelBuilder

Ola Bäckström, Pavel Krcal, Helena Troili

RiskSpectrum AB, Sweden. E-mail: {ola.backstrom, pavel.krcal, helena.troili}@riskspectrum.com

Model Based Safety Assessment (MBSA) tools encapsulate dependability expertise in the definition of high-level components. Detailed (formal) description of component behavior and interactions can be created by an expert and exposed to users only on the level required for building system models. Knowledge Bases in RiskSpectrum ModelBuilder (KB3) implement this separation by offering an analyst a library of graphical components with their properties and possible connections. Component behavior and interactions are pre-defined using the modeling language Figaro. This includes also operational rules that steer the system under study. We generalize our experience from real-life projects that developed such Knowledge Bases. We investigate how a common graphical formalism such as flow charts can be used, in connection with the Figaro language, to structure the Knowledge Base creation and facilitate quality of the final code. The proposed method takes a graphical specification of operational rules satisfying certain additional conditions on input and guides the Knowledge Base creation process. This is the first step towards automatic generation of the Figaro code from a graphical specification.

Keywords: Model based safety assessment, knowledge bases, Figaro, graphical specification.

1. Knowledge Base Approach

Performing dependability studies, for example reliability/risk assessment of a nuclear station, availability assessment of a production unit is a complex task. If the asset under consideration is large/complex, many persons are involved in such analysis. The model created might also be kept alive and modified as the asset is modified. There is a need for tools that can help to encapsulate knowledge, simplify updates, add unique or tailormade features in the assessment and facilitate the digitization also in the risk, reliability and availability domain.

There are several frameworks that can to different degree satisfy the above criteria. In this paper we discuss mainly the modelling language Figaro (Bouissou et al., 1991) which is used in RiskSpectrum ModelBuilder (KB3). Some other established MBSA frameworks include AltaRica (Point and Rauzy, 1999), Safety Analysis Modeling Language (SAML) (Güdemann and Ortmeier, 2010), Hierarchically Performed Hazard Origin and Propagation Studies (Hip-HOPS) (Papadopoulos and McDermid (1999)), and xSAP (Bittner et al. (2016)).

The Knowledge Base approach in RiskSpectrum ModelBuilder (KB3) is based on the concept that general modelling rules (which component types can be represented, which failure modes and data, how component types is allowed/expected to interact, etc.) are stored in a so-called Knowledge Base. The Knowledge Base is hence a "center for knowledge" and is typically setup by an expert of this type of problem – to simplify and quality assure the modeling across systems. This means that the analyst building the models of the systems can focus on the system itself and the generic modelling routines are managed by the Knowledge Base. Possible application areas for the Knowledge Base approach include production analysis of processing plants, power plants or downstream oil&gas industry. Analyses can evaluate and compare different designs by modifying the plant or its reliability parameters.

The Knowledge Base approach using the modeling language Figaro (Bouissou et al., 1991) offers all flexibility in adapting component type behavior and interactions exactly for the purpose of the dependability study. This includes not only rules for individual types of components, but also encoding relevant rules determining plant behavior based on the global state – operational rules of the plant (Krcal et al., 2022).

2. Graphical Representation of Operational Rules

As an addition to already existing design and debugging tools, this paper proposes a new method for encoding operational rules in the Figaro code of a Knowledge Base. A Knowledge Base creator specifies these rules in a commonly used formalism (e.g., flow charts), following certain restrictions on the conditions and commands. This specification together with the Figaro definitions of the other relevant classes is used by our method to structure the component interaction, information flow and plant decision steps. We also evaluate to what extent can the Figaro code be automatically generated from such high-level descriptions.

We illustrate the conceptual idea by an example. Assume a processing system with three units (*1*, *2*, and *3*) sending output product to a storage. Storage is emptied by another process according to its demand *D*. Storage level *L* should not fall below a threshold *T*.

Operational rules of the plan start and stop the processing units according to the priority order where *1* has the highest priority and *3* the lowest. Figure 1 depicts these operational rules in a flow chart. One can derive the following structures in the Figaro code of a Knowledge Base.

Interaction rules of the class representing the operation control unit can be derived directly from the flowchart. If we express conditions and commands in the Figaro language, then each path between two adjacent command represents one rule. Variables *level* and *i* become local attributes of the class. They are updated only in the interaction rules of this class and never used outside of this class.

Order of interaction rules execution across the classes (Steps Order) is determined by using attributes from other objects. E.g., *L* and *D* have to be calculated before we enter operational rules. Thus, their step order will be lower than that of the operational rules. The processing level of units *P(i)* also needs to be calculated before we use it.

There has to be a possibility to start and stop processing units. The control unit class needs an interface where processing units will be included. Interaction rules can change the state of the processing units, which update their processing capacity by own interaction rules in a step following interaction rules of the control unit.

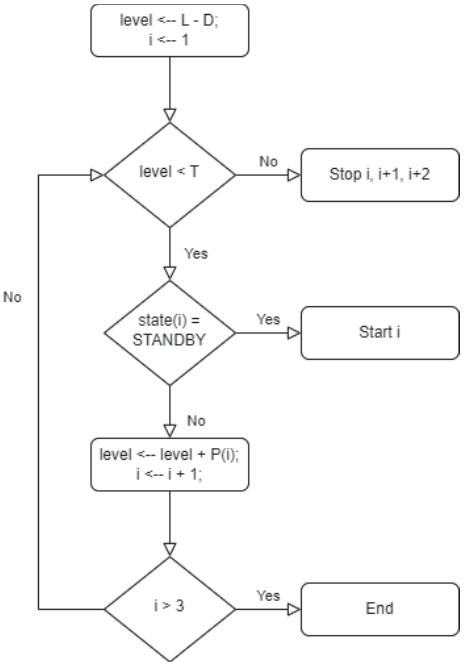

Figure 1. A flowchart for sample operational rules.

2. Conclusions

This work shows the first steps towards a structured or even automatic generation of the Figaro code for Knowledge Bases from a standard graphical representation of operating rules.

References

Bittner B., Bozzano M., Cavada R., Cimatti A., Gario M., Griggio A., Mattarei C., Micheli A., and Zampedri G. (2016). The xSAP Safety Analysis Platform. *In Proc. of TACAS'16.*

Bouissou M., Bouhadana H., Bannelier M., Villatte N. (1991). Knowledge modeling and reliability processing: presentation of the Figaro language and associated tools. *In Proc. of SAFECOMP'91.*

Güdemann M., Ortmeier F. (2010). A framework for qualitative and quantitative model-based safety analysis. *In Proc of HASE 2010.*

Krcal P., Troili H., and Bäckström O. (2022). Control Logic Encoding using RiskSpectrum ModelBuilder. *In Proc. of PSAM16.*

Papadopoulos Y., McDermid J. (1999). Hierarchically performed hazard origin and propagation studies. *In Proc. of SAFECOMP'99*.

Point G., Rauzy A. (1999). AltaRica: Constraint Automata as a Description Language. *Journal Européendes Systémes Automatisés* 33(8–9):1033–52.