Semantic DMN:
Formalizing Decision Models with Domain Knowledge
(Extended Abstract)*

Diego Calvanese[1], Marlon Dumas[2], Fabrizio M. Maggi[2], and Marco Montali[1]

[1]Free University of Bozen-Bolzano, Italy
[2]University of Tartu, Estonia

The Decision Model and Notation (DMN) is a recent OMG standard for the elicitation and representation of decision models, and for managing their interconnection with business processes, separating decision and control-flow logic. The standard is already receiving widespread adoption in the industry, and an increasing number of tools and techniques are being developed to assist users in modeling, checking, and applying DMN models. DMN builds on the notion of a *decision table*, which consists of columns representing the inputs and outputs of a decision, and rows denoting rules. Each rule is a conjunction of basic expressions, which in our case are captured in a language known as S-FEEL, which is also part of the DMN standard itself.

According to the standard, DMN models work under the assumption of complete information, and do not support integration with background domain knowledge. In this paper, we overcome this limitation, by proposing a combined framework, which we call *Semantic DMN*, that is based on *decision knowledge bases* (DKBs). In a DKB, decisions are modeled in DMN, and background domain knowledge is captured by means of an ontology expressed in multi-sorted first-order logic. The different sorts are used to seamlessly integrate abstract domain objects with the data values belonging to the concrete domains used in the DMN rules (such as strings, integers, and reals).

For the enriched setting of Semantic DMN, we provide a logic-based semantics, and we formalize how the different DMN reasoning tasks that have been introduced in the literature can be lifted to DKBs. We then approach the problem of actually reasoning on DKBs, and of devising effective algorithms for the different reasoning tasks captured by our formalization. For this purpose, we need to put restrictions on how to express background knowledge, and we consider the significant case where such knowledge is formulated in terms of an ontology expressed in a description logic (DL) equipped with *datatypes*. In such a DL, besides the domain of abstract objects, one can refer to concrete domains of data values (such as strings, integers, and reals) accessed through functional relations, and one can express conditions on such values by making use of *unary* predicates over the concrete domains. Specifically, we prove that for the case where the DL ontology is epressed in $\mathcal{ALC}(\mathfrak{D})$, i.e., $\mathcal{ALC}$ extended with multiple datatypes, all reasoning tasks can be actually decided in ExpTime.

We show the effectiveness of our framework by considering a case study in maritime security, arguing that our approach facilitates modularity and separation of concerns.

* The full version of this paper has been accepted for publication in the Proceedings of the 2017 International Joint Conference on Rules and Reasoning (RuleML+RR).