Scientific and Technical Report 2017

on the implementation of the project PN-III-P2-2.1-PED-2016-0494

December 4, 2017

The implementation of the first stage of the project *Formal Verification of Reconfigurable Systems* was performed between 17 August 2017 and 31 December 2017. The technical results for 2017 accord with the contract and are follows:

1. the definition of the language *H-spec*; this has been compiled in the document [CD17] (available also from the web page of the project);
2. a library of examples to illustrate and test the main features of *H-spec* that is available at https://ontohub.org/forver.

These are treated in separate sections as follows.

1 The definition of the specification langugage *H-spec*

In this section we provide an overview of the main features of the definition of *H-spec*, the full technical document may be read at the URL indicated above.

1.1 The features of *H-spec*

A *H-spec* document consists of either specification of new hybrid logics or specification of reconfigurable systems in a hybrid logic.

Hybrid logics. We introduce a declarative syntax for specifying the parameters of the generic hybridization method. They are:

- the name of the new hybridized logic,
- the name of the logic being hybridzed,
- the kinds of symbols allowed to appear in a quantification,
- the constraints made on the models of the logic.

We make the assumption that a library of possible constraints for each base logic is available, and the user must choose among the constraints of the specified base logic when making a new hybridization. Two types of constraints are possible:

- on the accessibility relations:
  - reflexive: $(\forall w)R(w,w)$
  - symmetric: $(\forall w_1,w_2)R(w_1,w_2) \implies R(w_2,w_1)$
  - transitive: $(\forall w_1,w_2,w_3)R(w_1,w_2) \wedge R(w_2,w_3) \implies R(w_1,w_3)$
  - serial: $(\forall w_1)(\exists w_2)R(w_1,w_2)$
  - Euclidean: $(\forall w_1,w_2,w_3)R(w_1,w_2) \wedge R(w_1,w_3) \implies R(w_2,w_3)$
  - functional: $(\forall w_1)(\exists! w_2)R(w_1,w_2)$
  - linear: $(\forall w_1,w_2,w_3)(R(w_1,w_2) \wedge R(w_1,w_3)) \implies (R(w_2,w_3) \vee R(w_3,w_2) \vee @_{w_2}w_3)$
  - total: $(\forall w_1,w_2)R(w_1,w_2) \vee R(w_2,w_1)$

where w, w_1, w_2, w_3 are worlds and R is the accessibility relation on worlds.

- on the local models:
  - the set of worlds of each local model is the same
  - the nominals are interpreted in the same way in each local model
  - symbols of some kind are interpreted in the same way in each local model
  - partial functions are defined on the same elements in each local model

Alternatively, one can add further semantic constraints or other kinds of symbols used in quantifications on an existing hybridized logic.

***H-spec* specifications.** *H-spec* basic specifications over a hybrid logic have three parts:

- the name of the hybrid logic
- the name of a specification in the base logic of the hybridized logic, containing the data part of the specification
- a configuration part, consisting of declarations of state names and events and sentences in the hybrid logic.

For structuring, we will make use of the Distributed Ontology, modeling and specification Language DOL [Mos+15]. DOL is a meta-language for structuring of ontologies, specifications and MDE models, independent of the formalism used at the basic level. A DOL structured specification can contain parts written in different logics. In our setting, we will only make use of homogeneous structuring, where all specifications appearing in a structured specifications are in the same logic.

1.2 The contents of the actual *H-spec* definition

1. The syntax of *H-spec* is given in BNF notation. This is structured in two parts, one for abstract syntax and another one for concrete syntax each of them being sub-structured in the following parts:
   a) the syntax for documents;
   b) the syntax for structured specifications; and
   c) the syntax for unstructured specifications.
2. *H-spec* semantics which contains the following parts:
   a) an overview of the mathematical foundations;
   b) the semantics for documents;
   c) the semantics for structured specifications;
   d) the semantics for unstructured specifications.

2 The initial library of specification examples with *H-spec*

This library is meant to illustrate and test the main features of *H-spec*. It is only the first stage of the library of examples as this will be further developed in the second stage of the project when the other objectives of the project will unfold. The library contains the following:

- A simple “hello-world” beginning example which features a system with only three states, all named. In two of them a proposition is true.

  The logic of the specification is an unconstrained hybridization of propositional logic.
- A refinement of the previous example which features a fully specified accessibility relation, i.e. no transitions other than the specified one are possible in a model.
- A specification of a reconfigurable calculator for natural numbers with a binary operation that in one state is sum and in the other one is multiplication.

  This uses a hybridization of first-order logic where the semantic constraint is given by user-defined rigid sorts and operations.
- A specification of a generation system with two modes of operation, each with two sub-modes: generate odd or even numbers or generate lowcase or uppercase letters.

This features a double layered hybridization where the semantic constraints consist of shared local worlds and shared nominals.

- A specification of a plastic buffer with two operation modes: in one of them it behaves as a queue, in the other as a stack. This is a benchmark example that was proposed in the paper [DM16] that lies the scientific foundations for the project.

  The specification features a rather sophisticated hybridization process that involves an uncommon semantic constraint, namely that partial function symbols must be interpreted in the local worlds as partial functions with the same domain.

- A specification of an ATM system with modes for PIN validation and operations on the account. The specification is deliberately faulty in its functionality and will be used to illustrate proofs of unintended consequences: Through validating the PIN of a card, we can operate on every account instead of only on the current account.

  The specification features a doubled layered hybridisation that models sub-states.

References

[CD17] Mihai Codescu and Răzvan Diaconescu. *H-spec language definition*. 2017.

[DM16] Răzvan Diaconescu and Alexandre Madeira. “Encoding Hybridized Institutions into First Order Logic”. In: *Mathematical Structures in Computer Science* 26 (2016), pp. 745–788.

[Mos+15] Till Mossakowski et al. “The Distributed Ontology, Modeling and Specification Language – DOL”. In: *The Road to Universal Logic*. Ed. by Buchsbaum A. Koslow A. Birkhauser, Cham, 2015.

Project Director,
Răzvan Diaconescu