The Combined SAVE Design and Verification Flow

1. Participants:

KTH: Wenbiao Wu, Axel Jantsch
LiU: Luis Alejandro Cortés, Petru Eles
Saab: Rolf Hälleberg
Cybertech: Lars Ödman

2. Moderators:

Wenbiao Wu and Luis Alejandro Cortés

3. Purpose:

The discussion focused on the search of a common design flow that integrates the specification and verification methodologies developed at ESDlab-KTH and ESLAB-LiU respectively, and suitable at the same time to Saab's industrial case.

4. Discussion

4.1. Design Flow:

The proposed design flow is the one shown in Figure 1. It starts with a Haskell description of the system and some transformations may be performed at this level (still within the Haskell-specification context). At this point validation is to be performed using simulation.

After the initial system validation, the Haskell description is translated into the PRES+ formalism. This allows the representation of the system to be verified using formal methods by model-checking the model against a set of required properties expressed by temporal logics. The kind of verification performed at this stage refers to check safety (no dangerous states are ever reached), liveness (absence of deadlocks, so that the functionality may eventually be completed), and timing properties. This formal verification approach does NOT deal with the functional correctness of the system in terms of the expected output values. From the result of the model checking procedure there could be feedback to the Haskell description. Also the PRES+ model can be simulated by using the tool SimPRES in order to study (validate) the functionality of the system in terms of correct output values.

In the next step of the proposed design flow, architecture decisions are taken and these must reflected in new details of the system, that is, the mapped PRES+ model. Also, at this point both formal verification and validation by simulation can be performed. In case that the system does not fulfill its required properties or does not have the expected behavior, it is possible to go back to previous phases of the design process.

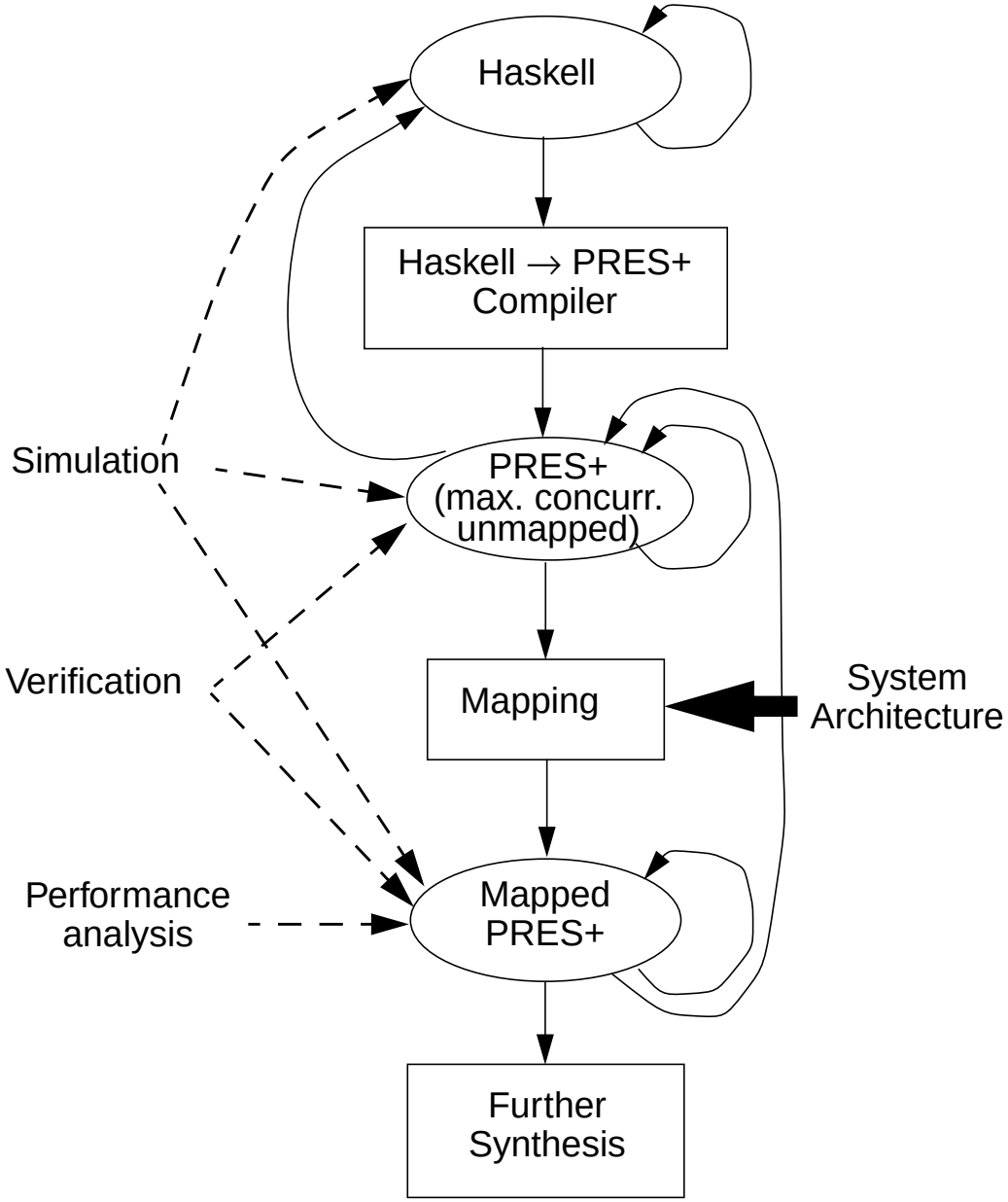

Figure 1. System Design Flow

Having the mapped PRES+ model it is possible to continue further down in the design process. Many approaches have been proposed in the literature for later stages.

4.2. Haskell -> PRES+ Compiler:

It was agreed that the first step in the cooperation between ESDlab-KTH and ESLAB-LiU should be the Haskell-> PRES+ translation. In order to do so, a compiler is the first tool needed. Therefore, the initial efforts of a common design flow will point towards the development of such a tool.

The first part of this subtask will be defining the basic translation procedures. We agreed on having PRES+ structures representing the skeletons already defined at ESDlab-KTH. W. Wu and L. A. Cortés will jointly work on this task. As a result a report should be generated by late January, due on January 31, 2000. In this report, a toy example will be studied and a manual translation will be performed, in order to convince ourselves that the compilation Haskell -> PRES+ is feasible.

In the mean time a M.Sc. (ex-jobb) project will be defined with the main objective of developing such a compiler. The idea is to post an announcement as soon as possible in KTH, LiU, and Saab.

5. M.Sc. Project:

(Note: This item was addressed during the discussion, but this first draft was elaborated afterwards).

KTH/LiU/Saab Ex-jobb

Title: Translating Haskell descriptions into PRES+, an extended version of Petri nets

Background: ESDlab at KTH, Stockholm, ESLAB at LiU, Linköping and Saab Dynamics are jointly developing a formal approach to specification, implementation and verification of heterogeneous electronic systems in the frame of the SAVE Project. The objective of the SAVE Project is to devise improved solutions and methods for high level electronic system specification, verification and refinement by using formal methods.

Project Description: In the current design flow of the SAVE project, the system is initially specified in Haskell, a purely functional language. In order to allow the specification to be analyzed and verified using model checking techniques, it is translated into the PRES+ formalism. Thus the system may be model-checked against a set of required properties expressed in temporal logics.

The main task of this project will be the development of a compiler that translates the system Haskell description into PRES+. Such a translation procedure shall include the following aspects:
1. Parse the Haskell specification, which is composed of skeletons and elementary functions.
2. Translate the basic skeleton Haskell structures into PRES+ structures.
3. Translate the elementary functions into the PRES+ representation.

4. Build up a PRES+ net representing the system.

It is appreciated if the candidate has experience in one or more of the following areas:
- Functional programming languages, especially Haskell.
- Petri nets.
- Compiler techniques.