Review of Ph.D. thesis

Ing. Tomáš Richta

Dynamic software architectures for distributed embedded control systems

This Ph.D. thesis deals with a topic that belongs to the area of distributed embedded control systems. As a typical example of such system we can imagine something like for example control systems for home automation. Such systems often consist of many different kinds of devices that must cooperate together. The computing power of such devices can be rather diverse, ranging from devices with very limited capabilities (like different kinds of sensors and actuators) to much more powerful kinds of computers. The systems consisting of such devices could be quite heterogeneous and their structure could change dynamically with the changing conditions. One of the main challenges in the design of such systems is the problem how to coordinate and reconfigure all the devices and how to ensure that they cooperate correctly as intended.

The thesis proposes one particular possibility how to solve this more general problem. The basic idea is to use a common system based on a particular variant of high-level hierarchical Petri nets called reference Petri nets. These Petri nets are used here not only for the purpose of modelling and simulation but instead they form in fact a whole platform for the implementation of the system — each device runs an interpreter that directly executes the given Petri nets and performs the corresponding actions. The Petri nets are used in a hierarchical way that allows to represent the behaviour of the system on several layers of abstraction. They are used also for the description and the execution for the way how a current configuration and a current code (in a form of a Petri net) is distributed and loaded to the devices.

The main topic of the thesis is a description of such proposed system and of a prototype implementation of it, together with a methodology for the construction of these kinds of systems. The particular system described in the thesis consists of several parts. One of them is called Petri Nets Operating System (PNOS). It deals with distribution, installation and loading of the code on the individual devices. This code is represented in a form of Petri nets. These Petri nets are encoded in a bytecode, called Petri Nets Byte Code (PNBC). This bytecode is used for distribution of the given Petri nets and for execution by a virtual machine called PNVM. It can be also translated into C and compiled directly into machine code that can be run by embedded devices with very limited capabilities.

The main contribution of the thesis is the description of this system and its implementation. It also describes some examples of its use.

The thesis is structured as follows. Chapter 1 is an introduction. Chapter 2 gives an overview of a related work. Chapter 3 contains description of several kinds of Petri nets and related formalisms like Workflow nets. Chapter 4 describes the overall structure of the proposed design of the system and Chapter 5 its implementation. Chapter 6 describes two use-cases — home automation and maritime logistics. Very short Chapter 7 contains some experimental measurements and Chapter 8 is a conclusion.

The topic of the thesis is an important and interesting problem and the proposed solution certainly has some merit. Also the number and quality of publications by the author seem to be sufficient for a Ph.D. thesis. However, in my opinion, the quality of the text of the thesis is not as good as it could be.

On one hand, the text is rather talkative, many general informal things and ideas are repeated several times, but without giving enough details. On the other hand, it sometimes presents many technical details without giving a clear global picture.

In Chapter 3 (and partly in Chapter 4), many formal definitions are introduced but they seem to be on one hand incomplete, and on the other hand unnecessary, since the remaining text almost does not use these formal definitions. The way how mathematical notation is used in the thesis is sometimes confusing.

Many examples given in the text are in fact not very informative and are not described in sufficient detail. In my opinion, it is not very useful to give long listings of bytecode (like Listing 5.3 on pages 56–58) without explaining their meaning and without explaining how such listings are related to corresponding pictures (like Figure 5.2). It is also often not clear what exactly the author tries to illustrate by such examples.

Chapter 6 presents two examples of the application of the system. However these descriptions are rather high-level with several illustrating examples representing some parts. To understand these parts was quite hard for me, since the precise details (e.g., what individual places and transitions represent, etc.) are not much discussed in the text.

Experimental results presented in Chapter 7 are, in my opinion, not very informative. Again, the author has not provided enough details of what was actually measured. There is no comparison with anything, so it is hard to say if the presented values are good or bad, or if they are an improvement with respect to something. In fact, it is not obvious what should be illustrated by these numbers and graphs.

The text contains quite a lot of typos and the quality of English is, in my opinion, not excellent.

Detailed comments:

- pp. 5: What is meant by "parts of the system specification migrate in the form of tokens"?
- pp. 7: There is said that "Unfortunately, it is usually very difficult to fix code mistakes in models and to change models if the code changes too". I think this needs some more explanation. I do not see why fixing bugs in a code is related to the corresponding model. Maybe it depends on the type of these bugs?
- pp. 24 — Definition 3.2: Why is a semicolon used in one place and commas in the other in the definition of HLPN? Has it some special meaning?
- pp. 24: The notion of *transition modes* is not explained in the text.
- pp. 24: The notation $\mu PLACE$ is not explained in the text. Does it represent the set of all (finite) multisets whose elements are from the set $PLACE$? (A similar notation is used on several other places in the text.)
- pp. 25: The definition of when a transition is enabled is not very clear. I suppose that for example in the definition of $Pre(T_\mu)$, T_μ is a (finite) multiset of elements of $TRANS$, $T_\mu(tr)$ represents the number of occurrences of tr in T_μ, multiplication of $Pre(tr)$ means taking the union of the given number of copies of the given multiset, using "$\leq$" means inclusion of multisets, etc. Things like that are not clear from the text.
- pp. 25: How are domains, types and elements of these types used in these definitions? Do they play any role in the transitions?
- pp. 25: How is the definition of a HLPN graph related to the previous formal definition of HLPN? It does not seem to use the formal notation introduced before. The correspondence is not very clear.

- pp. 27: The following sentence makes no sense: "In contrary to classical hierarchical Petri Nets, the static transition logic of the net is not refined, but instead of that a system state dynamics."

- pp. 27 — Definition 3.3: Maybe it would be better to say "a tuple" instead of "a n-tuple", since there is no n defined here. A similar use of "n-tuple" occurs also in other definitions in the text.

- pp. 27 — Why the definition of *elementary net system* uses B and E for places and transitions? How does this definition differ from the previous definitions of Petri nets?

- pp. 28 — Definition 3.4: This definition and the following description of the behaviour of *elementary object systems* is not very clear. For example, what is the meaning of the arc type function *type*? What $\hat{T}$ and N denote in $\hat{l} : \hat{T} \to (N \to C)$?

- pp. 29: The following sentences occur on two different places on this page:

  "... is very similar to the BPMN workflow models, so it might be easily adopted by business process modeling domain experts. For that reason we decided to use the Aalst's YAWL notation [3] and Workflow Petri Nets formalism [2] in the early beginning of the system construction process."

  (In the second occurrence some words were slightly changed.)

- pp. 30: What is the meaning of '$\not\to$' notation ? Does it denote a partial function? Then maybe something like '$\rightharpoonup$' would be a better alternative.

- pp. 30: Does $\mathbb{P}$ denote the powerset of a set?

- pp. 30 — Definition 3.8: What does map_{N_1} denote? Is it somehow related to *map*?

- pp. 33: "... which represents the main theoretical structure together with full-featured operational semantics for Turing-complete system definition." — Maybe, this full-featured operational semantics could be described in the text or at least some references could be provided where it could be found.

- pp. 34: It would be useful to have a more detailed informal description of Figure 4.1 in the text. For example, what different colors of arrows denote?

- pp. 34: Maybe also Figure 4.2 should be referenced in the following sentence: "The multi-layered nature of the system and responsibilities of particular levels are described in Figure 4.4."

- pp. 36: "obviously connected to radio" — Is some kind of a wireless connection meant here?

- pp. 37: The title in Listing 4.1 is misleading. Maybe such one line with a message could be written directly in a text where it is discussed (maybe on a separate line). Also description of this message in the text is not very clear.

- pp. 38: "Particular data types should be described in the terms dictionary" — I do not understand what "terms dictionary" means here. Maybe, it should be described in more detail.

- pp. 38 — Definition 4.1: Symbols i, o, T, and P_{EWF} probably refer to components of EWF. Maybe it should be described in the definition.
- pp. 39 — Definition 4.3: Definition of T^{WS} is not very clear. Maybe, it could be explained in more detail.
- pp. 41: Do words "double-sided arcs" mean a pair of arcs in opposite directions?
- pp. 41: "... translated into the PNBC pseudo-code" — probably, the word "bytecode" should be used instead of "pseudo-code".
- pp. 46 — Figure 4.10: It is not clear what this picture depicts, it is not explained in the text. It is also quite hard to read the small text in this picture.
- pp. 50: The following sentences are repeated twice:

  "In this representation, numbers are represented as text and also some spaces and line breaks are added. This means that the contents of the code memory is a bit more condensed. Each byte of the code is either an instruction for PNVM, or data."
- pp. 54: "Translation of Petri Nets to the target platform code assumes a set of simplifications within the Petri Nets formalism semantics that were necessary for the compliance with suggested target platform." — Maybe, some more detailed description of these simplifications would be useful, together with some discussion why they were necessary.
- pp. 56–58: There is the long Listing 5.3. I would appreciate some more detailed description of it and of the used syntax, and how it is related to net in Figure 5.2. Maybe, some simpler example with more detailed explanation would be more useful than this long listing without explanation.
- pp. 59: "PNVM also maintains a calendar for delayed postconditions." — What it exactly means? What is a calendar here?
- pp. 61: What is the difference between symbol identifier (`S`) and variable identifier (`V`)?
- pp. 65: "This part of the work is partially based based on work of one of our students, who translated original PNVM previously implemented in Smalltalk into the C code [64]." — It should be probably described in more detail, what parts are results of the work of the given student, and what parts are results of the work of the author.
- pp. 66: "the string objects in Lightweight Pattern-like style" — What is Lightweight Pattern? (Maybe, some reference would be appropriate.)
- pp. 67: "Later the interpret construction had undergo some more improvements that were not propagated into the original implementation yet." — Maybe, these improvements could be described in more detail. It is also not clear what is meant by the original implementation here.
- pp. 68: "Each step is a finite set of elementary operations ..." — maybe, something like "Each step consists of a finite set of elementary operations" would be more clear. It could be also mentioned here, that these elementary operations are described in the following subsection.

- pp. 68: The garbage collection should be described in more detail. The given description is not clear at all.
- pp. 68: "The search for unification during the execution is simmilar to Prolog [23]." — What exactly is similar to Prolog? The kind of unification needed here seems to be much simpler than the process of unification in Prolog. In fact, this unification needed here is not described in sufficient detail. Some more formal description would be suitable here.
- pp. 70: "all the discovered and tested application scenarios" — maybe some other word than "discovered" should be used here.
- pp. 76: What is a "swimline"? Is it some technical term? Maybe it should be a "swimlane"?

Typos:

(The list is probably not exhaustive.)

- pp. 5: "environments that dynamically changes"
- pp. 5: "idea prototype implementation"
- pp. 7: "the pure formal models (e.g., Petri nets, calculi, etc.) allows"
- pp. 13: "the system to flexible reflect"
- pp. 13: "Some of the nodes nodes"
- pp. 13: "the overall bussiness logic ... manifest itself"
- pp. 15: "All the presented formalisms is able to describe"
- pp. 16: "updated with with minimal delay"
- pp. 22: "languages tobe able to specific the problem domain"
- pp. 23: "Petri nets ... is specific mathematics language"
- pp. 24: "$P(P \cap T = \emptyset)$" — a space is missing
- pp. 27: "communications uses arguments"
- pp. 27: "In contrary" should be probably "On contrary"
- pp. 42: "*: input*" and "*: output*" — other type of font should be used to make it consistent with the rest of the text. Also there should not be spaces between the colon and the following identifier.
- pp. 58: "its operations set" should be probably "the set of its operations"
- pp. 60: "The interpretation details follows."
- pp. 61: Instead of "non/terminal", it would be better to write "nonterminal" or "non-terminal".
- pp. 63: "$< 0, 2^16 - 1 >$" should be "$< 0, 2^{16} - 1 >$"

- pp. 63: “Non-terminals ending with `id` represents”
- pp. 65: “This part of the work is partially based based on work”
- pp. 65: “we need to stat that”
- pp. 66: the symbol for division (probably ‘`/`’) is missing in the table
- pp. 66: “the rest after the division” — should be “the remainder”
- pp. 67: Instead of the word “interpret”, the word “interpreter” should be used. (The word “interpret” is a verb.)
- pp. 68: “All the event that execution time is older . . . ”
- pp. 68: “all the blocks following the removed template is moved according to the template size”
- pp. 68: “the firs possible choice”
- pp. 68: “simmilar” should be “similar”
- pp. 76: “I n our example” (an extra space)

Overall evaluation

In my opinion, the topic of the thesis is interesting and relevant, and the results presented there were presented on several conferences, so in this respect the requirements for a Ph.D. thesis are satisfied. The presented results have some merit and potential for a successful Ph.D. thesis but the overall quality of the text and the way how the given results are presented is not very good.

However, despite all these deficiencies, I think the thesis is defensible, so **I do recommend the Ph.D. thesis for the defense**.

Ostrava, August 2, 2021

doc. Ing. Zdeněk Sawa, Ph.D.
Department of Computer Science,
FEI, VŠB – Technical University of Ostrava