Modelling and Simulation of Parallel Systems Using CCS and Petri Nets: Major Concepts

Milan Urbášek[*]
urbasek@dcse.fee.vutbr.cz

Milan Češka[**]
ceska@dcse.fee.vutbr.cz

Abstract

The main aim of this contribution is to point on major concepts in modelling, simulation and analysis of parallel systems when Petri nets and process algebras of Milner's type, denoted CCS, are used. There exist a huge class of systems, whose behaviour may be defined as Petri nets and/or as CCS. Although both concepts are widely used for parallel applications in modelling, there is a difference how these models are used later in simulation and analysis of such systems. The first idea of both abstractions is to describe parallel systems; the analysis methods based on these ideas followed different paths. It sounds very promising to bring both theories together as far as possible, and to exploit advantages of both theories in order to get more detailed description of concurrent applications.

1. Introduction

Modelling and simulation of concurrent systems is the area of modelling and simulation which has been strongly investigated for many years and by many researchers. The natural complexity of parallel systems can be of various kind. Diverse models of parallel systems were introduced, e.g. Petri nets and process algebras. Many approaches can be found in literature. The questions of the potential usage in applications, in simulation, and questions of analysis of such models go hand in hand with new models introduced. This happened also in the case of Petri nets and CCS. Petri nets and CCS are used as models of concurrent systems. A surprise arises when the analysis methods of these models are compared. Petri nets have attracted systems and control engineers in the main, who have been concerned to prove liveness and identify dynamical invariants of systems designs. Process algebras have interested computer scientist and software developers with some interest in liveness and invariance but a greater interest in comparing the behaviours of different models. A possible source of the greater interests is the refinement that goes on during a software development project. Beginning with a coarse model of the system to be developed, finer and finer models are developed until a stage of refinement at code level is reached. There is a need to check that refinements satisfy bisimulation constraints [see, e.g. MIL 89].

Another difference is created by the main idea behind these concepts, as described later in detail. CCS system is described as several independent agents communicating to each other. Petri nets on the other side are a natural way for describing systems whose actions depends on some external and internal conditions in distributed environment. Distinguishing of independent agents in P/T Petri nets, however, is not an easy problem in general.

[*] Department of Computer Science and Engineering, FEECS, Brno University of Technology (VUT), Božetěchova 2, 612 66 Brno – Královo Pole.
[**] Prof., RNDr., CSc., Department of Computer Science and Engineering, FEECS, Brno University of Technology (VUT), Božetěchova 2, 612 66 Brno – Královo Pole.

2. CCS

The concept of Milner's process algebras, also known as CCS, was introduced in Milner's book Communication and Concurrency; see [MIL 89]. Since publishing a lot of authors have investigated behaviour of systems based on CCS description.

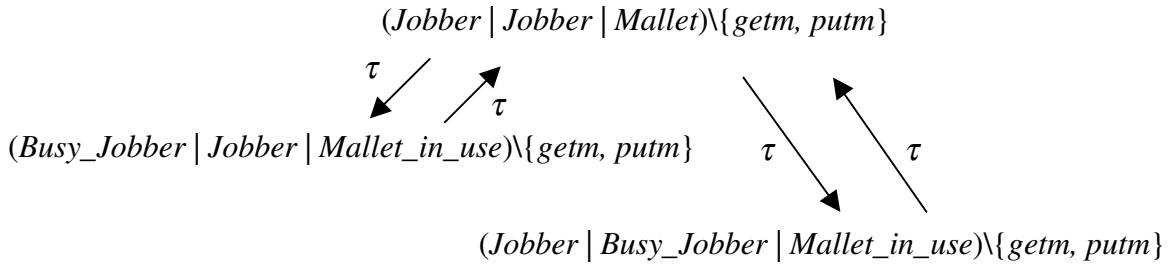
As a formal system, CCS reflects principal concepts we can find in parallel systems. A parallel system is taken to be a system having several more or less autonomous parts, such that there is a possibility of communication between those parts. When CCS are used, a system is exactly specified as a system of independent communicating agents. This idea is a natural way, how parallel systems are designed in the real world.

Before presenting main ideas behind the modelling with CCS, we will discuss a small example.

Jobshop - shared resource modelling. An example of a typical system in the Milner's book [MIL 89] is the workshop. There are two jobbers doing their work, but they need a mallet for their job and at the time there is only one mallet for both of them. This is a typical example of a shared resource. The way for description of shared resources in CCS is handshaking with restriction.

$$
\begin{aligned}
Jobshop &:= (Jobber \mid Jobber \mid Mallet) \setminus \{getm, putm\} \\
Jobber &:= \underline{getm}.Busy_Jobber \\
Busy_Jobber &:= \overline{putm}.Jobber \\
Mallet &:= \overline{getm}.Mallet_in_use \\
Mallet_in_use &:= putm.Mallet
\end{aligned}
$$

The equivalent behaviour expressed as a labelled transition system is shown below.



There are two dominant ideas shown in the example. The first one is *handshaking*. Handshaking is expressed as the possibility of performance of the silent action τ in the composition of two agents, what is the way of synchronisation of behaviour, as expressed, for example, in the transition $(Jobber \mid Mallet) \xrightarrow{\tau} (Busy_Jobber \mid Mallet_in_use)$. The idea of handshaking is that two agents perform a composite action, changing their states simultaneously. The another important idea is *restriction*. This idea allows to hide some actions, which the system can perform in order to internalise these action. These actions become invisible to the external viewer and are fully treated as internal actions. Such action is denoted as the silent action.

Previously mentioned concepts of specification of parallel behaviour work together as in *Jobshop* example. This is exactly what is pointed out by people interested in the research of CCS based models.

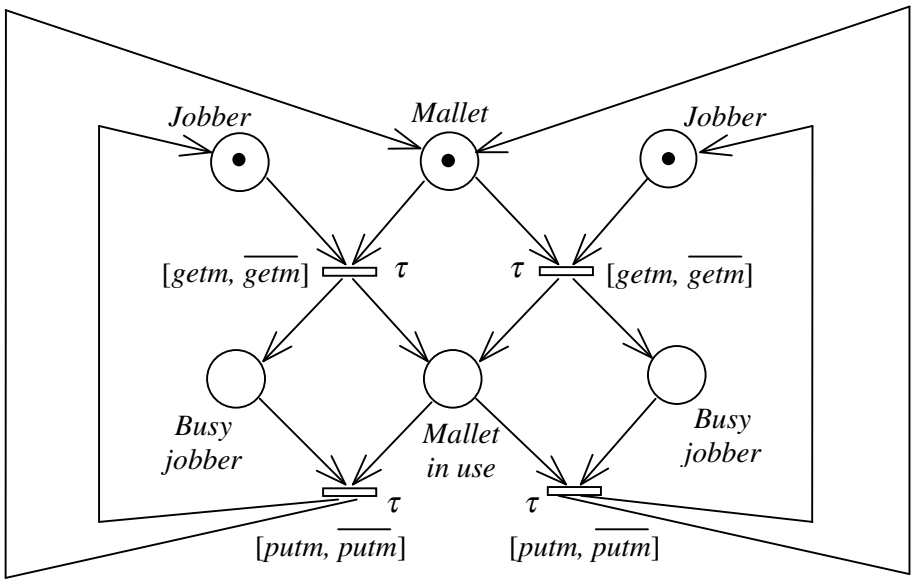
There is something more, what can be taken as an advantage of CCS models. This is a transparent expression of parallel parts. These parts are naturally expressed as members of composition operator '|'.

The theory of CCS is deeply investigated in following books [MIL 89], [FEN 96].

3. Petri nets

Since their introduction, Petri nets have been used to model many different systems and various classes of net have been developed. The standard class is the class of P/T Petri nets, but there are classes of coloured, timed and probabilistic nets introduced by many authors as extensions or restrictions of the standard class. The growing importance of Petri nets is related to their usefulness in modelling concurrent systems, distributed processing, communicating systems, all of whose dynamical behaviour can be easily expressed as movement of tokens in Petri nets.

Before going further, we will define the *Jobshop* system as a Petri net with equivalent behaviour. For such description we need to extend the traditional definition of P/T Petri net to the Petri nets, where each transition is labeled with a name of an event carried out, when the transition is fired.



Although the modelled system remained the same, the model has changed. There are another concepts, which are more naturally expressed as a Petri net. The first one is, how the enabled transition is defined. This is suitable for modelling of events, which are triggered when several conditions are satisfied. This is very important feature in the case of distributed system modelling. The another concept of big importance is that a global state of whole system is made up from the local states of every place. Every local change changes the system global state without affection the behaviour of independent parts of the system.

The usefulness of Petri nets in modelling and simulation has been discussed in many papers and books, e.g [ČEŠ 95], [PET 81]. Many times the advantage of graphical representation of the Petri nets has been mentioned.

There is an easy way, how to include the encapsulation of subsystems into a bigger system in Petri nets. This concept is formulated as hierarchical Petri nets. Many systems in

the real world are on this basis. Hierarchy is another concept, which can be found in the nature. To the theory of Petri nets were devoted many books, e.g. [PET 81].

4. Modelling power

For modelling purposes, the question of modelling power of a chosen model is important. The long time of investigation revealed the answer to this problem in both cases, for CCS and also for Petri nets.

In the case of Petri nets, the modelling power of C/E Petri nets is equivalent to the modelling power of finite automata. P/T Petri nets are stronger model, but to achieve a Turing power, it is necessary to make an extension of P/T Petri net class, as for example by adding inhibitor arcs. The detailed description can be found in [PET 81].

On the other side, the situation in the case of CCS is not so easy as shown in Taubner's dissertation [TAU 89]. Taubner showed that modelling power of CCS is equivalent to the modelling power of Turing machine. However, this power is the result of extraordinary cases, which are possible to specify according to the definition of CCS expressions.

The original idea behind CCS as presented in Milner's work [MIL 89] is very transparent idea. But the definition of CCS expressions and inference rules allow more. It is possible to express extraordinary cases in CCS behaviour. Although these boundary cases extend the modelling power to Turing power, the usefulness of such expressions is extremely low, see [TAU 89], [URB 99].

5. Simulation tools

Petri nets and process algebras such as CCS [MIL 89] are used to model the behaviour of non-deterministic systems. Both have an action semantics expressed in the form of state transition behaviour, and both have inspired the development of support tools that assist the analysis of model behaviour. Some examples are The Concurrency Workbench [PAR 89] for CCS, PESIM [ČEŠ 95] for Petri nets.

The analysis methods can be applied algorithmically and a lot of systems deal with a analysis and simulation of Petri nets and CCS.

The very important feature of Petri nets is the possibility of graphical representation of the model. This allows easier and intuitive comprehension and the graphic outline is, of course, more suitable for presentation. This is nice feature, but in general, the development of the graphical editor and suitable graphical user interface is not a trivial problem.

In order to allow better simulation and model description, several extensions and adaptations of ordinary models have been defined.

For Petri nets, hierarchical Petri nets have been defined. As a strong formalism, hierarchical Petri nets are the basis for object oriented Petri nets. For practical purposes, the timed and time extensions of Petri nets have been introduced. Very important definition is the extension to the stochastic Petri nets, which allows the analysis of probabilistic behaviour.

In the similar way extensions of CCS have been introduced. The timed CCS have been defined. The stochastic concept is not well investigated yetin CCS environment.

The idea of handshaking is the leading idea in CCS concept. Standard CCS allows the two-way handshaking only. To allow multi-way handshaking, new classes of CCS have been defined. The concept of restriction has been also changed, the inference rules for CCS behaviour too, see [FEN 96].

Software systems based on CCS are mainly focused on the idea of proving behavioural and observational equivalence of two models described in CCS formalism. The main advantage of CCS formalism is that description is more formal in mathematical way, due to the expressiveness of system as algebraic definitions using CCS combinators.

6. Analysis

There are, however, some differences between the communities which employ these ways of approaching concurrency. Petri nets have attracted systems and control engineers in the main, who have been concerned to prove liveness and identify dynamical invariants of systems designs. Process algebras have interested computer scientist and software developers with some interest in liveness and invariance but a greater interest in comparing the behaviours of different models.

In general, Petri net analysis methods were focused on language equivalence. Petri nets were investigated as language acceptors and their relationship to Chomsky hierarchy of formal languages.

Theory of CCS was focused on bisimulation relations between two systems. The behavioural and observational equivalence and congruence are mainly investigated throughout the theory of CCS.

As large systems are increasingly build as society of communicating agents, the CCS community in particular has developed ways of combining agents to hide or silence their internal handshaking communications and focus on the external behaviour of 'social' system. The Petri net community has not yet moved towards bisimulation studies nor to silent action. The purpose of this work is to start preparing the ground for the introduction of such abstractions into theories of Petri net state dynamics.

Both theories CCS and Petri nets are deeply developed and are still being enriched by many researchers definitions of both models we presented and some types of extensions and restrictions were made in order to suit to applications better.

The biggest similarity is that both formal models are used for concurrency modelling, both are used in simulation and verification of systems. On the other side, as said before, the development of formal methods amongst these models followed different paths.

A big contribution is made to solve hard problems of the analysis of a modelled system, its verification and safety or security proving. CCS like Petri nets can describe the same system, but from this point the analysis is different. It would be nice bring these formal methods of developing a system closer together and use advantages of both in order to say more about the system at the level of a proposal.

7. Unification of CCS and Petri nets

As discussed above, there are common points throughout theories of CCS and Petri nets. There are, however, several different points between major concepts. As mentioned before, it sounds promising to join both theories together in order to obtain deeper insight into theory of concurrent systems in general.

The unification has been approached from categorical point of view. Category theory is formal mathematical theory with strong background in universal algebra. The description of new formalism in terms of category theory satisfies strong conditions, which are necessary from mathematical point of view, when new concept is introduced.

Conversely, it is necessary to find more exact definition of Petri nets and CCS, which is closer to algebraic point of view.

To bring all theories together, it is useful to concentrate on similarities between CCS and Petri nets. Both have semantics based on transition behaviour. There is an obvious attempt to convert one model into another.

The view of translation of CCS into Petri net is nothing new in general. We can find many papers and presentations dealing with such subject. One of the most complex is the dissertation of Dirk Taubner [TAU 89], discussing several classes of Petri nets. However, the possible power of P/T Petri net class is not used. The syntax-driven construction for safe Petri

nets is restrictive on some types of CCS expressions, which bring problems during the automatic translation, but still can be treated without loss of generality.

Taubner's construction generates a Petri nets, whose graph of markings is isomorphic to the transition graph of CCS. The isomorphism is defined as usual isomorphism of graphs.

If we relax the obvious definition of graph isomorphism, one can define isomorphism of two algebraic structures in different way. This is natural notion in category theory.

The notion of morphisms in usual functional sense is discussed, for example, in [JOY 94], [MES 90], [WIN 84].

Different approaches to theory of Petri nets and CCS is under development. In [URB 99] there is made an attempt to define morphisms as bisimulation relations, which can bring these theories closer together.

8. Conclusion

This contribution didn't bring new ideas at all. The aim is to point on the relationship between two models of concurrency, namely Petri nets and CCS, and describe the abstraction, which has inspired authors to concern on similar problems in both theories. An attempt was made to discuss similarities, differences and alterations in approaches in both theories on a few pages of this report. The main idea has been adumbrated only and it is a big source of inspiration for a future work. Although some foundations have been laid, there are many questions still opened. One of the most important ones is the question of algebraic representation of both models in order to suit better to the aims of category theory. There are many classes of models even in Petri nets and CCS itself, e.g. initialised Petri nets, non-initialised and so on. These are reasons why the research in this area is not trivial and needs a lot of care.

There are many other problems that arise from the attempt of unification CCS and Petri net using category theory. However, we can see from many papers that category approach to Petri nets and CCS is useful not only for description of Petri nets from a different point of view but also for understanding of deeper relationships to other models and analysis techniques. Although this work laid just foundations, one can see that it pays to spend time in designing and investigating categories of Petri nets.

Acknowledgements

This work was supported by GAČR under the contract No. 102/98/0552 „Research and Applications of Heterogenous Models", was carried out under the financial support of Research intention No. CEZ: J22/98: 262200012 – „Research in information and control systems" and under the financial support of School of Computing and Mathematics, University of Huddersfield, Great Britain.

References

[ČEŠ 94] Češka M.: Petriho sítě - Základy teorie a nástroje pro jejich aplikaci, Akademické nakladatelství CRM, Brno 1994.

[ČEŠ 95] Češka M., Skácel M.: *The Petri Net Tool PESIM and Its Application in VLSI and ASIC Design*, In: Proc. of The 7th Microcomputer School VLSI and ASIC Design, Wydawnictwo Prac Naukowych FORMAT, Wroclaw 1995, pp. 185-194.

[FEN 96] Fencott C.: *Formal Methods for Concurrency*, Cambridge University Press, 1996.

[JOY 94] Joyal A., Nielsen M., Winskel G.: *Bisimulation from Open Maps*, 42 pp, 1994. Available at *http://www.brics.aau.dk/RS/94/7/index.html.*

[LAW 97] Lawvere F. W., Schanuel S. H.: *Conceptual Mathematics: A first introduction to categories*, Cambridge University Press, 1997 (translated from Italian edition 1991).

[MES 90] Meseguer, J., Montari, U.: *Petri Nets Are Monoids*, In: Information and Computation, Volume 88, Number 2, Academic Press, Brugge, Belgium, 1990.

[MIL 89] Milner R.: *Communication and Concurrency*, Prentice Hall (UK), 1989.

[PAR 89] Parrow J., Cleaveland R., Steffen B.: *The concurrency workbench: A semantics based tool for the verification of concurrent systems*, Technical Report ECS-LFCS-89-83, Laboratory for the Foundations of Computer Science, Edinburgh University, 1989.

[PET 81] Peterson J. L.: *Petri Net and the Modelling of Systems*, Prentice Hall (USA), 1981.

[TAU 89] Taubner D.: *Finite Representations of CCS and TCSP Programs by Automata and Petri Nets*, Lecture Notes in Computer Science, Volume 369, Springer-Verlag, New York/ Berlin/Heidelberg, 1989.

[URB 99] Urbášek M.: *Bisimulation relations in categories of A-labelled Petri nets and the connection of CCS and Petri nets*, Technical report, Department of Comp. Science and Engineering, FEECS, Brno University of Technology; to appear (http://www.fee.vutbr.cz/~urbasek).

[WIN 84] Winskel G.: *Categories of models for concurrency*, Lecture Notes in Computer Science, Volume 197, Springer-Verlag, New York/Berlin, report, Computer Laboratory, University of Cambridge.