

Response from authors on revised version of the paper:

CASM-14-02-1903

K. Jensen and L.M. Kristensen: Coloured Petri Nets: A Graphical Language for Formal Modelling and Validation of Concurrent Systems"

We have now submitted the revised version of the paper.

Firstly, we would like to thank the reviewers for their careful reading and for the constructive comments that have been very useful in improving the paper. Before going into details on how we have addressed the individual comments from the reviewers, we find that it is appropriate with a discussion of the overall content of the review reports.

Reviewer 1 gives a very positive review of the paper and finds that: "... the paper can be published as is". The primary issue raised is item 7 in the report on some references to other high-level formalisms and tools. In response to this, we have added references to other high-level Petri nets formalisms in the last section of the paper.

Reviewers 2 has "mixed feelings" about the paper primarily on the basis of questioning whether the paper with the current focus on CPNs should be published in CACM, but basically leaves this decision to the editor taking into account the content of other review reports.

Reviewer 3 has also provided a positive review raising two main issues in the report. One issue is that more detail on the verification part of the paper, and the second issue is references to other high-level formalisms. The latter comment being consistent with the main issue raised by reviewer 1 (see above).

It should be noted that there is a **strict limit** of 8 pages for manuscripts of CACM. This has impacted which of the reviewers comments that we have been able to address. In particular, there were three areas in which the reviewers suggested expansion of the content of the paper: 1) references to other high-level Petri net formalisms; 2) expansion of sidebar I on Petri nets; and 3) more on the verification and analysis techniques.

In order to stay within the page limit, we have decided to only expand on 1) and not add further discussion of basic Petri nets and analysis techniques to the paper. In addition to the page limit, a main reason for this choice is that our aim was to write a paper focusing on the CPN modelling language itself and how it extends Petri nets so that practical modelling of concurrent software systems becomes possible. We do link the paper to ordinary Petri nets in Sidebar I and in Section 3 where the relationship to ordinary Petri nets is discussed in the context of unfolding to make explicit the relationship between CPNs and ordinary Petri nets. State space exploration and model checking is indeed an important analysis technique for CPNs, but most model checking techniques are transferable between modelling languages as they are formulated at the level of labelled transitions systems and Kripke structures. This means that state space exploration and model checking is not specific for CPNs. A similar comment applies to simulation-based performance analysis which is also a techniques widely used across many modelling formalisms, in particular in the area of protocols and networks. CACM do offer the option of online appendices for papers, but we do not find that an elaborate presentation of Petri nets and analysis techniques has a nature that naturally fits in an

appendix. Instead we have added a couple of references to existing papers where more information on this can be found.

Below we explain in detail our response to the comments provided in the three review reports. The original reviewers comments has been prefixed with a ">" whereas our response has been written in blue.

Reviewer 1:

>Report on „Coloured Petri Nets. A graphical Language for Formal Modeling and Validation of
>Concurrent Systems“ submitted by Kurt Jensen et al to CACM. Manuscript ID CASM-14-02-1903

>The paper is technically sound and professionally interesting. The paper presents an adequate
>picture of the “Coloured Petri Nets” modeling technique. This technique is of interests for many
>readers of the CACM. The paper meets the requirements of the author guidelines of the CACM for
>contributed articles. Summing up, the paper can be published as it is.

We thank the reviewer for the positive comments.

>I nevertheless see chances for improvement. The authors may consider the following
>recommendations; I do however not require them to be followed necessarily.

>1. page 1, column2, line 36 (I follow the numbering on the margin): Here and at some other places you
>use “Petri nets” with singular “is”, as a generic term. This is a matter of taste; however, in
>line 45 “Petri nets (plural) ... is a graph (singular)”, This principle is very confusing. Furthermore,
>you sometime use “Petri nets” with plural. Be consistent!

We have gone through the paper with the intention to consistently use the plural form and the terms
“Coloured Petri Nets” , “CPNs”, “Place/Transitions Nets”, “PTNs”, “Petri nets” and “PrT nets”.

>2. page 1, column 2, line 58: “conflict” is not for the general audience. Add “non determinism” in
>brackets.

We have added “non-determinism”

>3. page 2, column 2, line 25: Motivate the term “substitution”. For readers with some familiarity
>with Petri nets it would be convenient here to understand that the firing rule is not applicable
>here.

We now motivate this when substitution are introduced at the beginning of Section 2 saying that
substitution transitions are purely syntactical constructs.

>4. A general remark to the graphics: All Figures follow the rules of the CPN-tools. But this article
>would be more readable if you just skip some redundancy. Figure 1: “coordinator” is written
>inside as well as on the edge of the left box. Likewise, “workers” at the right box.

We decided to keep the figures as they appear in CPN Tools, but in order to shorten the paper we have
removed the explicit explanation of the tag names and added instead a motivation for the term
“substitution” cf. the comment above.

>Figure 3: The
>token () is mentioned three times: As the type of the place, as the actual token load, and as the

>number of actual tokens. His confuses newcomers a lot. Generally, in your entire example, the
>“multiset” aspect is irrelevant. You never have two or more identical tokens. So just skip
>multiplicities everywhere.

We have decided to keep the figures identical to how they appear in CPN Tools. To partly address the issue, we have removed the use of the word “multi-set” in a number of places in the text as it was not strictly needed.

>Furthermore, the “Out” and “In” information at the places in the right
>column of Figure 3 is anyway given by the arrow heads. Skip the small boxes and the
>corresponding text. For the newcomer, this is just boring and confusing.

We decided to keep them as there is no other way to see that the places in questions are port places. Indeed the In/Out information can be derived from the direction of the arcs, and in order to shorten the paper we removed the explicit discussion of the In/Out tags from the text.

>page 4, Figure 4: The place *Can commit* carries the circled “2” on the margin, whereas “Waiting
>Votes” carries the corresponding “1” inside. Many Figures are inconsistent this way.

We have now consistently placed the circle indicating the number of tokens on the margin/border of places.

>5. page 5/6, Sect. 3: The relation to P/T nets is valid. But there have been many more wellmotivated
>proposals for unfoldings and for high-level nets.

We are not sure exactly what is meant by this comment. We have, however, added a sentence saying that CPN Tools works directly on the high-level net representation without performing unfolding.

>6. page 7, column 2, line 41/42: “.... properties expressed using ...” is no good English style.

We have changed the formulation to “specified in”.

>7. It’s good scientific praxis to also discuss the limits of one’s achievements. For example, Petri nets
>offer a whole bunch of analysis methods (Place invariants, transition invariants, traps,
>coverability graphs, ...), supported by other tools, and lifted to high-level versions of Petri nets.
>You should shortly mention this. Likewise, the state explosion problem, that you frequently
>mention, has successfully been attacked for Petri nets, by exploiting symmetries, reversibility,
>etc. Finally, there are other, successful Petri net based industrial software tools. It is more
>impressive for the audiences to present CPN-tools as “the flagship among many” than presenting
>them as solitaire.

We have added references to other high-level net formalisms in the last section of the paper, and we now mention in sidebar I that Petri nets support analysis based on both structural techniques (e.g., invariants and unfoldings) and dynamic techniques (e.g., state spaces and coverability graphs). We have added a reference where the reader can find more information about basic Petri nets.

Reviewer 2:

Referee report on the paper CACM -14-2-1903

Coloured Petri Nets: A Graphical Language for Formal Modeling and Validation of Concurrent Systems

>I have mixed feeling about this paper.

>Pros:

>The paper is well written and a pleasant reading at least for the readers who appreciate this type of contributions. (Apart from a few minor but not irrelevant issues pointed out below). Colored Petri Nets (CPNs) are certainly a major contribution in the field of modeling concurrent systems, with particular reference to the models rooted in the original Petri's work. In the decades they evolved effectively in terms of modeling features, supporting tools, and applications to practical problems.

Many thanks for the positive comments.

>Cons:

>CPNs have been around for quite a while and many papers in the open literature already document all aspects –and history- of this formalism; there is even a full, rather recent, Springer monograph authored by the same authors of this paper. CPNs are certainly not the only formalism that addresses the issue of modeling and analyzing concurrent systems; neither are they the only general extension of the original Petri's model; the authors properly point out that many of such extensions were ad-hoc, but there are others that are as general and well thought as CPNs, such as, e.g., CO-OPN, or the now numerous UML-based formalisms, i.e. those formalisms that starting from UML notations enrich them with various form of formal semantics; notice, BTW, that some UML notations are clearly inspired by Petri nets.

We agree that there are many other languages that targets concurrency systems. We have added some selected references to examples of such formal modelling languages for concurrency systems in the introduction of the paper. A reference to timed automata (to include a language that in addition to concurrency focus on real-time), Promela to include a textual and also widely used language for verification purposes, and StateCharts to include a graphical language linked linked to UML. In addition, we have added reference to other high-level net formalism in the conclusion part of the paper.

>So, my main question is: should CACM publish a paper which certainly address an important issue and describes *just one* among many proposed approaches, all already well documented and illustrated in the open literature?

>Given that we do not expect original results from such a paper, which type of CACM readers could be attracted and could obtain an added value from this paper? Certainly not readers who already know about CPNs; also readers who are already in the field of modeling concurrent systems most likely would be aware of the many existing approaches, and, even if not knowing the special contribution of CPNs would not find any difficulty in reaching the abundant and widely accessible documentation. Many readers are probably aware of the increasing relevance of concurrent and distributed systems and could be interested in a quick but not trivial understanding of the problem and of the solutions offered by the state of the art. For this type of readers, however, I think a broader paper would be more useful: even assuming to give a privileged focus on CPNs, and avoiding the style of a typical Computing Surveys paper, I would favor a better statement of the critical issues in concurrency (and real-time!), a kind of progressive “zoom-in” from general formal approaches, to Petri nets-based ones and, finally to CPN. This would probably make an already long paper (for CACM) a little longer, but some space could be saved by giving up a few “historical remarks” about CPNs and some details about the rather standard tool interface.

We find that the broader paper on modelling concurrent / reactive (possibly real-time) systems being requested here would be a very different paper from what we intended with the submitted paper. Hence, we have not attempted to revise the paper to make it a more general paper on formal modelling languages for concurrent and distributed systems. We also find that there are other researchers in the community of formal methods that would be in a much better position and have more credibility in writing such a paper. Furthermore, given the two other positive reviews, we decided to not fundamentally change the content of the paper.

One goal of the present paper was to make the CPN language visible to a broader computing community than what one normally reaches with textbooks and with papers published at formal methods oriented conferences. Hence, we find that the paper serves a difference purpose than already published material. We agree that in the end this is an editorial decision to be made. It seems, however, that CACM publishes such papers. A recent example is in Vol. 57 04/2914 of CACM, 2014 in which there is a paper on the Scala programming languages (no comparison with CPNs otherwise intended) for which there also exist numerous published papers and textbooks.

>On the basis of these remarks I am rather against publishing this paper, with all respect and >appreciation for the authors and their long standing research. However, since it is not easy for me to >"impersonate" the average CACM reader, I would certainly not object if the editor, possibly on the >basis of a different report, decides to overwhelm my opinion.

See our response to the previous item.

>In case the paper is accepted for publication, here are a few minor but not irrelevant points, besides >the above general remarks.

>1. On the basis of the hypothesis that this paper should attract and be well-understood by general >readers, I would recommend to expand considerably the sidebar about basic PNs, including some >discussion on their ability to capture problems such as deadlock, conflicts, and their limitations so >that it will be later clear where are the advances introduced by CPNs (presently such general >comments are scattered within the CPN presentation).

As we have written initially, then given the page limit and our intended focus of the paper we decided to only expand sidebar I slightly in response to a comment from reviewer 1.

>2. From a theoretical point of view the authors state that CPN "do not add expressive power" to basic >PTNs; but properly add "(possibly infinite)". This is due to the unfolding process; however, infinite >processes are not exactly *abstract machines*, rather, roughly speaking, they represent *machine* >*behaviors*. In fact, whereas original (finite but unbounded) PNs have not the power of Turing >machines, many forms of extended PNs, including CPNs which can simulate, e.g. PNs with inhibitor >arcs, do have such a power.

We have changed the discussion of unfolding at the beginning of section 3 to make it clear that (as correctly pointed out) already PTN + inhibitor have the expressive power of Turing machines.

>3. The above point has a natural impact when dealing with tools supporting the analysis of the model. >Whereas a simulator can clearly produce *sample* behaviors of the modeled system even if the >abstract machine is undecidable, a model checker can recursively state system properties only for >decidable formalisms. I understand that it is now common practice to use model checkers to >"semidecide" >properties of models which in principle are undecidable; this can be done in several ways: >either by accepting the risk of non-termination of the tool, or by restricting the state space, or etc....; >it would be better, however, that the choices taken when applying tools to "in-principle- >undecidableproblems" >are clearly and explicitly stated. In this case the authors only speak about state explosion, >i.e., complexity, but do they assume *a priori* some hypotheses that guarantee decidability (typically a >finite state space, such as bounded number of tokens and finite domains for colors, as it happens in >the paper's example)? Or, is it the user who has the responsibility to write graphs whose properties >are decidable? Or again, is the user who must choose and use the appropriate tool depending on >the system she is modeling and the properties she wants to analyze? All this should be clarified,

>mainly for users who might not be theoretically well equipped.

We have expanded the sidebar II on State Spaces and Model Checking to briefly clarify this as it is basically the user that is responsible for ensuring that the state space of the analyzed model is finite (often this is done by restricting data types and introducing bounds on the number of tokens on places).

>4. The issue of dealing with timing aspects is briefly touched at the end of section 2 but leaves
>several
>unanswered questions, for instance: “The time stamp of a token specifies the earliest model time at
>which the token can be removed by the occurrence of a transition” seems to hint at a weak time
>semantics; in this case is it possible for the user to specify a timeout? Again, a considerable amount
>of literature on dealing with time critical aspects in Petri nets is ignored. If the intricacies of the
>timing semantics require going too deep into the issue, it is perhaps better to ignore it at all.

It would take a considerable amount of space in the paper to properly discuss this, and hence we have decided not to expand on this as also discussed in the beginning of this document. We have moved the parts briefly introducing time at the end of section 2 into sidebar II in order to have the discussion of time in one place and to make it clear that this is not the main focus of the paper.

>5. The value (color) of the tokens is stated by the firing of the transition that produces the token;
>this
>is consistent with the general philosophy of PNs. However, in the example *it seems* that in some
>cases such a value is completely specified by the label of the output arc, e.g. SendCanCommit
>produces exactly two tokens with the two colors; in the case of ReceiveCanCommit, instead the
>production of the pair (w, vote) is a little less clear: I would *guess* that there is a nondeterministic
>choice between Yes and No about the vote, which is produced by the transition firing, whereas the
>value of w is the same as the one received from the input token. Later, we read that the tool, in the
>automatic simulation applies random choices whenever the behavior is not determined –which is
>typical of PNs-; but random choices can rarely be useful to analyze properties such as reachability
>or safety, i.e. questions existentially or universally quantified w.r.t. possible behavior. This could be
>a nice complementary use between interactive, automatic simulation (which should however
>include some statistical analysis) and model checking; but it should be clearly explained to the
>nonspecialized
>reader.

In state space exploration, the tool considers all possible transition firings – not only one selected by random. We expanded Sidebar II on state space to make it clear that in this case all the possible enabled binding elements are being computed (and explored).

>6. Neither Firefox nor Explorer have been able to compile the full example in [http:// goo.gl/HY4ZNQ](http://goo.gl/HY4ZNQ)
>which is redirected to <http://home.hib.no/ansatte/lmkr/cpncommitprotocol.xml>: Firefox reported
>“the XML file has no associated style sheet”.

This seems to be a Firefox problem. The authors have been able to download it with IE explorer and Chrome. The file may alternative be downloaded from the URL using a different browser and/or command line HTTP clients such as wget. In case the paper is eventually accepted for publication, we will make the complete CPN model including a short video demonstration (if possible) available as an online appendix. For now we have removed the link from the paper, but the CPN model is still available via the above URL.

>7. At p.7, r. 29, II column: constributed ---> contributed.

Fixed

Reviewer 3:

>Comments to the Author

>This article introduces the well-established formalism of Colored Petri Nets (CPNs), and the two
>authors are eminently qualified to write it, as they are some of the most important contributors to
>the definition, development, and use of CPNs.

>The article succeeds in not being overly technical, while providing a fairly clear introduction to the
>key concepts of CPNs. In particular, the modularity and the type-enforcing/checking features of
>CPNs are gently introduced through the running example.

>I believe that an introductory article on CPNs to “wet the appetite” of CACM readers is appropriate,
>and this submission might just do that. In other words, interested readers could then read more in
>depth about CPNs through longer articles or even books (and multi-volume books like the one
>authored by K. Jensen), and proper pointers to these publications are given in the references.

We thank the reviewers for the positive feedback.

>However, I have important two criticisms to the article:

>The first criticism is that, while the article does a good job of giving an idea of why CPNs might be a
>good “language for formal modeling of concurrent systems”, it does not go into enough detail into
>the types of techniques that can be employed for the “validation” (or is it “verification”?) of such
>systems.

>The paper is thus unbalanced in my opinion, and it left me wanting more on the, let’s call it,
>“analysis” part. I believe that this aspect is fixable by adding some more material. Ideally, it would
>be good to refer the techniques to the running example, in the form: “if the user is interested in
>analyzing X, techniques A, B, or C can be used, with the following tradeoffs...”, for a few X’s.

As we have discussed in the beginning of our response, then our aim of the paper was to focus on the CPN language itself and only briefly discuss the analysis techniques. Given this, and the strict 8 page limit we have not been able to expand significantly on this. We do provide references to papers where the verification techniques are discussed in detail.

>The second criticism is that the authors have not compared CPNs to other related formalisms,
>including other high-level formalisms related to Petri Nets, such as well-formed nets. The article
>reads like CPNs (and the tools described for it) are the only game in town, while they are not. This is
>understandable as the authors have been extremely involved with CPNs for most of their career,
>but I still would have liked to see other alternatives mentioned.

>On the other hand, this strict focus on CPNs is perhaps acceptable for a CACM submission, as I
>imagine the goal is to popularize a notion (the use of CPNs in this case), rather than to have a survey
>of the entire field.

We have added references to other high-level net formalism in the last section of the paper.

>In conclusion, I think that my first criticism should be addressed, but I leave it to the Editor to decide
>whether the second criticism needs to be addressed as well.

>Regarding the writing, I have only a few minor editing suggestions:

[We have fixed the minor modifications that were suggested.](#)

>CPN Tools which => CPN Tools, which

>distributed system was => distributed systems were nets) that were introduced => nets), which
were introduced

>Petri nets is => Petri nets are

>and is still able => and are still able

>is a directed bi-partite => are a directed bi-partite

>“staggering” is the wrong word, if you mean “slowing down”

>“zoo” is too informal a word, “variety” might be better

>is comprised of => is composed of

>in Fig 6 is => in Fig 6 are

>cannot without exchanging messages know the vote of another worker => cannot know the vote of
>another worker without exchanging messages

>Above => So far,

>[19] that => [19], which

>It is based => This is based

>The sentence “and the model stays at the current model time until no more transitions are
>enabled” is unclear to me, I did not understand what you mean.

>10.000 => 10,000 (but it might depend on CACM typesetting standards)

>got the idea => had the idea

>Are sidebar II and III called in the main text? If not, they should be.

>The sentence “Many advanced techniques exist to combat the state explosion problem, and most of
>these can be applied also in the context of CPN models” is vague:
>what techniques are actually available in CPN tools? I would like more details.

>The research into => Research into