Summary of Thesis and Evaluation.

This thesis focuses on theory and implementation for client server systems with unbounded number of clients and unbounded concurrent interactions between clients and server. The goal is to identify different abstract models to capture the behaviour, explore logics to specify properties that are interesting for systems and develop verification tools to verify these properties not just in theory but in practice. For the model, the author considers different variants and extensions of Petri nets a classical model for concurrent systems. The main contributions are then to provide a bounded model checking approach with an encoding into Satisfiability-modulo-theory (SMT) solvers leading to a practical implementation.

The theory and background is developed nicely with Chapter 2 introducing Petri net models as well as \nu-net and Elementary Object Systems (EOS) extensions. This chapter also describes a basic encoding into constraint systems that can be handled by state of the art Satisfiability-modulo-theory (SMT) solvers. In Chapter 3, the author introduces different logics, starting from basic linear temporal logic (LTL) to counting variants and First order logic with Monodic restrictions. A brief encoding into SMT is also written. Chapters 4 -7 are the main novel contributions of the thesis. In Chapter 4and 5 , a bounded model checking (BMC) approach is developed for verifying LTL over Petri nets, with so-called sequential and concurrent semantics and this is implemented with some experimental results presented comparing with some other tools. In Chapter 6 this is lifted to an extended logic and Petri nets with names or \nu-Petri nets, where a bounded semantics and SMT encoding is presented. Finally, in Chapter 7, the author considers an extension to a hierarchical Petri net model where different lossy extensions are considered. Theoretical decidability and expressiveness results are shown for these models in a comprehensive manner. This final model is also implemented using an answer-set programming approach and some results shown.

Overall, I believe the thesis contains a decent body of work, that can be accepted once the following points are addressed, which I classify into major and minor points.

Major points

1. In a few places, notions and notations are used before being introduced. It would be good to do a careful check throughout the thesis. For instance APS is used on page 18 as an example but is not defined till… 39 and even there it is rather unclear. It is only really defined in pg 79 (sec 6.1.1), but many properties about it are used from pg 18 onwards. This case-study should be introduced much earlier as it is referred to often.

2. The prelims chapter needs a careful going over. There are many notational and other significant errors. For instance weighted Petri nets are used but only unweighted are formally defined. Buchi automata and acceptance are never defined but used in many places.

3. In Chapter 4 algorithm 1 and 2 are not described except as pseudocode. Novelty in Algo 2 and notations must be made clear. Also some proof of correctness is needed. It may be good to possibly formulate a theorem saying that Algo2 is sound but not complete. Also, it seems that the Unfolding of a net for doing BMC is not defined formally.

4. In Chapter 5, in the experiments, I had two main questions: (i) why does your approach have false positives? I expected DCModelChecker to be sound but not complete, hence I am surprised to see ~9% false positives which show lack of soundness. Did you examine these examples? Can you identify their source and fix it? (ii) From Expt 2/3 it seems DCModelChecker2.0 is not much better than DCModelChecker1.0… Especially in Table 5.3 it seems 1.0 is much better than 2.0. In Expt4 in sec 5.4.4, there is a statement saying “our experiments demonstrate 2.0 is better than 1.0” but

these experiments don't seem to be presented in the thesis. Can you add these and comment on them, given this shows novelty and effectiveness of the 2D-BMC algorithm?

5. In Chapter 6, the experimental results of DCModelChecker3.0 seem to be missing. I agree there is nothing to compare with, but it would be good to present the results in a table/plot. I am a bit confused actually whether this tool is available or not since it is listed in Relatedwork section as well, but section 6.2 talks about tool and implementation. If it has been implemented some results should be presented, else it should be made clear that the contribution is the bounded semantics and the SMT encoding of the semantics but implementation is part of future work.

6. In Chapter 7, the results are nice but some of the proofs of theorems are referred to a conference paper. These should be perhaps included here. Also in the proofs it would be good to highlight where conservativeness is used to obtain decidability. Currently this seems a bit mysterious.

Minor

- glossary is empty

Chapter 1
pg3: Buchi automata are not defined, accept states or condition not present in def 1.1.1
pg4: fig1.1 equivalent BA to what? Omega words not defined
pg6,7- contributions are out of order? 5 should come earlier than 2-4?

Chapter 2
pg11: where do you use conservativeness in entire thesis? If not needed it can be removed.
Pg12: fig 2.3: the picture does not match the caption in particular the marking.
Pg13: fig 2.4: same issues
pg14: SMT solvers are used without any introduction.
Pg14: you seem to be using weighted petri nets in many places throughout the thesis (even in sec 2.2) but definition in 2.1 and semantics also are not weighted.
Pg17: unbounded Petri nets are used but boundedness and unbounded was not clearly defined earlier?
Pg17: formal definition of unfolding is not present. This is also slightly confusing since in petri nets unfolding has a slightly different meaning from unfoldings for BMC...

Chapter 3
pg 29: Buchi automata are not defined.
Pg30: Kripke structure should be defined and also an examples of syntax and semantics will help.
Pg34: what is a live agent? The fact that agents and clients are same needs to be written earlier.
Pg 38-41: since APS is not defined fully yet, many of these parts don't make clear sense.

Chapter 4
pg 47: model cannot be a subpoint of verify? Maybe write both?
Pg 50: unbounded net not defined in prelim?
Pg 52: state space explosion defined after its used.

Chapter 5
pg58: 2D-BMC encoding needs to be defined clearly and explained better. Maybe an example will help?
Pg63: DCModelchecker1.0 properties should perhaps be written earlier and emphasized. PNML, ANTLR etc are used before they are defined.

Pg77: what tapaal used earlier?

Chapter 6
pg 79: example of APS is coming too late?
Pg 86: only few points in the semantics are described. If the rest are routine/similar and hence left out it should be made clear. Else they should be added.
Pg 87: what is UCSTL? Not defined earlier. What is MFOTL? If it is important it should be defined earlier, and not in related work. Proof of Lemma 1 details should be written as this is a thesis; not just a sketch.

Chapter 7
pg91: standard Pns are defined earlier… if you are not changing the notation, can just refer to that.
Pg95: can recall conservativeness and explain where it is needed maybe? Or how it is used in [65]
pg 114: why switch from SMT to ASP? May add some justification.

Section B

(Evaluator's Detailed Comments)

Please provide the following as an attachment

1. General comments on the thesis, including a critical survey and evaluation of the quality and quantity of the work reported in the thesis.
2. Points which require clarification, and suggested amendments or revisions (if any)
3. Questions to be asked at the time of viva-voce.

I have provided 1. and 2. as a separate attachment.

Questions for Viva-voce: Many questions are already given in the attached file.

In addition, the following questions may be considered:

1. How does the answer set programming approach compare to the bounded model checking approach for normal Petri nets? Could this be a way to unify the different approaches of the thesis?

2. Instead of using SMT over LIA, could we use Pseudo Boolean solvers or (M)ILP solvers for the various problems addressed in the thesis?

3. From a modeling point of view are there are other case-studies other than the autonomous parking system for which the ideas in this thesis can be applied?