Abstract

Workflow management systems (WMS) provide convenient ways of visualization, analysis and automation of work processes. We wish to improve the quality of workflow systems by applying formal verification techniques. We present an automatic translator for workflow models designed as Petri net models to a state space, and we develop a tableau based model checking algorithm to verify properties of that model. Tableau based theorem provers are both computationally and resource intensive. Introducing a timed and temporal logic increases the complexity. To cope with this problem we discuss a mechanism to distribute the verification procedure among multiple processors.

High Performance Computing

- Divides computation tasks into smaller pieces and distributes them to multiple processors to process concurrently.
- Reduces time to completion.
- Increases available memory.

What We Need

- An explicit model of the system showing all possible states.
- A system specification language to express timed properties.
- A strong proof procedure to verify properties of the system.

Timed temporal logics permit the explicit reference to time in order to specify quantitative properties, needed by safety critical systems. Our property specification language is timed CTL logic, which in Backus-Naur form is:

\[
\begin{align*}
\phi &::= \top \mid \neg \phi \mid \phi \lor \psi \mid \phi \land \psi \\
&\mid \phi \rightarrow \psi \\
&\mid \exists t \psi \\
&\mid \forall t \psi \\
&\mid \phi U \psi
\end{align*}
\]

Our framework uses distributed computing as follows:

- The Petri net model is distributed over processors to generate the state space.
- Each processor will search only its local states.
- The tableau MC algorithm will search the distributed state space.

Proposed Framework

Our framework provides an efficient combination of model checking and theorem proving to achieve the automatic style of model checking as well as the logical foundation of theorem proving. It also applies distributed computing to manage the state space explosion problem and to make the verification process more efficient.

Problems

- Manual translation of a large Petri net model to a state space is very time consuming.
- Timed temporal logic increases the computational complexity.
- Model checking suffers from the state explosion problem.
- Fully automatic theorem proving is quite tedious and impractical for large workflow systems.
- Even a simple workflow can have many possible execution paths and may result in a very large state space, which is hard to manipulate in a reasonable amount of time by a single processor.

Verification Process

- Step 1 The XML parser generates a simple text file containing the places, transitions and arcs extracted from the XML file generated by the CPN tool.
- Step 2 The state space generator program takes the text file and generates the state space by executing the PN model.
- Step 3 The tableau model checker then verifies the properties on that state space.
- Step 4 The output of the model checker shows either the specification properties are true or generates a counter model.
- The framework also incorporates an ontology reasoner to guide the verification process.

Discussion

One major difference in the proposed framework is the implementation language is now MPI C++, which makes the system architecture easier to extend than the unstructured Prolog implementation. Also C++ has been successfully used and proved to be efficient in various theorem provers. The distributed memory computation makes the system capable of handling very large model efficiently.

This project is the first author’s M.Sc. thesis, the completed project will have important implications to safety critical system design.

Related Works

K. Miller (in 2009) developed a model checker for ontology driven workflow verification. It involved a manual translation of a PN model to its state space, the specification language was Timed BDI, and the implementation used Prolog in a shared memory architecture. J. Dallien (in 2007) developed a model checker for health care workflow verification that involved a manual translation of a PN model to its state space, the specification language was ACTL and a serial Prolog implementation was used.

Background

- Workflow

A workflow can be depicted as a sequence of tasks. It gives a virtual representation of an actual work.

- Workflow Management Systems (WMS)

WMS provide visualization, analysis and automation of business processes to identify and correct potential errors in a system.

- Properly verified workflows ensure better systems.
- Safety critical systems like health care must ensure an error free procedure.
- Properly verified workflows ensure better systems.
- Safety critical systems like health care must ensure an error free procedure.
- Petri Net

Having a strong mathematical foundation, Petri nets are a popular tool to model workflow systems in the industry. A Petri net consists of the following:

- A finite set of places
- A finite set of transitions
- A finite set of directed arcs. An arc can only connect a place and a transition.
- A finite number of tokens (representing resources) in places.

A transition can only be enabled when all the input places have tokens. An enabled transition may fire (if there is no other contraints) by consuming a token from each of its input places and produces a token in each output place. Extended formalisms of Petri nets support modelling time. However Petri net tools provide an implicit representation of the system and have limited verification capability.

Formal Verification

The process of showing that a formal design specification (model) satisfies its formal requirement specifications (properties). Formal verification uses mathematical techniques to ensure the correctness of a system design. Various formal verification approaches include:

- Interactive theorem proving: It is the most powerful verification method. But it needs logicians to guide the proof procedure.
- Automated theorem proving: It does not require human interactions. The implementation of ATP is very complex. It provides better assurance than model checking.

- Model checking: In principle it is automatic. However, the application is limited to finite state transition systems because of the state explosion problem.

Workflow management systems (WMS) provide convenient ways of visualization, analysis and automation of work processes. We wish to improve the quality of workflow systems by applying formal verification techniques. We present an automatic translator for workflow models designed as Petri net models to a state space, and we develop a tableau based model checking algorithm to verify properties of that model. Tableau based theorem provers are both computationally and resource intensive. Introducing a timed and temporal logic increases the complexity. To cope with this problem we discuss a mechanism to distribute the verification procedure among multiple processors.

High Performance Computing

- Divides computation tasks into smaller pieces and distributes them to multiple processors to process concurrently.
- Reduces time to completion.
- Increases available memory.

What We Need

- An explicit model of the system showing all possible states.
- A system specification language to express timed properties.
- A strong proof procedure to verify properties of the system.

Timed temporal logics permit the explicit reference to time in order to specify quantitative properties, needed by safety critical systems. Our property specification language is timed CTL logic, which in Backus-Naur form is:

\[
\begin{align*}
\phi &::= \top \mid \neg \phi \mid \phi \lor \psi \mid \phi \land \psi \\
&\mid \phi \rightarrow \psi \\
&\mid \exists t \psi \\
&\mid \forall t \psi \\
&\mid \phi U \psi
\end{align*}
\]

Our framework uses distributed computing as follows:

- The Petri net model is distributed over processors to generate the state space.
- Each processor will search only its local states.
- The tableau MC algorithm will search the distributed state space.

Proposed Framework

Our framework provides an efficient combination of model checking and theorem proving to achieve the automatic style of model checking as well as the logical foundation of theorem proving. It also applies distributed computing to manage the state space explosion problem and to make the verification process more efficient.

Problems

- Manual translation of a large Petri net model to a state space is very time consuming.
- Timed temporal logic increases the computational complexity.
- Model checking suffers from the state explosion problem.
- Fully automatic theorem proving is quite tedious and impractical for large workflow systems.
- Even a simple workflow can have many possible execution paths and may result in a very large state space, which is hard to manipulate in a reasonable amount of time by a single processor.

Verification Process

- Step 1 The XML parser generates a simple text file containing the places, transitions and arcs extracted from the XML file generated by the CPN Tool.
- Step 2 The state space generator program takes the text file and generates the state space by executing the PN model.
- Step 3 The tableau model checker then verifies the properties on that state space.
- Step 4 The output of the model checker shows either the specification properties are true or generates a counter model.
- The framework also incorporates an ontology reasoner to guide the verification process.

Discussion

One major difference in the proposed framework is the implementation language is now MPI C++, which makes the system architecture easier to extend than the unstructured Prolog implementation. Also C++ has been successfully used and proved to be efficient in various theorem provers. The distributed memory computation makes the system capable of handling very large model efficiently.

This project is the first author’s M.Sc. thesis, the completed project will have important implications to safety critical system design.