Modeling and Verification for Different Types of System of Systems using PRISM

Dongwon Seo, Donghwan Shin, Young-Min Baek, Jiyoung Song, Wonkyung Yun, Junho Kim, Eunkyoung Jee, Doo-Hwan Bae

School of Computing, KAIST, Republic of Korea

2016.05.16

Presenter: Doo-Hwan Bae

http://se.kaist.ac.kr
bae@se.kaist.ac.kr
Contents

- Introduction
- Related work
- Background
- Overview
- Modeling & Verification
  - Modeling different types
  - Verification property and results
- Conclusion
  - Lesson learned
  - Summary
Managerial & operational independence of various CSs*

*CS: Constituent System

- Maier\(^1\) proposed the notion of managerial & operational independence of CSs for SoS.
- SoS may not have full authority to manage or operate their CSs → Lack of authority

---

Four types of SoS are classified by Maier\textsuperscript{[1]} and Dahmann et al.\textsuperscript{[2]}

- Directed SoS, acknowledged SoS, collaborative SoS, and virtual SoS.

Introduction

- Lack of studies considering different types of SoS in modeling and verification.
- We do a case study for the modeling and verification of several types of SoS.
  - Use “probabilistic model” to represent uncertainty of autonomous CS’ behaviors.
Related Work

- Modeling and verification attempts for each type of SoS
  - Bryans et al: Collaborative SoS, SysML ➔ CML
  - Hammand et al: Directed SoS, SysML

- PRISM, used for verifying SoS goals
  - Calinescu et al: policies
  - Zhou et al:

  ➔ Have not considered various types of SoS and their characteristics in verification
Background

- **Types of SoS**
  - *The degree of authority* determines the adaptability of CSs to SoS-level goals.
  - SoS-level managers could consider which type is the most appropriate to perform the desired behaviors.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SoS-level goal</strong></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Explicit objectives</td>
<td>Explicit/Implicit</td>
<td>No common goal</td>
<td></td>
</tr>
<tr>
<td><strong>SoS-level organization</strong></td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Enforce</td>
<td>Recommend</td>
<td>Not exist(operational independent)</td>
<td></td>
</tr>
<tr>
<td><strong>SoS-level ownership</strong></td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Own</td>
<td>Not exist(managerial independent)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Background

- **Statistical Model Checking (SMC) for verification**[1]
  - Hypothesis testing on given samples (i.e., system simulation traces)
  - Give a statistical verification result from probabilistic models
  - No state explosion problem

---

Overall Approach

Define SoS scenario

Mass Casualty Incident (MCI) scenario

SoS-level goal

CS-level goal

Verify SoS model

SoS Model

Statistical Model Checker (PRISM)

Verification Results

Analyze SoS under types

SoS-level Goal(s)

Model each type of SoS

Directed MCI-SoS

Acknowledged MCI-SoS

Collaborative MCI-SoS

MCI-SoS: MCI response system

DIR: directed SoS

ACK: acknowledged SoS

COL: collaborative SoS

Degree of authority
An MCI is an incident where the resources of the emergency services are overwhelmed by the number and severity of casualties.

An SoS tries to rescue as many patients as possible using various Patient Transferring Systems (PTSs) as CSs.

- **SoS-level goal**: to rescue patients in a local area including the MCI area.
- **CS-level goal**: to rescue patients in the area, which the CS covers.

→ The CS-level goal can be yielded to achieve the SoS-level goal.
**MCI-SoS: Abstracted SoS Components**

- **Two areas**
  - There are two: an MCI area and a non-MCI area.
  - Patients occur more often in an MCI area.

- **Two PTSs**
  - Originally, each PTS rescues patients in the non-MCI area.
  - After the MCI happens, each PTS decides whether to follow the SoS-level goal (move to the MCI area) or the CS-level goal (move to the non-MCI area).

- **Patients**
  - Each patient has the three states: occurred, rescued, and dead.
In DIR-SoS, CSs simply follow the order from the SoS-level manager.
  - If there is a patient in the MCI area, a PTS serves the MCI patient first.

1. 
   \[ \text{saved}_\text{MCI} + \text{dead}_\text{MCI} < \text{MCI}_\text{MAX} \Rightarrow \text{moveTo}_\text{MCI}; \]

2. 
   \[ \text{saved}_\text{MCI} + \text{dead}_\text{MCI} = \text{MCI}_\text{MAX} \Rightarrow \text{moveTo}_\text{non}_\text{MCI}; \]
**Modeling - Acknowledged SoS**

- In ACK-SoS, CSs decide their own actions using \( \text{rate}_\text{SoS} \) in PRISM.
- The difference between DIR-SoS and ACK-SoS is the decision making process.
  - Based on the \( \text{rate}_\text{SoS} \), a CS in ACK-SoS decides autonomously its own action.
  - If \( \text{rate}_\text{SoS}=0.9 \), the PTS will move to the MCI area with a 90% probability.

\[
1 \ [\text{true}] \rightarrow \text{rate}_\text{SoS}:(\text{moveToMCI}=1) + (1-\text{rate}_\text{SoS}):(\text{moveToMCI}=0);
\]

*rate\_SoS*: affinity to the SoS-level goal (move to the MCI area)
In COL-SoS, each PTS decides its action using both $\text{rate}_\text{SoS}$ and $\text{rate}_\text{info}$. The difference between ACK-SoS and COL-SoS is the decision making of a PTS.

- In COL-SoS, the information of the MCI area can be degraded.
- If $\text{rate}_\text{SoS}=0.9$ and $\text{rate}_\text{info}=0.3$, the PTS will move to the MCI area with a 27% probability.

\[
\text{1} \quad \text{true} \rightarrow (\text{rate}_\text{SoS}*\text{rate}_\text{info})(\text{moveToMCI}=1) \\
\quad + (1-\text{rate}_\text{SoS}*\text{rate}_\text{info})(\text{moveToMCI}=0);
\]

* $\text{rate}_\text{SoS}$: the affinity to the SoS-level goal (move to the MCI area)
* $\text{rate}_\text{info}$: the quality of the information for the MCI area
Verification Property

- Verify whether the **SoS-level goal** is achieved or not.

```
Our SoS-level goal is to save at least 225 out of 250 patients (>=90%).
```

- **Environmental settings**
  - Patients in the MCI area appear *more frequently than* patients in the non-MCI area.
  - The probability of death in the MCI area is *five times higher than the* probability of death in the non-MCI area.
  - SMC takes 10,000 samples and verifies the property with 99% confidence.
Verification Result - Types of SoS

- The degree of authority (i.e., SoS type) affects the SoS-level goal achievement.

- The better the authority to CSs, the higher the probability of achieving the SoS-level goal.
In a collaborative SoS, the increased possibility of acquiring MCI-related information \((\text{rate}_{\text{info}})\) affects the probability of goal achievement.

- The better the information, the higher the probability of achieving the SoS-level goal.
Lessons Learned

- Different types of SoS can be modeled via probabilistic models and variables.

- Statistical Model Checking (SMC) is a proper way to verify the SoS-level goal achievement in a quantitative way.

- To analyze the pros and cons of types of SoS, another aspect of SoS (e.g., cost) should be added to model.
We demonstrated differences between the three types of SoS with probabilistic models.

We quantitatively verified the SoS-level goal achievement using SMC.
Future work

Extend the SoS scenario

Mass casualty Incident (MCI)

CS level goal VS. CS level goal

SoS-level Goal(s)

Add heterogeneous CSs

Directed MCI-SoS

Acknowledged MCI-SoS

Collaborative MCI-SoS

Degree of authority

Compare and analyze other SMC tools

Verification Results

Model Checker (PRISM)

SoS Model

Verify SoS model

Extend the SoS scenario

Statistical Model Checker (PRISM)

Verification Results

Add heterogeneous CSs
SE for SoS Research Group

SW Star Lab (http://se.kaist.ac.kr/starlab)

- Software R&D for Model-based Analysis and Verification of Higher-order Large Complex System (2015.03-2023.02, funded by Institute for Information & communications Technology Promotion)
  
  A. SoS Modeling & Goal Specification
  B. Model-based Statistical Verification of SoS
  C. Dynamic Reconfiguration of SoS

- S/W tools will be released as open source. Please join us!

Research Plan (2015-2023)
Q&A

- We appreciate for kind and valuable comments from reviewers.
- Thank you for listening.
module env_MCI

  total_MCI:[0..MCI_MAX] init 0;  // cumulative number of patients.
curr_MCI:[0..MCI_MAX] init 0;   // number of patients in the queue.
saved_MCI:[0..MCI_MAX] init 0;  // number of saved patients.
dead_MCI:[0..MCI_MAX] init 0;   // number of dead patients.

  // MCI patient occurs.
  [] (saved_MCI+dead_MCI+curr_MCI<MCI_MAX) ->
  (total_MCI'=min(total_MCI+MCI_OCCUR_RATE,MCI_MAX)) &
  (curr_MCI'=min(curr_MCI+MCI_OCCUR_RATE,MCI_MAX));

  // MCI patient dead.
  [] curr_MCI>=1 ->
  PR_MCI_DEAD:(dead_MCI'=min(dead_MCI+1,MCI_MAX)) & (curr_MCI'=curr_MCI-1)
  + (1-PR_MCI_DEAD):true;

  // MCI patient served by PTS1
  [serve_MCI_PTS1] curr_MCI>=1 ->
  (saved_MCI'=min(saved_MCI+1,MCI_MAX)) & (curr_MCI'=curr_MCI-1);

  // MCI patient served by PTS2
  [serve_MCI_PTS2] curr_MCI>=1 ->
  (saved_MCI'=min(saved_MCI+1,MCI_MAX)) & (curr_MCI'=curr_MCI-1);

endmodule
module PTS1

    // Case: DIR
    [serve_MCI_PTS1] saved_MCI+dead_MCI<MCI_MAX -> true;
    [serve_ETC_PTS1] saved_MCI+dead_MCI=MCI_MAX -> true;

    // Case: ACK or COL
    // s1: [0..1] init 0; // 0: CS purpose, 1: SoS purpose
    // [serve_ETC_PTS1] s1=0 -> true;
    // [serve_MCI_PTS1] s1=1 -> true;

    // Case: ACK
    // [] true -> rate_SoS:(s1'=1) + (1-rate_SoS):(s1'=0);

    // Case: COL
    // [] true -> rate_info*rate_SoS:(s1'=1) + (1-rate_info*rate_SoS):(s1'=0);

endmodule
Thank you

Managerial & operational independence of various CSs

Four types of SoS depending on the degree of authority

Modeling & verification of different types of SoS

mailto: bae@se.kaist.ac.kr