An Approach for System of Systems Executable Architecture Structure Analysis

Jing-jing LIAO and Chun-lei ZHANG

China Shipbuilding Industry Corporation No.722 Research Institute, Wuhan 430205, China

Keywords: System of systems, DoDAF, Executable architecture, Extension, Structure analysis.

Abstract. In order to obtain the structure properties of System of Systems (SoS), a structure analysis framework of SoS executable architecture based on view models of DoDAF is presented. At first, advantages and disadvantages of Petri nets, automata and event structure graphs are compared after analyzing tools of state-oriented approaches and event-oriented approaches. Then respective extensions of the referred three approaches are proposed owing to the occurred probabilities and delays of the activities of SoS executable architecture. In the end, a new framework for System of Systems executable architecture structure analysis is brought forward combining the advantage of obtaining the structure properties from executable architecture models for Petri nets and easily modeling behaviors for automata and event structure graphs. The theory basis is provided for capability evaluation by the approach proposed.

Introduction

In order to research complex system engineering problems, the U. S. military proposed the concept of SOS [1] at first. Architecture is the top level of SoS and the basis of SoS working and developing.[2] Therefore, if we don’t study the architecture of SoS, the behavior characteristic could not be deeply understood and grasped. SoS composed of related component systems which pass through all the life period operates how to produce the correct demand.[3] But SoS models built under the standard of DoDAF are always graphs, tables and matrices. Thus each architecture viewpoint is described by a static way, and could not be analyzed by dynamic simulations. Therefore, in order to show and analyze the dynamic characters, the SoS executable model should be considered, which is called SoS executable architecture. [4] And building SoS executable architecture means building concurrent, synchronous and event driven discrete event system executable models. [5]

The current simulation analysis approach lacks the support of ideal event logic structure analysis methods. And structure analysis is a kind of performance analysis approach[6] with the notable advantage of independence on the system’s initial state, and it can wholly reveal the system’s structural characters.

This paper advises an approach for the SoS structure analysis by obtaining structure properties from SoS executable architectures and then computing some sample data. That means we can describe the SoS behavior characters based on structure analysis from the top architecture design viewpoints to study the SoS’s capability.

Architecture Modeling Approaches

DoDAF defines a kind of universal way for the descriptions, denotations and operations of the Department of Defense Apartment Framework. [7,8] And it is known as a kind of normal standard for the SoS architecture top design, support and validation, which is widely applied to guide the modeling in the economic and military field.[9,10] An executable architecture means a dynamic model which reveals activity orders produced by operational nodes after consuming sources and information. However, the DoDAF executable architecture means dynamic simulation models which can be used to analyze the architecture.
Professor Levis is the first scholar who studies analyzing and evaluating complicated event systems using Petri nets. He presents the method and steps of transferring DoDAF architectures to Colored Petri nets (CPNs) in [11]. And the corresponding relation between DoDAF models and CPN models is introduced in [12]. Hence we would not describe that in this paper.

Therefore, the research on approaches for SoS executable architectures has a long way to go. This paper follows how to build SoS executable architectures based on DoDAF operation viewpoint models, and discusses the structure analysis on the SoS executable architectures.

### Comparison on Modeling Approaches

Automata, CPNs and Event Structure Graphs are all formally modeling methods based on states and events, which complement each other with different viewpoints and descriptive form.

Petri nets is a kind of formally discrete event system modeling method with concurrent and dynamic characteristics. Petri net models are executable which include two kinds of nodes named place and transition. There have been some kinds of extended Petri nets such as Stochastic Petri nets, CPNs and so on. And CPNs is a kind of high-level Petri nets with colored tokens which are folded based on Ordinary Petri nets (OPNs) and the standard Meta Language is used in it. Hence CPNs can be used in engineering system modeling. There are abundant data types which can be defined as arc functions and guard functions. So far, CPN Tools is a typical tool which can be used to build, simulate, and analyze CPNs. This kind of CPNs is executable after placed initial markings.

Automata is a kind of state transition oriented modeling method, and is a formal tool to validate systems with finite states. Timed Automata (TA) is a kind of extended Automata with real data as time variables and finite time restrained mechanisms. TA runs from the system logic correctness and time constraints, and the most outstanding study is to use TA to validate the time systems. UPPAAL is a kind of tool to model and validate time systems based on TA. UPPAAL has a favorable solution to state explosion problem.

Event structure is a kind of interaction semantic modeling method. In Process Algebra, any behavior expression can be defined as operator interpretation functions of Event structure. The most popular application is the binding event structure, in which every event could be enabled by different event without contradictions. Probabilistic Event Structure endows some probabilistic events with certain probabilities. Then we can consider performance evaluation parameters such as the probability indexes.

<table>
<thead>
<tr>
<th>Modeling</th>
<th>Petri net</th>
<th>Automata</th>
<th>Event Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Meta Language+Graphs</td>
<td>Graphs</td>
<td>Algebra Expressions+Graphs</td>
</tr>
<tr>
<td></td>
<td>No stochastic delays and no choice probabilities for transitions</td>
<td>No stochastic delays and no choice probabilities for transitions</td>
<td>No stochastic delay and no choice probabilities for event transitions</td>
</tr>
<tr>
<td></td>
<td>Focused on logic structure hierarchies</td>
<td>Focused on state transitions</td>
<td>Focused on event structure relations</td>
</tr>
<tr>
<td>Analysis</td>
<td>CPN Tools, PnetLab and so on</td>
<td>UPPAAL, Kronos, SPIN and so on</td>
<td>Seldom tools for nonexponential delays</td>
</tr>
<tr>
<td></td>
<td>Behavior statistic analysis and simulation analysis</td>
<td>Safety and time analysis</td>
<td>π operation validation and behavior description</td>
</tr>
<tr>
<td></td>
<td>Transition delay and probability simulation</td>
<td>Control message sequence graph simulation</td>
<td>Unsuitable for simulation</td>
</tr>
<tr>
<td></td>
<td>Describe state transition and firing event based on states and events</td>
<td>Describe states of systems based on states</td>
<td>Describe event logic structure and dynamic behavior based on events</td>
</tr>
</tbody>
</table>

Automata, Petri nets and Event structure graphs are three different modeling methods. According to their semantic mapping, we can ensure the behavior sequences be identical. And the extending of these three methods can be used to build SoS executable architectures and be easily developed the structure analysis of SoS executable architectures.
Stochastic Extending of Modeling Methods

SoS composed with numerous component systems are stochastic concurrent systems with multiple attributes. The importance of time and probability aspects in the modeling and analysis of stochastic concurrent systems has been widely researched. That’s because the correctness of systems is due to the time index and probability index. And the activity delays and probabilities make systems more reasonable. But Automata, Petri nets and Event Structure Graphs could not simulate the delays and probabilities of system events. Hence we need extending the modeling approaches to describe SoS more precisely.

Most of the specification approaches has a two-level structure: the level of symbolic semantics and the level of concrete semantics. For the first level, we introduce a new method to represent timed systems with general distributions—Free Probabilistic Automata (FPA), which includes clocks and probabilities. And for the second level, we show two new modeling methods—Extended Colored Stochastic Petri Nets (ECSPNs) and Generalized Stochastic Extended Bundle Event Structures (GSEBESs). The latter two approaches both have the advantages of intuitionist graphs and expressing true concurrent semantics. Besides that, ECSPNs have strong modeling abilities and many good performance evaluation results, while GSEBESs describe systems by event/action sequences instead of time specification, which can be easily used to analyze system event structures and behavior tracks.

Free Probabilistic Automata (FPA) is a kind of symbolic semantic specification approach of SoS executable architecture. There have been a lot of research results about its verification, but little on its performance evaluation has been studied. Therefore, we need to develop concrete semantic specification approached, two of which we will introduce in the next two sections-Extended Colored Stochastic Petri Nets (ECSPNs) and Generalized Stochastic Extended Bundle Event Structures (GSEBESs).

Event structure is a prominent noninterleaving model for concurrency. Bundle event structures consist of events labeled with actions—an event modeling the occurrence of its action—together with relations of causality and conflict between events. It is difficult to simulate a system’s behaviors only depending on the sequences of activities. For concurrent systems, the performance analysis is a more important problem. So it is necessary to consider some performance evaluation parameters (e.g., time metrics and probability metrics). In the following, we will obtain Stochastic Extended Bundle Event Structures by decorating events and bundles with random time and probabilities.

The SoS systems could be described by probability Process Algebra and be graphed by Generalized Stochastic Extended Bundle Event Structures based on stochastic Process Algebra to simulate the behaviors of SoS systems. And stochastic Process Algebra, which gives every activity a certain probability, develops based on time Process Algebra and probability Process Algebra.

Structural Analysis Framework

The static analysis of SoS systems is mainly composed with their structures, logic meanings and efficiency analysis, which ensure the completeness, correctness, rationality of the SoS architecture design to satisfy the requirement. According to the intrinsic dependence of the structure, logic and efficiency, we should finish the structure analysis at first, and then estimate the correctness of the logic structure of models. The remarkable advantage of structure analysis is that it is undependable of the initial states of SoS systems, and we could obtain the whole logic confirmation which reveals the behavior features of SoS.

Also, structure analysis can provide support for decisions, which can feedback the structure check and analysis results to the SoS architecture designers for the reference of revisions and improvements. And then the time and other attribute information could be added into the SoS executable architecture with correct structure to validate the system behaviors. The the SoS executable architecture structure analysis is finished.
The structure analysis mainly focuses on the structure compositions and complexities. Different syntaxes can execute the same logic process and achieve the same efficiency. But under different restrain conditions, such as funds, schedules and so on, the SoS architecture will perform differently.

FPA can simulate concurrent SoS systems which conclude activities with general stochastic distribution. The attributes of start clocks, end clocks, activities firing and probability choices all can be described by symbol transitions. FPA is a kind of model building tool for concurrent SoS. But it mainly focuses on system validation. Hence we cannot use it alone to finish the SoS system structure analysis. It is necessary to adopt essential semantic tools to assist finishing the SoS system structure analysis.

There are mature structure analysis approaches for ECSPNs. Also, some SoS properties is decided by the structures but not the initial markings, which are named the structure properties of ECSPNs. The structure properties of ECSPNs include boundedness, incidence matrixes, S (T) invariants, liveness and so on. We can validate the structure correctness by structure analysis methods of OPNs. The information conservativeness structure and loop event structure can be studied by S (T) invariants. Moreover, the familiar system structure properties conclude conservativeness, repetitiveness, security, consistency, deadlock free and so on, which can be obtained from the incidence matrixes of Petri nets.

As introduced above, the SoS executable architecture is developed as ECSPNs with Standard Meta Language, thus the study for structure analysis of ECSPNs becomes a notable difficult problem. So at first, we can use the equivalent OPNs structural analysis methods to obtain the structure properties to finish the structure analysis. This method is shown in Figure 1, which steps are as follows.

Step 1: The equivalent OPNs incidence matrix can be obtained by unfolding the incidence matrix of ECSPNs based on resolving the Standard Meta Language binded to the tokens and transitions. Then the SoS’s behaviors can be deduced by the incidence matrix of OPNs, from which S (T) invariants and trap (siphon) sets can be computed.

Step 2: The structure properties obtained in step 1 need a confirmation mechanism as a platform. The FPA models transferred from ECSPNs in the structure level can be used to validate the SoS’s behaviors. This structure transferring is based on no loss of logic structure and data information, so the validation of SoS based on FPA is on the basis of the structure transferring of ECSPNs, which focuses on the activities, transitions in ECSPNs. After validation, attributes can be added to analyze the SoS’s behavior.

Step 3: GSEBESGs are structure transferred from the DoDAF architecture and ECSPNs. After adding the attributes of probabilities and delays, we can study the SoS’s behavior, such as key events and key events paths of SoS.

Figure 1. A framework for System of Systems executable architecture structure analysis.
Summary
The SoS executable architecture can be modeled by CPNs. To represent the SoS executable architectures with general distributions and conflict activities, we introduce a kind of symbol semantics level approach—FSA, and two concrete semantics level approaches—ECSPNs and GSEBESs. Through comparing the characters of the latter two methods, we brought forward the approach of combining them to analyze the SoS’s structures, which develops as the structure analysis framework of the SoS executable architecture.

Acknowledgment
This work was supported by a grant from the National Natural Science Foundation of China (No. 61273207).

References