An asset-based development approach for availability and safety analysis on a flood alert system

Fumio Machida¹, Jianwen Xiang², Kumiko Tadano¹ and Shigeru Hosono¹
¹ Knowledge Discovery Research Laboratories, NEC Corporation, Kawasaki, Japan
² School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China
jwxiang@whut.edu.cn

Abstract— Dependability design of IT services including safety and availability analysis requires expertise and often takes long time to carry out. Efficient analysis of system dependability is thus a key to increase the productivity and quality of system development project in service provider. In this paper, we propose an approach to improve the process of system dependability analysis through asset-based development concept in which safety constraints, system designs, availability models, parameter values and empirical data are incorporated into project asset on the premise of reuse. A structure of asset for dependability analysis along with other software artifacts is presented. Through an example of availability and safety analysis of a flood alert system, we characterize the effectiveness of the asset-based approach.

Keywords—asset, availability, flood alert system, reuse, safety.

I. INTRODUCTION

Continuous improvements of the quality of service and productivity are key concerns of IT service providers. Use of cloud computing and adoption of open source software enable easier and more cost-effective IT service production. The role of IT system integrator, who has been delivering systems required by customers, is shifting toward a total solution provider making use of the commodities of IT systems. Under such circumstances, it is an impending issue for system integrators to improve the productivity of IT system integration and to focus more on the works closer to customer values, such as consultation, analysis, prediction and planning for customer’s business. On the other hand, in the process of productivity enhancement through common IT services, the quality of services including performance, availability and safety should not be sacrificed. Quality assurance is also an essential aspect of customer values.

To improve both the productivity and quality of the service, organizational assets of system integrator in terms of IT service production process need to be managed in an effective manner. An asset is composed of any artifact generated during the system integration process that is potentially useful in future projects [1]. In addition to reusable software components or libraries, an asset can include project-associated documents, designs, programs, and test cases, which are created and used in the same project. All of the contents in the asset are annotated and are linked together so that the content can be easily retrieved from an asset repository by means of a search method. Well-structured and well maintained assets enable us to reuse the past experiences and knowledge of system integration effectively in the subsequent projects.

In this paper, we propose an approach to improve system dependability through system integration project by taking advantage of the assets related to availability and safety analysis conducted in the past projects. Availability assurance of IT service is a big challenge in system design as a low-availability of service results in the loss in customer values. For mission critical systems, safety is also an important attribute of dependability, which ensures that there are no catastrophic consequences on human or environment [2]. Oversight in availability and safety risk in system design might cause unacceptable work for recovery in the later phases of the project and thus ensuring the quality and accuracy of the design in the early phase is crucial. Based on the existing availability and safety analysis approaches, we exploit the reuse of assets for system design to improve the accuracy as well as productivity. Several artifacts associated with availability and safety analysis can be incorporated into an asset in a structured manner along with other reusable software artifacts. For instance, parameter values used in the availability analysis are included into the asset so that the following projects can rely on or learn from the past experiences. We discuss the advantages of the asset-based development approach for availability and safety analysis with an example of a flood alert system (FAS) design.

The paper is structured as follows. A concept of asset-based development for system integration is introduced in Section II. In Section III, an approach to availability and safety analysis using assets are described. Section IV gives an example of dependability analysis on a flood alert system, and discusses the advantage and drawbacks of the asset-based approach. Conclusions and future work are provided in Section V.

II. ASSET-BASED DEVELOPMENT

A common system integration process is divided into phases from requirement definition followed by basic design, detailed design, implementation, integration test and delivery. Several intermediate artifacts, such as requirement specifications, design documents, programs and test cases are generated in and between the phases. These artifacts are not only essential for completing the project but also important for
building tangible knowledge of the system integrator. Such project-associated artifacts can be packaged together as an asset and should be maintained in the organization repository. Using a repository search technique, one can find the past project asset, which would be useful in several phases of other system integration projects. We call such a development process for system integration taking advantage of built asset as asset-based development process. Asset-based development is analogous to domain engineering [3] discussed in software engineering researches and recently advances in industries.

Traditional approach for software reuse in system development project often fails due to the lack of the information about the context that implicitly restricts the function of the software. Functions and behaviors of a system are typically designed under specific context, constraints and/or assumptions. Software reuse is likely to fail when the context is not consistent with original one. Documents such as requirement definitions and design specifications are helpful to understand the gap between the projects contexts. Once the difference between the contexts is clarified, engineers can judge the reusability of the software with more confidence. Packaged assets can help such a context-awareness in software reuse.

The benefits of packaged assets are not limited to reusing software programs. In system design phase, semi-formal modeling language is often used to specify its system architecture and the component interactions. UML is widely-adopted in software development projects and SysML has been used in the design of systems [4][5]. The designed model by such semi-formal languages also can be included in the asset and reused in other projects. Reusing such model enables us to reduce the complexity of system design and shorten the design process. Similarly, test program incorporated in the asset also helps improve the productivity and quality. When the unit test program for software component is stored in the asset, we can shorten the testing process as well.

III. ASSET-BASED AVAILABILITY AND SAFETY ANALYSIS

In this section, we present an approach to system availability and safety analysis along with asset-based development that makes the analysis more efficient by reuse.

A. Availability analysis

Availability is a quantitative dependability measure represented by a probability a system is in a functioning condition. While the required functions are correctly implemented in the system, unavailability of the system loses the actual values of customers or end-users. The system availability is affected by various factors including system configuration, component failure-recovery processes, maintenance schedule, administrative operations, network connectivity and so on. Therefore system design considering availability is critical for providing a dependable service.

Model-based availability analysis is an approach to quantitatively assess the availability of the system from its design. Composition of availability model often relies on experts who have knowledge of stochastic models and reliability logic. In order to correctly capture the system configuration, behavior and system environment, the model experts also require domain specific knowledge that may be obtained through the communication with domain experts. Due to this expert-dependent nature, modeling process for availability analysis is often time consuming.

Once the expert knowledge is devoted to composing the availability model in a design, the experiences and outcomes can be reused in future similar projects. For reuse purpose, we incorporate the availability model and the associated context information considered in the modeling process into the asset of the project. The model is linked to the design documents, requirements and software programs in the asset so that engineers can effectively retrieve them together. There are two aspects of benefits from reusing the availability model as described below.

1) Benefit of cross-department reuse

While a whole availability model for a system is difficult to reuse, a part of model might be reusable in another project in a different department. In system integration projects, a part of system components such as server, software, storage, sensor, is commonly used across the projects even in different departments. Therefore the part of availability model associated with a common part of the system could be reusable. However, a technical issue resulting from this approach is statistical dependency of part of the model to the external components existed in the original system. For instance, the failure probability of a software process statistically depends on the failure probability of the hosting server. Unless the part of the model is statistically independent of the other external components, we may not extract the reusable model. Extracting reusable part could be more difficult than composing new one in such case.

Component-based availability modeling approach presented as Candy [5] provides a solution to this issue by generating availability model from model components which are common parts of availability model corresponding to specific system elements. Dependency information can be specified in the semi-formal language, and thus it is easily adapted to a new system configuration. In order to take advantage of model reuse across departments, it is important to employ such a component-based modeling approach in earlier projects to build assets continuously.

2) Benefit of repetitive reuse

Regardless of cross-department or inside a department, repetitive reuse of the asset produces another benefit in terms of the accuracy of availability assessment. Model-based availability analysis relies heavily on the parameter values used in the analysis as well as the correctness of the model. When the parameter values chosen for evaluation are far from the reality, the analytical result does not provide helpful information for system design. The initial project may not have any data contributing to system availability and hence the system needs to be designed with reasonable estimates for parameter values. In the successive projects, however, the real data observed in the previous system, such as component failure rate and system recovery time, can be supplied to the parameter values for availability analysis. The empirical data may not be statistically sufficient especially in failure data, but repetitive reuse of models and data collection significantly
improves the confidence of estimation. Given the confidence intervals of input parameter values, we can also estimate the confidence intervals of availability by applying uncertainty propagation [6]. In order to improve the accuracy of model-based availability analysis, it is important to collect data for parameter values in each project, store them in conjunction with the availability model in the asset and reuse them in the following projects.

B. Safety analysis

Safety is one of the qualitative aspects of dependability which concerns the assurance of safe condition for users or environments. Safety analysis needs to be conducted by experts from different domains because the threats against safety are posed by diverse factors. Common safety analysis methods usually require team-based time-consuming procedure. Hazard and Operability study (HAZOP) is a well-known hazard analysis method, which is undertaken by a team through a series of brainstorming sessions. It is commonly accepted that HAZOP analysis is an extremely time consuming process [7]. However, part of results from HAZOP for the previous cases can be utilized for another project in the similar domain that could improve the efficiency of the analysis [8].

System Theoretic Process Analysis (STPA) is an alternative new approach to safety design in which the hazardous scenarios are analyzed in light of control structure of safety [9]. In STPA, hazard is treated as a violation of safety constraints and hazard can occur in the consequence of inadequate control actions rather than simple component failures or event chains. In the analysis, a control structure of the system and associated process models are determined. Based on the understanding of the control structure, users investigate hazardous control actions with the help of guidewords. Since the STPA does not necessarily require the system architecture, the analysis can be started in the requirement definition phase.

Although the view of safety in STPA differs from that in HAZOP, similar efforts to improve the efficiency of safety analysis can be considered. The outcomes, such as control structures, process models and identified unsafe actions can be stored in the asset for reusing in future similar projects. Repetitive reuse of the safety analysis results potentially help improve the quality of safety design.

C. Asset structure

To consider efficient reuse of artifacts related to availability and safety analysis presented in the previous sections, the artifacts should be organized in a structured manner along with other software artifacts. We introduce the information scheme that defines the category of artifacts and their conceptual relationships. Figure 1 shows a proposed information structure of asset in the form of UML class diagram.

At the topmost class, an asset is associated with a system from which the asset is created. An asset is composed of one or more artifacts, which can be design documents, source codes, test cases. In the same level, “Dependability analysis model” is defined as a type of artifact that is divided into further categories by dependability attributes like availability, reliability and safety as defined in the taxonomy [2]. For availability analysis, component-based availability analysis method is suitable for reuse as discussed in section III-A in which availability model components and associated parameter values are incorporated. Thus the “availability model component” and “parameter values” are aggregated to “availability model” as shown in Figure 1. For safety analysis, a result of HAZOP analysis is directly incorporated into the asset as an artifact of “HAZOP result” which is a child class of “Safety model”. “STPA model” is also underneath the safety analysis and it may contain control structure, process models and identified unsafe actions. The relationships between these artifacts are defined as in Figure 1. Note that any artifact has associations to other artifacts in the same asset as denoted in the self-loop attached to the “Artifact” class (See Figure 1). For example, there should be an association between availability model and design document since availability model is constructed from the system design model. Similarly, STPA model might have an association to requirement documents when hazardous conditions are specified in the requirements.

![Figure 1. Structure of asset including dependability analysis model](image)

The efficiency of asset-reuse depends on search capability, which exploits the meta-data or tags assigned to artifacts. Open Services for Lifecycle Collaboration (OSLC) Asset Management specification [10] provides a standardized notation method for meta-data of artifacts in an asset. According to the OSLC, any associations between artifacts are represented by linked data, and hence artifacts can be traversed and investigated systematically by asset management tools.

IV. A CASE STUDY: FLOOD ALERT SYSTEM

To show how the asset-based availability and safety analysis works in practice, a design of Flood Alert System (FAS) is discussed in this section. We are witnessing an increasing trend of flood disasters especially in cities in Asian countries partly caused by climate changes and rapid urbanizations. FAS provides a proactive measure to such flood disasters by disseminating alert messages prior to a potential flood via Short Message Service (SMS). The people who
receive the alert SMS are able to prepare for the predicted flood by for example going out of the dangerous zones.

Orchard road in Singapore city is one of the vulnerable areas to flood because of its characteristic landscape and pluvial climate [11]. Against repetitive flood damages encountered in recent years, an FAS has been introduced by the city government. However, it was reported that SMS were not delivered to the users in a timely manner and consequently the FAS did not help prevent flood damages [11]. In addition, users of SMS alerts are allowed to subscribe only one server, which could become a single point of failure of the system. Indeed, this example exhibits the importance of safety and availability analysis of disaster prevention systems such as FAS to accomplish their primary goal (i.e., protecting citizen and important property from flood disaster).

System providers having experienced enterprise system integrations are not familiar with FAS development. The parts of experiences of dependability design for enterprise systems, however, could be useful even in different application domain. Here we focus on availability and safety analysis of FAS and compare the two approaches; i) analysis from scratch and ii) analysis with asset.

A. Safety analysis for FAS

First we consider a safety analysis, which is assumed to commence in requirement definition phase to unveil unsafe scenarios in the early stage of the project. A major hazard to be considered in FAS is that a flood damages the citizen or important properties in the city. Any scenarios causing hazardous state need to be identified so that FAS properly achieves its mission. STPA is used for this purpose that allows us to find the unsafe control actions resulting in hazard and derive design requirements to avoid the unsafe control actions.

1) Analysis from scratch

When the project is still in a proof-of-concept phase or there is no asset available in the company, the analysis needs to start with a blank sheet of paper. To apply STPA, first we need to construct a functional control structure. Figure 2 shows a core part of the functional control structure designed in which server computer controls the SMS sender with the input from the water level sensors.

![Figure 2. Control structure for safety analysis of FAS](image)

The computer server has the process model which consists of two variables: water level and SMS sender. The water level is either low, middle or high, while the state of the SMS sender is either in up, down or unknown. The basic control actions taken by the computer server are start, stop and send alert. With the functional control structure, potential hazardous control actions are investigated using guidewords; a control action a) is not provided, b) is provided, c) is provided wrong timing, and d) is stopped too soon or applied too long [9].

The obtained hazardous control actions can be recorded in the table as shown in TABLE I. As can be seen in the table, undesirable control scenario encountered in the Orchard road’s case is identified as “Alert mail is delivered too late” in the row of “send alert”. All the identified hazardous control actions should be addressed in the safety requirements so that the subsequent system design ensures the safety constraints.

![TABLE I. HAZARDOUS CONTROL ACTIONS IN FAS](image)

<table>
<thead>
<tr>
<th>Control action</th>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Too early/too late</th>
<th>Wrong order</th>
<th>Stopping too soon/too long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>SMS sender is not started</td>
<td>N/A</td>
<td>SMS sender starts too late</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td>N/A</td>
<td>SMS sender downs before alert sending</td>
<td>SMS sender downs before alert sending</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Send alert</td>
<td>Alert mail is not delivered N/A</td>
<td>Alert mail is delivered too late</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2) Analysis with asset

Next we consider the case that the system provider has no experience in FAS integration but has experiences in other system integration. Even in this case, safety analysis process can be assisted from the previous experiences in different projects when the knowledge is maintained as asset in the organization. Suppose the provider has assets of factory monitoring system that aims to supervise a chemical factory and is designed to deliver alert messages when the temperature exceeds the predetermined threshold. The control structure of the monitoring system and the process model for alert senders are shown in Figure 3.

![Figure 3. Control structure for factory monitoring system](image)

We can apply the control structure and the process model for the monitoring system to conduct STPA analysis for FAS as they have similar functional components with subtle differences in the terminology. Asset query technique must support structural search capability to identify such a functional similarity between different domains. Once the retrieved control structure and the process model are adjusted into the context of FAS, we can derive the same hazardous control actions as obtained in TABLE I. Compared to the analysis without asset, the times and efforts to produce the control structure and process model are significantly reduced.
Moreover, the unsafe control actions identified by STPA in the previous project can also be reused. When the factory monitoring system has a safety-critical mission and the STPA is conducted for safety analysis purpose, the identified unsafe actions potentially are hazardous actions in FAS as well. "Alert mail is delivered too late" can be identified easily as it also causes hazardous consequences in the factory monitoring system. In brief, the asset-based approach can identify the same hazard action in a reduced time for analysis.

B. Availability analysis for FAS

Availability analysis is conducted in the system design phase after basic system architecture is determined. A system architecture of FAS consists of sensors to monitor the water level, servers collecting the sensor data and GSM module to send SMS messages [12]. The server hosts a database server to maintain the telephone numbers of subscribers. Each component is essential for providing system functionality. The system architecture can be modeled by SysML as shown in Figure 4 where MySQL server is used for a database server and gnokii [13] is used for GSM module. SysML Internal Block Diagram (IBD) is used to represent the layers of system architecture including logical functions, software processes and hardware components. In the FAS example for Orchard road, seven sensors are installed in the canal. There is no redundancy in the server, database and GSM modules.

![Figure 4. A system architecture of FAS in SysML block diagrams](image)

1) Analysis from scratch

In order to predict the availability of the FAS, redundant configurations as well as the reliability of system component should be considered from its design. Candy [5] enables us to translate SysML-based system design into availability model to be analyzed. However, if there is no model components available for FAS, system engineers have to define model components for each SysML block or use the common model component. The translated model components are then assembled together and the composed availability model is to be analyzed with parameter values such as MTTF and MTTR of system components. Figure 5 shows results of numerical analysis on the composed availability model, where the availability of FAS is plotted by varying the recovery rate of database process (from 0.5 hour to 10 hours) with different MTTF of the database process (3 months, 6 months or 1 year) under the assumed parameter values.

The parameter values used in the analysis are estimates by system engineers or real values observed in test systems. Since there is no reusable asset in terms of data for availability analysis, the accuracy of the estimation highly rely on the assumed parameter values.

<table>
<thead>
<tr>
<th>FAS availability</th>
<th>Parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MTTF</td>
</tr>
<tr>
<td>Sensor process</td>
<td>1 year</td>
</tr>
<tr>
<td>Server process</td>
<td>4 months</td>
</tr>
<tr>
<td>Database process</td>
<td>3 months</td>
</tr>
<tr>
<td></td>
<td>6 months</td>
</tr>
<tr>
<td>GSM module</td>
<td>3 months</td>
</tr>
<tr>
<td>Sensor device</td>
<td>10 years</td>
</tr>
<tr>
<td>Host server</td>
<td>5 years</td>
</tr>
</tbody>
</table>

![Figure 5. Availability analysis results by varying database server recovery rate with different MTTF for the database process](image)

2) Analysis with asset

Asset can contain availability model components in association with SysML block which represent a specific system element. Once availability model component is defined, it can be used in any project using the same system components as far as the role and behavior of the components are not changed. For example, an availability model component for MySQL server has been defined in enterprise three-tier web application system [5] that can be used for FAS case as well. Water level sensors and GSM module are not defined in the web application system, but the corresponding availability model components can be defined in other types of systems. The benefits of component-based approach become more significant as many availability model components are defined.

Besides the enhanced efficiency of model composition, the accuracy of availability analysis might be improved due to the data for parameter values recorded in association with the defined model components. As mentioned in the previous section, repetitive reuse of the model components and collecting data for parameter values into asset improve the statistical confidence of availability estimation. For instance, whenever we encounter database process failure, the observed time to failure from the time process starts can be recorded in the asset, which is used to estimate the MTTF of database process. Assume that the time to failure of database server follows exponential distribution with MTTF=\( \hat{m}_d \) and let \( \hat{m}_d \) be the sample mean of MTTF from \( n \) observations. 100(1 – \( \alpha \))% confidence interval of MTTF is given by

\[
\hat{m}_d \pm \frac{2n}{\chi^2_{2n;1-\alpha/2}} \left( \hat{m}_d \right) < \hat{m}_d < \frac{2n}{\chi^2_{2n;1-\alpha/2}} \left( \hat{m}_d \right)
\]

where \( \chi^2_{2n} \) represents the \( \chi^2 \) distributions with \( 2n \) degrees of freedom. Figure 6 characterizes how the 90% confidential interval of MTTF is narrowed down by the increased samples assuming that the computed \( \hat{m}_d \) is equal to 6 months regardless of the number of samples. The confidence interval shortens significantly by one additional sample especially when the number of samples is small.
C. Discussion

The advantages of the asset-based approach on FAS example can be summarized as below. In safety analysis, parts of outcomes in previous safety analysis such as control structure, process model and identified hazardous actions can be reused that make the analysis faster than analysis without asset. The efficiency of availability analysis can also be improved by reusing model components with the help of a modeling framework like Candy. In addition, asset-based approach is beneficial to improve the accuracy of the availability estimation owing to increased samples for model parameters.

On the other hand, the approach with asset might have the following issues. In safety analysis, in particular for STPA, it is argued that the traditional analytical reduction is not suitable for analyze system safety because the safety is considered as an emergent property which is not being able to be defined solely in terms of properties of the individual components [9]. The asset-based approach partly owes reduction approaches as control structure and process models are treated as reusable components in safety analysis for system using similar components. Therefore asset reuse without thinking an entire system view might cause oversights of hazards to be identified. Although the safety analysis process could be enhanced when the asset is relevantly reused, the approach itself has risk to promote the traditional reduction approach.

In availability analysis, since the analysis approach relies on the availability model component, the definition and maintenance of model components can become an overhead. The use case and associated behavior of system components are not always identical as they are defined in the availability model components. Defining a common model component which is applicable to many cases is not easy. In addition, maintaining the availability model components becomes problematic when the components necessitates updates according to the change of corresponding system elements. With these issues, when a large part of availability model components is not yet defined, composing availability model from scratch can be more efficient than the asset-based approach.

Overall, the drawbacks discussed above are related to the operation how the asset-based approach is implemented and managed in the organization. In addition to install an asset repository and store the artifacts for dependability analysis, it is important to add accurate and concise context data to each artifact. Educational programs, guidelines and the processes are also important to reuse the asset effectively and correctly.

V. Conclusion

Asset-based development is an important practice for system integrator to improve their productivity and quality of the service. Each phase of system integration process can gain benefit from the reusable asset built from the past project experience. In this paper, we propose an approach to incorporate safety and availability analysis into an asset-based development process. Through the FAS design example, we show how safety and availability analysis can be improved by reusing the asset obtained in the previous FAS project or other projects using common system components.

In order to effectively reuse the models, data, and documents for safety and availability analysis in a system integration process, the related content needs to be organized in a structured asset framework as studied in service production process [1] and software production engineering [14]. Defining a common process for safety and availability analysis in adoption to the framework will be addressed in future work.

REFERENCES