XSTAMPP: An eXtensible STAMP Platform As Tool Support for Safety Engineering

Asim Abdulkhaleq, Stefan Wagner
Institute of Software Technology
University of Stuttgart, Germany
{*Asim.Abdulkhaleq, Stefan.Wagner*}*@informatik.uni-stuttgart.de*

***Abstract*—STPA (Systems-Theoretic Processes Analysis) is a new hazard analysis technique based on STAMP. STPA is already being used in different industrial domains (e.g. space, aviation, medical or automotive). To support the application of STPA and make using STPA more efficient, we developed an open tool called A-STPA. However, the current usage of A-STPA by safety analysts in different areas shows a number of shortcomings in terms of documenting unsafe control actions, drawing different levels of control structure diagrams, documenting the causal factors in STPA Step 2 and supporting the application of STPA in different areas.**

In this paper, we present an extensible STAMP platform called XSTAMPP as tool support designed specifically to serve the widespread adoption and use of STPA in different areas, to facilitate STPA application to different systems and to be easily extended to include different requirements and features. Moreover, XSTAMPP has the potential to be extended in the future to support the application of CAST for accident analysis. We believe that XSTAMPP is a useful first step toward establishing a base platform to support the application of STAMP methodologies in different domains.

***Keywords*-STAMP; STPA; CAST approach; safety analysis; Eclipse**

I. INTRODUCTION

STAMP (Systems-Theoretic Accident Model and Processes) [1] is a modern approach to safety engineering and a representative of modern, system-theoretic accident models that promise to overcome the problems of traditional safety analysis techniques. STAMP strives to provide a deep understanding about the possibility of accidents occurring in systems. The STAMP approach can be divided into two different analysis methodologies. While STAMP acts as an underlying theory, the methods STPA (Systems-Theoretic Process Analysis) and CAST (Causal Accident Analysis based on STAMP) are to be practically used for safety analysis. STPA is designed for safety analysis in the system development and operation stage; the goal here is to identify hazards existing in the system and providing so-called safety constraints to mitigate those hazards. CAST is designed for accident analysis; the goal here is to identify causal factors which lead to an accident.

Recently, STAMP methodologies have been successfully applied in different areas in industry (e.g. STPA for security [2], Interval Management in NextGen [3], Nuclear Power Plants [4], U.S. Coast Guard Aviation Mishap [5] and Engineering Financial Safety [6]) to address different kinds of contributing factors to accidents in the different application areas.

Over the last years, STAMP has received increasing attention. Many experiences of applying the STAMP methodologies to different systems in different areas have been documented. The increase in the usage of STAMP methodologies has fostered the need for developing a tool support to assist safety analysts in performing STPA as well as CAST.

A. Problem Statement

A-STPA was our first attempt to implement STPA activities. A-STPA is already being used by safety analysts in different industrial domains. However, the current practices in using A-STPA face considerable hurdles: 1) STPA has different application areas, 2) A-STPA was developed based on the basic steps of STPA, and 3) the architecture of A-STPA is not extendable to include new requirements and further improvements. Consequently, these obstacles prevent A-STPA from supporting the application of STPA in different systems in different areas as well as to extend it to support the application of CAST.

B. Research Objectives

The overall objective of our research is to develop an extensible platform support for STAMP methodologies (STPA and CAST) to encourage the widespread adoption and use of STAMP by safety analysts in different applications areas. In particular, we aim to provide a base platform for STPA that could be easily extended in the future to include CAST and new requirements of safety analysts.

C. Contribution

The contribution of this research constitutes an extensible platform based on the A-STPA tool. XSTAMPP is built to be flexible to be extended by including different user interface editors for the STAMP components and to be used by different users in different application areas. The platform gives safety analysts different views on an STPA hazard analysis. Furthermore, the new platform can be used during the application of STAMP methodologies to different systems in industry.

II. Background

A. Existing Tool Support

There exist three different tools supporting STPA which have been developed and were introduced in the 3rd STAMP Conference 2014 at MIT: 1) STPA Tool, 2) SafetyHAT and 3) A-STPA.

*1) **STPA Tool**:* Suo and Thomas [7] developed a tool called STPA Tool to automate Thomas's extended approach [8] to STPA step 1 of identifying the unsafe control actions based on the combination of process model variables. The prototype of the tool allows users to specify hazards, draw the safety control structure and perform STPA step 1 to generate the context table templates automatically based on the control structure and generate XML files for storing analysis results and interoperation.

*2) **SafetyHAT**:* Hommes [9] developed a tool called SafetyHAT as transportation systems safety hazard analysis tool to facilitate using STPA in the transportation domain. SafetyHAT was developed for a specific project to include transportation-oriented guide phrases and causal factors that tailor the STPA method to transportation systems. SafetyHAT guides safety analysts through the preparatory and analysis steps of STPA by providing a streamlined data entry process, directing analysts through STPA with a wizard-like format with preloaded transportation-specific guidewords and enabling customization for other domains.

*3) **A-STPA**:* We developed A-STPA [10] as an open tool based on the Eclipse platform to implement the activities of STPA and support its application. We developed A-STPA to make using STPA more efficient, to simplify and implement the major STPA steps that will reduce the time and effort in conducting STPA analysis for complex systems. In A-STPA, we have implemented the main STPA steps as proposed by Leveson. A-STPA allows the safety analyst to perform all STPA activities e.g. document the fundamentals of analysis, draw a control structure diagram, document unsafe control actions and safety constraints, draw the process models and document the causal factor analysis.

To date, A-STPA has been used by a hundred of safety analysts in 45 countries[1]. However, there are many issues that prevent A-STPA to support the application of STPA in different domains. The reasons behind that are: 1) We developed A-STPA based on the basic procedure of STPA and included only basic notations of STPA; 2) An extended approach to STPA was proposed by Thomas [8] as an alternative way to document the unsafe control actions based on process models and more safety analysts want to document their results in A-STPA based on this approach; and 3) there exist two different ways to document the causal factors in the STPA Step 2 which are: a) based on the components in the control structure diagram; or b) based on each unsafe control action. However, up to now, it is not clear how to document the casual factors and there is no systematic way to identify causal factors and build scenarios in STPA Step 2.

III. A-STPA Shortcomings

In the following, we itemized the major shortcomings of A-STPA in terms of extensibility, functionality, designing and editing issues.

***Extensibility Issues**:* The major extensibility issues of A-STPA are: 1) The A-STPA navigation cannot be extended to include a new user interface editor, and 2) the A-STPA architecture does not support to be extended by plug-in libraries or integrated with other existing tools.

***Designing Issues**:* The main design shortcomings of A-STPA are: 1) The workbench of A-STPA is specified only to show one user interface view in the workbench UI, and 2) A-STPA does not have a project explorer to allow safety analysts to create or open more projects in the workbench UI.

***Functionality Issues**:* The main functionality issues of A-STPA are: 1) A-STPA does not allow drawing additional levels of the control structure diagram (hierarchical and detailed diagrams), 2) A-STPA does not implement a context table for all process models variables, similar to Thomas' extended approach, 3) A-STPA does not allow safety analysts to document causal factors and refine safety constraints based on unsafe control actions which are identified in STPA Step 1, and 4) A-STPA does not give safety analysts (e.g. a safety management analyst) the capability to draw sub-blocks within blocks in the control structure diagram.

***Editing Issues**:* It is difficult to edit a large number of unsafe control actions in the unsafe control action table in A-STPA. The unsafe control action table in A-STPA seems beneficial only for a small number of unsafe control actions.

IV. The STAMP Platform Overview

Considering the aforementioned problems and shortcomings, we have improved the architecture of A-STPA to an extensible platform. We developed XSTAMPP as an open source, plug-in-based, extensible software platform using the Eclipse Rich Client Platform (RCP)[2] which makes our platform easier to extend and to integrate independent components.

A. XSTAMPP Architecture

As shown in Figure 1, the XSTAMPP platform architecture mainly consists of four components:

***STAMP Components**:* The main components which are used during the application of the STAMP/STPA approach: 1) STAMP data lists (e.g. Hazards list, accidents list, system goals and design constraints, safety requirements, corresponding safety constraints and control actions), 2) STAMP

[1] http://sourceforge.net/projects/astpa/files/stats/timeline

[2] http://www.eclipse.org/

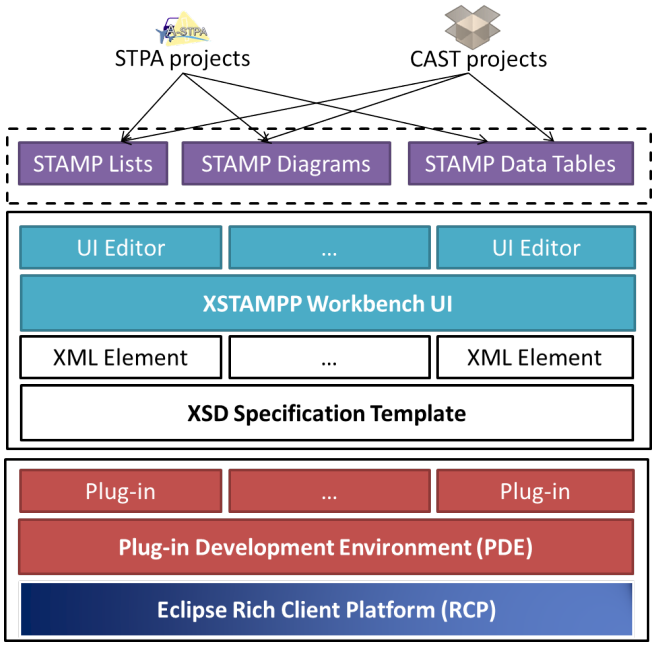

Figure 1. The XSTAMPP Architecture

diagrams (e.g. hierarchical and detailed safety control structures, and process models diagram); and 3) STAMP Tables (e.g. unsafe control actions table and causal factors analysis table).

STAMP Components Editors: As an external representation, each STAMP component is represented by an independent Eclipse user interface editor which is tightly integrated into the platform workbench UI. Each editor allows a safety analyst to edit a STAMP component in a separate user interface.

Workbench User Interface: The workbench UI contains the infrastructure for views and UI editors. All UI editors of STAMP components, views and perspectives are located in the Workbench UI.

XSD Specification Template: As an internal representation, each STAMP component editor is always associated with an XML element that documents the input data from a safety analyst in the user interface editor. All XML elements will be saved to and restored from a saved XSD[3] file with extension *.haz for a whole project.

Plug-in Development Environment (PDE): PDE provides custom extension points which can be extended with new software components. A software component called a plug-in is a component that provides a certain type of service within the context of the Eclipse workbench[4].

Eclipse Rich Client Platform (RCP): RCP provides an inherently extensible application framework that allows the seamless integration of independent software modules into a software application.

[3]http://www.w3.org/XML/Schema
[4]https://eclipse.org

B. New Features of the STAMP platform

The XSTAMPP platform has the same major functions of A-STPA as well as including the following additional main functions:

- Allows safety analysts to create and open more projects in the project explorer.
- Enable safety analysts to add new user interface views to the project explorer to customise his/her project based on the domain of the project.
- Provide alternative template views for additional requirements such as a view of hierarchical control structure or a template for the context table of process model variables.
- Open and arrange different user interface editors and views in the workbench.
- Integrate, combine and update easily by additional plug-in libraries.
- Export the whole project data as a PDF file and each individual user interface view as an Excel sheet or a JPEG image.

C. Design and implementation

The following is a description of the implementation and design details of the STAMP platform:

From the implementation point of view, we developed the STAMP platform based on the Eclipse RCP platform and XSD specifications to facilitate versioning, backup and possible future integration with other tools. Built upon RCP, XSTAMPP provides core functionality that makes it easier to extend in the future. The new architecture of our platform supports to add new plug-ins into the workbench UI. We developed each STAMP component editor as a plug-in which can be easily integrated and extended. For each STPA component, we also provided an XML element template which acts as an internal representation of the STAMP component data. These features make our platform easy to extend in the future and to implement new requirements and extensions for STPA and CAST as well.

From the design point of view, the STAMP platform allows the safety analyst to create and open many projects by a New/open project wizard. The current version of XSTAMPP supports only to create and open STPA projects. However, XSTAMPP has a potential to include the CAST project as well. Each project will be viewed in the project explorer as a tree which contains the basic components of the main steps of STPA (figure 2). For instance, an STPA project will appear with three sub-trees which are: a sub-tree of the fundamentals of analysis (e.g. system description, accidents, hazards), a sub-tree for the control structure diagram and a sub tree the STPA data tables e.g. unsafe control actions and causal factors tables.

From the functionality point of view, the new platform allows safety analysts to select and add new views to his/her project explorer such as a view of the hierarchical control structure, a view of the control structure diagram at a detailed level and the context table of the process model variables. Unlike A-STPA, the new STAMP platform allows the safety analysts to open different user interface editors in the platform workbench at the same time, order and manage them in one view. That provides a safety analyst with the capability to view many user interface editors of the project in the workbench. Furthermore, the new platform enables the safety analysts to export the results of analysis in different formats such as PDF, JPEG or Excel for a whole project or for each user interface view.

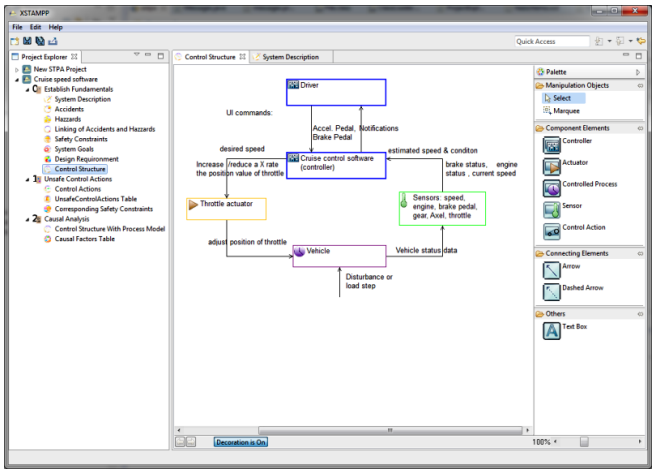

Figure 2. The Main Workbench UI of STAMP Platform

During developing XSTAMPP, a big challenge that we faced was in reusing the A-STPA code and adapting all A-STPA functions which implement all necessary functions of STPA. This challenge is addressed in the first version of XSTAMPP.

V. Conclusion and Future Work

We have developed an extensible platform support for applying STAMP methodologies in different industrial environments based on our earlier tool A-STPA [10]. Our platform aims at implementing the steps of STAMP/STPA to serve different safety analysts in different industry areas in performing STAMP/STPA methodology. The STAMP platform has a potential to implement the activities of STAMP/CAST. We developed the STAMP platform based on the Eclipse Rich Client Platform (RCP) as an extensible software platform that can be extended in the future with new requirements and improvements to STAMP methodologies. The platform is implemented in Java as open source.

As future work, we aim to benefit from the new architecture to implement the CAST steps and provide them in the upcoming version of the platform. Moreover, we aim to integrate support for safety analyst to transform the safety requirements which are derived by STPA automatically to formal specifications such as Linear Temporal Logic (LTL). Providing a formal specification of STPA safety requirements will help safety analysts to verify design models of the system against the STPA safety requirements as well as verify the software code against the STPA software safety requirements.

We have provided an early version of the new platform which included only the main A-STPA functions to get valuable feedback[5] from the safety analysis experts about the new design. The first prototype of the new platform is available online in the project repository[6].

Acknowledgments

The authors would like to express their gratitude to Lukas Balzer who worked with us to improve and build the STAMP platform. We are grateful for his effort and time.

References

[1] N. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*, ser. Engineering Systems. MIT Press, 2011.

[2] W. Young and N. Leveson, "Inside risks-an integrated approach to safety and security based on system theory: Applying a more powerful new safety methodology to security risks," *Communications of the ACM*, vol. 57, no. 2, pp. 232–242, 2014.

[3] C. Fleming, M. Placke, and N. Leveson, "Technical report: Stpa analysis of nextgen interval management components: Ground interval management (gim) and flight decn interval management (fim)," *MIT*, 2013.

[4] J. Thomas, F. Lemos, and N. Leveson, "Evaluating the safety of digital instrumentation and control systems in nuclear power plants," *MIT Technical Report*, November, 2012.

[5] J. Hickey, "A system theoretic safety analysis of u.s. coast guard aviation mishap involving cg-6505," Master's thesis, MIT, 2012.

[6] M. B. Spencer, "Engineering financial safety: A system-theoretic case study from the financial crisis," Master's thesis, MIT, 2012.

[7] D. Suo and J. Thomas, "An STPA tool," *STAMP 2014 Conference at MIT*, 2014.

[8] J. Thomas, "Extending and automating a systems-theoretic hazard analysis for requirements generation and analysis," *SANDIA National Laboratories report 2012-4080*, 2012.

[9] Q. Hommes, "SafetyHAT a transportation systems safety hazard analysis tool," *STAMP 2014 Conference at MIT*, 2014.

[10] A. Abdulkhaleq and S. Wagner, "A-STPA: Open tool support for system-theoretic process analysis," *2014 STAMP Conference at MIT*, 2014.

[5]http://a-stpa.limequery.org/index.php/791994/lang-en

[6]https://sourceforge.net/projects/stampp/