# Open Tool Support for System-Theoretic Process Analysis

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

***Abstract*—STPA (System-Theoretic Process Analysis) is a new hazard analysis technique which builds on STAMP, a process and accident model using concepts of system and control theory. In this paper, we present A-STPA an open tool to help transform STPA to an executable STPA which automates the activities of STPA. We develop the A-STPA tool to assist safety analysts in performing STPA. Moreover, it will give the safety analysts different views on the STPA hazard analysis process. We discuss the design of the tool and illustrate its usage. So far, it is still an early version but it can already help the safety analysts in avoiding consistency defects. We are confident that A-STPA will become a powerful tool support for STPA.**

***Index Terms*—STAMP/STPA approach; safety analysis; hazard analysis technique; tool support; Eclipse**

## I. Introduction

Hazard analysis is one of the most important elements in developing safe systems. One of the most challenging issues in safe system development today is to identify all potential failures modes as well as unsafe interactions among components of the system to ensure its safe function under all conditions. Many hazard analysis techniques have been proposed to investigate system design models and to elicit hazards and design flaws [1].

The last few years have seen an increase in activity in using ideas from systems thinking and systems theory to identify hazards of systems. STPA is one of the approaches based on these concepts [2]. The STPA approach has now been used extensively on real-world systems (e.g. Japanese Exploration Agency (JAXA) [3], space shuttle operations [4], railroad safety in China [5] and the Darlington Shutdown System [6]) in different areas and there are indications that it is more effective, less expensive and easier to use than traditional approaches. Various researchers have used STPA for analysing the hazards of complex systems in different areas in industry such as space, aviation, medical, defence, nuclear, automotive and food. This has led to increase the need for tool support which helps in automatint the STPA approach.

We present the tool we have developed as well as the way it can automate the steps of STPA approach: A-STPA. It is based on the STPA hazard analysis technique and aims to automate all activities during STPA such as establish fundamentals of analysis, define/refine safety constraints, draw the control structure diagram and edit the different tables of analysis as far as possible. We developed the current version A-STPA in a student project with third-year software engineering students. Our tool is developed in Java and built on the Eclipse platform, an open and extensible tool platform [7]. A-STPA aims to provide an appropriate tool to support for conducting STPA hazard analysis with controllability, completeness and consistency to establish safety requirements completely and consistently in the system design process.

### A. Problem Statement

STAMP/STPA is a powerful hazard analysis approach which has proven to be effective on real systems. Since STPA is a new method, so far there is little tool support specifically for STPA and its application was often done with standard office suites, which needs unnecessarily high effort and is error-prone.

### B. Research Objectives

The overall objective of our research is to better understand hazard analysis with STPA and improve its application in practice. In this paper, we concentrate on providing tool support to make using STPA more efficient. Hence, the goal is to develop tool support to automate the STPA approach as far as possible.

### C. Contribution

We present a first version of A-STPA, a software to automate and support the application of STPA. This first version contains the possibility to create all necessary lists (accidents or hazards), diagrams and links between those documents. This enables safety engineers to work efficiently with the lists and diagrams created in STPA as well as avoid defects such as inconsistencies between different documents.

### D. Context

A-STPA is an open-source tool based on the Eclipse platform which is developed as a student project in the software engineering programme of the University of Stuttgart. The student project started in April 2013 and will finish in February 2014. Our team consisted of 9 students and 3 teaching assistants.

## II. STAMP and STPA

STPA (System-Theoretical Process Analysis) is a modern hazard analysis method that is based on a new system-theoretical model of accidents (STAMP) for identifying hazards in large and complex systems. With STPA, the system

is viewed as an interaction of control loops and accidents are described as results from inadequate enforcement of security constraints in the design, development and operation. Over the last years, this approach has received increasing attention. Many experiences of applying the STAMP/STPA approach to different systems in different areas have been documented. STAMP/STPA as it was described in [8] is a top-down analysis approach that considers the unsafe interactions between software, hardware, operators, management and regulatory bodies. We support the main steps of STPA, which are described in [2], in our tool.

## III. A-STPA Overview

A-STPA is our proposed tool for automating and supporting conducting the STPA steps to help the analyst by ensuring that important information is near at hand. In this section, we will present how A-STPA supports the three steps of STPA.

We implemented A-STPA as open-source software on the Eclipse platform. A-STPA follows directly the steps from the STPA hazard analysis technique. It has the following main functions:

1) Edit the fundamentals of the analysis
2) Link the conducted information during step 1 to the other components in the next steps such as the hazards link to the accidents and safety constraints which are derived from the hazards.
3) Draw the control structure diagram
4) Edit tables such as the control actions table, unsafe control action table and causal factors table
5) Augment the control structure diagram with a process model
6) Export and import the STPA hazard analysis results

The overall structure for A-STPA is depicted in Figure 1. There are four phases: (i) We provide the STPA components data model for recording information for each component that is generated during an STPA analysis with short/long description; (ii) After editing this information, we add internal mapping rules to map the information between the STPA steps to facilitate the tracing between the three steps. For example, we link hazards to certain accidents and extract the control actions from the control structure diagram into the control action table. (iii) We tabulate the analysis information in the tables such as unsafe control action, corresponding safety constraints and causal factors. (IV) We provide a semantic control structure diagram editor in which the STPA components (e.g. controller, sensor, actuator or controlled process) can be sketched and linked. The process model can also be only augmented into the controller components. It provides semantic links between different components of the control structure diagram according to the standard control loop which was proposed by Leveson [2]. As the STPA approach is an iterative process in which the three steps can be repeated, we ensure the consistency between these three steps in our tool. If the user changes something in a previous step or update the control structure diagram, it will influence the next steps.

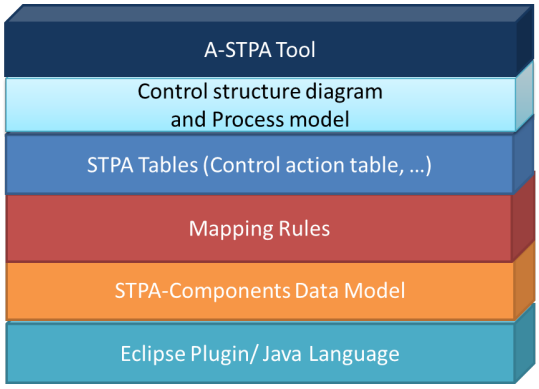


Figure 1. Overview of the Structure of A-STPA

## IV. Views in A-STPA

In this section, we describe the user interface of the tool and its usage.

The A-STPA user interface consists of several views, as shown in Figure 2. In the *welcome* view and the *main menu* view, the user can manage the A-STPA project (e.g. creating a project and editing its description or loading and saving a project). In the *STPA explorer*, a user can edit and manage all activities of the STPA steps such as creating and editing the fundamentals of a projects (e.g. managing accidents, hazards, system goals, design requirements and safety constraints). In the *accident and hazard links* view, the user can link the hazards that can lead to a certain accident and vice versa. The accident can be linked to the hazards to show how the accident could occur. The *control structure* view is used to draw and manage the control structure of the system. The components model of a control structure diagram consists of several elements as shown on the top right in Figure 2 (e.g. controller component, actuator component, sensor component, controlled process component, the links between them and the text-field component).

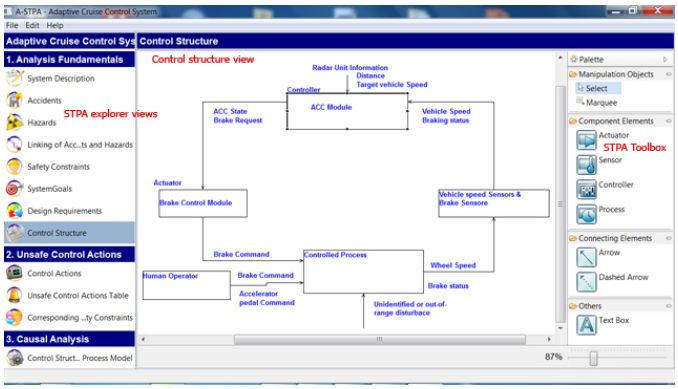


Figure 2. Control Structure Diagram View of A-STPA

In the *control actions* view, all control actions (links between components) in the control structure diagram will automatically import in the control action table. Also, the user can manage and add more control actions into this table. The

view of *unsafe control actions* is used to manage the control actions according to four types of inadequate control which are described in more detail in [2]: *Not Given, Given Incorrectly, Wrong Timing or Order and Stopped Too Soon or Applied Too Long*. The users have to evaluate each unsafe control action and determine whether it is hazardous or not as defined by system-level hazards (hazards view, step 1). If the unsafe control action leads to a hazardous state, then the user can link the unsafe control action to its related hazards.

In the view *corresponding safety constraints*, the unsafe control actions which are checked with the hazardous states in the unsafe control action view will automatically import in the corresponding safety constraints table. The user can refine and edit the safety constraints here derived from the unsafe control actions. To augment a process model into a controller, the control structure diagram, which is sketched in the control structure view, is automatically imported into this view and the user can drag and drop the process model component with its variables only into a controller component. The*causal factor* view is used to manage the causal factors for the various parts of the control structure diagram (control structure diagram view). The user can refine the safety constraints of the system and document them in the causal table. Also, the users can edit their recommendations for additional mitigations.

The results of A-STPA are documented in an XML-specifications [9] as an external representation to facilitate versioning, backup and possible future integration with other tools. The results can be also exported as a PDF file in a human-readable form to give them to other stakeholders.

## V. Challenges and Problems

A big challenge that we faced at the beginning of the A-STPA development was that all our students had no idea or previous experiences on the topic of safety and hazard analysis. To overcome this challenge, we arranged tutorials on using STPA and ran a seminar on STAMP/STPA to increase the understanding of the approach and how it might be implemented it as an executable tool.

The major issue which we are facing during the development of A-STPA is that there are no rigorous procedures, guidances and rules on drawing control loop and standard notations to document casual factors. From that, several questions arose: (1) Who can connect to whom in the control loop? (2) Can the actuator connect directly to the sensor? We spend much effort on deciding on these issues. Especially how we shall implement the drawing rules and links between different components in the control loop as well as on which way we shall document the causal analysis results during step 3 of STPA caused many discussions. It is also not clear in the current publications how the process model parameters are used during the causal factor analysis to identify inadequate control.

## VI. Conclusion and Future Work

Based on our earlier work on applying STPA to the automotive domain [10], we have developed tool support to assist safety analysts during the STPA hazard analysis process. Our tool called A-STPA aims at automating the steps of STPA. This tool is implemented in Java as an open source tool on the Eclipse platform. This paper presents the first functional version of our tool. We are currently working on improving the usability and test it intensively.

A major topic of our future work is to provide an integrated simulation of the causal analysis process (step 3) to aid safety analysts in performing the causal analysis to determine how potentially inadequate control action could occur. We aim also to conduct an online empirical evaluation study on using this tool by safety experts during their work on applying STPA to real systems in industry. Finally, as we will release A-STPA as open source, we hope to attract others in working with us on various improvements of the tool.

## Acknowledgments

We would like to thank all students who built A-STPA during their student project: Aliaksei Babkovich, Lukas Balzer, Adam Grahovac, Jarkko Heidenwag, Benedikt Markt, Jaqueline Patzek, Sebastian Sieber, Fabian Toth and Patrick Wickenhaeuser. We are grateful to Ivan Bogicevic, Daniel Kulesz and Jasmin Ramadani for supervising this project.

## References

[1] C. A. Ericson, *Hazard Analysis Techniques for System Safety*. Wiley, 2005.

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

[3] T. Ishimatsu, N. G. Leveson, J. P. Thomas, C. H. Fleming, M. Katahira, Y. Miyamoto, R. Ujiie, H. Nakao, and N. Hoshino, "Hazard analysis of complex spacecraft using systems-theoretic process analysis," *Journal of Spacecraft and Rockets*, 2013, accepted.

[4] B. Owens, M. Herring, N. Dulac, N. Leveson, M. Ingham, and K. Weiss, "Application of a safety-driven design methodology to an outer planet exploration mission," in *Aerospace Conference, 2008 IEEE*, 2008, pp. 1–24.

[5] A. Dong, "Applicaton of CAST and STPA to railroad safety in China," Master's thesis, Massachusetts Institute of Technology, 2012.

[6] Y. Song, "Applying system-theoretic accident model and processes (STAMP) to hazard analysis." Master's thesis, McMaster University, 2012.

[7] Eclipse. (2013) The eclipse foundation open source community website. [Online]. Available: http://www.eclipse.org/

[8] N. Leveson, M. Daouk, N. Dulac, and K. Marais, "A systems theoretic approach to safety engineering," in *DEPT. OF AERONAUTICS AND ASTRONAUTICS, MASSACHUSETTS INST. OF TECHNOLOGY*. MITesd, 2003, 30 October.

[9] W3C. (2013) W3C recommendation – extensible markup language (XML). [Online]. Available: http://www.w3.org/TR/REC-xml/

[10] A. Abdulkhaleq and S. Wagner, "Experiences with applying stpa to software-intensive systems in the automotive domain," *2013 STAMP Conference at MIT, Boston, USA*, 2013.