



(ITEA 3 – 17003)

PANORAMA

Boosting Design Efficiency for Heterogeneous³ Systems

Deliverable: D6.6

Automated Traceability Tool

Work Package: 6

Design Flow and Traceability

Task: T6.3

Standards-compliant traceability tools and practices

Document Type:	Software	Classification:	Public
Document Version:	Final	Contract Start Date:	2019-04-01
Document Preparation Date:	2022-03-31	Duration:	2022-03-31



INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT



Contents

1	Introduction	1
2	Traceability in Heterogeneous Environments with Eclipse Capra	2
3	Eclipse Capra 0.8.0	4
4	Eclipse Capra 0.8.1	6
5	Eclipse Capra 0.8.2	7
6	Eclipse Capra 0.9.0	8
7	Impact of the Changes on Safety Practices	9

List of Figures

3.1	The Sunburst and Matrix views introduced in Eclipse Capra 0.8.0	5
7.1	A traceability matrix showing how hazards are addressed by safety goals.	10
7.2	A sunburst view of a failure and its linked artifacts, showing that the failure is connected to a hazard and the hazard is connected to a safety goal.	11

List of Tables

1 Introduction

This deliverable summarizes the changes that Eclipse Capra has received since the start of the Panorama project. It therefore is an extension of Deliverable D6.2 [Ste22] which described the status of Eclipse Capra after half of the project runtime. In particular, the deliverable describes the changes made in four releases:

Eclipse Capra 0.8.0 released on July 5, 2020

Eclipse Capra 0.8.1 released on January 4, 2021

Eclipse Capra 0.8.2 released on May 5, 2021

Eclipse Capra 0.9.0 the current development version

Based on a description of Eclipse Capra in Chapter 2, this document reviews the most important changes in version 0.8.0 in Chapter 3, version 0.8.1 in Chapter 4, and the ones in version 0.8.2 in Chapter 5. The current development version of Eclipse Capra is described in Chapter 6. Finally, an overview of the impact of these changes on safety analysis practices is presented in Chapter 7.

2 Traceability in Heterogeneous Environments with Eclipse Capra

Eclipse Capra is a traceability management tool. It allows the creation of traceability links between arbitrary artefacts, provides features to edit them and keep them consistent, and visualise the relationships between them. This includes traceability matrices and graph visualisations that are helpful for reporting and change impact analysis. A special focus was on the support of different domain-specific languages that are used in software and systems engineering in the automotive, avionics, and healthcare domains.

In essence, Eclipse Capra allows the creation of trace links between arbitrary artefacts, as long as an adapter for these artefacts is available. Eclipse Capra currently natively supports elements from UML, SysML, AADL, EAST-ADL, as well as ReqIF models created in, e.g., Eclipse Papyrus, Eclipse EATOP, Microsoft Excel, and ProR. Furthermore, adapters for test case executions managed by a continuous integration server like Hudson or Jenkins allow traceability to concrete continuous integration runs. There is also support for source code files supported by the Eclipse Platform (e.g., Java, C, Python) and tasks from an issue tracking system such as Jira and Bugzilla as supported by Eclipse Mylyn. External artefacts for which the Eclipse Platform does not offer built-in support can also be linked if a fitting adapter is provided. Built-in capabilities allow linking to Office documents and documents hosted by Google Docs. A generic file handler and a handler for EMF models also provide capabilities for formats that currently have no native support.

Once traceability links are established, Eclipse Capra offers features to manage them. If a model element that is linked to is moved, e.g., Eclipse Capra will notify the user and allow changing the link accordingly. The same support is given for model elements that are deleted or renamed. Quick fixes are available to fix most issues in a semi-automatic fashion.

Eclipse Capra also offers visualisations of the traceability links which allows developer to traverse the relationships established through the traceability links and understand how the different artefacts are connected. This is helpful when assessing the impact a change has (e.g., which design artefacts need to be adapted when a requirement has changed?) or when trying to understand how the design artefacts in a complex development project are connected. In addition, Eclipse Capra can display traceability matrices, as requested by standards like ISO 26262.

The tool is highly extensible. The meta-model used for the traceability links can easily be adapted to a specific end-user's needs. Eclipse Capra's modular architecture allows exchanging the persistence, the visualisation, and the management modules easily. New adapters for additional artefacts can easily be added without re-compilation. This allows end-users to customise almost every aspect of the tool if needed. At the same time, we

provide sensible defaults that will allow the majority of users to use Capra out of the box without extensive configuration.

Eclipse Capra is available for download at <https://www.eclipse.org/capra>.

3 Eclipse Capra 0.8.0

The focus of the 0.8.0 release of Eclipse Capra was to add additional traceability link visualisation techniques and to increase operability with Eclipse APP4MC. For this purpose, the following new features were included into the tool:

- Added ability to create trace links from and to Amalthea models. While Amalthea models had been supported through the generic EMF handler before, this new feature now adds the ability to define that trace links should specifically include elements from Amalthea models. The change also makes it possible to navigate directly to the Amalthea model from within the Eclipse Capra visualisations.
- Added dedicated traceability matrix view. It is a common request to produce a traceability matrix that shows how one set of artifacts (e.g., requirements) are traced to another set of artifacts (e.g., test cases). The new traceability matrix view allows to easily create such matrices and even shows an overview of all defined traceability links. The matrix can be exported as a spreadsheet for easy inclusion into reports.
- Added sunburst trace model visualisation. A sunburst visualises all trace links that originate in a selected artifact as arcs located in a circle around it. Each arc represents an artifact that is traced to. Clicking on an arc focuses the view on that artifact and shows the traceability links that originate from it. In addition, it is possible to see chains of traceability links (i.e., transitive traceability links) up to a configurable depth. This allows easy navigation of the trace model in a compact and intuitive view.
- Other relevant changes make it easier to identify the traced artifacts from different editors and views in the Eclipse IDE and to include additional artifacts (e.g., Xtext documents) in the traceability links.

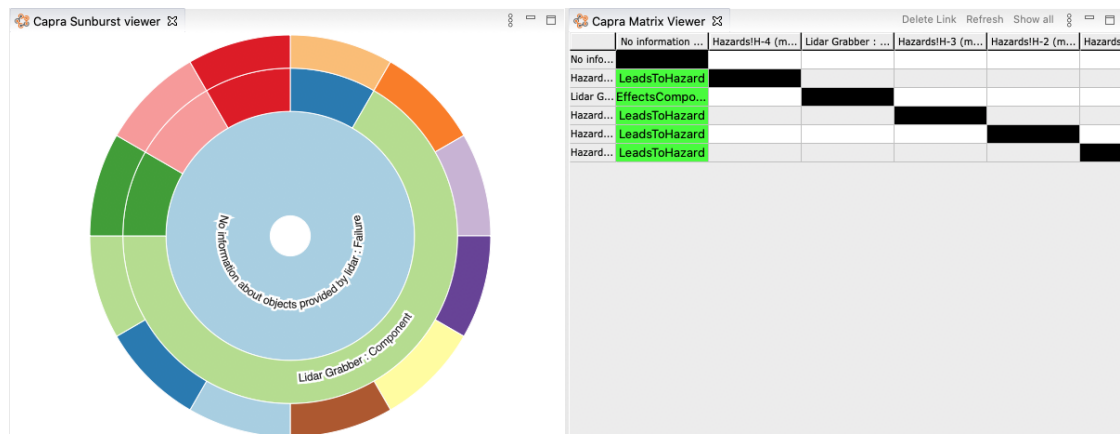


Figure 3.1: The Sunburst and Matrix views introduced in Eclipse Capra 0.8.0

4 Eclipse Capra 0.8.1

Eclipse Capra 0.8.1 was a maintenance release that focused on improving the user experience with a particular focus on traceability link visualisations. It included the following noteworthy changes:

- Improved visualisation of artifacts in views by removing line breaks. Certain artifacts can contain line breaks in their display name. For instance, Eclipse Capra can show the entire text of a requirement that is the origin or target of a traceability link. However, in many visualisations, the line breaks break the layout and make the artifact unreadable. The traceability matrix, e.g., does not support multi-line text as a header. Therefore, line breaks are now removed before displaying an artifact, leading to a better user experience.
- Improved selection of artifact handler. The artifact handler determines how an artifact is treated by Eclipse Capra. In this version, the component that selects which artifact handler to use for a certain artifact has been improved so that the most specific artifact handler gets the job (e.g., the handler for APP4MC instead of the generic one for EMF). This improves the user experience since, e.g., users now see more specific artifact descriptions.
- User interface improvements. A number of views included in Eclipse Capra have received updates to improve the user Experience. The view used to select elements in Microsoft Office documents now has icons in its toolbar instead of text. The user instructions for the user in the PlantUML view have been made clearer. Selected elements are now more reliably visualised in the traceability viewers.

5 Eclipse Capra 0.8.2

Eclipse Capra 0.8.2 was a maintenance release that focused on extending the abilities of Eclipse Capra for safety analysis and traceability maintenance. It included the following noteworthy changes:

- Added ability to create traceability links to and from Open Dependability Exchange (ODE) models which enables traceability to safety cases and other safety-relevant artifacts supported by ODE.
- Added the ability to delete traceability links from the Traceability Matrix view to support trace maintenance tasks directly in the user interface.
- Handler for MS Office files now uses relative paths to ensure that trace models are exchangeable across different users with different directory structures but using the same relative paths (e.g., when files are synchronised using version control systems).
- Trace links now differentiate origins and targets by default thus providing support for directed trace links which are necessary for some forms of change impact analysis and coverage analysis.
- Improved visualisations in PlantUML and the Sunburst view, especially when the underlying trace model is changed, when large trace models are visualised, or when the size of the user interface changes.

Eclipse Capra 0.8.2 is the version of Capra that is recommended to use with the MobSTr dataset [SKB+21].

6 Eclipse Capra 0.9.0

Eclipse Capra 0.9.0 is the current development version that has a number of additional features that focus on providing more information about traceability links and improve automation and consistency management. The highlights include:

- Added ability to store metadata about a traceability link. This allows recording additional information such as the creation date, the creator, or a rationale as a comment. The feature is fully integrated into the GUI and the metadata can be shown and edited using standard Eclipse user interface elements such as the Properties View.
- Listeners for traceability link creation and changes of traceability links allow automatically adding metadata to the traceability links when they are created and updated. This allows extensions of Eclipse Capra to store and manipulate the metadata on demand. The functionality is currently used to store the creation date and the creator of the traceability link.
- Improved handling of Microsoft Excel files shows the different workbooks of an Excel file as a table rather than as lines of text which makes it easier to work with large spreadsheets and to find relevant information to trace to.
- Improved compatibility with modern versions of the Eclipse IDE. Eclipse Capra is now using Java 11 as the standard Java runtime environment. This means that older versions of Eclipse before 2018-12 are no longer supported, but more modern version up to the current 2022-03 are. Dependencies have also been updated for full compatibility with current versions of Eclipse.

Eclipse Capra 0.9.0 makes changes to the way traceability links are stored. Therefore, existing traceability models need to be transformed to conform to the new version. This transformation is still under development and a complete version is expected in Spring 2022.

7 Impact of the Changes on Safety Practices

Eclipse Capra supports safety analysis and the construction of a safety case by providing engineers with the ability to establish and manage the relationships between the artifacts and by visualising these relationships to establish evidence that safety concerns have been addressed correctly. ISO 26262, e.g., requires engineers to provide evidence that all hazards have been addressed by safety goals which in turn have been implemented by concrete measures. Eclipse Capra allows engineers to trace between hazards, safety goals, measures, their implementation, and corresponding test cases to establish this evidence.

Concretely, the new adapter for the Open Dependability Exchange (ODE) meta-model [RFC+20; STP+15] that landed in Eclipse Capra 0.8.2 allows the creation of traces to models containing such elements. ODE is a domain-specific modeling language for dependability in general which contains all elements required for the modelling of safety cases. By adding the ability to trace to such elements, Eclipse Capra now provides full support for the safety engineering lifecycle. It is now possible to connect safety requirements with hazards, failure modes, effects, and safety cases as well as with analysis and other verification entities such as test cases.

These abilities are demonstrated in the MobSTr dataset [SKB+21]. The dataset contains artifacts in many different formats that together describe an autonomous driving system. Requirements, including safety goals and safety requirements, are stored in Microsoft Excel spreadsheets, design artifacts such as a component model are stored in UML and Amalthea, and safety artifacts, including the results of a failure tree analysis (FTA) and a failure mode and effects analysis (FMEA) are stored in ODE. The safety case is described using the Structured Assurance Case Meta-Model (SACM) which forms the backbone of ODE.

Once the traceability links between these artifacts have been created, engineers need to be able to access them and to include reports based on them into their safety cases. The additional functionalities provided by Eclipse Capra support this task. The traceability matrix view can be used to show, e.g., that all hazards have been addressed by a corresponding safety goal (cf. Figure 7.1). Likewise, the traceability matrix can be used to show that each safety requirements has associated test cases or that each component has a corresponding implementation. Eclipse Capra also offers the option to export the traceability matrix, which makes it easy to include the view as evidence into a safety report.

Likewise, the sunburst view is helpful in determining chains of traceability links. Figure 7.2 shows, e.g., the artifacts a failure from a failure mode and effects analysis (FMEA) is connected to. The arcs in this figure also indicate that all hazards the failure

	Requirements!SG-3 (mobstr-requirements.xlsx)	Hazards!H-1 (mobstr-hazards.xlsx)	Requirements!SG-2 (mobstr-requirements.xlsx)	Hazards!H-4 (mobstr-hazards.xlsx)	Hazards!H-3 (mobstr-hazards.xlsx)	Requirements!SG-5 (mobstr-requirements.xlsx)	Hazards!H-2 (mobstr-hazards.xlsx)	Requirements!SG-1 (mobstr-requirements.xlsx)	Hazards!H-5 (mobstr-hazards.xlsx)	Requirements!SG-4 (mobstr-requirements.xlsx)
Requirements!SG-3 (mobstr-requirements.xlsx)					AddressedBy...					
Hazards!H-1 (mobstr-hazards.xlsx)										
Requirements!SG-2 (mobstr-requirements.xlsx)							AddressedBy...			
Hazards!H-4 (mobstr-hazards.xlsx)										
Hazards!H-3 (mobstr-hazards.xlsx)										
Requirements!SG-5 (mobstr-requirements.xlsx)									AddressedBy...	
Hazards!H-2 (mobstr-hazards.xlsx)										
Requirements!SG-1 (mobstr-requirements.xlsx)		AddressedBy...								
Hazards!H-5 (mobstr-hazards.xlsx)										
Requirements!SG-4 (mobstr-requirements.xlsx)				AddressedBy...						

Figure 7.1: A traceability matrix showing how hazards are addressed by safety goals.

is traced to have a corresponding safety goal. Therefore, such a visualisation makes it easy for the engineer to see any gaps in the artifacts.

The improved handling of spreadsheets also supports the aims of a safety engineer. Since the data is now shown as a table rather than lines of text, it becomes easier to navigate to the relevant data within a spreadsheet that, e.g., includes safety requirements. The safety engineer can more easily identify the relevant rows to trace to. This improves the usability of Eclipse Capra and increases the engineer’s efficiency.

The automated inclusion of metadata when traceability links are created or changed also supports the safety argument. Part of a safety case is that the engineers creating the safety case have the right credentials and follow best practices. When the creator of a traceability link is stored with the link, an argument can be made that competent engineers were involved in the safety analysis and the creator metadata can be used as evidence. Likewise, the age of traceability links and comments left on the links, e.g., as the rationale why the links exists, can help in making an argument about the quality of the trace model, e.g., that traceability links have been created and updated in a timely fashion and that their existence is justified and necessary.

Overall, these changes improve the way engineers interacts with the artifacts, how they create evidence, and how they identify gaps in the set of artifacts of the system under construction. Future versions of Eclipse Capra will include further improvements, e.g., specialised tools for coverage analysis that also show which artifacts are currently not sufficiently included in the traceability model. This means that engineers will no longer have to search for gaps, but that the tool provides a way to identify them semi-automatically.

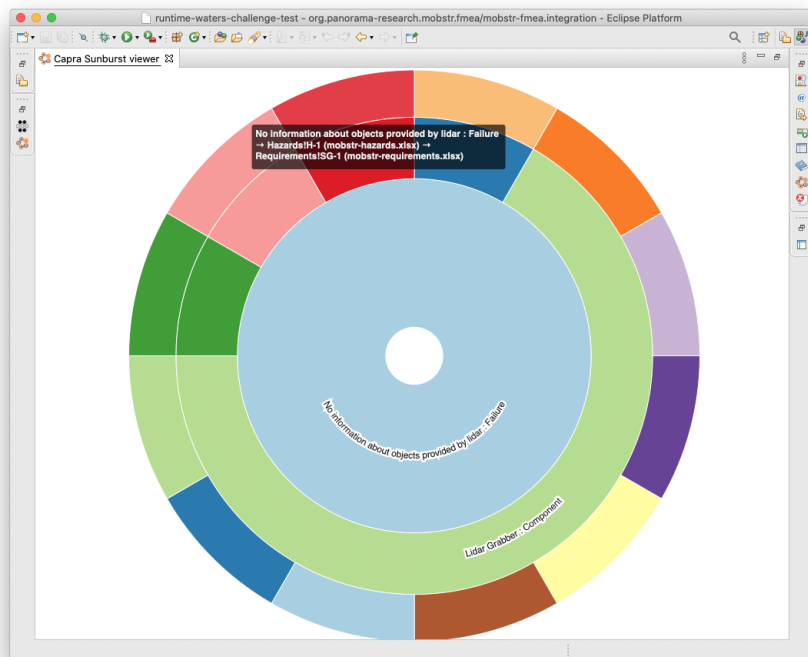


Figure 7.2: A sunburst view of a failure and its linked artifacts, showing that the failure is connected to a hazard and the hazard is connected to a safety goal.

Bibliography

- [RFC+20] J. Reich, J. Frey, E. Cioroica, M. Zeller, and M. Rothfelder, “Argument-Driven Safety Engineering of a Generic Infusion Pump with Digital Dependability Identities,” in *7th Int. Symposium on Model-Based Safety and Assessment*, ser. LNCS, vol. 12297, Springer, 2020, pp. 19–33.
- [SKB+21] J.-P. Steghöfer, B. Koopmann, J. S. Becker, I. Stierand, M. Zeller, M. Bonner, D. Schmelter, and S. Maro, “The mobstr dataset - an exemplar for traceability and model-based safety assessment,” in *29th IEEE International Requirements Engineering Conference, RE 2021, Notre Dame, IN, USA, September 20-24, 2021*, IEEE, 2021, pp. 444–445. DOI: 10.1109/RE51729.2021.00062. [Online]. Available: <https://doi.org/10.1109/RE51729.2021.00062>.
- [Ste22] J.-P. Steghöfer, “D6.2: Traceability tool,” ITEA3 PANORAMA Project, Tech. Rep., 2022. [Online]. Available: <https://itea4.org/community/project/workpackage/document/download/7375/D6.2%20Traceability%20Tool.pdf>.
- [STP+15] D. Schneider, M. Trapp, Y. Papadopoulos, E. Armengaud, M. Zeller, and K. Höfig, “WAP: Digital Dependability Identities,” in *IEEE Int. Symposium on Software Reliability Engineering (ISSRE)*, 2015, pp. 324–329.