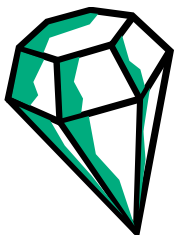
crystal_facet_uml

1.4.0

Generated by Doxygen 1.8.11

Contents

1 crystal_facet_uml user manual

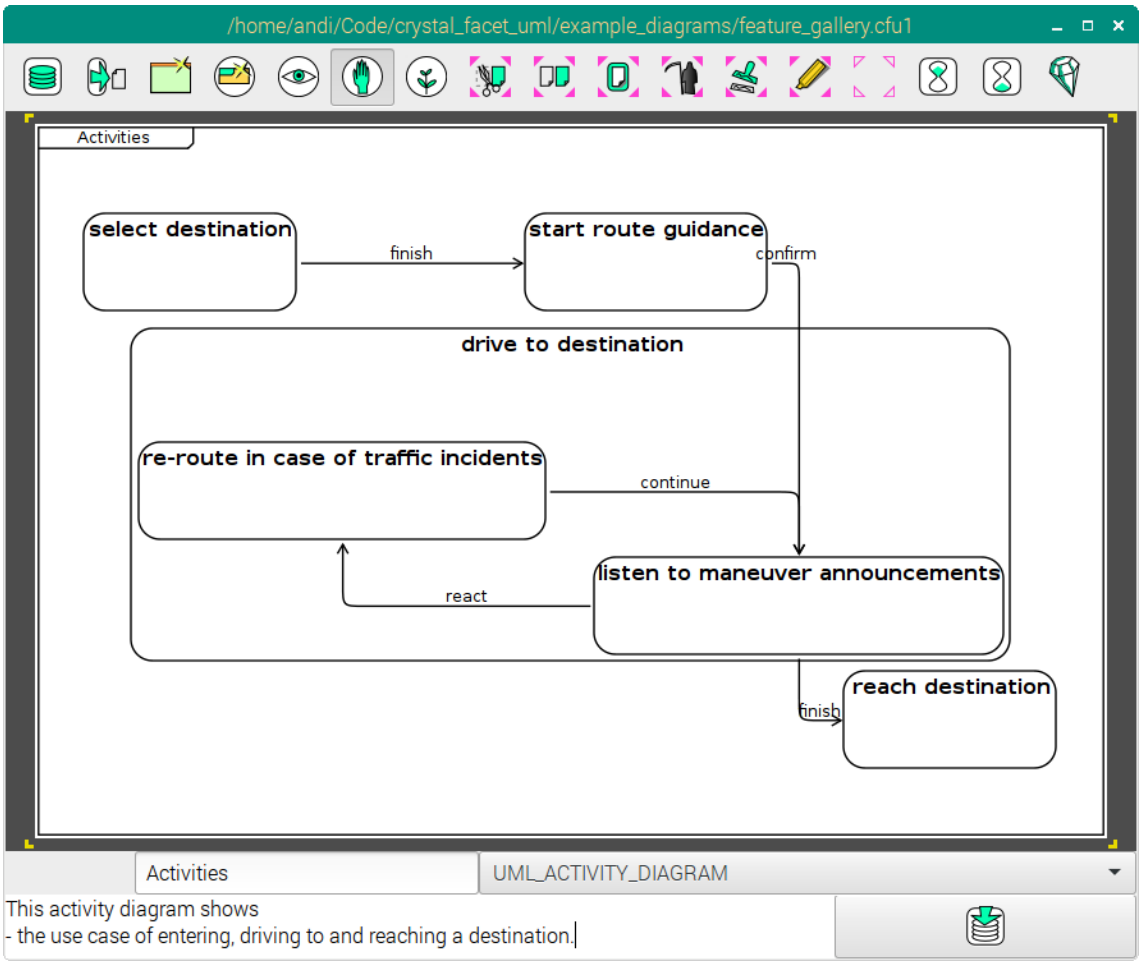


1.1 Introduction

crystal_facet_uml is a uml diagram drawing tool that creates a set of consistent uml diagrams.

If started in graphical mode, it shows a window with

- toolbar on top,
- drawing area in the center,
- element configuration widgets below and
- an optional notification bar at the bottom.

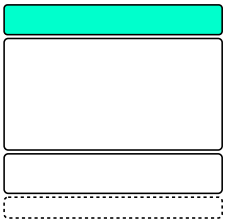


Additionally, crystal_facet_uml can be started from command line to check and repair database files. Run

```
1 ./crystal_facet_uml -h
```

to get a list of supported parameters.

1.2 Tool Bar



1.2.1 Create/Use DB

- Opens an existing database file or creates a new database file

1.2.2 Export

- Exports all diagrams to the selected folder (supported formats are txt, png, pdf, ps and svg)

1.2.3 New Window

- Opens another window on the same database.

1.2.4 New Diagram

- Create a new diagram

1.2.5 Navigate

- Navigate to parent or child diagrams

1.2.6 Edit

- Modify elements in the diagram

1.2.7 New Object

- Create elements in the diagram

1.2.8 Cut

- Cut all pink-cornered elements to the clipboard (features of classifiers are cut independantly of their corner-colors)

1.2.9 Copy

- Copy all pink-cornered elements to the clipboard (features of classifiers are copied independantly of their corner-colors)

1.2.10 Paste

- Pastes diagrams and classifiers from the clipboard to the uml model. (Relationships are not pasted) If id and name are identical to an existing element, an instance of the existing element is pasted to the diagram. Otherwise a new element is created.

1.2.11 Delete

- Deletes all pink-cornered elements. This operation may fail if marked elements have unmarked children.

1.2.12 Instantiate

- Toggles the pink-cornered classifiers between classes and instances. (Does not work for relationships and features)

1.2.13 Highlight

- Toggles the pink-cornered classifiers between yellow-marked, greyed-out and normal. (Does not work for relationships and features)

1.2.14 Reset Selection

- Resets the pink-cornered selection

1.2.15 Undo

- Un-does the last operation (Opening a database and exporting files cannot be undone)

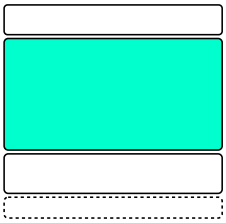
1.2.16 Redo

- Re-does the last un-done operation

1.2.17 About

- Shows version, license and copyrights

1.3 Drawing Area



Diagrams are layouted automatically. You can influence the locations of classifiers only. When adding too many classifiers or relations, auto layouting may not achieve the expected results. In many cases, splitting the diagram into two or more diagrams solves the layouting issues and at the same time improves understandability by focusing on one aspect/topic per diagram.

1.3.1 New Diagram

- To create a child-diagram, click into the children-area of the current view.

1.3.2 Navigate

- To navigate to the parent diagram, click on the parent diagram.
- To navigate to a child diagram, click on the child diagram.
- To restructure the diagram tree by shifting a child diagram up and the parent down, click on the child diagram and press F7.

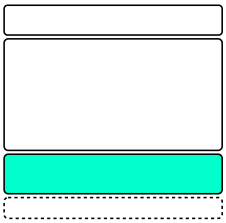
1.3.3 Edit

- To move classifiers within the diagram, press, drag and release the mouse button.
- To select the diagram or a classifier or a feature or a relationship with yellow corners, click on this object.
- To mark diagram and/or classifiers and/or features and/or relationships with pink corners, click on these objects twice.

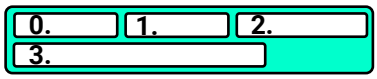
1.3.4 New Object

- To create a classifier, click at an empty space in the diagram.
- To create a child classifier, click into the white space of a classifier. Alternatively, create a classifier and a containment relation.
- To create a relationship, press on the source classifier and drag it to the destination classifier.
- To create a feature, click onto a classifier (name or border).

1.4 Element Configuration



Edit the properties of the yellow-cornered object.



- 0.: stereotype of the focused object (deactivated depending on object-type)
- 1.: name of the focused object
- 2.: type of the focused object
- 3.: description of the focused object

1.4.1 Maximum stringlengths

All strings (names, descriptions, stereotypes) have a maximum length.

Ascii characters require one, most other characters two bytes. Current sizes in bytes are:

Classifiers:

- DATA_CLASSIFIER_MAX_NAME_LENGTH = 47,
- DATA_CLASSIFIER_MAX_STEREOTYPE_LENGTH = 47,
- DATA_CLASSIFIER_MAX_DESCRIPTION_LENGTH = 4095,

Features:

- DATA_FEATURE_MAX_KEY_LENGTH = 47, (name)
- DATA_FEATURE_MAX_VALUE_LENGTH = 255, (type)
- DATA_FEATURE_MAX_DESCRIPTION_LENGTH = 1023,

Relationships:

- DATA_RELATIONSHIP_MAX_NAME_LENGTH = 47,
- DATA_RELATIONSHIP_MAX_DESCRIPTION_LENGTH = 1023,

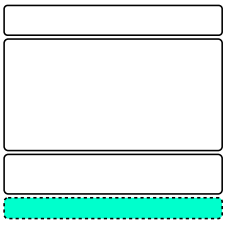
Diagrams:

- DATA_DIAGRAM_MAX_NAME_LENGTH = 47,
- DATA_DIAGRAM_MAX_DESCRIPTION_LENGTH = 8191,

1.4.2 Commit

- Stores the latest changes to the database immediately. This feature is optional, it is not necessary to explicitly save the file.

1.5 Notification Bar



1.5.1 Information

- Informs on success of an operation, e.g. an export

1.5.2 Warning

- Informs on a possible problem

1.5.3 Error

- Informs on an error

1.6 Further Information

1.6.1 Download

Find the latest version at:

https://sourceforge.net/projects/crystal-facet-uml/

https://github.com/awarnke/crystal_facet_uml

https://build.opensuse.org/package/show/home:awarnke/crystal_facet_uml

User documentation is available here:

http://www.andreaswarnke.de/crystal_facet_uml/crystal_facet_uml_user_documentation.↩
pdf

https://github.com/awarnke/crystal_facet_uml/blob/master/doxygen_build/crystal↩
_facet_uml_user_documentation.pdf

1.6.2 License

License of crystal_facet_uml is Apache-2.0. crystal_facet_uml contains sqlite which is Public Domain. Unit tests are based on embunit (MIT/X Consortium License).

Author

(c) 2016-2018 A.Warnke; Email-contact: cfu-at-andreaswarnke-dot-de

2 crystal_facet_uml modelling guidelines

2.1 Modelling Introduction

This page lists remarks on creating a software architecture and design document in general and it lists hints on getting along with the tool crystal_facet_uml.

As all tools, this program has its strengths and weaknesses. This page helps in making use of the strenghts.

2.2 crystal_facet_uml hints

- Diagrams are organized as a tree:
  - Start the root of the tree showing the major use cases/features of your software.
  - At the second level of the tree, list the main areas to be modelled, e.g.:
    - context (problem space) of development,
    - context (problem space) of operating environment, system boundary,
    - design decisions, alternatives, rationales
    - tools and dataflows of development (solution space),
    - system in operating environment (solution space).
- Put only few elements into each diagram. This increases understandability of the main purpuse of the diagram.
- Put further aspects of a topic into a separate diagram. Do not hesitate to copy an element from one diagram to the next. This is what crystal_facet_uml is good at: it keeps the model in sync.

2.3 general hints on architecture documentation

- Distinguish things that are
  - given constraints (problem space),
  - decisions, chosen and rejected alternatives and
  - the designed solution
- Names of things are crucial: If the reader gets a wrong understanding by the name of an element, a hundred correct sentences of describing text cannot set this straight again.
- Every design element needs a description, maybe a list of responsibilities: What shall this element do, what is it for? Names alone cannot explain a system part.
- Things that are similar but not the same shall be different entities when modelling. E.g. The process in which an example application runs may be different from the storage location and may be different from the software-component. These are three things: Example_App_Process (Type: Node), Example_App_Object↩File (Type:Artifact) and Example_App_SWComponent (Type:Component).

2.4 Further Information

Author

(c) 2016-2018 A.Warnke; Email-contact: cfu-at-andreaswarnke-dot-de