crystal_facet_uml user documentation

COLLABORATORS

	TITLE : crystal_facet_uml user documentation		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Andreas Warnke	2020-10-18	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1   Introduction

crystal_facet_uml creates diagrams to document system and software architecture.
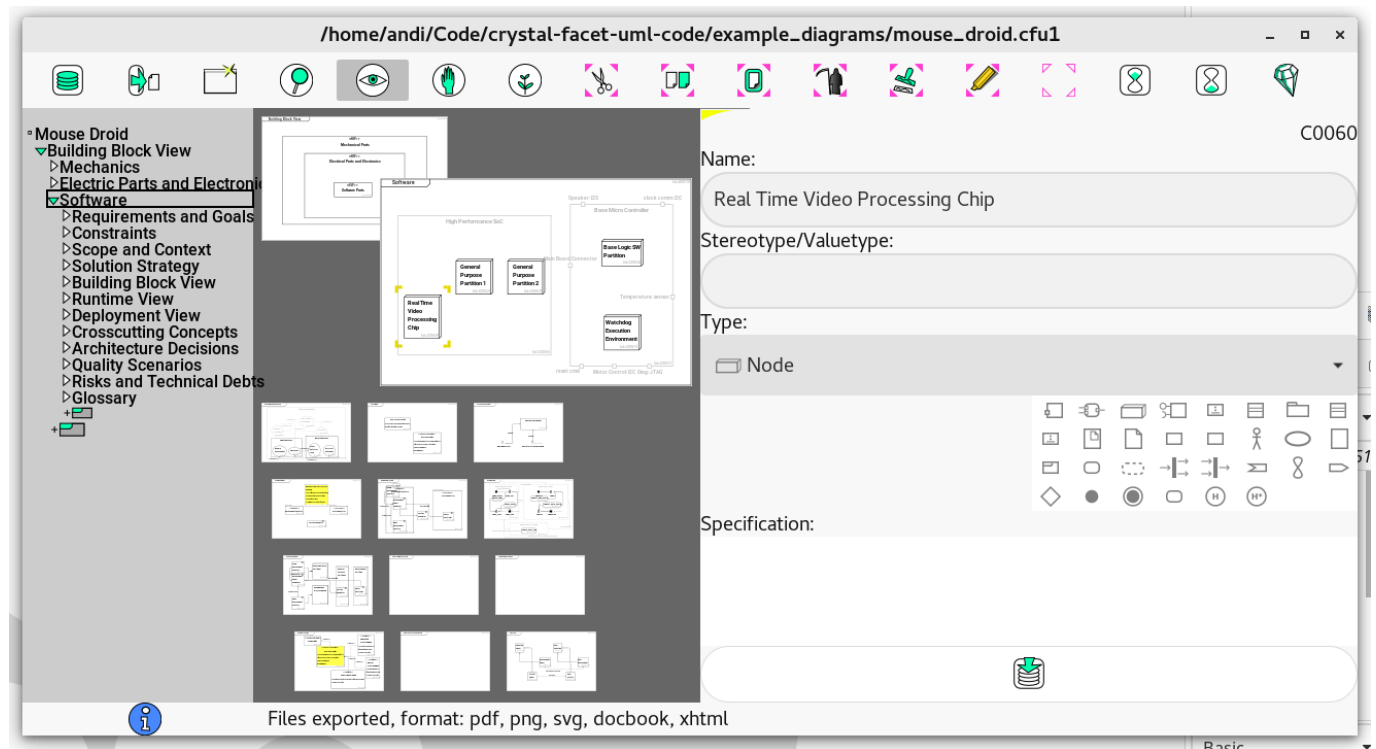
Like a crystal shows different facets of the same thing, this application shows different views of the same system.

1.1   Goal

As software architect, you create a set of diagrams describing use-cases, requirements, structural views, behavioral and deployment views.

crystal_facet_uml keeps element names and element hierarchies consistent. It exports diagrams in svg, pdf, ps and png formats to be used in text processing systems like docbook, html, latex. This tool runs on your local PC and is based on glib, gdk, gtk, cairo, pango, sqlite.

1.2   Features

crystal_facet_uml provides a graphical user interface to

• create diagrams

  (use-case, deployment, component, composite-structure, package, class, activity, state, timing, communication, sequence)

- create uml elements

  (actor, system-boundary, use-case, node, component, part, interface, package, class, activity, state, object, artifact, comment, requirement)

- move, modify and delete uml elements

- create, modify and delete relationships

  (dependency, association, aggregation, composition, generalization, realization, contains, sync-call, return-call, async-message, communication-path, control-flow, object-flow, deployment, manifest, include, extend)

- create, modify and delete features

  (port, field, operation)

- cut, copy, paste uml elements between diagrams

- undo and redo are supported

- multiple windows can show different or same parts of the uml model

Diagrams are layouted part-automatically:

- The user chooses the relative location of uml elements towards others

- crystal_facet_uml selects the exact locations of uml elements

- The user controls the positions of messages/transitions in sequence and timing diagrams

- crystal_facet_uml auto-layouts relationships in other diagrams

crystal_facet_uml manages a meta model:

- Diagrams are organized as a tree, similar to a book's table-of-contents

- Uml(TM)/SysML(TM) elements exist only once even if shown in many diagrams

- Relationships and features are consistent between all diagrams

- Diagram-local messages/transitions are supported in scenario-based diagrams

  (sequence, communication, timing)

crystal_facet_uml exports diagrams as

- vector graphics

  (pdf, ps, svg)

- pixel graphics

  (png)

- textual representation

  (utf-8-txt, docbook, xhtml)

- machine-readable model

  (xmi(TM))

crystal_facet_uml can also be started from command line to export all diagrams automatically or to check and repair database files.

1.3 Usage Overview
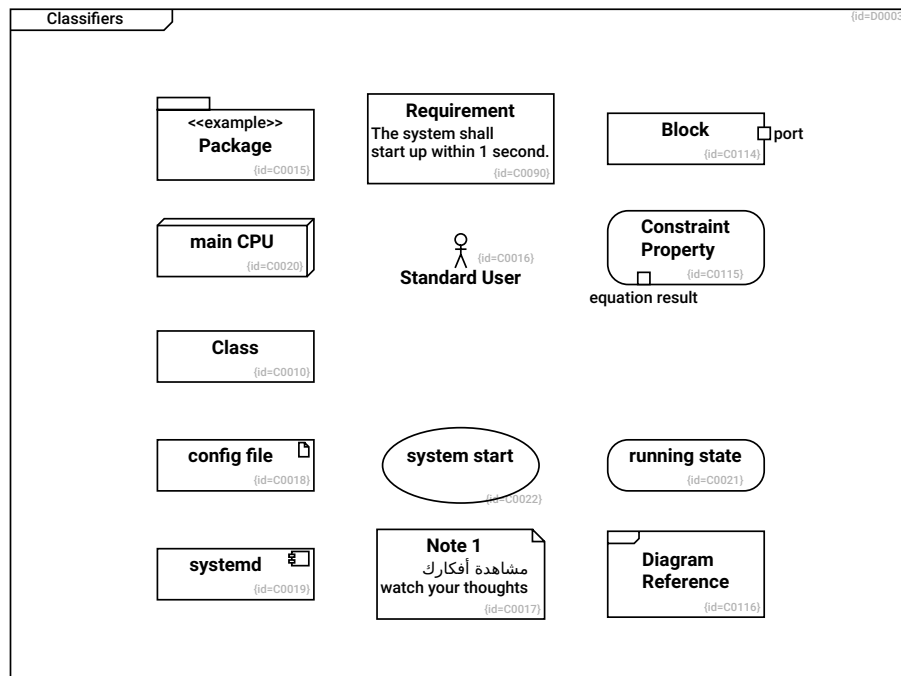

crystal_facet_uml can be started in graphical mode (see Section 3) or from command line (for help run **crystal_facet_uml -h**).
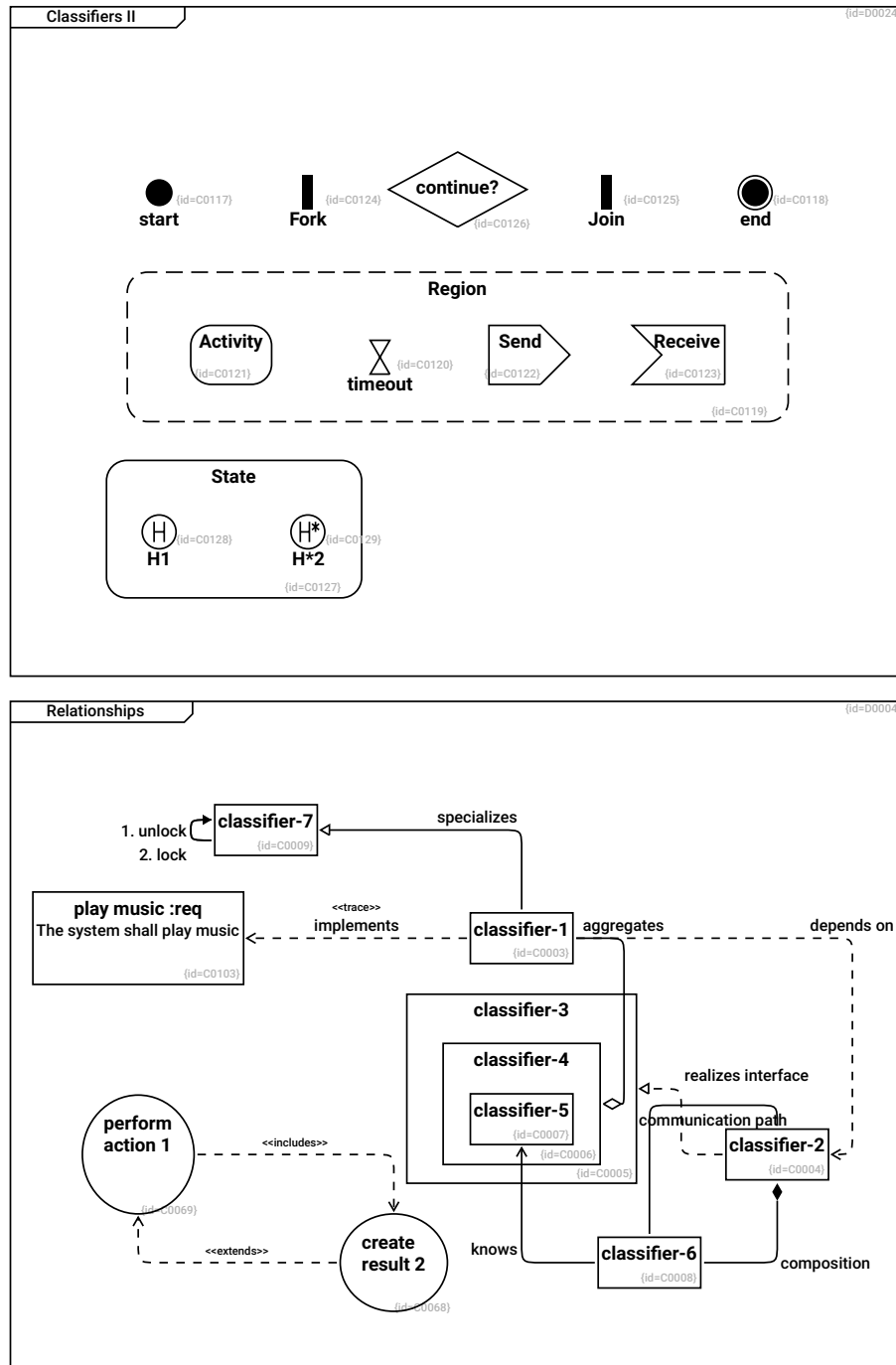
2 Example Diagrams
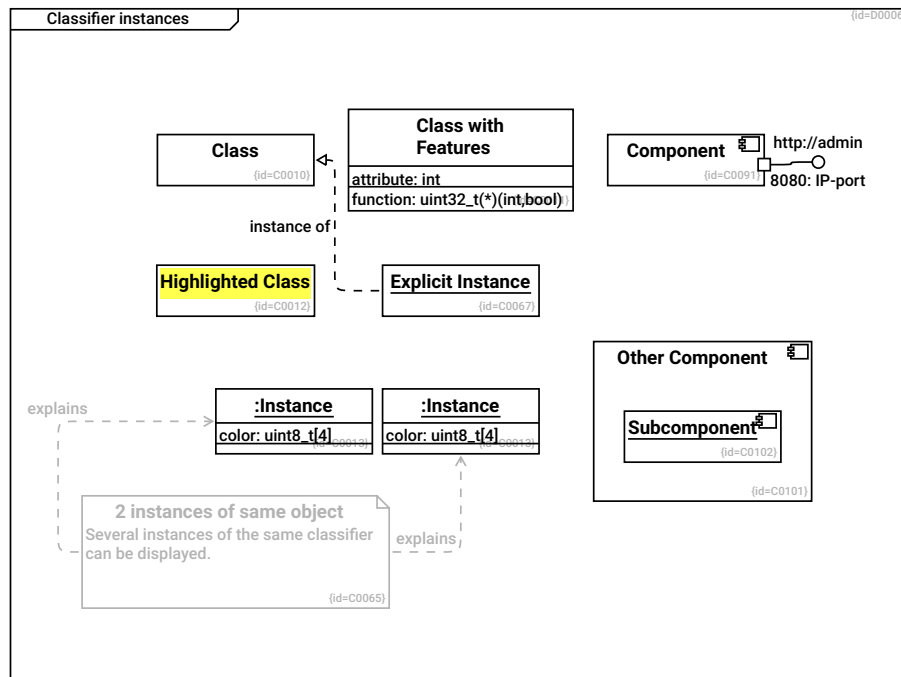
This sections presents the features of crystal_facet_uml.

2.1 Feature List

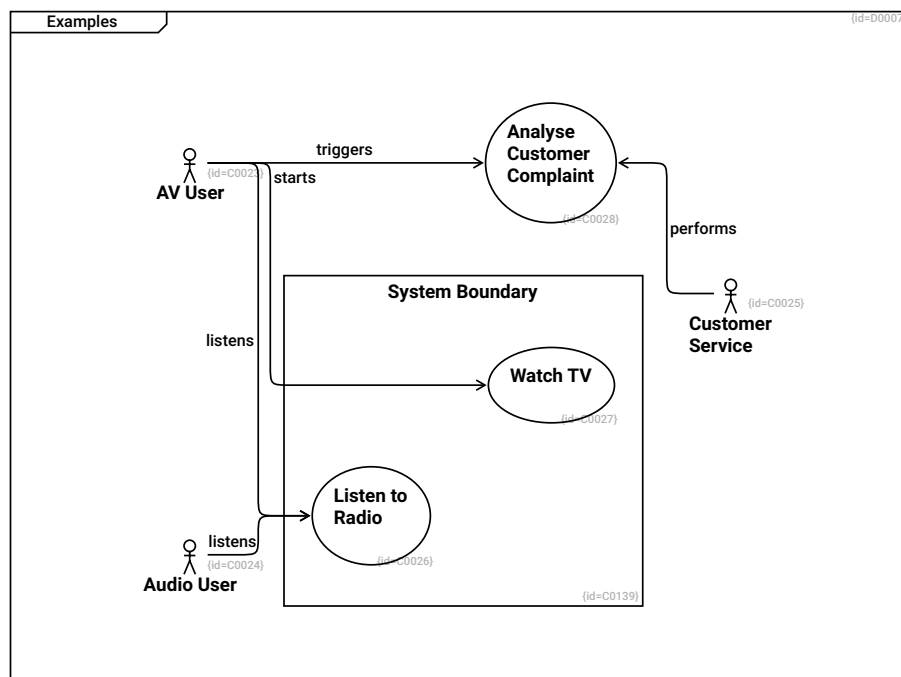This section lists what kind of elements crystal_facet_uml can draw in diagrams.

Classifiers II {id=D0024}

start {id=C0117}

Fork {id=C0124}

continue? {id=C0126}

Join {id=C0125}

end {id=C0118}

Region {id=C0119}

Activity {id=C0121}

timeout {id=C0120}

Send {id=C0122}

Receive {id=C0123}

State {id=C0127}

H1 {id=C0128}

H*2 {id=C0129}

Relationships {id=D0004}

classifier-7 {id=C0009}

1. unlock
2. lock

specializes

play music :req
The system shall play music {id=C0103}

<<trace>>
implements

classifier-1 {id=C0003}

aggregates

depends on

classifier-3

classifier-4

classifier-5 {id=C0007}

{id=C0006}

{id=C0005}

realizes interface

communication path

classifier-2 {id=C0004}

perform action 1 {id=C0069}

<<includes>>

create result 2 {id=C0068}

<<extends>>

knows

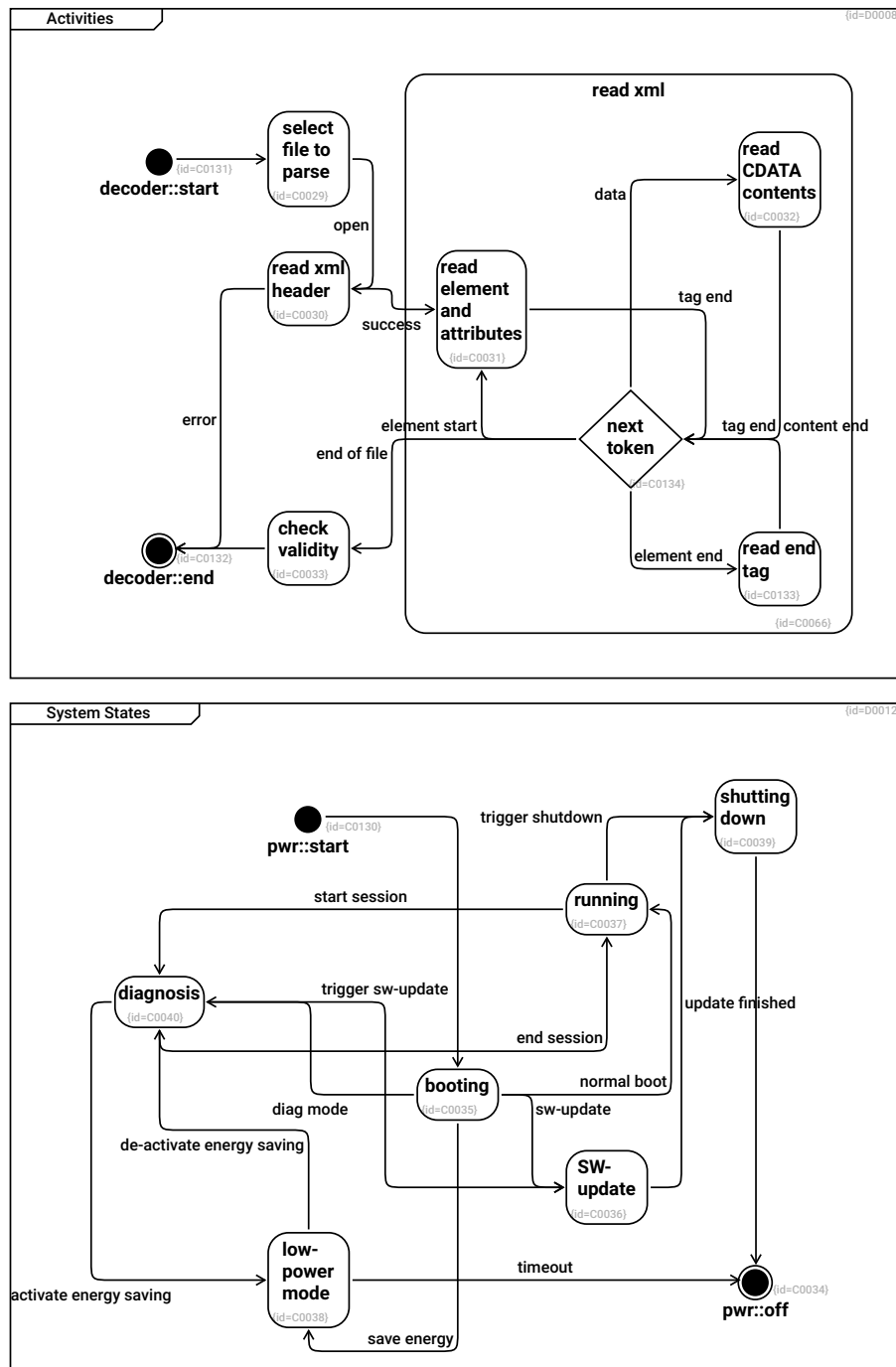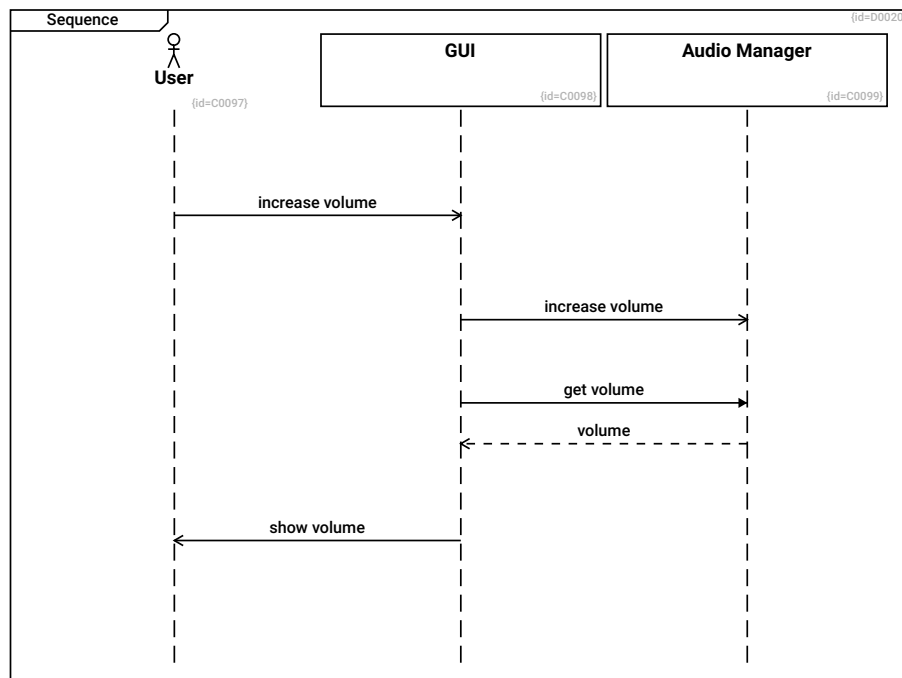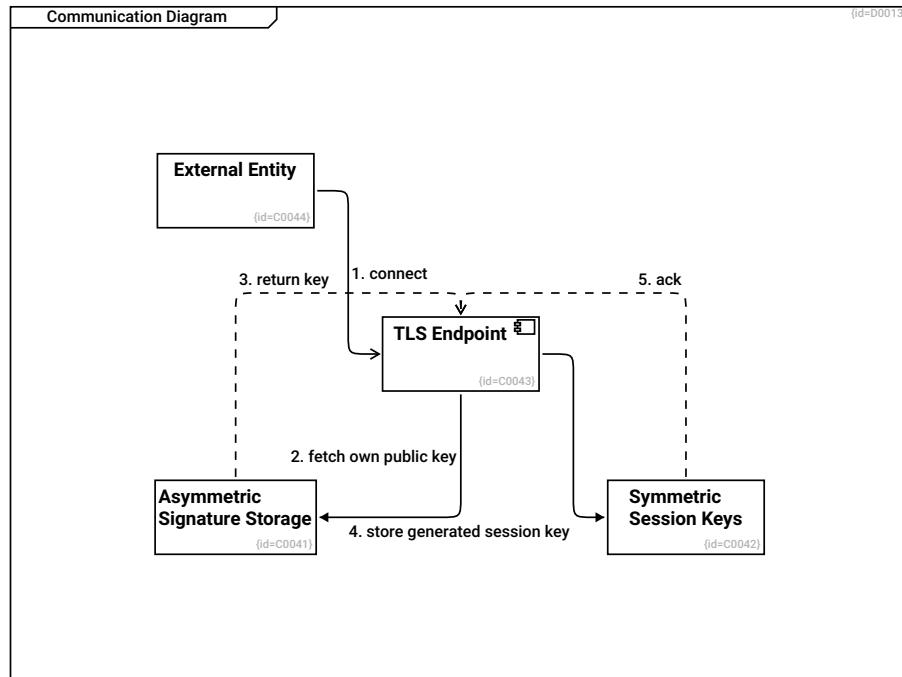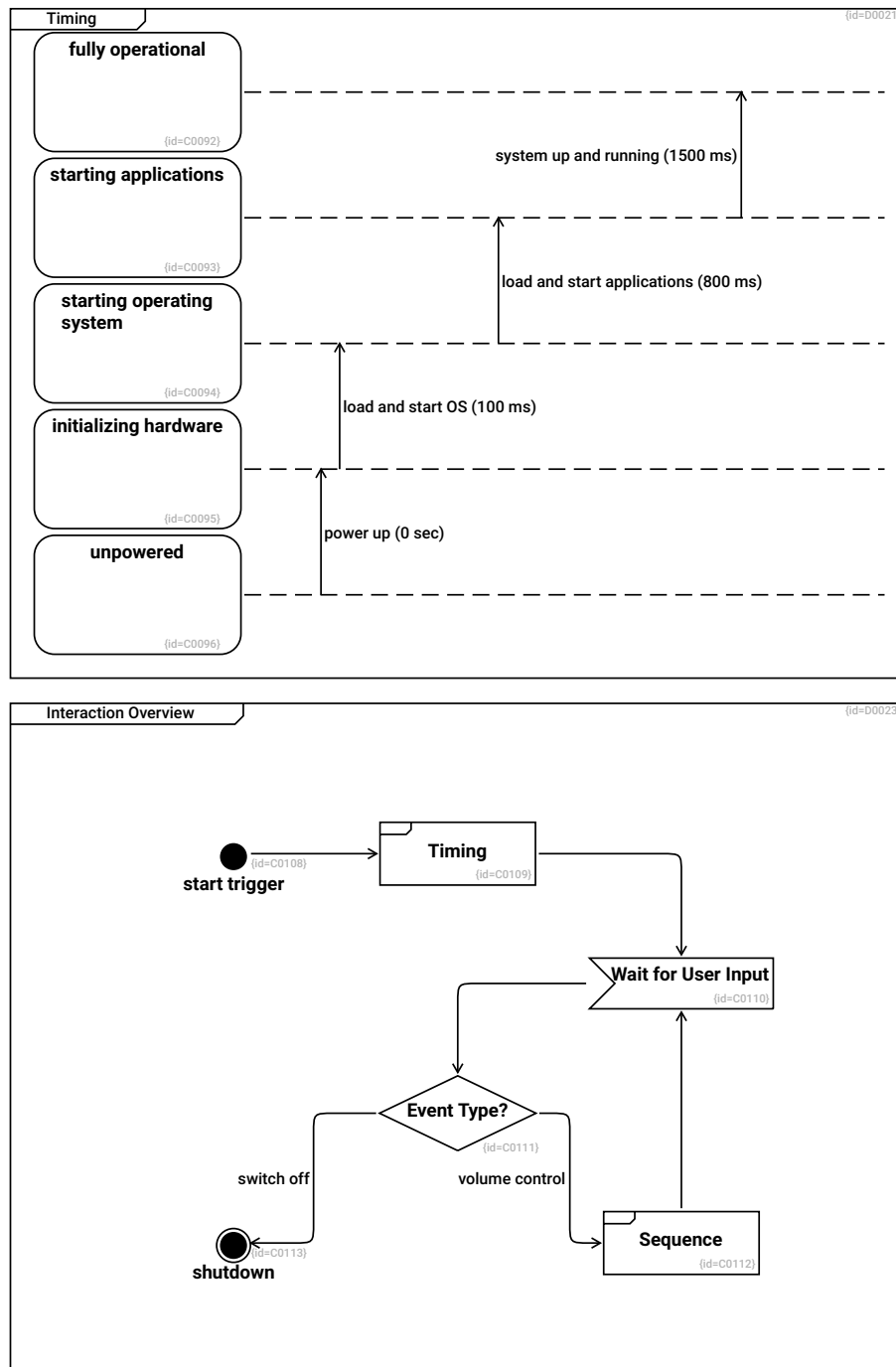**classifier-6** {id=C0008}

composition

2.2 Example UML Behavioral Views

This section lists what kind of elements crystal_facet_uml can draw in diagrams.

Communication Diagram   {id=D0013}

External Entity

{id=C0044}

3. return key   1. connect   5. ack

TLS Endpoint

{id=C0043}

2. fetch own public key

**Asymmetric
Signature Storage**   **Symmetric
Session Keys**

{id=C0041}   4. store generated session key   {id=C0042}

Sequence   {id=D0020}

User   **GUI**   **Audio Manager**

{id=C0097}   {id=C0098}   {id=C0099}
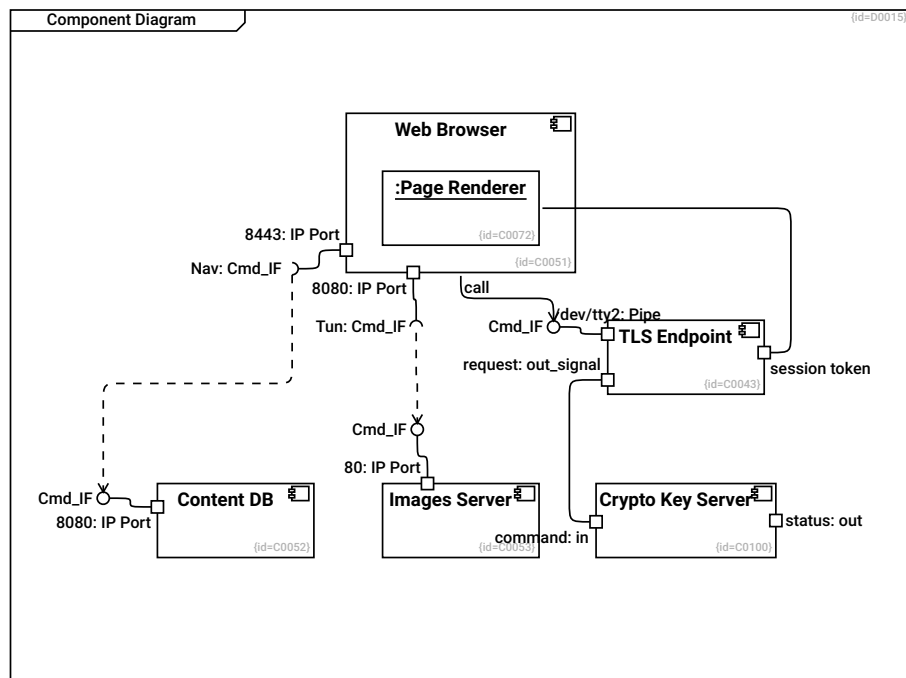
increase volume

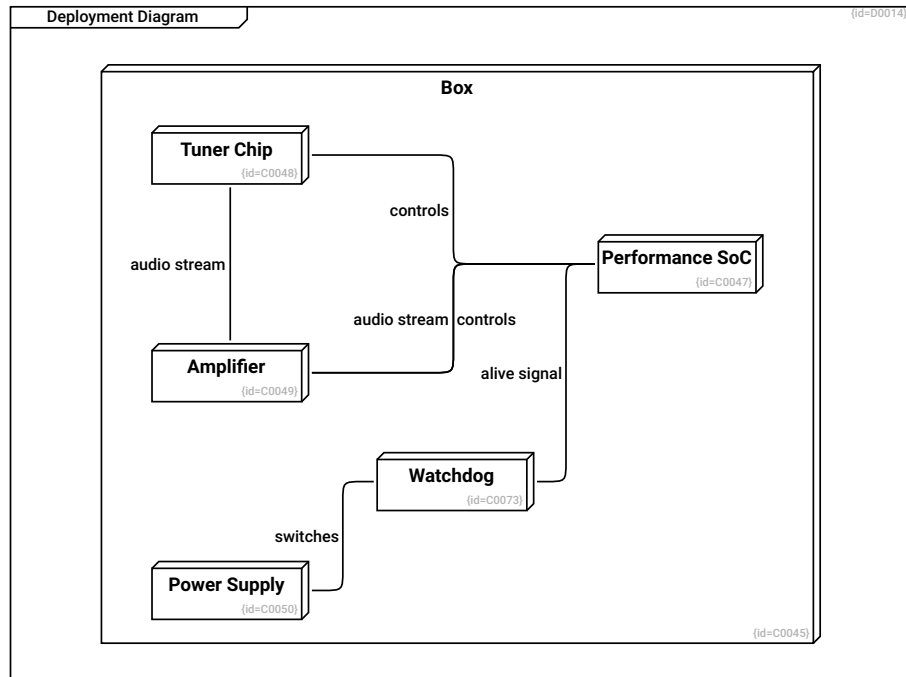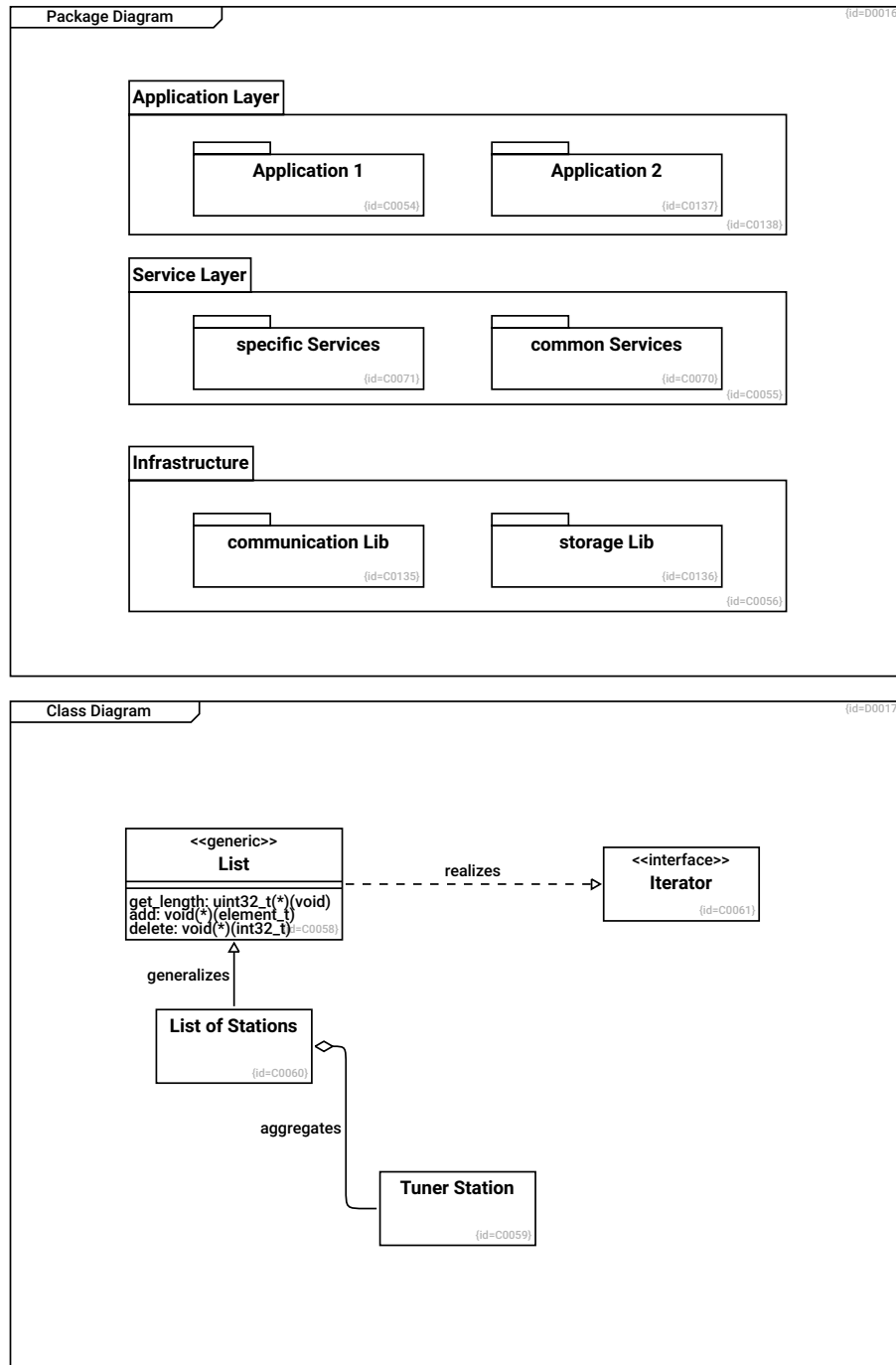increase volume

get volume

volume

show volume

2.3 Example UML Static Views

This section lists what kind of elements crystal_facet_uml can draw in diagrams.

Deployment Diagram {id=D0014}

Box

Tuner Chip {id=C0048}

controls

audio stream

Performance SoC {id=C0047}

audio stream controls

Amplifier {id=C0049}

alive signal

Watchdog {id=C0073}

switches

Power Supply {id=C0050}

{id=C0045}

Component Diagram {id=D0015}

Web Browser

:Page Renderer {id=C0072}

8443: IP Port

Nav: Cmd_IF

{id=C0051}

8080: IP Port

call

Tun: Cmd_IF

/dev/tty2: Pipe

Cmd_IF

TLS Endpoint {id=C0043}

request: out_signal

session token

Cmd_IF

80: IP Port

Cmd_IF

Content DB {id=C0052}

8080: IP Port

Images Server {id=C0053}

command: in

Crypto Key Server {id=C0100}

status: out

2.4   Example SysML Views
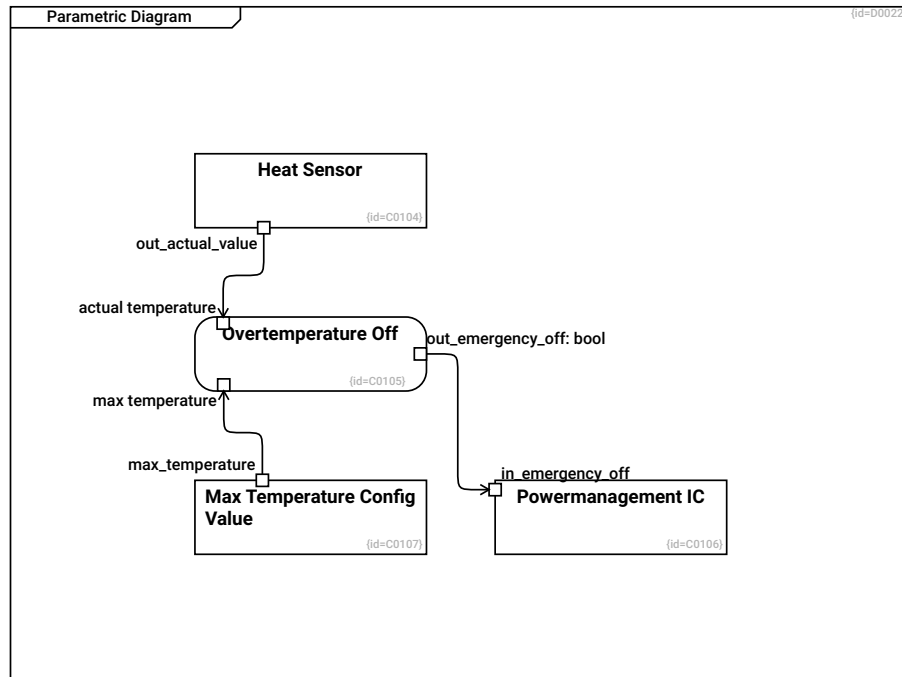
This section lists what kind of elements crystal_facet_uml can draw in diagrams.

Requirements   {id=D0018}

Phone Call
{id=C0078}

REQ-1004
The system shall allow a user to dial a number   <<trace>>
{id=C0079}

REQ-1001
The system shall stream audio
{id=C0075}

<<refine>>
audio_in

REQ-1002
The system shall record the mic signal
{id=C0076}

<<refine>>
audio_out

REQ-1003
The system shall playback audio on speakers
{id=C0077}

Block Diagram   {id=D0019}

Telephone

Display
{id=C0085}

coordinates: SPI   **Dial Pad**
{id=C0081}
touch input

power supply

video_1: LVDS_in

data in   **Power
Management
IC**
{id=C0084}

display signal   **CPU**

RISC 64 bit: type

V+   0

audio_out: i2s   audio_in: i2s

in: i2s   amp out

Speaker

16 bits @ 24000 Hz: playback rate   power supply

data out

pwr

out: i2s   mic in

Microphone   **Network
Connector**   **Power
Connector**

16 bits @ 48000 Hz: sample rate   {id=C0086}   {id=C0087}

{id=C0080}

2.5   More Examples

There are further examples available as html:

- http://www.awarnke.keepfree.de/crystal_facet_uml/mouse_droid/mouse_droid.xhtml

- http://www.awarnke.keepfree.de/crystal_facet_uml/quality/quality.xhtml

And in crystal_facet_uml binary format:

- https://github.com/awarnke/crystal_facet_uml/tree/master/example_diagrams

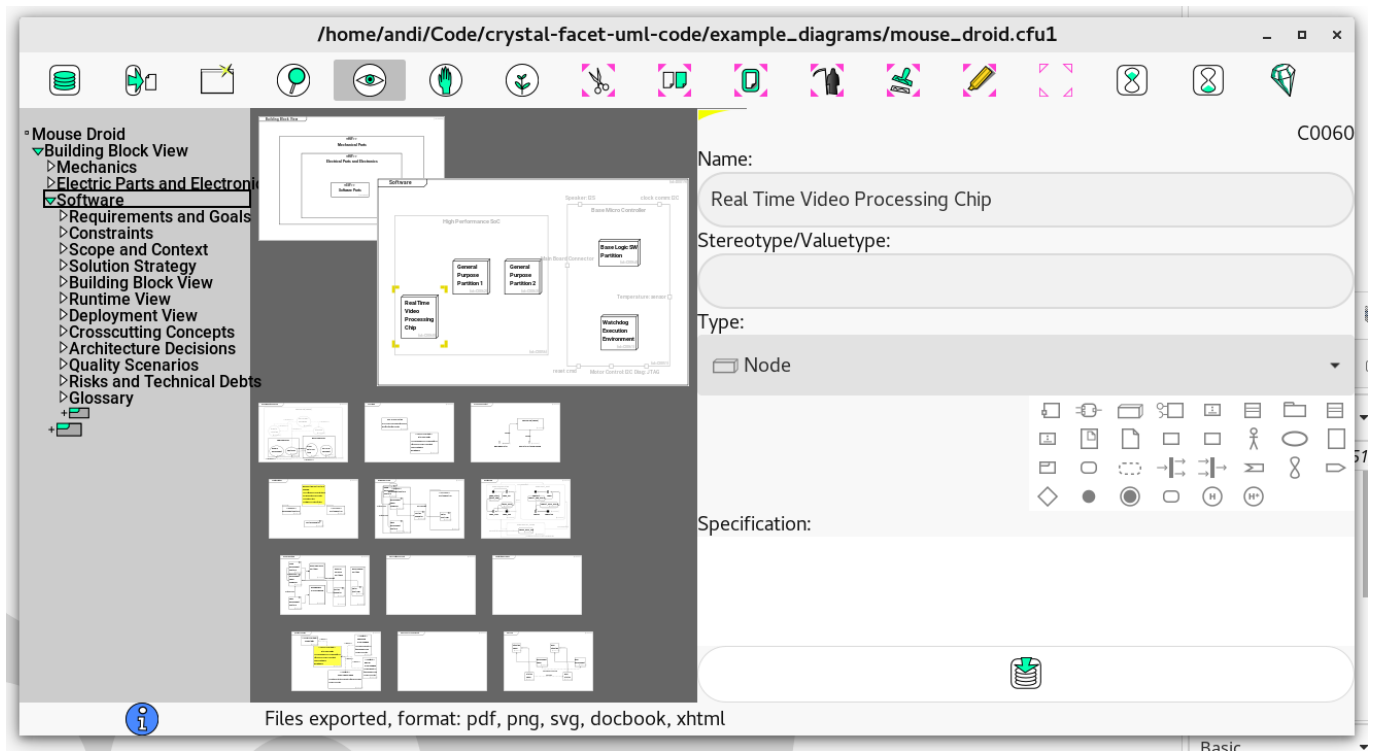- https://github.com/awarnke/crystal_facet_uml/tree/master/architecture/doc
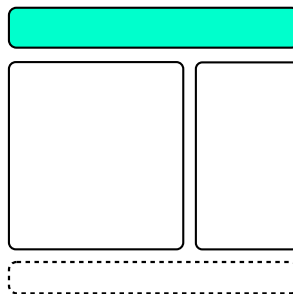
3   GUI / Usage Manual

3.1   Window Area Overview

If started in graphical mode, crystal_facet_uml shows a window with

- toolbar on top,

- drawing area in the center,

- element configuration widgets to the right and

- an optional notification bar at the bottom.

3.2 Tool Bar

3.2.1 Create/Use DB

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

3.2.2 Export

- Exports all diagrams to the selected folder (supported formats are png, pdf, ps, svg, txt. xhtml, docbook, xmi)

3.2.3 New Window

- Opens another window on the same database.

This new window allows you to work reliably with multiple windows on the same database.

3.2.4 Search

- Find diagrams that contain the searched elements (see Section 3.3.1)

3.2.5 Navigate

- Navigate to parent or child diagrams
- Create a new diagram (see Section 3.3.2)

3.2.6 Edit

- Modify elements in the diagram (see Section 3.3.3)

3.2.7 Create

- Create elements in the diagram (see Section 3.3.4)

3.2.8 Cut

- Cut all selected (pink-cornered) elements to the clipboard (features of classifiers are copied if the classifier is selected)

3.2.9 Copy

- Copy all selected (pink-cornered) elements to the clipboard (features of classifiers are copied if the classifier is selected)

3.2.10 Paste

- If the clipboard contains a diagram, this diagram is pasted below the current diagram. All other elements are pasted into the new diagram.

- If the clipboard does not contain diagrams, classifiers and relationships from the clipboard are copied into the current diagram.

- If the name of a classifier is identical to an existing one, an instance of the existing classifier is pasted to the diagram. Otherwise a new classifier is created.

3.2.11 Delete

- Deletes all selected (pink-cornered) elements. This operation may fail if a selected diagram contains non-selected elements.

3.2.12 Instantiate

- Toggles the selected (pink-cornered) classifiers between classes, named instances and anonymous instances.

- No effect on relationships and features.

3.2.13 Highlight

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

3.2.14 Reset Selection

- Resets the (pink-cornered) selection

3.2.15 Undo

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

3.2.16 Redo

- Re-does the last un-done operation

3.2.17 About

- Shows version, license and copyrights

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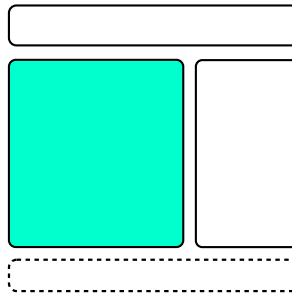
Diagrams are layouted automatically. You can influence the locations of classifiers only. When adding too many classifiers or relations, auto layouting may not achieve the expected results. In many cases, splitting the diagram into two or more diagrams solves the layouting issues and at the same time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

3.3.1 Search

- Enter the ID of an element (e.g. C0001) or a part of its name or description to find diagrams containing this element.

3.3.2 Navigate

- To navigate to parent, sibling or children diagrams, click on the diagram.
- To create a new diagram, click on the ⁺🖿 icon, or the smaller ⁺🖿 icon for a new child-diagram.
- To restructure the diagram tree, drag a diagram name to the new location.

3.3.3 Edit

- To focus the diagram or a classifier or a feature or a relationship (yellow corners), click on this object.

- To select an element (pink corners), click on these objects twice.

- To move classifiers within the diagram, 1.) press, 2.) drag and 3.) release the mouse button.

  Note: When moving a classifier, this is moved in all diagrams where it appears. Order and locations of things stay consistent between different views.

3.3.4 Create

1. To create a classifier, click at an empty space in the diagram.

2. To create a child classifier, click into the white space of a classifier. (Alternatively, create a classifier (see 1) and a containment relationship (see 4).)

3. To create a feature, click onto a classifier (name or border).

4. To create a relationship, press on the source classifier and drag it to the destination classifier.
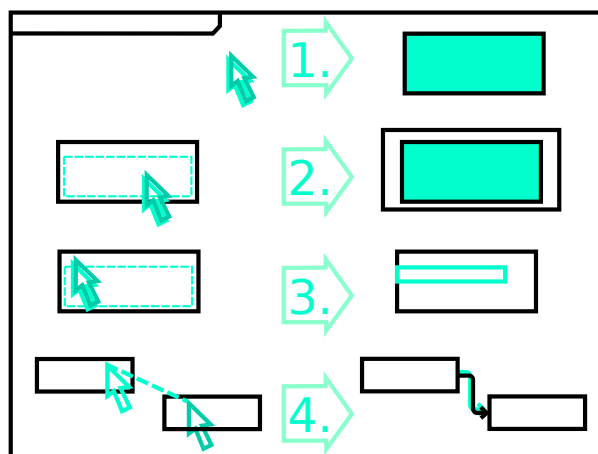
3.4 Element Configuration Area
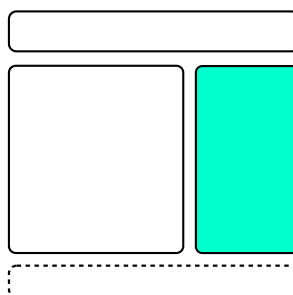
Edit the properties of the focused (yellow-cornered) object.

- name of the focused object

- stereotype/valuetype of the focused object (deactivated depending on object-type)

- type of the focused object

- description of the focused object.

  For xhtml and docbook export, use a double linebreak to create a new paragraph, start lines with *, + or - to format a list, use D0001#id and D0001#name to create a link to the diagram D0001 (showing either the id or the name).

3.4.1 Commit

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

3.5 Notification Bar
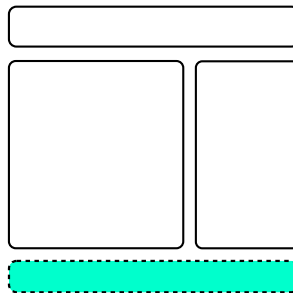
3.5.1 Information

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

3.5.2 Warning

- Informs on a possible problem

3.5.3 Error

- Informs on an error

4 Diagrams and Elements Spec

This program creates diagrams that strive for compatibility to

- UML 2.5

- SysML 1.5

- MOF 1.4.1

In some cases, it deviates from these standards for several reasons:

- Reduce complexity to be able to handle such models in a small open source project

- Reduce feature-set to improve understandability of diagrams even to non-software-architects

- Reduce feature-set to enhance usability of the program

This section gives an overview on standards and implementation-status of crystal_facet_uml. It may be incomplete.

4.1 Classifiers

Classifiers are the nodes in the model-graph.

The table shows the classifier types introduced by different specifications, if they filter/hide their features and a comment stating how this is implemented in crystal_facet_uml.

	Spec	Diagram Context and Filter	Comment
Block	SysML	* / -	Limitations: Compartment Order is "properties, operations" instead of "constraints, operations, receptions, parts, (bound) references, values, properties, stereotype-tagged-values, behavior, namespace, structure" Limitations: No labeled compartments Limitations: no Multiplicities of Block-Instances.
Constraint Block	SysML	Parametric / -	Limitations: Only the rounded-rect symbol is supported.
Node	UML	Deployment / -	
Component	UML	* / -	
Part	UML	* / -	

	Spec	Diagram Context and Filter	Comment
Interface	UML	* / -	
Package	UML, SysML	* / -	
Class	UML	* / -	Limitations: No active classes
Object	UML	* / -	
Artifact	UML	* / -	
Comment	UML, SysML	* / unconditional features	
Requirement	SysML	* / -	
Actor	UML, SysML	Use Case, Sequence / unconditional features	
Use Case	UML, SysML	Use Case / -	Limitations: No SysML extension points
System Boundary	UML, SysML	Use Case / unconditional features	
Diagram Reference	UML	Interaction Overview / unconditional features	Hint: To easily find the referenced diagram, name the reference identical to the diagram.
Activity	UML 2.5 (ch15.2)	Activity / -	Limitations: Object Node Pin notation not supported, draw the object between activities
Interruptable Region	UML	Activity / unconditional features	XMI-Export: For xmi export, all regions belonging to the same set of activities need an outer, enclosing activity.
Fork	UML, SysML	Activity / unconditional features	XMI-Export: For xmi export, all activity-nodes belonging to the same set of activities need an outer, enclosing activity.
Join	UML, SysML	Activity / unconditional features	XMI-Export: For xmi export, all activity-nodes belonging to the same set of activities need an outer, enclosing activity.

	Spec	Diagram Context and Filter	Comment
Accept Event	UML, SysML	Activity / unconditional features	XMI-Export: For xmi export, all activity-nodes belonging to the same set of activities need an outer, enclosing activity.
Accept Time Event	UML, SysML	Activity / unconditional features	XMI-Export: For xmi export, all activity-nodes belonging to the same set of activities need an outer, enclosing activity.
Send Signal	UML, SysML	Activity / unconditional features	XMI-Export: For xmi export, all activity-nodes belonging to the same set of activities need an outer, enclosing activity.
Decision/Choice	UML 2.5 (ch14.2.4,15.3), SysML	Activity, State / unconditional features	In activity diagrams, this is called decision, in statesmachines it is called choice. XMI-Export/State-context: For xmi export, all states belonging to the same statemachine need an outer, enclosing state. XMI-Export/Activity-context: For xmi export, all activity-nodes belonging to the same set of activities need an outer, enclosing activity.
Initial Node	UML 2.5 (ch14.2.4), SysML	Activity, State / unconditional features	Limitations: There is no distinction in ActivityInitial and FlowInitial. Limitations: There is no separate entryPoint state-type. XMI-Export/State-context: For xmi export, all states belonging to the same statemachine need an outer, enclosing state. XMI-Export/Activity-context: For xmi export, all activity-nodes belonging to the same set of activities need an outer, enclosing activity.
Final Node	UML 2.5 (ch14.2.4), SysML	Activity, State / unconditional features	Limitations: There is no distinction in ActivityFinal and FlowFinal. Limitations: There is no separate exitPoint and terminate state-type. XMI-Export/State-context: For xmi export, all states belonging to the same statemachine need an outer, enclosing state. XMI-Export/Activity-context: For xmi export, all activity-nodes belonging to the same set of activities need an outer, enclosing activity.
State	UML 2.5 (ch14.2), SysML	State, Timing / -	Limitations: No symbol for hidden decompositions, no regions (swimlanes) in composite states. Limitations: entry/exit/do list. Limitations: entryPoint and exitPoint states cannot be drawn on parent state border line. XMI-Export: For xmi export, all states belonging to the same statemachine need an outer, enclosing state.
Shallow History	UML 2.5 (ch14.2.4), SysML	State / unconditional features	XMI-Export: For xmi export, all states belonging to the same statemachine need an outer, enclosing state.

	Spec	Diagram Context and Filter	Comment
(H*) Deep History	UML 2.5 (ch14.2.4), SysML	State / unconditional features	XMI-Export: For xmi export, all states belonging to the same statemachine need an outer, enclosing state.
× Value Type	SysML	- / -	not supported. Limitations: Compartment Order of Classifiers is "properties, operations" instead of "operations, properties, stereotype-tagged-values"
× Enumeration	UML, SysML	- / -	not supported. Note: Use a class instead.
× ActivityParameterNode	SysML	- / -	not supported.
× MergeNode/Junction	UML 2.5 (ch15.3), SysML	Activity, State / unconditional features	In activity diagrams, it is called merge, in state diagrams junction node. This is not supported. Note: You may directly connect the arrows to the target activity/state.
× ActivityPartition	UML, SysML	Activity / unconditional features	not supported. Note: Use a parent activity instead.

LEGEND

Filter Defines which elements related to a classifier are not visible

An InstanceSpecification (UML) denotes an instantiation of a classifier. crystal_facet_uml allows any classifier to appear in different diagrams as classifier, as anonymous InstanceSpecification or as named InstanceSpecification. (Rationale: If a classifier is an instance may depend on the context: An M1-class may be an instance if shown in an M2-meta-class diagram, an XML-parser-class may be an instance if shown in the context of stream processors.)

4.2 Features

Features are elements attached to one classifier.

The table shows the feature types introduced by different specifications, if they are visible in any diagram or just once, and a comment stating how this is implemented in crystal_facet_uml.

	Spec	Scope	Comment
▤ Property	UML, SysML	unconditional	Limitations: no SysML Flow-Properties refinement.
▤ Operation	UML, SysML	unconditional	

	Spec	Scope	Comment
Port	UML, SysML	unconditional	Limitations: no SysML-compartment Notation supported. Limitations: no SysML-nested-ports, SysML-proxy-port, SysML full-ports supported. Limitations: no flow property, no compartment notation, no port-compartments. Limitations: no UML behavior ports.
Provided Interface	UML, SysML	unconditional	
Required Interface	UML, SysML	unconditional	
Lifeline	UML 2.5 (ch17.2), SysML	scenario, 1 per diagram	Limitations: One lifeline is visible only in one diagram. Limitations: Lifelines start and end only at diagram border. Limitations: ExecutionSpecification (ActivityBar) are not supported. XMI-Export: For xmi export, lifelines are not yet supported.

LEGEND

Scope scope is unconditional if a feature belongs to a classifier unconditionally, scenario if only applicable in 1 diagram

4.3 Relationships

Relationships are the edges of the model-graph.

The table shows the relationship types introduced by different specifications, a classification in which diagram type to use them preferably, and a comment stating how this is implemented in crystal_facet_uml.

	Spec	Diagram Context	Comment
Dependency	UML, SysML	any	
Containment	UML, SysML	Deployment, Package, Internal Block, Composite Structure, Activity, State	
Deploy	UML	Deployment	
Manifest	UML	Deployment	
Communication Path	UML, SysML	Component, Composite Structure, Block, Internal Block	

	Spec	Diagram Context	Comment
Association	UML, SysML	Class, Use Case	Note: SysML calls this ReferenceAssociation Limitations: no AssociationClass(SysML: ParticipantProperty) exists. Limitations: no AssociationEnd Classes exist, no multiplicities, no roles, no ownership (dot notation). Limitations: no ternary associations (only two ends supported). Limitations: no non-navigateable ends (crosses) suported yet - see todo.txt.
Aggregation	UML, SysML	Class	Note: SysML calls this SharedAssociation
Composition	UML, SysML	Class	Note: SysML calls this PartAssociation
Generalization	UML, SysML	Class, Use Case(?)	Limitations: no Generalization-Sets supported
Realization	UML	Class	
Trace	SysML	Requirement	
Refine	SysML	Requirement	
Extend	UML, SysML	Use Case	Limitations: no SysML-condition-notes can be attched to this relationship
Include	UML, SysML	Use Case	
Control Flow/Transition	UML, SysML	Activity, State	In activity diagrams, this is called control flow, in statesmachines it is called transition.
Object Flow	UML, SysML	Activity	
Async. Call	UML, SysML (?)	Sequence, Timing, Communication	
Sync. Call	UML, SysML (?)	Sequence, Timing, Communication	

	Spec	Diagram Context	Comment
←-- Return Call	UML, SysML (?)	Sequence, Timing, Communication	
× Connector	UML, SysML	Internal Block	not supported. Limitations: No Bi-directional Connectors Note: SysML calls this BindingConnector Note: Use a Communication Path instead.
× Item Flow	SysML	Block Definition	not supported. Note: Use an Object Flow instead.
× Exception Flow	UML 2.5 (ch15.5)	Block Definition	not yet supported, see todo.txt.

4.4 Diagrams

Diagrams are views on the model-graph. They select classifiers and may filter their features and relationships.

The table shows the diagram types introduced by different specifications, if they filter/hide their features and/or relationships and a comment stating how this is implemented in crystal_facet_uml.

	Spec	Filter	Comment
List Diagram	-	any feature, any relationship	This is an overview diagram showing only classifiers without features and without relationships
Box Diagram	-	any feature, any relationship	This is an overview diagram showing only classifiers without features and without relationships
Block Definition Diagram	SysML	lifelines	
Internal Block Diagram	SysML	lifelines	
Parametric Diagram	SysML	lifelines	
Deployment Diagram	UML	lifelines	
Component Diagram	UML	lifelines	
Composite Structure Diagram	UML	lifelines	
Package Diagram	UML, SysML	lifelines	

	Spec	Filter	Comment
Class Diagram	UML	lifelines	
Profile Diagram	UML	lifelines	not supported
Requirements Diagram	SysML	lifelines	
Use Case Diagram	UML, SysML	lifelines	
Interaction Overview Diagram	UML	unconditional relationships (Scenario), unconditional feature	Limitations: There is no link from Diagram-References to referenced Diagrams
Activity Diagram	UML 2.5 (ch15.2), SysML	lifelines	Limitations: Swimlanes not supported
State Machine Diagram	UML, SysML	lifelines	
Communication Diagram	UML	unconditional relationships (Scenario), unconditional features	
Sequence Diagram	UML, SysML	unconditional relationships (Scenario), unconditional features	
Timing Diagram	UML	unconditional relationships (Scenario), unconditional features	

LEGEND

Filter Defines which elements are not visible in the diagram

Scenario Diagrams show only relationships associated with a lifeline of a visible classifier.

4.5 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,

5   Modeling Guidelines

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

5.1   crystal_facet_uml Hints

5.1.1   Tree Structure

Diagrams are organized as a tree. Start the root of the tree explaining the document structure. At the second level of the tree, list the main areas to be shown, for example based on the arc42 template https://arc42.org/overview/ :
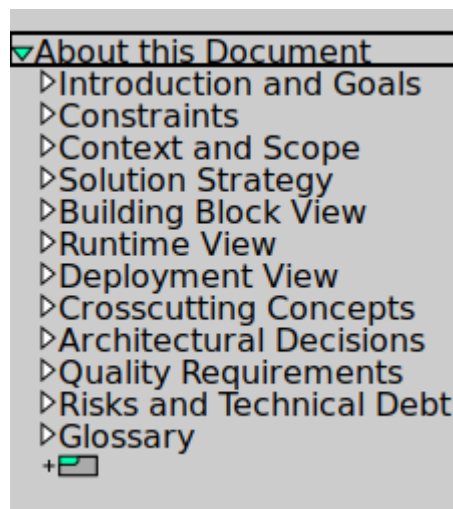
In case you show several layers of abstraction, each building block may contain its sub-blocks, sub-blocks may again show sub-sub-blocks. In this case, structure the specification of the sub-blocks in the same way: apply the proposed folder structure recursively, omitting possibly empty or superfluous folders.
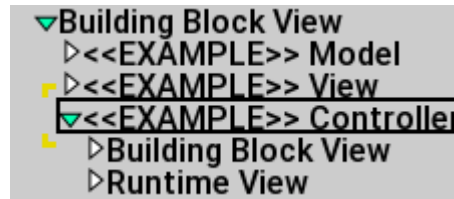
5.1.2 Focus

Put only few elements into each diagram. This increases understandability of the main purpose 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.

5.1.3 Namespaces

Put a prefix to all your elements denoting its namespace. You can then distinguish a GLOBAL_START_STATE from an AUDIO_START_STATE. Or global::start from audio::start.

To achieve a more compact layout of an element, one may insert space characters into names. (In case names get long, the space allows for a linebreak).

5.1.4 Attic/Storage room

If you are not sure if you really want to delete elements, 1) copy them to an attic-diagram and then 2) delete them from the original diagram.

5.2 General Hints on Architecture Documentation

5.2.1 Problem vs. Solution

Distinguish things that are

- given constraints (problem space),
- decisions, chosen and rejected alternatives and
- the designed solution

5.2.2 Names

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.

5.2.3 Description

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.

5.2.4 Precise sentences

Be precise: Write in active form, e.g. The persistence component shall store and retrieve binary data records indentified by string-based keys.

5.2.5   Distinguish similar things

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_ObjectFile (Type:Artifact) and Example_App_SWComponent (Type:Component).

A   Configuration

A.1   Download, Installation and License

A.1.1   Download Links

Find the latest version of crystal_facet_uml at:

- https://www.heise.de/download/product/crystal-facet-uml/

- 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

- https://download.opensuse.org/repositories/devel:/tools/openSUSE_Leap_15.1/x86_64

- https://salsa.debian.org/debian-edu-pkg-team/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/user_doc/crystal_facet_uml_user_documentation.pdf

A.1.2   Installation on Linux

The .deb and .rpm packages can be installed by the package installers of your system.

For installation on ubuntu, debian or raspbian, you may e.g. invoke **sudo dpkg --install <filename>** on the command line:

```
andi@debian1zotac:~/Downloads$ sudo dpkg --install crystal-facet-uml_1.12.0-1_amd64.deb

    We trust you have received the usual lecture from the local System
    Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for andi:
Selecting previously unselected package crystal-facet-uml.
(Reading database ... 198990 files and directories currently installed.)
Preparing to unpack crystal-facet-uml_1.12.0-1_amd64.deb ...
Unpacking crystal-facet-uml (1.12.0-1) ...
Setting up crystal-facet-uml (1.12.0-1) ...
Processing triggers for gnome-menus (3.13.3-9) ...
Processing triggers for desktop-file-utils (0.23-1) ...
Processing triggers for mime-support (3.60) ...
Processing triggers for man-db (2.7.6.1-2) ...
andi@debian1zotac:~/Downloads$
```

For installation you may use a gui-installation tool like yast. Because the packages are not signed, you may want to ignore the warning.
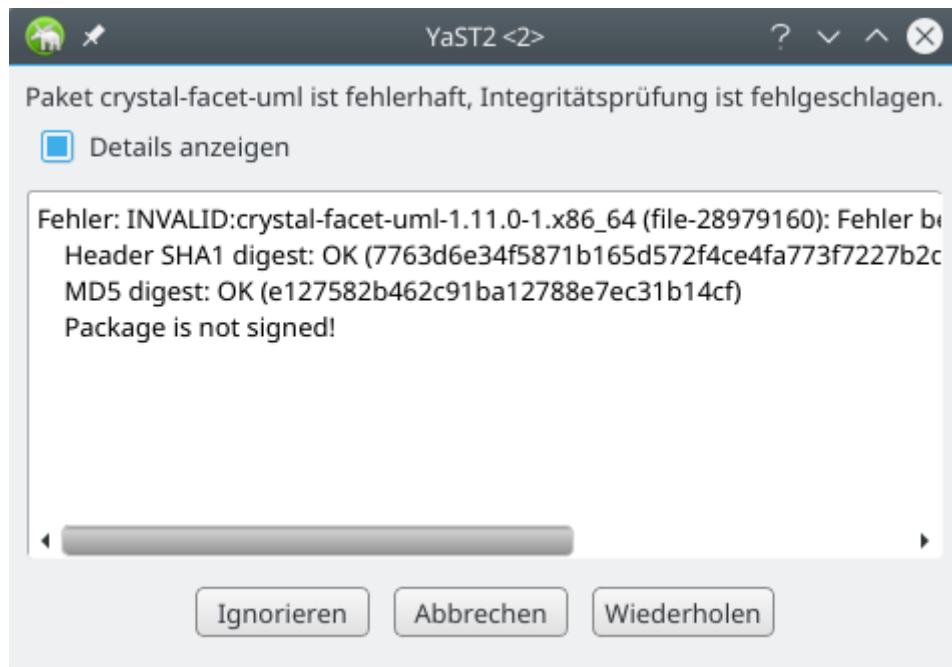
For installation on opeSuSE, you may e.g. invoke **sudo zypper install <filename>** on the command line:

```
andi@linux-uv90:~/Downloads>  sudo zypper install crystal-facet-uml-1.13.1-1.x86_64.rpm
[sudo] Passwort für root:
Repository-Daten werden geladen...
Installierte Pakete werden gelesen...
Paketabhängigkeiten werden aufgelöst...

Das folgende Paket wird aktualisiert:
crystal-facet-uml

1 Paket wird aktualisiert.
Gesamtgröße des Downloads: 698,6 KiB. Bereits im Cache gespeichert: 0 B. Nach der  ←
    Operation werden zusätzlich 8,8 KiB belegt.
Fortfahren? [j/n/...? zeigt alle Optionen] (j): j
Paket crystal-facet-uml-1.13.1-1.x86_64 abrufen  ←
                                                                      (1/1), 698,6 KiB  ←
    (  1,2 MiB entpackt)
crystal-facet-uml-1.13.1-1.x86_64.rpm:
Package is not signed!

crystal-facet-uml-1.13.1-1.x86_64 (Einfacher Cache für RPM-Dateien): Fehler beim  ←
    Überprüfen der Signatur [6-File is unsigned]
Abbrechen, wiederholen, ignorieren? [a/w/i] (a): i
Überprüfung auf Dateikonflikte läuft:  ←
    ...................................................................................
    fertig]
(1/1) Installieren: crystal-facet-uml-1.13.1-1.x86_64  ←
    .................................................................................[ ←
    fertig]
andi@linux-uv90:~/Downloads>
```
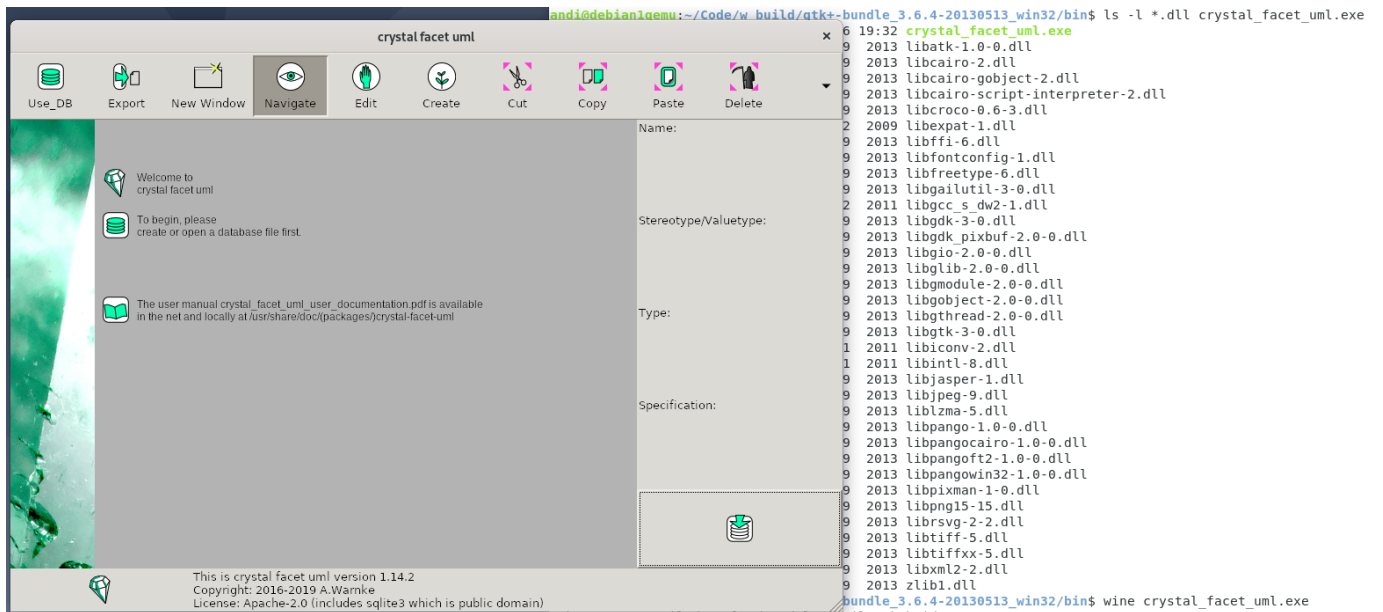
Alternatively, you may want to build the software from the .orig source-package and then install it by **sudo make install**; see the readme file for more information.

A.1.3   Installation on Windows/Wine

- On windows, doubleclick on **crystal_facet_uml.exe**,

- or using the wine emulation, call **wine crystal_facet_uml.exe** to start.

A.1.4   License

License of crystal_facet_uml is Apache-2.0. Copyright 2016-2020 Andreas Warnke; Email-contact: cfu-at-andreaswarnke-dot-de