

^{SK +} **OO**Other

Object Oriented Documentation Tool

Release 1.06

Help File Release 1.06.1

© Copyright Roman M. Zielinski Metod&SystemUtveckling, 1992-1994

Warning:

OOOther is *not* a Free Software

License is required after the free evaluation period (of 4 weeks).

Contents

About OOther

License, Fee, Juridical & Economic Conditions

How to Contact the Author

News in This Release

System Requirements

Introduction to the Tool

Limitations

Ideas for Future Releases

OO Diagram (OOA) Editor

Task Management Component (TMC) Editor

Finite State Machine (FSM) Editor

Use Case Overview Diagram (UC) Editor

Use Case Sequence Diagram (SEQ) Editor

Common Utilities

Clipboard

C++ Header Generation

Printing

Reference Manual

Perception of an object costs

Precise the Object's loss -

Perception in itself a Gain

Replying to its Price -

The Object Absolute - is nought -

\$ Contents

K Contents

+ AA:000

*Perception sets it fair
And then upbraids a Perfectness
That situates so far -*
[Emily Dickinson, 1866]

#_{SK} About OOther

OOther is an easy to use and cheap documentation tool for Object Oriented analysis/modelling and design. It is almost a CASE. The main purpose for OOther is to be a drawing tool & data base, a kind of a note book for OO designers.

It is also a powerful tool for education and training in OO-area.

The current release contains graphical editors and printout support for:

- + OO models according to Coad/Yourdon's OOA and OOD
- + State Machine Diagrams using a dialect of SDL
- + Use Case notation mainly according to Ivar Jacobson's OOSE/Objectory.

The history of the tool is simple: 1991-92 I taught some courses on OOA and OOD and I saw a need to give my colleges a tool simplifying all the hand-drawing and/or Post-It (tm) note handling. I needed a cheap and reliable tool for the notation/methods I use... I have worked as a professional system architect and consultant for many years, and I am dismayed at the unfriendly pricing of some methods and tools licences/fees.

Thus, you can see this tool as an attempt to:

- + prove that good professional software does not need to be expensive
- + help all analysts and designers to work methodically
- + motivate the use of OO in education/training (the reason why the tool is free for students and universities)

This version is based on a *hybrid* methodology I found usable to start with. The object charts are expressed in form of Peter Coad's notation as described in his superb OOA and OOD books (read them!) presenting a simple method, especially suitable as the *starting* OO idiom:

<p>Object Oriented Analysis Second Edition Peter Coad, Edward Yourdon Yourdon Press, Prentice-Hall ISBN 0-13-629981-4</p> <p>Object Oriented Design Peter Coad, Edward Yourdon Yourdon Press, Prentice-Hall ISBN 0-13-630070-7</p> <p>The books are marketed in hardback (OOA is blue and OOD is yellow) and paperback/student (red colour) editions.</p>

In the release 1.06 of the tool I introduced a Use Case notation. It contains two kind of diagrams:

_aboutOOther

\$ About OOther

K OOther;Coad/Yourdon;OOA;OOD;OOSE;SDL;Jacobson;Objectory;SDL;Book
References;Reading Guide

+ AA:010

- + overview diagrams showing correspondence between use cases and actors
- + sequence diagrams showing how the objects co-operate in a use case.

The Use Case idea is mostly a *common sense* and *engineering art*. It is mainly an attempt to show the dynamics of the objects in typical situations - it helps to understand the system and roles of objects. If you want to dig deeper see the following book:

Object-Oriented Software Engineering,
A Use Case Driven Approach
Ivar Jacobson et al
Addison-Wesley Publishing Company
ISBN 0-201-54435-0

The book is 520 pages, filled with discussions on development models, processes, analysis and design methods, technology, project control, quality assurance, etc., etc.

I also recommend books on OMT and Shlaer-Mellor methods. As it looks like today, we are still in the starting phases of the OO aeon, no method is 100% perfect, so keep your eyes open...

#S- **Juridical & Economic Conditions**

Shareware Distribution

Direct Distribution

License Conditions

Fee

Registration Form

Copyright References

#_juridic

\$Juridical & Economic Conditions

+ AA:020

#\$ Shareware Distribution

This version is distributed as **shareware** - i.e. NOT as Public Domain. This means that the software is copyright protected by law. The user may without charge evaluate the software and then decide to purchase it or not. If she/he decide to continue to use the software a registration and a corresponding fee. Using the software without registration after the specified evaluation period is regarded as a copyright violation.

The user may freely distribute **unmodified** copies of the product to other persons for evaluation purposes. You are not allowed to take any charge for distribution, except the cost of the distribution medium.

New versions of the tool may be requested from the author, or from SIMTEL20, or other databases of public domain software and shareware (BBS).

Look in their "win3", "win3/programming" or "win3/utilities" directories!

File packages normally have the name format:

"OOT-" <release> <maintenance> ".ZIP"

E.g. OOT-106a.ZIP means: release 1.06, maintenance level a, etc. ZIP is the compression encoding suffix.

The maintenance level designations are: (none), a, b, c.... etc. I step the maintenance numbers when fixing bugs, or adding very small improvements.

If you have access to USENET e-mail, the mail server at SIMTEL20 will send to you the OOT package as answer to a request, e.g. to fetch the *1.06a* release just e-mail:

To: listserv@vm1.nodak.edu

Subject: SIMTEL20-request

/PDGET MAIL PD1:<msdos.windows3>oot-106a.zip UUENCODE

Earlier I used to distribute garbo.uwasa.fi, but the site seems collapsed under own administrative problems, so I do not care to update them any longer.

(Note: You will need to use the Unix uudecode command and an unzip/dezip/pkunzip/zip utility to uncompress the package. Shareware/public domain versions of such a tool can be found on any BBS system. The tool unzip you may also find on Borland's distribution disks. Or ask the nearest "hacker"...))

Beside the mail servers, e.g. SIMTEL20, you can use FTP directly or via Archie ftpmail.

To get information about how to run ftpmail, send an e-mail to an Archie site, e.g. to the British one:

To: archie@doc.ic.ac.uk

Subject:

[help](#)

(There is an Archie server in almost any country, not only in UK).

_sharedistr

\$ Shareware Distribution

K shareware;SIMTEL20;ZIP;uudecode;e-mail;archie;FTP;BBS;obtaining an evaluation copy

+ AA:030

As an alternative to downloading files from a BBS or SIMTEL20 the evaluation copies of the product may also be ordered from the author, see [Direct Distribution](#).

#SK+ **Direct Distribution**

The tool is also distributed via direct marketing by the author.

The shareware and direct distribution versions **may differ** in functionality, documentation and registration/license fee. However until now (= January 94) the versions were identical. When I distribute diskettes the tool is uncompressed and ready to run.

I distribute:

- + **ooa.exe**, the executable file
- + **ooahelp.hlp**, the help file (this)
- + **ooahelp.doc**, word for windows file with the same contents as ooahelp.hlp
- + **readme** and **information files** about what you got and how to install
- + a few **examples**

#_directdistr
\$Direct Distribution
Kobtaining an evaluation copy
+ AA:040

#SK+ Evaluation and License Conditions

The software OOther is available "as is" without any warranty.

The software may be [freely evaluated during a period of four \(4\) consecutive weeks](#). To continue using the software after end of the period the user is requested to register. Otherwise the software must be removed from all the workstations, systems and network servers where software was installed.

The user may freely distribute OOther to other users - of course all the original and unmodified files must be supplied. They may also evaluate the software during maximally 4 weeks and also requested to register or remove the software.

To register the software and obtain a license the user is requested pay a registration fee.

The licenses are constructed in three ways ([different fees](#)):

- + student license and university course/education/training license
- + single user, home user license
- + 1-5 user commercial site license, including commercial training/courses

Universities using the tool for education purposes are however kindly [recommended](#) to sign a commercial license and in so way support further development of the tool.

The tool is distributed as usual in this business 'as-is' without any warranty.

Support is only given to licensed users.

Even if the author tries to perform a reliable design and testing, some bugs may be present - this is a one-man development, so I have problems to get other than auto-critical internal reviews & testing. Please inform me if You find something strange, or if you have some ideas how to make the tool more user friendly.

#_license

\$License Conditions

K license; guaranty; warranty; evaluation

+ AA:060

#SK+ **Fees**

free (\$0!)

License fee for university students and non-commercial education purposes at technical universities - but please register!

Universities are however *kindly recommended* to sign a site license, \$USD170.

free (\$0!)

Free upgrade to 1.06 if You have registered earlier and fetch the files from a BBS.

USD\$70 (World) or 450 SEK (Scandinavia) or 450 SEK +moms (Sweden)

Distribution fee when ordering disks from the author. The home users will automatically obtain a home user license if they have paid the distribution fee.

USD\$70 (World) or 450 SEK (Scandinavia) or 450 SEK +moms (Sweden)

License fee for non-commercial home users.

USD\$170 (World) or 1100 SEK (Scandinavia) or 1100 SEK +moms (Sweden)

Site License for others, including any use in all kinds of courses/seminars/workshops with a course fee, any use in development projects, etc. License covers max. 5 simultaneous users. Additional 1-5 users USD\$100 / SEK 700 / SEK700+moms.

The license is without time limits within the release 1.x product family. Free upgrades no newer 1.x releases if You fetch files.

USD\$1000 or 7000 SEK

source code licence. The source code licence gives you unlimited use and allows you to modify the software. The kit contains full source code, Software Design Description and object descriptions. Please contact the author for further details.

Users in Sweden please add the current tax (**MOMS**) when sending the fee.

Please contact the author for an agreement for site licenses, distribution license or a source code license.

Invoice is sent on request, e.g. if You need my company's organisation number for MOMS accounting.

Cheques, payable in Sweden, are to be sent to the author :

**Roman Zielinski,
Tre Källors väg 7
S-145 65 Norsborg
Sweden**

You can also direct your payment to my **Swedish PostGiro** account

#_fee

\$ Fee

K fee;license;registering

+ AA:080

PG: 436 78 78-8.

I do **not** accept any credit cards.

#SK+ **Registration Form**

Please send me the following registration data:

- + **Name** (or Company Name + name of a contact person),
- + **Mailing Address**,
- + **E-mail address** (if you have one). If I have your e-mail address, it is easier to send you new updates!
- + **Fax number**
- + **Version of OOther (e.g. 1.06a)**
- + **License Type:**
 - Student License** (free), Please specify your age and name + address of the School
 - Non-Commercial Home User License**, enclose cheque
 - Commercial User License**(specify how many 5-user licences), enclose cheque
- + **Specify Your equipment** (optional information), e.g.: PC model, CPU type (286/386/386/486/SX/DX), Windows release, memory size, esoteric memory boards, printer type.

--- ADDRESS LABEL --- cut here -----

**Roman Zielinski,
Tre Källors väg 7
S-145 65 Norsborg
Sweden**

--- REGISTRATION FORM ---cut here -----

Name:

Company:

Contact Person:

Mailing Address:

E-mail address:

Fax number:

Version of OOther:

License Type:

- Student License** (free)

School Name:

Address:

- Non-Commercial Home User License** (enclose cheque)

- Commercial User License** (enclose cheque)

Number of 5-user licenses:

#_register

\$Registration Form

K fee;license;registering;author's address

+ AA:090

Specify Your equipment (optional information), e.g.: PC model, CPU type (286/386/386/486/SX/DX),
Windows release, memory size, esoteric memory boards, printer type.

#SK- Copyright References

OOTher is a trade mark of Roman Zielinski Metod&SystemUtveckling in Sweden.

The OO Documentation Tool is written in Borland C++ 2.0 using MS-Windows Software Development Kit (SDK) 3.1, Whitewater Resource Toolkit.

The original idea and design are entirely by Roman Zielinski **without any stolen** impressions from other tools like Coad's *OOATool* (tm) and Objective Systems Objectory ORY/SE (tm).

Copyright (parts):

Borland Intl (C++)

Microsoft (MS-Windows environment, SDK 3.1, WinHelp,
OpenFile/SaveAs/Font/PrintSetup-dialogues)

Whitewater Group Inc (Resource Toolkit)

Roman M. Zielinski Metod&SystemUtveckling (Windows Classes "WinClass")

Roman M. Zielinski (OO Documentation Tool, (OO Documentation Tool, help file, user
interfaces, design & implementation)

#SK + **How to Contact the Author**

Mail:

**Roman M. Zielinski Metod&SystemUtveckling
Tre Källors Väg 7
S-145 65 Norsborg
Sweden**

E-mail (USENET):

It is my consultant address, probably You can find me there during 1994, at least until July:
conrozi@KK.ericsson.se

#_author
\$How To Contact the Author
K author; author's address; e-mail
+ AA:110

#SK- **News in This Release**

The **release 1.06** of **OOther - Object Oriented Documentation Tool** contains powerful functional enhancements as compared with the release 1.05. The most important news are:

- + a new symbol "message to/from many objects" has been introduced in the OOA Editor
- + Finite State Machine editor handles separate FSM diagrams for each service and common for all the model
- + Improved editing functions in the FSM editor (align, grouping for copy and move)
- + Use Case Overview Diagram editor
- + Use Case Sequence Diagram editor
- + Use Case Overview Diagram printout formatter
- + Use Case Sequence Diagram printout formatter
- + consistency check verification tools (Sequence diagrams vs. OOA, FSM and UC diagrams)
- + generation of simple header files from information tags inside the description texts (demo)
- + the header files assigned to an object (generated by OOther or user supplied) can be presented via notepad
- + a listing of contents of the models can be generated - to help You to know which parts of the model exist or not.
- + updated the help file (this one)
- + text strings has been moved to Window's string resources. It allows easier adaptation to national languages, if someone wants to translate
- + file name can be passed as call argument (e.g. when clicking from file manager)
- + a dialogue to customise all the paths required by OOther and stored in WIN.INI has been added
- + template files can be composed of national characters in OEM (= IBM PC) or ANSI (= MS Windows) character set. If the flag TemplateOemToAnsi=1 in win.ini the template files will be translated to Windows character set when read in, e.g. the national characters äöÄÖeñê etc. Value 0 of the flag means no conversion as earlier.
- + corrected bugs around handling of the remembered path names of the recently edited files
- + corrected bugs, loosing the FSM texts and disorder in connectors

The **release 1.05** of **OOther - Object Oriented Documentation Tool** contains functional enhancements as compared with the release 1.04. The most important news are:

- + standardised open/save dialogues
- + free font selection
- + editing in zoom mode
- + simplified/improved printout dialogues
- + SDL-dialect finite state machine (FSM) editor
- + speeded up copy-paste transfer to Excel and Word for Windows 2.0
- + printed page boundaries are visible on the edited sheet
- + printer set-up dialogue
- + correction of problems from earlier versions
- + new help file

#_news

\$News in This Release

K News in This Release;Release 1.05;Release 1.06;UC Editor;SEQ Editor

+ AA:120

#SK- System Requirements

- + MS-Windows 3.1 **required**.
- + 386 or 386-SX with 4MB is **recommended** (286 works fine too, unless you're using one of the terrible memory boards around).
- + MS-Windows Compatible mouse is **required**.
- + Colour VGA, preferably Super-VGA running higher resolutions like 800x600 or higher is **recommended**. Higher resolutions in turn require larger screens (>15") and/or good eyes.
- + Printer with graphical possibilities (9-pin, 24-pin, Ink Jet, Laser...) is **recommended**.

You should have at least **6-10 Megabytes free disk space** for windows swap and for the spool files. Print files can be some megabytes long (formatted graphics+text) - don't blame me, it's High Tech (?). The tool is small, and the created files are also small, if you run MS-DOS 6 with double space the are well compressed.

```
# _sysreq
$ System Requirements
K system requirements;memory;disk;MS-Windows;COMNDLG.DLL;VGA;SVGA;printer
+ AA:130
```

#S- Introduction to the Tool

Common

Other Editors

File Handling

Views

Mouse

Auto Save

Templates

Configuring of the Tool

#_introTool

\$Introduction to the Tool

+ AINTRO:010

Introduction, Common

This chapter contains a short introduction. Read it first, then continue with the more detailed rest of the Help file.

OOther is simple to use. You communicate with OOther directly via a graphical user interface, just use your mouse, point and click-click-click.... Most of the work is done by clicking on graphical symbols. The tool presents some dialogue boxes when it needs some data from you. Some less frequent tasks are invoked by using the menus.

The OO Documentation Tool will generate some warning and error messages when it cannot perform a requested function or something happens that concerns the user.

OOther is simple and thus it is fast and reliable.

All OOther Editors have a menu at the top and scroll bars at bottom and right edges of the window. Menus have a similar disposition in all editors.

The OO Documentation Tool stores all information about your model & objects in an ASCII files. The suffix *.OOT* is *recommended* for file name. You may also associate the suffix OOT to the ooa.exe executable under the MS-Windows File Manager (use File, Associate... under the MS-Windows File Manager). However you may only run one incarnation of the OOther.

_introToolCommon
\$ Introduction, Common
+ AINTRO:000

#SK+ Introduction, OOther Editors

OOther contains in the current version *five editors* working on data contained in a *single file*.. Each Editor works on a **model**. The models are (more or less) referring to each other's contents.

1. Object Oriented Analyse (OOA) Editor, presenting an static OO-model diagram. (See also [OOA Editor](#).)
2. Task Management Component (TMC) Editor, presenting how objects are allocated to tasks/processes and computers/processors. (See also [TMC Editor](#).)
3. Finite State Machine (FSM) Editor, presenting details on the dynamic behaviour of each object.. (See also [FSM Editor](#).)
4. Use Case Overview (UC) Editor, presenting Use Case/Scenarios for the analysed domains.. (See also [UC Editor](#).)
5. Sequence Diagram (SEQ) Editor, showing how objects co-operate in a Use Case.. (See also [SEQ Editor](#).)

When you start or when you open a file You are always in the OOA Editor. It's because all the other editor relay on the model created in the OOA Editor.

You can only edit only *one file* at a time. and You are using *one of the editors* at a time. But you can easily switch between them by clicking on the menus.

_introToolEditors

\$ Introduction, OOther Editors

K editors;OOA Editor;TMC Editor;FSM Editor;UC Editor;SEQ Editor

+ AINTRO:010

#SK+ Introduction, File Handling

The **FILE** menu option contains the primary choices NEW, OPEN, MERGE, SAVE and SAVE AS.

The NEW option empties the contents of the OOther editors. If the editors contain an unsaved model, you will be informed, and have a chance to cancel the NEW-request and can save your work.

The OPEN option invokes a standard Windows dialogue. After specifying a file name OOther clears the contents of the editor (after warning you, of course) and reads a OO-model from file. OOther remembers the recently edited files - You can see the names in the FILE menu.

The MERGE option is used to copy another OOA and OOD/TMC model into the currently open one. The user will be asked for X,Y co-ordinates where to read the file, OOther proposes a suitable non-overlapping location.

The SAVE and SAVE AS options allow you to store the contents of the work sheet to disk with keeping the original file name or specifying a new one. OOther creates a backup copy (.BAK) of the previous file contents when you save.

#SK+ Introduction, Views

The views menu options allow You to control appearance of the model on the screen.

In the OO-Diagram (OOA) Editor according to Coad/Yourdon method you can filter, i.e. present or suppress the structure, attribute and service level of the OO model, by choosing VIEW, STRUCTURE, VIEW,ATTRIBUTE or VIEW,SERVICE. The current selection is marked as check marks on the menu. Object names are always presented and not affected by the VIEW options.

In all editors you can get an optional indication where go the boundaries between pages on the printer, option VIEW, PAGE BOUNDARIES.

All the OOther editors allow editing in two modes: normal and zoomed. The applied zoom factor can be specified by using VIEW, ZOOM 2x ... 5x.. Switching between normal and zoom mode is performed by pressing the **right mouse button** or the menu ZOOM.

_introToolViews

\$ Introduction, Views

K view;structure level view;attribute level view;service level view;filtering;zoom

+ AINTRO:050

#SK+ Introduction, Mouse Conventions

Left mouse button:

single click, select the object/symbol the mouse cursor points upon

double click, invoke dialogue box for the pointed symbol

SHIFT + left mouse button, select the second object/symbol to be connected to that one being selected earlier.

CTRL + left mouse button, groups/selects the object for multiple moving, i.e. moving together with other selected objects, only in OO and FSM -Diagram Editors. Max 30 symbols/objects can be grouped. Grouping of the objects can also be used to select which objects are to be copied to the clipboard (OOA and FSM Editors only). In the FSM editor the grouped symbols can also be aligned or deleted.

press the left button & drag, move graphical object(s)

Right mouse button:

click the right mouse button, zoom/unzoom the document (toggle)

SHIFT + right mouse button, show all the inherited attributes and services, only in OO-Diagram Editor

#_introToolMouse

\$ Introduction, Mouse Conventions

K mouse;click;double-click;left-button;right-button;drag;zoom;select object/symbol;connect objects/symbols;inheritance (show)

+ AINTRO:060

#SK+ Introduction, Auto saving

The tool has a auto saving facility, activated every **10:th minute**. If the file in the OO Documentation Tool has been modified, it will be saved into a file with the default name **C:\AUTOSAVE.OOT**.

If windows crashes or you make a mistake, examine this file. If you wish to recover its contents just OPEN it from the OO Documentation Tool and use FILE, SAVE AS to store it under another file name.

(If the tool fails to start the auto saving process, it will inform you when starting the tool - this might happen if other applications have started too many timers in MS-Windows).

The path name of the auto save file may be specified in the WIN.INI file. Default path is "**C:\autosave.oot**"

If you run OOther via a network file system be sure that the auto save file name is selected in such a way that no collisions between users occur, e.g. on your private disk or private catalogue on the shared disk.

_introToolAutoSave
\$ Introduction, Autosaving
K autosave;recovery;crash;lost data;failure
+ AINTRO:070

#SK+ Introduction, Templates

All default description templates in OO Diagram (OOA) Editor and TMC Editor can be replaced by user supplied ASCII files. Names may be specified in the file WIN.INI. It is also possible to run without any templates by specifying \DEV\NUL as a template file.

The template files are translated from PC's OEM character set to Window's ANSI character set (which also is used by OOther). It affects only the national characters.

The FSM, UC and SEQ Editors do not use templates - the expected information in the descriptions entered there is quite trivial.

#_introToolTemplates
\$ Introduction, Templates
K template;description;user supplied template;empty template;character set
+ AINTRO:080

#SK+ Introduction, Configuring the Tool

Configuration of the tool is performed via three options under FILE menu:

- + FILE PRINTER SETUP...
- + FILE CONFIGURE...
- + FILE COLOR...

The option FILE PRINTER SETUP... allows selection of printer and printout orientation (landscape/portrait). Printer Set-up changes are only affecting this session of OOther, i.e. they do not affect other applications. To set up printer options permanently use the Control Panel, Printers from MS-Windows Program Manager.

The FILE CONFIGURE... option allow You to specify grid, the fixed steps used to position the graphical objects/symbols on the sheet. Fonts can be selected by the two font buttons. All fonts available on your system may be selected, but the best result will be achieved when using the scaleable fonts like TrueType or Postscript..

Colours of graphical symbols can be customised by invoking the FILE COLOR... option.

The customisation may also be done via WIN.INI file.

```
# _introToolConfig
$ Introduction, Configuring the Tool
K configuring the tool;color;printer;printer setup;page
orientation;landscape;portrait;grid;font;
+ AINTRO:090
```

#SK+ Introduction, Limitations

- + only OOA-notation (Coad/Yourdon) is supported for object models. Lot of people want OMT object models, it would be not so difficult to update the symbols and add the missing few, so why not? The only problem is time...(my time).
- + there is no *subject filtering* of what is presented (i.e. objects are not grouped into subject areas). However you will get the same effect by entering different subjects into different files. Via Import/Copy Object you can create links to other object files or just copy the object to your current file. (Coad is somewhat unclear if an object may participate in many subjects, but the import function is powerful enough!).
- + printing accuracy differs from that on the screen, so some text and connectors will appear differently on paper
- + the strategy to draw connectors is rather simple, resulting in some non-optimal lines which may need some manual help.
- + the names of objects, attributes, services, messages, use cases etc. are limited to 32 characters, while description texts for objects, attributes, services and associations are limited to 5000 characters. This is a memory economy constraint. In reality, during modelling this limit is seldom reached.
- + number of processed objects is limited to 120 (if you find this value too limited tell me it's just a compile time constant that easily can be changed)

#_limits

\$ Introduction, Limitations

K limitations;number of objects;description;filtering;subject filtering

+ AINTRO:100

#SK · **Introduction, Ideas for Future Releases**

While the release 1.06 makes OOther quite complete I do not really know what comes next. The release 1.07 will probably contain:

- + extended analysis of your model, criticisms, hints, warnings...(You find already some in the Use Case and Sequence diagram editors!)
- + optional views with abstract or concrete object in Sequence diagrams (view filter)
- + presenting TMC processor and task allocation in the Sequence diagram in Use Case
- + maybe improved generation of C++ classes and some code generation?
- + maybe verification of state machines (FSM)?
- + I will not implement too many dramatic changes before the release 2.0, but if You have some ideas let me know!

I still was not able to figure out how to format clipboard metafile buffers to open possibilities to edit graphics under MS-Word for Windows 2.0. Who can help me?

Later releases (release 2.x): It will be a totally redesigned CASE tool with multiple method support!

#_ideas
\$ Introduction, Ideas for Future Releases
K consistency check;FSM Editor;SEQ Editor
+ AINTRO:110

#^S [OO Diagram \(OOA\) Editor](#)

Coad/Yourdon Notation

Creation of Symbols

OO Diagram Utilities

OOA Editor Menus

Mouse in OOA Editor

#_OOAeditor
\$OO Diagram (OOA) Editor
+ EDITOOA:000

#SK+ Coad/Yourdon OOA Notation

Objects

Every object is represented as a rectangle divided by two horizontal lines into the following parts:

- + *object identity part* (top)
- + *attribute part* (middle)
- + *service part* (bottom)

An optional dash-dot line around the object rectangle indicates that the object represents a set of instances. If this line is not present, the symbol represents a class, i.e. an uninstantiated abstract entity.

The object identity part contains the name of the object, a descriptive text and type of the object (object-&-class or class).

Connectors and Associations

Two objects may be associated by a *connector*. These can be of type gen-spec, whole-part, instance-to-instance or message.

In the *gen-spec association* the generic object contains common properties of some objects, while the specific object contains the unique properties. The association presents the inheritance structure.

In the *whole-part association* the whole part contains (or keeps track of) its parts.

The gen-spec and whole-part associations are structure associations and thus controlled by the Structure flag in the View menu.

In the *instance-to-instance associations*, object instances know about each other in some way (e.g. by keeping track of each other's identities or addresses). The association is a attribute level association and is thus controlled by the Attribute flag in the View menu.

In the *message associations*, objects can exchange messages. This association is not a structure association but a service level association. The messages are used to initiate services! Thus you control it via the Services flag in the View menu. If the message connections make the drawing unreadable, e.g. when sending messages from an object to almost all other objects, you may use the NewSymb multi-head messages. The symbol that you send widely (without closer specification in the graphical syntax). There is also an opposite direction of multi-head message symbol, i.e. reception from an unspecified number of objects.

Note that in general the two objects involved in a connection have different roles, so keep track of who is who when connecting them together. The "Existing Connectors" section of the dialogue box presented when creating a new connection has a REVERSE button which allows you to reverse the direction of the connection (the first object selected defaults to

_OOAnotation

\$ Coad/Yourdon OOA Notation

K Coad/Yourdon;attribute;service;object;class;instance;gen-spec association;whole-part association;instance-instance association;message association;connector;association

+ EDITOOA:010

being the senior).

#SK+ Creation of OOA Symbols

Objects/Object-&-Classes are created by means of the menu NEWSYMB,OBJECT. OOther OO Diagram Editor presents a dialogue where You have to enter name of the object, its type class or object-&-class, and optional description text.

The created object symbol is located on the screen at a predefined location and has to be moved by dragging with mouse to a desired location.

By **double-clicking** on the relevant part you may edit the contents of that part. For example, double-clicking on the object identity part will enable you to read or modify the name of the object and the descriptive text, and to change a class to a class-&-object (and vice-versa).

Similarly, double-clicking on the attribute or service part allows you to **add, browse** or **edit** definitions of **attributes** and **services** contained in the class.

Associations between objects are implemented as connectors.

To **connect/associate two objects**: Select one object by clicking on its object identity part. If it is black you do not need to click on it again. Select the object to which it is to be connected by pointing to its object identity part with the mouse and pressing SHIFT + left mouse button. OOther OO Diagram (OOA) Editor will present a connector dialogue, where you can choose type of connector, direction and, if applicable, a cardinality specification. Also a verbal description may be stored.

See also mouse conventions.

The template for descriptions can be specified by referring in WIN.INI to a file containing the template.

```
#_OOAsymbols
$ Creation of OOA Symbols
K object;mouse;double-click;template
+ EDITOOA:020
```


#SK+ OOA Diagram Utilities

Under the menu for Utilities you find options to invoke the other editors (TMC, FSM and UC) and also two handy tools showing inheritance for the currently selected object (UTILITIES, SHOW INHERITANCE) and one showing statistics about your OO model (UTILITIES, STATISTICS)

_OOAutils

\$ OOA Diagram Utilities

K editors;switching to TMC editor;switching to FSM editor;OOA editor utilities; switching to UC editor

+ EDITOOA:030

#SK+ **OO Diagram (OOA) Editor Menus**

File

- New
- Open...
- Merge (read)...
- Save
- Save As...
- Exit
- Print (All/Selections)...
- Print Parts
 - Print OOA Graphics...
 - Print TMC Graphics...
 - Print FSM Graphics...
 - Print Use Case Overview Graphics...
 - Print Use Case Overview Text...
 - Print All Use Case Seq.Diagr Graphics...
 - Print OOA Text...
 - Print TMC Text...
- Printer Setup...
- Configure...
- Customize Paths...
- Color...

Edit

- Copy All Graphics to Clipboard
- Copy Selected Objects
- Copy Connectors (Internal Clipboard)
- Paste Connectors (Internal Clipboard)
- Delete Connectors
- Search...

View

- Structure
- Attribute
- Service
- Page Boundaries
- Selected zoom factor:
 - Zoom 2x (50%)
 - Zoom 3x (33%)
 - Zoom 4x (25%)
 - Zoom 5x (20%)

NewSymbol

- Object...
- Import/Copy Object...
- Title...

Utilities

- Task/Process Components Chart

_OOAmenus

\$ OO Diagram (OOA) Editor Menus

K OOA Editor;menus

+ EDITOOA:040

Finite State Machine Flow Chart

Use Case Chart(s)

Show Inheritance

Statistics

Generate C++ Header file for this class

Generate ALLHeader files

Show C++ header file for this object

List of Contents

Zoom

About

Help

#SK+ **Mouse in OO Diagram (OOA) Editor**

Selecting an Object, See [mouse conventions](#)

Moving Objects, [See mouse conventions](#)

Selecting Group of Objects, [See mouse conventions](#)

Editing/Deleting of the title is performed by double clicking on the title string in the top of the chart. Double clicking invokes the same dialogue as invoked by the menu [Title...](#)

Editing/Deleting Object and its Name and Description is performed by double clicking on the up-most part of the object symbol, i.e. inside the rectangle containing object name. Double clicking invokes the same dialogue as invoked by the menu [Object...](#)

Adding/Editing/Deleting Attributes and their Names and Descriptions is performed by double clicking on the middle part of the object symbol, i.e. inside the rectangle containing attribute names. Double clicking invokes the [Attribute dialogue](#)

Adding/Editing/Deleting Services and their Names and Descriptions is performed by double clicking on the bottom part of the object symbol, i.e. inside the rectangle containing service names. Double clicking invokes the [Service dialogue](#)

If You first select an object by clicking on it and then shift-click on a second one, You may Add/Modify/Delete Object Associations. In this case the tool invokes a [connector dialogue](#)

Setting/Resetting from Zoom View is performed by pressing the right mouse button, [See mouse conventions](#) and [menu Zoom](#).

Presenting Inheritance is performed by pressing SHIFT and the right mouse button, [See mouse conventions](#) and dialogue for menu [Show Inheritance](#).

_OOAmouse
\$ Mouse in OO Diagram (OOA) Editor
K editors;OOA editor;mouse
+ EDITOOA:050

#SK+ **Task Management Component (TMC) Editor**

OOther/TMC Notation

Creation of Symbols

OOther/TMC Utilities

TMC Editor Menus

Mouse in TMC Editor

_TMCeditor

\$Task Management Component (TMC) Editor

K editor;TMC editor

+ EDITTMC:000

#SK- OOther/TMC Notation

The TMC chart shows allocation of objects in the OO-diagram to **tasks/processes** and **computers/processors**. To define how the objects are handled in the executable space in a **design problem**.

The symbols are ordered into three columns of boxes. The boxes are ordered in alphabetic order.

The leftmost boxes represent processors, the middle tasks, and the rightmost all the objects defined in the OO Diagram.

Connectors between boxes represent allocations. When You select a box (object, task or processors) the corresponding allocation connectors will be illuminated.

When you print the text parts of the OOA and TMC models OOther shows which message associations need interprocess communication (IPC) and which interprocessor communication (datacommunication) transfer.

#_TMCnotation
\$OOther/TMC Notation
K processor;task;object;allocation
+ EDITTMC:010

#SK + **Creation of TMC Symbols**

Symbols are created by selecting the menu option NEWSYMB, PROCESSOR and NEWSYMB, TASK. OOther presents then a pop-up menu where the user may specify name and a description of entity.

The symbols may be modified by double-clicking with the left mouse button.

Allocation connectors are created/removed by selecting a processor, task and object and shift-clicking on the other entity.

See also mouse conventions.

_TMCsymbols
\$ Creation of TMC Symbols
K TMC editor;processor;task;process;allocation
+ EDITTMC:020

#SK + **TMC Utilities**

The menu **OO-diagram** is used to switch back to the OOA editor. The TMC contents are still there even if not visible.

#_TMCutils
\$TMC Utilities
K switching to OOA editor
+ EDITTMC:030

#SK+ **TMC Diagram Editor Menus**

File

- Save
- Save As...
- Print (All/Selections)...
- Print Parts
- Print TMC Graphics...
- Print TMC Text...
- Printer Setup...
- Configure...
- Customize Paths...
- Color...

Edit

- Copy All Graphics to Clipboard

View

- Selected zoom factor:
- Zoom 2x (50%)
- Zoom 3x (33%)
- Zoom 4x (25%)
- Zoom 5x (20%)

NewSymbol

- Processor...
- Task...

Utilities

- List of Contents

OO-diagram

Zoom

About

Help

_TMCmenus
\$ TMC Diagram Editor Menus
K editors;TMC Editor;menus
+ EDITTMC:040

#SK+ **Mouse in TMC Diagram Editor**

Selecting an Object, Task and Processor, just click on the symbol. See [mouse conventions](#)

Editing/Deleting Task and Processor and its Name and Description is performed by double clicking on the symbol rectangle. Double clicking invokes the same dialogue as invoked by the menu under NewSymb [Processor...](#) and [Task...](#)

If You first select a symbol (processor, task or object) by clicking on it and then shift-click on a second one of another kind, You may Add/Delete Allocation Associations, [see mouse conventions](#)

Setting/Resetting from Zoom View is performed by pressing the right mouse button, [see mouse conventions](#) and [menu Zoom](#).

_TMCmouse
\$ Mouse in TMC Diagram Editor
K editors;TMC Editor;mouse
+ EDITTMC:050

#SK+ **Finite State Machine (FSM) Editor**

The FSM Editor can be invoked in two ways:

+ via the menu in the OOA Editor, there you find a common diagram not connected to any service.

+ by pressing the button "FSM diagram" in the Service dialogue in the OOA Editor (double click on the bottom part of an object to get the dialogue and then select the service for which you want edit the FSM diagram.

OOther/SDL Dialect and Notation

Creation of Symbols

FSM Utilities

FSM Editor Menus

Mouse in FSM Editor

#_FSMeditor

\$ Finite State Machine (FSM) Editor

K editors;FSM editor

+ EDITFSM:000

#SK- OOther FSM/SDL Dialect and Notation

The FSM Editor is used to define the dynamic behaviour of each of your objects, e.g. state machines. The graphic language used is a simplified (standard) SDL.

The work is performed both during Object Oriented Analysis and Design.

Meaning of symbols follows:

μ §

State

The *state* symbol denotes a stable and reproducible situation in the finite automata.

Received message

A *received message* corresponds to a received information or signal. In OO languages it might correspond to a call to a method/service/member function, or reception of a letter.

Sent message

A *sent message* corresponds in OO languages to invoking a method/service/member function in a class, or sending of a letter.

Synchronous message sending + reception

It is a aggregation of send and a receive symbol, however, it is performed immediately without involving the state handling mechanisms, i.e. a synchronous inquiry (asking).

Action/Processing

The *action* symbol describes that some kind of processing is performed synchronously.

Condition/Selection

The *condition* symbol allows selection of alternative processing (if then/else or a switch/case block). Symbols following the condition symbol must be completed by a condition text meaning true/false. However this version does not perform any syntax checks.

μ §

Iteration

The *iteration* symbol denotes a repetitive processing. You may specify loop condition within the box. Everything to the corresponding next/label symbol is performed.

Next iteration/label

Denotes end of a iteration block. It may also be used as a label.

Text

Allows adding a *comment* to the chart.

Style of FSM/SDL diagrams

There are two ways to draw the diagrams:

1. Start with a state and show which received messages/events can occur in that state and what is done (state driven FSM). I.e. that a starting state is only present in one place in the diagram, but a message reception symbol can practically be directly under any of the states.

_FSMnotation

\$ OOther FSM/SDL Dialect and Notation

K SDL;FSM;state machines;flow charts;state;message;

+ EDITFSM:010

2. Start with an received message/event and branch on the current state (event driven FSM), here the received message is present only once in the diagram but the branching on a particular state may be present for every message type.

Which style you use is up to You... The style 1 makes it easier to check that no event is forgotten for every state, the style 2 is more similar to the programming languages.

[Informal note: Theoretically it is possible to automatically translate these styles to each other...]

#* Creation of FSM Symbols

You create symbols by choosing one of options under NewSymb. The symbols are explained above.

To *connect the symbol to a previous one*, select it and press SHIFT-left-Button on the previous symbol. To disconnect two symbols select it and press SHIFT-Left-Mouse-Button while pointing on itself.

By using the multiple selection (grouping) mouse function, CTRL-left mouse button you can select up to 30 objects which can be moved, copied to clipboard, aligned or deleted. To switch off the grouping operator click outside any symbol.

See also mouse conventions.

#SK+ **FSM Utilities**

The menu **OO-diagram** is used to switch back to the OOA editor. The FSM contents are still there even if not visible.

#_FSMutils
\$Finite State Machine (FSM) Editor, FSM Utilities
K switching to OOA editor
+ EDITFSM:030

#SK+ **FSM Diagram Editor Menus**

File

Save
Save As...
Print (All/Selections)...
Print Parts
Print FSM Graphics...
Printer Setup...
Configure...
Customize Paths...
Color...

Edit

Copy All Graphics to Clipboard
Copy Selected Graphics to Clipboard
Align to Top
Align Center
Delete selected symbols
Delete ALL symbols

View

Selected zoom factor:
Zoom 2x (50%)
Zoom 3x (33%)
Zoom 4x (25%)
Zoom 5x (20%)

NewSymbol

State...
Receive...
Send...
Synch send+receive...
Action...
Condition / Selection...
Iteration (Start of Iteration)...
Next (End of Iteration) / Label...
Text comment...

Utilities

List of Contents
Customize Paths...
OO-diagram
Zoom
About
Help

#_FSMmenus
\$FSM Editor Menus
K editors;menus;FSM editor
+ EDITFSM:040

#SK+ **Mouse in FSM Editor**

Selecting a symbol - just click inside the symbol, See [mouse conventions](#)

Moving Symbols, just select and drag with left-button pressed, [See mouse conventions](#)

To move a group of symbols, select them one by one by pressing CTRL-left mouse button and then click inside a symbol and drag to the new location. Click outside a symbol to finish moving.

Editing/Deleting Symbols is performed by double clicking inside the symbol. Double clicking invokes the same dialogue as invoked by any of the menus under NewSymb: [State...](#), [Receive...](#), [Send...](#), [Synch send+receive...](#), [Action...](#), [Condition / Selection...](#), [Iteration \(Start of Iteration\)...](#), [Next \(End of Iteration\) / Label...](#), [Text comment...](#)

Use Edit Align functions to arrange group of selected symbols (by CTRL-left mouse button).

If You first select a symbol by clicking on it and then shift-click on a second one, You may create a **connector line**. By connecting the symbol to itself you remove the connector line between two symbols.

The symbol Next/Label may carry up to 20 connectors to preceeding symbols, the others just to one. You may not connect a Next/Label symbol directly to a Condition or Iteration symbol. If you need it, add a "dummy" Action box.

Setting/Resetting from Zoom View is performed by pressing the right mouse button, [See mouse conventions](#) and [menu Zoom](#).

#_FSMmouse
\$Mouse in FSM Diagram Editor
K editors;FSM editor;mouse
+ EDITFSM:050

#SK+ Use Case Overview Diagram (UC) Editor

Use Case Notation

Creation of Use Case Symbols

Invoking the Sequence Diagram (SEQ) Editor

Use Case Utilities

UC Editor Menus

Mouse in UC Editor

#_UCeditor

\$ Use Case Overview Diagram (UC) Editor

K editors; Use Case Editor

+ EDITUC:000

#SK+ Use Case Overview Notation

Use Cases help to understand system dynamic. One of the lessons learned in the early OO was that none is happy by just having many objects described, and not really knowing how they co-operate.

You can also in some cases use the use cases as domains.

The notation uses two symbols:

+ ellipses, symbolising **Use Cases**

+ **actors**, symbolising *operators* and *external systems* or parts of the system not embedded in the use case

μ §

The use cases may inherit or extend each other's contents. So also the actors.

The arrows between the symbols have directions.

Read the texts in direction of the arrow.

_UCnotation

\$ Use Case Overview Notation

K Use Case;Scenario;Actor;uses;extends;inherits;domain

+ EDITUC:010

#SK+ Creation of Use Case Overview Symbols

To create an actor or use case symbol select one of the options under the menu NewSymb: Actor..., Use Case.. or Text.... All three invoke a dialogue box where you can specify a short name and a descriptive text.

The button OK accepts the entered text, while the button DELETE removes the symbol. The Use Case dialogue contains also a button for invoking the Sequence Diagram (SEQ) Editor.

The entered information will be printed when ordering **Print Use Case Overview Texts**.

To move any of the symbols to a new location simply select it by pressing the left mouse button and dragging to a new location.

By double clicking on the symbol you will invoke the dialogue for editing the name and description texts.

#SK+ **Invoking the Use Case Sequence Diagram (SEQ) Editor**

In the dialogue for editing use case texts You can find the button **Sequence Diagram**. By pressing it you will enter a new editor. See [Sequence Diagram \(SEQ\) Editor](#)

_UCseq
\$ Invoking the Use Case Sequence Diagram Editor
K Use Case;Scenario;Sequence Diagram
+ EDITUC:030

#SK+ Use Case Utilities

You can verify consistency of all use cases or just the selected one by selecting the option **UTILITIES, Verify Use Cases...**

When verifying all use cases you may also get the coverage percentage for objects and services.

Values <100% may mean that there are aspects you not discussed in the use cases. (100% says that you maybe covered the significant parts).

See [Verifying Sequence Diagram \(SEQ\)](#)

#SK+ **UC Diagram Editor Menus**

File

- Save
- Save As...
- Print (All/Selections)...
- Print Parts
 - Print Use Case Overview Graphics...
 - Print Use Case Overview Text...
 - Print All Use Case Seq.Diagr Graphics...
 - Print Current Use Case Seq.Diagr.Graphics...
- Printer Setup...
- Configure...
- Customize Paths
- Color...

Edit

- Copy All Graphics to Clipboard

View

- Selected zoom factor:
 - Zoom 2x (50%)
 - Zoom 3x (33%)
 - Zoom 4x (25%)
 - Zoom 5x (20%)

NewSymbol

- Use Case...
- Actor...
- Text comment...

Utilities

- Verify Use Case
- List of Contents

OO-diagram

Zoom

About

Help

_UCmenus

\$ UC Diagram Editor Menus

K editors;UC editor;menus

+ EDITUC:050

#SK+ Mouse in UC Diagram Editor

Selecting a symbol - just click inside the symbol, See mouse conventions

Moving Symbols, just select and drag with left-button pressed, See mouse conventions

Editing/Deleting Symbols is performed by double clicking inside the symbol. Double clicking invokes the same dialogue as invoked by any of the menus under NewSymb: Use Case..., Actor..., Text comment...

If You first select a symbol by clicking on it and then shift-click on a second one, You may create a connector line. By connecting the symbol to itself you remove the connector line between two symbols. A dialogue is invoked allowing to specify some descriptive text and name of the connector, see UC Connection.

Setting/Resetting from Zoom View is performed by pressing the right mouse button, See mouse conventions and menu Zoom.

^{#SK+} [Use Case Sequence Diagram \(SEQ\) Editor](#)

Note! Jacobson uses another name for the Sequence Diagrams: *Interaction Diagrams*.

Use Case Sequence Diagram Notation

Creation of Use Case Symbols

Verifying Use Case Sequence Diagram

SEQ Editor Menus

Mouse in SEQ Editor

#_SEQeditor

\$Use Case Sequence Diagram (SEQ) Editor

K editors;SEQ editor; Use Case; Sequence Diagram;Interaction Diagram

+ EDITSEQ:000

#SK+ Use Case Sequence Diagram (SEQ) Notation

The notation uses four symbols:

- + objects
- + actors
- + services
- + messages

μ §

Objects and actors are expressed as vertical lines, services as rectangles located above the objects, messages are arrows going between the objects. Messages start services. You may also add an optional response message transferred to the object who sent the message starting the service, e.g. to tell that something went wrong.

μ §

#_SEQnotation

\$ Use Case Sequence Diagram (SEQ) Notation

K Use Case; Scenario; Object; actor; service; message; Interaction Diagram

+ EDITSEQ:010

#SK- Creation of Use Case Sequence Diagram (SEQ) Symbols

To assign an object or an actor to a use case select the menu option NewSymb: Actor or Object.

When assigning an object, OOther will present a list of all the objects defined in the OOA chart. I.e. that you cannot create any use case sequence diagram until You have defined Your objects and their relations.

The button OK accepts the entered text, while the button Cancel keeps the symbol unchanged. By means of the **Utilities** menu You can move the objects/actors to left or rite, or even delete them.

To add a message/service you have first select an object by clicking on the vertical object line. Then connect it with an other object by a SHIFT-Left-click. Than You will see a dialogue presenting all the defined services in the receiving object and a edit box for the name of the message. Again You can only choose one of the existing services in the OOA diagram. Name of the message can be set to any value.

To edit a message or service double click on the upper part of the service rectangle.

To move the service rectangle to another location just click and drag in a vertical direction.

To make a service rectangle longer/shorter click and drag on the lower part of the rectangle.

#_SEQsymbols

\$ Creation of Use Case Sequence Diagram (SEQ) Symbols

K Use Case;Scenario;Object;Actor;service;message;Interaction Diagram

+ EDITSEQ:020

#SK- Verifying Use Case Sequence Diagram

You can verify consistency of the use cases by selecting the option Utilities, Verify Use Case

The results are written to a file which is viewed by Windows Notepad.

#_SEQverify
\$ Verifying Use Case Sequence Diagram
K Use Case;Scenario;Verify;Consistency
+ EDITSEQ:030

#SK+ **SEQ Diagram Editor Menus**

File

- Save
- Save As...
- Print (All/Selections)...
- Print Parts
- Print Current Use Case Seq.Diagr.Graphics...
- Printer Setup...
- Configure...
- Customize Paths...
- Color...

Edit

- Copy All Graphics to Clipboard
- Delete Connectors
- Search...

View

- Page Boundaries
- Selected zoom factor:
- Zoom 2x (50%)
- Zoom 3x (33%)
- Zoom 4x (25%)
- Zoom 5x (20%)

NewSymbol

- Object...
- Actor(s) / SystemBorder(s)...
- Text comment...

Utilities

- Move Object to Left
- Move Object to Right
- Delete Service
- Delete Object
- Verify Use Case
- List of Contents

UC-diagram

Zoom

About

Help

_SEQmenus
\$ SEQ Diagram Editor Menus
K editors;UC Editor;menus
+ EDITSEQ:040

#SK+ **Mouse in SEQ Diagram Editor**

Selecting an object - just click on the vertical line corresponding to the object, see [mouse conventions](#)

Selecting a service - just click on the upper part of the service rectangle, see [mouse conventions](#)

Moving a service rectangle, just select it by clicking on the upper part of the service rectangle and drag with left-button pressed in a vertical direction (up or down), [see mouse conventions](#)

Resizing a service rectangle, just select it by clicking on the lower part of the service rectangle and drag with left-button pressed in a vertical direction (up or down), [see mouse conventions](#). If the rectangle is small, the resizing has precedence before moving.

Moving a text, just select it by clicking and drag with left-button pressed to a new location, [see mouse conventions](#)

Editing services and texts is performed by double clicking inside the symbol. Double clicking invokes dialogues as [Text comment...](#) or [SEQ Service and Message](#)

If You first select an object or actor by clicking on it and then shift-click on a second one, You may create a message and service symbol, see [SEQ Service and Message](#)

Setting/Resetting from Zoom View is performed by pressing the right mouse button, [See mouse conventions](#) and [menu Zoom](#).

Common Utilities

List of contents may be invoked from all editors. It gives a short list of items in all the defined models. It can be usable to quickly take look of what was the name of an attribute or service. Especially the possibility to have the list of contents beside OOther windows can make your life easier - as you noticed this version of OOther shows one diagram at a time.

#_commonUtils
\$Common Utilities
+ UUTILS:000

#SK+ Clipboard

Clipboard may be used to transfer copy of the graphics to other tools under MS-Windows, e.g. Word for Windows. All three editors (OO Diagram, TMC, FSM, UC and SEQ Editors) allow copying of all graphics, EDIT, COPY ALL.

OO Diagram Editor allows also selection of some objects by CTRL-left-button-click and copying by using **EDIT, COPY SELECTED OBJECTS**. I.e. the menu location has a dual behaviour.

The clipboard transfer (copy-paste) uses windows metafiles (i.e. commands to draw the picture) rather than a screen copy. The receiving application must in so way understand windows metafiles written with ANISOTROPIC formatting. MS-Word for Windows 2.0 and MS-Excel 3.0/4.0 accepts these clipboard contents.

Unfortunately, due to lack of information on clipboard formats expected by MS-Word for Windows Draw the pictures imported to WfW cannot be edited in Draw.

If You are the lucky one who knows, please tell me how to format clipboards for Draw - or give me a hint where to read about it!

There is no way to copy graphics or text to OOther from other applications.

To copy connectors between objects in the OO Diagram editor there is also a simple *internal* clipboard. See also EDIT COPY CONNECTORS and EDIT PASTE CONNECTORS

#_clipboard

\$Clipboard

K clipboard;metafile;copy-paste;editors;OOA Editor;TMC Editor;FSM editor;UC Editor;SEQ Editor

+ UCLIP:010

#SK- Printing

There are two ways to order printing: via FILE, PRINT (All/Selections) and via FILE, PRINT PARTS

The menu option , **File Print (All/Selections)** formats graphics and object descriptions for your printer. The dialogue box is quite complex; it contains a large set of flags where you can define the data to be printed. For increased flexibility the dialogue box also contains a list of all defined objects.

The menu options **FILE, PRINT PARTS, Print OOA Graphics, FILE, PRINT PARTS, Print TMC Graphics, FILE, PRINT PARTS, Print TMC Graphics, FILE, PRINT PARTS, Print FSM Graphics, PRINT PARTS, Print Use Case Graphics** and **PRINT PARTS, Print Use Case Seq.Diagr. Graphics** allow to print a selected range of pages, e.g. page 3 through 5.

The menu options **FILE, PRINT PARTS, Print OOA Text, FILE, PRINT PARTS, Print TMC Text**, and **FILE, PRINT PARTS, Print Use Case Overview Text** direct all text to printer.

To configure the printer for the current OOther session, e.g. select printer and/or orientation, use **FILE,PRINTER SETUP.**

#_printing
\$Printing
K printing;printer;printer setup;configuring the tool
+ PRINT:000

#S- [Reference manual](#)

Mouse:

[Mouse invoked functions](#)

Menus:

[File](#)

[Edit](#)

[View](#)

[NewSymb](#)

[Utilities](#)

[OO-diagram](#)

[UC-diagram](#)

[Zoom](#)

[About](#)

[Help](#)

Other:

[Parameters in Win.ini](#)

[File Format](#)

[Limitations & Known Bugs](#)

[Postscript printers](#)

#SK+ **Mouse Invoked Functions**

Access to:

OOA Object

OOA Attribute

OOA Service

OOA Connection

SEQ Service and Message

UC Connection

TMC-Symbols Processor

TMC-Symbols Task

FSM-Symbols State

FSM-Symbols Receive

FSM-Symbols Send

FSM-Symbols Synch Send+Receive

FSM-Symbols Action

FSM-Symbols Condition/Selection

FSM-Symbols Iteration

FSM-Symbols Next/Label

FSM-Symbols Text

UC-Symbols Use Case

UC-Symbols Actor

UC-Symbols Text

SEQ-Symbols Text

Zooming

#_mouseCall

\$Mouse Invoked Functions

K mouse;double-click;select object;attribute;object;service

+ ZREFAA:010

#SK+ **Mouse Invoked Function - Object**

The function is invoked when double-clicking with the left mouse button on the up-most part of the object symbol. The function invokes the same dialogue as NewSymb Object and allows modifying characteristics of the object/object-&-class.

See also introduction, Mouse Convention and Coad/Yourdon Notation

#_mouseObject
\$Mouse Invoked Function - Object
K mouse;double-click;select object;attribute;object;service
+ ZREFAA:020

#SK+ **Mouse Invoked Function - Attribute**

The function is invoked when double-clicking with the left mouse button on the middle part of the object symbol. The function invokes the dialogue allowing modification of characteristics of the attributes.

Purpose:

To add/modify/remove attributes in an object/object-&-class

Editor:

OO-Diagram Editor

Dialogue Attributes for Object xxxx:

The dialogue operates attributes.

Dialogue Items in Attributes for Object xxxx:

New-button - creates a new attribute

Attribute selection list - is used to point out an existing attribute

Name - is used to new enter name of attribute at renaming

Description - window containing textual description of the attribute. Template may be specified by the user, see [templates](#)

Return-button - returns to the editor

Delete-button - deletes the selected attribute

Limitations:

-

Short Cuts:

-

See also:

Introduction, [Mouse Convention](#) and [Coad/Yourdon Notation](#)

#_mouseAttribute

\$Mouse Invoked Function - Attribute

K mouse;double-click;attribute

+ ZREFAA:030

#SK+ **Mouse Invoked Function - Service (in OOA Diagram)**

The function is invoked when double-clicking with the left mouse button on the bottom part of the object symbol. The function invokes the dialogue allowing modification of characteristics of the services.

Purpose:

To add/modify/remove services in an object/object-&-class

Editor:

OO-Diagram Editor

Dialogue Services for Object xxxx:

The dialogue modifies services.

Dialogue Items in Services for Object xxxx:

New-button - creates a new service

Service selection list - is used to point out an existing service

Name - is used to new enter name of service at renaming

Description - window containing textual description of the service. Template may be specified by the user, see [templates](#)

Return-button - returns to the editor

Delete-button - deletes the selected service

Limitations:

-

Short Cuts:

-

See also:

Introduction, [Mouse Convention](#) and [Coad/Yourdon Notation](#)

#_mouseService

\$Mouse Invoked Function - Service (in OOA Diagram)

K mouse;double-click;service;OOA Diagram

+ ZREFAA:040

#SK+ **Mouse Invoked Function - Service and Message (in SEQ Diagram)**

The function is invoked when connecting two objects/actors in the Use case Sequence Diagram:

- + select the sending object/actor by clicking on its vertical line
- + SIFT-LeftClick on the receiving object/actor.

It can so be invoked when double clicking on the top of the service rectangle in the diagram.

Purpose:

To add/modify services+messages in a Use Case Sequence Diagram.

Editor:

SEQ Editor

Dialogue Add a Service to an Object in the Use Case Sequence Diagram:

The dialogue modifies services.

Dialogue Items in Add a Service to an Object in the Use Case Sequence Diagram:

Message - name string for the message

Service Name selection list - is used to point out an existing service

Description - read/only window containing textual description of the service

Return-button - returns to the editor and accept changes

Cancel-button - returns dialogue without any changes

Limitations:

-

Short Cuts:

-

See also:

Introduction, [Mouse Convention](#) and [UC SEQ Notation](#)

#_mouseSEQservice

\$Mouse Invoked Function - Service and Message (in SEQ Diagram)

K mouse;double-click;service;Use Case Sequence Diagram;Interaction Diagram

+ ZREFAA:050

#SK+ **Mouse Invoked Function - Connection (in OOA Diagram)**

The function is invoked when first selecting an object by clicking with the left mouse button and then by pressing the shift key and clicking on an another object. The function invokes the dialogue allowing modification of characteristics of the connections/associations.

Purpose:

To add/modify/remove connections/associations between two objects/object-&-classes

Editor:

OO-Diagram Editor

Dialogue Services for Object xxxx:

The dialogue modifies connections.

Dialogue Items in Services for Object xxxx:

New-button - creates a new connection

Manual-button - allows manual adjustment of the connecting line between the objects

Reverse-button - reverses direction of connection

Connection selection list - is used to point out which of the existing connections is to be changed/browsed

Connection type list - is used to define/change type of the connection

Multi - edit box - allows entry of cardinality (instance and whole-part connections)

Description - window containing textual description of the service. Template may be specified by the user, see [templates](#)

Return-button - returns to the editor

Delete-button - deletes the selected connection

Limitations:

-

Short Cuts:

-

See also:

Introduction, [Mouse Convention](#) and [Coad/Yourdon Notation](#)

#_mouseConnection

\$Mouse Invoked Function - Connection

K mouse;click;select object;connector

+ ZREFAA:060

#SK+ **Mouse Invoked Function - Connection Use Case/Actor (in UC Overview Diagram)**

The function is invoked when first selecting an Use Case or Actor by clicking with the left mouse button and then by pressing the shift key and clicking on an another use case/actor. The function invokes the dialogue allowing modification of characteristics of the connections/associations.

Purpose:

To add/modify/remove connections/associations between two use cases and actors

Editor:

UC Editor

Dialogue Use Case Relation Description:

The dialogue modifies connections.

Dialogue Use Case Relation Description:

Name string - is used to assign name visible in the UC Overview diagram

Short Description - a short description of the relation

Templates are not supported here.

OK-button - accepts the input data and returns to the UC editor

Delete-button - deletes the selected connection

Limitations:

-

Short Cuts:

-

See also:

Introduction, [Mouse Convention](#) and [UC Notation](#)

#_mouseUCconnection

\$Mouse Invoked Function - Connection

K mouse;click;use case;actor;connector

+ ZREFAA:070

#S- **File**

Menus:

File

- New
- Open...
- Merge (read)...
- Save
- Save As...
- Exit
- Print (All/Selections)...
- Print Parts
 - Print OOA Graphics...
 - Print TMC Graphics...
 - Print FSM Graphics...
 - Print Use Case Overview Graphics...
 - Print Use Case Overview Text...
 - Print All Use Case Seq.Diagr Graphics...
 - Print Current Use Case Seq.Diagr.Graphics...
 - Print OOA Text...
 - Print TMC Text...
- Printer Setup...
- Configure...
- Customize Paths...
- Color...

#_file

\$File

+ ZREFFI:000

#s:File New

Purpose:

To create an empty model.

Editor:

OO-Diagram Editor

Dialogue Are You Sure?, File is modified - skip your changes?:

If there is an modified model in any of editors, the user will be warned and have a chance to cancel the **FILE,NEW** request by pressing the No-button.

Dialogue Items in Are You Sure?, File is modified - skip your changes?:

Yes-button - clears the editors without saving your work

No-button - goes back to editor

Limitations:

-

Short Cuts:

-

See also:

-

#_fileNew

\$File New

+ ZREFFI:010

#S-File Open

Purpose:

To read an existing file into editors.

The recently opened files are also listed on the FILE menu - the last file in the top, the oldest in the bottom.

You may also pass the name of the file to be open when starting OOther by clicking on an OOT-file under MS-Windows File Manager or when starting MS-Windows from DOS (e.g. win oot c:\projfoo\anal\bar.oot)

Editor:

OOA-Dialogue Editor

Dialogue Are You Sure?, File is modified - skip your changes?:

If there is an modified model in any of editors, the user will be warned and have a chance to cancel the **FILE,NEW** request by pressing the No-button.

Dialogue Items in Are You Sure?, File is modified - skip your changes?:

Yes-button - clears the editors without saving your work

No-button - goes back to editor

Dialogue Open

The standard MS-Windows 3.1 Open dialogue is used

Dialogue Items in Open:

File Name: specify file name by selecting in the list or typing in the requested name, or specify name using regular expressions (wild cards).

Directories: specify the directory, as default the home directory is used (see WIN.INI)

Drivers: specify the drive

List Files of Type: - not relevant

OK-button - goes back to editor

Cancel-button - goes back to editor

Limitations:

-

Short Cuts:

-

See also:

-

#S- **File Merge**

Purpose:

To merge (concatenate) a file to the edited one.

Editor:

OOA-Dialogue Editor

Dialogue Merge, Where (Pixel location)?:

The dialogue allows You to change the proposed co-ordinates where to read the external OO-diagram.

Dialogue Items in Merge, Where (Pixel location)?:

X= horizontal pixel address (0...32767)

Y= vertical pixel address (0...32767)

OK-button - accepts the X and Y values

Cancel-button - cancels the merging

Dialogue Merge with File

The standard MS-Windows 3.1 Open dialogue is used

Dialogue Items in Merge with File:

Same as in FILE,OPEN.

Limitations:

Only OO-Diagram and TMC-diagram is read (i.e. FSM, UC and SEQ parts are skipped)

Short Cuts:

-

See also:

FILE,OPEN

#_fileMerge

\$File Merge

+ ZREFFI:030

#S- **File Save**

Purpose:

To save contents of all the editors to file.

Editor:

OO-Diagram Editor, TMC Editor, FSM Editor, UC Editor, SEQ Editor

Dialogue:

If no file name is assigned OOther will perform the **FILE, SAVE AS** dialogue. If the file name is known the file will be stored without any messages (except if it fails).

Dialogue Items:

-

Limitations:

-

Short Cuts:

-

See also:

FILE, SAVE AS

#S- **File Save As**

Purpose:

To save contents of the editors to a file with specifying the file name.

Editor:

OO-Diagram Editor, TMC Editor, FSM Editor, UC Editor, SEQ Editor

Dialogue Save As:

The standard MS-Windows 3.1 Save As dialogue is used. The dialogue checks for accidental overwriting of existing files.

Dialogue Items in Save As:

File Name: specify file name by selecting in the list or typing in the requested name, or specify name using regular expressions (wild cards).

Directories: specify the directory, as default the home directory is used (see WIN.INI)

Drivers: specify the drive

List Files of Type: - not relevant

OK-button - goes back to editor

Cancel-button - goes back to editor

Limitations:

-

Short Cuts:

-

See also:

FILE, SAVE

#S* **File Exit**

Purpose:

To leave the OOther

Editor:

OO-Dialogue Editor

Dialogue Leaving OOther:

The dialogue inform the user that the OOther session has been requested to be terminated. If there is an modified file the user can save it.

Dialogue Items in Leaving OOther:

Save + Exit button save the current editor contents by using FILE,SAVE and exit the tool

Don't Save + Exit button skips Your changes and exits the tool

Don't Exit button cancels the exit request, and returns back to OOther

Update settings in WIN.INI check box, when checked (default) all modification of configuration parameters will be remembered to the next session, see also WIN.INI for format of stored configuration parameters.

Limitations:

-

Short Cuts:

Alt-F4

See also:

FILE,SAVE

#_fileExit

\$File Exit

+ ZREFFI:060

#S* **File Print (All/Selection)**

Purpose:

To format a graphical and/textual printout of the models in editors.

Editor:

OO-Diagram Editor, TMC Editor, FSM Editor, UC Editor, SEQ Editor

Dialogue Print:

The dialogue allows to specify which kind of information to be printed, and which object names are to be selected. To better fit the varying printers You can zoom.

The dialogue is quite complex, so check one more time before pressing the **Print-button**, it may be lot of pages...

Dialogue Items in Print:

Check boxes under **What to print:**

OOA Graphics, print OO-Diagram

All objects in graphics, when checked all objects are printed, if not only these marked (as black) the **Object list**

Object Descriptions, when checked the textual object descriptions will be printed for all or selected objects depending on what is marked in the **Object list**.

Attribute Descriptions, when checked the textual attribute descriptions will be printed for all or selected objects depending on what is marked in the **Object list**.

Service Descriptions, when checked the textual attribute descriptions will be printed for all or selected objects depending on what is marked in the **Object list**.

Descriptions, when checked the textual service descriptions will be printed for all or selected objects depending on what is marked in the **Object list**.

List of associations, when checked the textual service descriptions will be printed for all or selected objects depending on what is marked in the **Object list**.

Task Mgmt Comp graph, when checked the graphical TMC diagram is printed

TMC, descriptions, when checked the entire textual TMCdescriptions will be printed

FSM, flow/state diagram, when checked the entire graphical FSM diagram is printed

UC graphics when checked the entire graphical UC diagram is printed

UC Descriptions when checked the entire textual UC descriptions will be printed

UC SEQ graphics when checked all the graphical Use case Sequence diagrams will be printed

Object list, the list contains names of all objects in the OO-diagram. By clicking you can select an object or a set of consecutive objects. by CTRL-left-mouse-button you may add/remove non-consecutive objects.

ALL-button, marks/selects all objects in the list

None-button, unmarks/deselects all objects in the list

Zooming list, the zoom factor scales size of the printout, 1 makes the drawing/text minimal, higher values increase the size. Experiment for a suitable value for your printer, see also [zooming factor in WIN.INI](#)

Print-button, orders printing of the selected information

Cancel-button, returns to editor without any printout

Setup button, allows You to change printer and page orientation

#_filePrint

\$File Print

+ ZREFFP:000

Limitations:

-

Short Cuts:

-

See also:

FILE, PRINT SETUP, FILE, PRINT PARTS OOA GRAPHICS, , FILE, PRINT PARTS TMC GRAPHICS, , FILE, PRINT PARTS FSM GRAPHICS, , FILE, PRINT PARTS UC GRAPHICS, , FILE, PRINT PARTS UC SEQ GRAPHICS, , FILE, PRINT PARTS UC SEQ ALL GRAPHICS, , FILE, PRINT PARTS OOA TEXT, , FILE, PRINT PARTS TMC TEXT, , FILE, PRINT PARTS UC TEXT,

#s. **File, Print Parts, Print OOA Graphics**

Purpose:

To print all or some consecutive pages of the OO-Diagram

Editor:

OO Diagram Editor

Dialogue:

Standard MS-Windows 3.1 Print and Print Setup Dialogues.

Dialogue Items:

Print Range, is used to specify what to print:

All - print all pages

Pages From: nnn To: nnn - print selected range of pages

Print Quality, available for printers supporting resolution control

Print to File, if checked the dialogue will prompt for a file name. This file can be routed to printer later.

OK-button, starts spooling/printing

Cancel Button, cancels the print request

Setup Button, invokes the FILE,PRINTER SETUP dialogue

Limitations:

Zoom factor cannot be modified, the value set in File Print or WIN.INI is used.

Short Cuts:

-

See also:

FILE,PRINT, , FILE, PRINT SETUP, FILE, PRINT PARTS OOA TEXT,

#_filePrOOAgra

\$File, Print Parts, Print OOA Graphics

+ ZREFFP:020

#_File, Print Parts, Print TMC Graphics

Purpose:

To print the Task Management Component diagram

Editor:

TMC Editor

Dialogue:

Standard MS-Windows 3.1 Print and Print Setup Dialogues.

Dialogue Items:

Print Range, is used to specify what to print:

All - print all pages

Pages From: nnn To: nnn - print selected range of pages

Print Quality, available for printers supporting resolution control

Print to File, if checked the dialogue will prompt for a file name. This file can be routed to printer later.

OK-button, starts spooling/printing

Cancel Button, cancels the print request

Setup Button, invokes the FILE,PRINTER SETUP dialogue

Limitations:

Zoom factor cannot be modified, the value set in File Print or WIN.INI is used.

Short Cuts:

-

See also:

FILE,PRINT,, FILE, PRINT SETUP, FILE, PRINT PARTS TMC TEXT,

#_filePrTMCgra

\$File, Print Parts, Print TMC Graphics

+ ZREFFP:030

#_File, Print Parts, Print FSM Graphics

Purpose:

To print the Finite State Machine / Flow chart

Editor:

FSM Editor

Dialogue:

Standard MS-Windows 3.1 Print and Print Setup Dialogues.

Dialogue Items:

Print Range, is used to specify what to print:

All - print all pages

Pages From: nnn To: nnn - print selected range of pages

Print Quality, available for printers supporting resolution control

Print to File, if checked the dialogue will prompt for a file name. This file can be routed to printer later.

OK-button, starts spooling/printing

Cancel Button, cancels the print request

Setup Button, invokes the FILE,PRINTER SETUP dialogue

Limitations:

Zoom factor cannot be modified, the value set in File Print or WIN.INI is used.

Short Cuts:

-

See also:

FILE,PRINT, , FILE, PRINT SETUP

#_filePrFSMgra

\$File, Print Parts, Print FSM Graphics

+ ZREFFP:040

#_File, Print Parts, Print UC Overview Graphics

Purpose:

To print the Use Case Overview chart

Editor:

OO-Diagram Editor, UC Editor

Dialogue:

Standard MS-Windows 3.1 Print and Print Setup Dialogues.

Dialogue Items:

Print Range, is used to specify what to print:

All - print all pages

Pages From: nnn To: nnn - print selected range of pages

Print Quality, available for printers supporting resolution control

Print to File, if checked the dialogue will prompt for a file name. This file can be routed to printer later.

OK-button, starts spooling/printing

Cancel Button, cancels the print request

Setup Button, invokes the FILE,PRINTER SETUP dialogue

Limitations:

Zoom factor cannot be modified, the value set in File Print or WIN.INI is used.

Short Cuts:

-

See also:

FILE,PRINT,, FILE, PRINT SETUP, FILE, PRINT PARTS UC TEXT,

#_filePrUCgra

\$File, Print Parts, Print Use Case Overview Graphics

+ ZREFFP:050

File, Print Parts, Print UC SEQ Graphics

Purpose:

To print the current Use Case Sequence Diagram

Editor:

UC Editor, SEQ Editor

Dialogue:

Standard MS-Windows 3.1 Print and Print Setup Dialogues.

Dialogue Items:

Print Range, is used to specify what to print:

All - print all pages

Pages From: nnn To: nnn - print selected range of pages

Print Quality, available for printers supporting resolution control

Print to File, if checked the dialogue will prompt for a file name. This file can be routed to printer later.

OK-button, starts spooling/printing

Cancel Button, cancels the print request

Setup Button, invokes the FILE,PRINTER SETUP dialogue

Limitations:

Zoom factor cannot be modified, the value set in File Print or WIN.INI is used.

Short Cuts:

-

See also:

FILE,PRINT,, FILE, PRINT SETUP, FILE, PRINT PARTS, All Use Cases SEQ Diagrams

,

#_filePrSEQgra

\$File, Print Parts, Print UC SEQ Graphics

+ ZREFFP:060

#* **File, Print Parts, Print All Use Case SEQ Graphics**

Purpose:

To print the all Use Case Sequence Diagrams

Editor:

OO-Diagram Editor, UC Editor, SEQ Editor

Dialogue:

Standard MS-Windows 3.1 Print and Print Setup Dialogues.

Dialogue Items:

Print Range, is used to specify what to print:

All - print all pages

Pages From: nnn To: nnn - print selected range of pages

Print Quality, available for printers supporting resolution control

Print to File, if checked the dialogue will prompt for a file name. This file can be routed to printer later.

OK-button, starts spooling/printing

Cancel Button, cancels the print request

Setup Button, invokes the FILE,PRINTER SETUP dialogue

Limitations:

Zoom factor cannot be modified, the value set in File Print or WIN.INI is used.

Short Cuts:

-

See also:

FILE,PRINT, , FILE, PRINT SETUP, FILE, PRINT PARTS, Use Cases SEQ Diagrams, ,

#_filePrSEQAllgra

\$File, Print Parts, Print All UC SEQ Graphics

+ ZREFFP:070

#**File, Print Parts, Print OOA Text**

Purpose:

To print **all** Textual descriptions of the OO model.

Editor:

OO Diagram Editor

Dialogue:

Standard MS-Windows 3.1 Print and Print Setup Dialogues.

Dialogue Items:

Print Quality, available for printers supporting resolution control

Print to File, if checked the dialogue will prompt for a file name. This file can be routed to printer later.

OK-button, starts spooling/printing

Cancel Button, cancels the print request

Setup Button, invokes the FILE,PRINTER SETUP dialogue

Limitations:

Zoom factor cannot be modified, the value set in File Print or WIN.INI is used.

Short Cuts:

-

See also:

FILE,PRINT, , FILE, PRINT SETUP, FILE, PRINT PARTS OOA GRAPHICS,

#_filePrOOAtxt

\$File, Print Parts, Print OOA Text

+ ZREFFP:080

#_File, Print Parts, Print TMC Text

Purpose:

To print **all** Textual descriptions of the TMC model.

Editor:

OO Diagram Editor, TMC Editor

Dialogue:

Standard MS-Windows 3.1 Print and Print Setup Dialogues.

Dialogue Items:

Print Quality, available for printers supporting resolution control

Print to File, if checked the dialogue will prompt for a file name. This file can be routed to printer later.

OK-button, starts spooling/printing

Cancel Button, cancels the print request

Setup Button, invokes the FILE,PRINTER SETUP dialogue

Limitations:

Zoom factor cannot be modified, the value set in File Print or WIN.INI is used.

Short Cuts:

-

See also:

FILE,PRINT, , FILE, PRINT SETUP, FILE, PRINT PARTS TMC GRAPHICS,

#_filePrTMCtxt

\$File, Print Parts, Print TMC Text

+ ZREFFP:090

#**File, Print Parts, Print UC Text**

Purpose:

To print **all** Textual descriptions of the Use Case Overview diagram.

Editor:

OO-Diagram Editor, UC Editor

Dialogue:

Standard MS-Windows 3.1 Print and Print Setup Dialogues.

Dialogue Items:

Print Quality, available for printers supporting resolution control

Print to File, if checked the dialogue will prompt for a file name. This file can be routed to printer later.

OK-button, starts spooling/printing

Cancel Button, cancels the print request

Setup Button, invokes the FILE,PRINTER SETUP dialogue

Limitations:

Zoom factor cannot be modified, the value set in File Print or WIN.INI is used.

Short Cuts:

-

See also:

FILE,PRINT, , FILE, PRINT SETUP, FILE, PRINT PARTS Use Case GRAPHICS,

#_filePrUCtxt

\$File, Print Parts, Print UC Text

+ ZREFFP:100

#SK+ **File Printer Setup**

Purpose:

To select a printer and page orientation. The changed values are only applied for this OOther session. To perform a change affecting all application use the Control Panel in Program Manager.

Editor:

OO-Diagram Editor, TMC Editor, FSM Editor

Dialogue

Standard MS-Windows Printer Setup dialogue.

Dialogue Items in:

Limitations:

-

Short Cuts:

-

See also:

-

#S* **File Configure**

Purpose:

To define granularity of the grid, and fonts.

Editor:

OO-Diagram Editor, TMC Editor, FSM Editor, UC Editor, SEQ Editor

Dialogue:

Two dialogues are used, Configure and Fonts.

Dialogue Items in Configure:

Horizontal Grid, selects granularity of the horizontal locations for graphical symbols, see also WIN.INI.

Vertical Grid, selects granularity of the horizontal locations for graphical symbols, see also WIN.INI.

Fixed Object Width check box, makes boxes of equal size (probably no value function, because the used texts strings cannot be long), see also WIN.INI

Screen&Printer Font, **Font Graphics button**, to invoke the standard font set-up dialogue. The selected fonts are used for all texts in graphics, see also WIN.INI.

Printer Font, **Font Descriptions button**, to invoke the standard font set-up dialogue. The selected fonts are used for all texts in descriptions outputs to the printer

OK button, remembers the changes (note font changes are remembered when their OK button is pressed)

Cancel Button, skips changes (note font changes are remembered when their OK button is pressed)

Limitations:

-

Short Cuts:

-

See also:

-

#S* **File Customize Paths...**

Purpose:

To define all the paths used by OOther and defined in WIN.INI. The changes are written to WIN.INI.

Editor:

OO-Diagram Editor, TMC Editor, FSM Editor, UC Editor, SEQ Editor

Dialogue Customize Paths:

Home, the default directory for Your model files

AutoSave, name of the auto save path (created periodically)

VerifyFile, name of the temporary file with verification results

TocFile, name of the temporary file containing list of contents of the model

Headers, name of the directory where You store (C++) header files for the model

TemplateObj, name of the template file for object descriptions or empty string if default is expected, You may also specify the \DEV\NUL device

TemplateAttr, same for the attribute descriptions

TemplateServ, same for the service descriptions

TemplateConn, same for the connection descriptions

TemplateProcessor, same for the processor descriptions

TemplateTask, same for the task/process descriptions

Convert Templates to ANSI, checked if the templates contain national characters in IBM PC char set

Limitations:

-

Short Cuts:

-

See also:

WIN.INI

#_fileCustom

\$File Customize Paths...

+ ZREFFR:010

#SK+ **File Color**

Purpose:

To modify/reset the colours used in the diagrams.

Editor:

OO-Diagram Editor, TMC Editor, UC Editor, SEQ Editor

Dialogue Color:

The contents of the colour dialogue varies for each of the editors. FSM Editor is using colours defined in OO diagram Color set-up.

Dialogue Items in Color:

Graphic Object - is used to specify which object's colours are to be changed

Colors, RGB: there are three slides one for three primary colours, Red, Green, Blue, see also WIN.INI.

OK button, accepts changes

Cancel button, returns to editor without changes

Default button, reload s default settings

Limitations:

FSM, UC and SEQ Editors reuse colors of OOA,TMC Editors

Short Cuts:

-

See also:

WIN.INI

Edit

Menus:

Edit

- Copy All Graphics to Clipboard
- Copy Selected Objects
- Copy Connectors (Internal Clipboard)
- Paste Connectors (Internal Clipboard)
- Delete Connectors
- Search...
- Align to Top
- Align Center
- Delete selected symbols
- Delete ALL symbols

#_edit

\$Edit

+ ZREFGE:000

#SK+ **Edit Copy All Graphics**

Purpose:

To copy the entire diagram to clipboard. The receiving application must be able to understand the ANISOTROPIC MetaFile format of the clipboard.

In the other application you may need to clip and resize the received picture.

E.g. in *MS-Word for Windows 2.0* use SHIFT-LeftButton and drag the corners to clip away any unwanted empty space. Then use LeftButton and drag to resize the picture. While the clipboard buffer is formatted for the screen, the font dimensions in WfW may differ, so it happens that the graphics and texts overlay each other.

I had not been able to figure out how to format the metafile so WfW would allow editing. Thanks anyone who can give me information about convention required by WfW. As a bypass I use to paste the OOT-clipboard to Excel 3.0, and copy it out immediately and then paste to WfW. In so way WfW will allow editing the picture. (magic!)

The Copy All Graphics function is available if you have *not* made any group selection of the graphical symbols by CTRL-Left-Mouse-click.

Editor:

OO-Diagram Editor, TMC Editor, FSM Editor, UC Editor, SEQ Editor

Dialogue:

-

Dialogue Items:

-

Limitations:

-

Short Cuts:

-

See also:

EDIT COPY SELECTED OBJECTS

#SK+ **Edit Copy Selected Objects**

Purpose:

To copy a part of the OO diagram to clipboard. The receiving application must be able to understand the ANISOTROPIC MetaFile format of the clipboard.

The function is available if you have group selection of the graphical symbols by CTRL-Left-Mouse-click.

Editor:

OO Diagram Editor

See also:

Edit Copy All

#_editCopySelected
\$Edit Copy Selected Objects
K clipboard;copy-paste
+ ZREFGE:020

#SK+ **Edit Copy Connectors**

Purpose:

To duplicate connectors from a selected object to another one. The connectors are copied to an internal clipboard for later **EDIT PASTE CONNECTORS**. The object must be first selected by clicking.

Editor:

OO Diagram Editor

See also:

EDIT PASTE CONNECTORS

EDIT DELETE CONNECTORS

#SK+ **Edit Paste Connectors**

Purpose:

To duplicate connectors earlier copied by **EDIT COPY CONNECTORS** to a selected object. The receiving object must be first selected by clicking.

Editor:

OO Diagram Editor

See also:

EDIT COPY CONNECTORS

EDIT DELETE CONNECTORS

#SK + **Edit Delete Connectors**

Purpose:

To remove all connectors from a selected object. The receiving object must be first selected by clicking.

Editor:

OO Diagram Editor

See also:

EDIT COPY CONNECTORS

EDIT PASTE CONNECTORS

#_editDeleteConn
\$Edit Delete Connectors
K delete
+ ZREFGE:050

#SK+ **Edit Search**

Purpose:

To find the object having a specified name or containing attribute or service with such a name.

Editor:

OO Diagram editor

Dialogue Search Name:

Dialogue Items in Search Name:

Name, specify name of the searching

whole name specified check box, if not checked beginning of the name may be specified
case sensitive, if checked it differs between ABC and abc

searching from beginning, if not checked it searches forward via objects in alphabetic order

object name, if checked object names are searched

attribute name, if checked attribute names are searched

service name, if checked service names are searched

OK button starts searching

Cancel button, returns to editor

Limitations:

it's quite simple searching without any regular expressions (wild cards)

Short Cuts:

-

See also:

-

#_editSearch

\$Edit Search

K search;attribute;object;service

+ ZREFGE:060

#SK+ **Edit Align to Top**

Purpose:

To arrange selected FSM symbols, by CTRL-left mouse button, for aligned top coordinates.

The symbols are moved to the highest location occupied by the symbols.

Editor:

FSM Diagram editor

Limitations:

Up to 30 symbols may be moved

Short Cuts:

-

See also:

-

#SK+ **Edit Align Center**

Purpose:

To move the object selected by CTRL-left mouse button so they become centred to the leftmost center position of the objects.

Editor:

FSM Diagram editor

Limitations:

Up to 30 symbols can be handled.

Short Cuts:

-

See also:

-

#SK+ **Edit Delete selected symbols**

Purpose:

To delete symbols selected by CTRL-left mouse button.

Editor:

FSM Diagram editor

Dialogue Are You Sure:

Dialogue Items in Search Name:

Button Yes to perform deletion

Button No to cancel deletion

Limitations:

Up to 30 symbols can be deleted.

Short Cuts:

-

See also:

-

#_editFSMdelSel

\$Edit Delete selected symbols

K delete

+ ZREFGE:090

#SK+ **Edit Delete ALL symbols**

Purpose:

To remove ALL symbols in the current FSM diagram, i.e. for the current service or for the common FSM diagram.

Editor:

FSM Diagram editor

Dialogue Are You Sure:

Dialogue Items in Search Name:

Button Yes to perform deletion

Button No to cancel deletion

Limitations:

-

Short Cuts:

-

See also:

-

#SK+ **View**

Menus:

View

Structure

Attribute

Service

Page Boundaries

Selected zoom factor:

Zoom 2x (50%)

Zoom 3x (33%)

Zoom 4x (25%)

Zoom 5x (20%)

#_view

\$View

K view;structure level view;attribute level view;service level view;filtering

+ ZREFHV:000

#SK- **View Structure**

Purpose:

To control presentation of the object structures on the screen. If checked all whole-part and gen-spec associations are presented.

Editor:

OO Diagram Editor

See also:

Views

#_viewStruct
\$View Structure
K view;structure level view;filtering
+ ZREFHV:010

#SK+ **View Attributes**

Purpose:

To control presentation of the attribute information on the screen. If checked all instance-instance associations and attribute names are presented.

Editor:

OO Diagram Editor

See also:

Views

#_viewAttr
\$View Attributes
K view;attribute level view;filtering
+ ZREFHV:020

#SK- **View Services**

Purpose:

To control presentation of the service information on the screen. If checked all message associations and service names are presented.

Editor:

OO Diagram Editor

See also:

Views

#_viewServ
\$View Services
K view;service level view;filtering
+ ZREFHV:030

#SK+ **View Page Boundaries**

Purpose:

To inform the user where the page breaks occur.

Editor:

OO Diagram Editor, TMC Editor, FSM Editor, UC Editor, SEQ Editor.

Limitations:

It might be some smaller differences in what OOther believes and what the printer does.

See also:

Views

#SK+ **View Zoom Factors**

Purpose:

To specify zoom factors for editing in the zoom mode. The zooming is performed by invoking the menu ZOOM or pressing the right mouse button.

Editor:

OO Diagram Editor, TMC Editor, FSM Editor, UC Editor, SEQ Editor.

See also:

ZOOM and WIN.INI

#_viewZoom
\$View Zoom Factors
K zoom
+ ZREFHV:050

\$NewSymbol

Menus:

NewSymb

Object... (in OOA Editor)
Import/Copy Object...
Title...
Processor...
Task...
State...
Receive...
Send...
Synch send+receive...
Action...
Condition / Selection...
Iteration (Start of Iteration)...
Next (End of Iteration) / Label...
Text comment... (in FSM Editor)
Use Case... (in UC Editor)
Actor... (in UC Editor)
Text comment... (in UC Editor)
Object... (in SEQ Editor)
Actor(s) / SystemBorder(s)... (in SEQ Editor)
Text comment... (in SEQ Editor)

_newSymb
\$NewSymbol
+ ZREFNS:000

#SK+ **NewSymb Object**

Purpose:

To create an object symbol.

Editor:

OO Diagram Editor

Dialogue Object:

The dialogue is used to define the name and characteristics of the object. Also a short textual description may be added. The dialogue is also used for browsing and updating existing objects.

Dialogue Items in Object:

Name - specify name of the object

Type - select type of object or object-&-class (Coad/Yourdon definition)

Description - a short textual description, the template may be a default one, or user specified, see WIN.INI

OK button - accepts changes

Cancel button - skips changes

Delete button - removes the object

Limitations:

-

Short Cuts:

Double clicking on an existing objects identification part invokes the Object dialogue

See also:

#_newSymbObject
\$NewSymbol Object
K object;description;class
+ ZREFNS:010

#SK+ **NewSymb Import/Copy Object**

Purpose:

To fetch an object from another model file. Import just refers to that file, Copy performs a physical copying of that object.

Editor:

OO Diagram Editor

Dialogue Import/Copy Object:

The user selects an object contained in an existing file for copying or import.

Dialogue Items:

Import/Copy from file - specify file name

Directory - specify directory

Contained objects - list of objects, click for one of objects in the list

Import button - perform copying by reference

Copy button - perform physical copy

Cancel button returns to editor without doing anything

Limitations:

-

Short Cuts:

-

See also:

-

#SK+ **NewSymb Title**

Purpose:

To specify/update a short title for the model.

Editor:

OO Diagram Editor

Dialogue Title:

Dialogue Items in Title:

Enter the title string - place for your string

OK button - accepts the string

Cancel button - returns to editor without changing anything

Limitations:

The string is short and always located on the same place

Short Cuts:

-

See also:

-

#SK+ **NewSymb Processor**

Purpose:

To define a processor in the TMC domain.

Editor:

TMC Editor

Dialogue Processor:

The user enters/modifiers characteristics of the processor symbol.

Dialogue Items in Processor:

Name - specify/modify name of the processor

Number of Incarnations - specify a string or number

Description - a short description. A default template or user specified templates may be used, see WIN.INI

OK button - accepts changes

Delete button - deletes the symbol

Limitations:

-

Short Cuts:

The dialogue can be invoked by double clicking on the processor symbol

See also:

-

#_newSymbProcessor
\$NewSymbol Processor
K processor
+ ZREFNS:040

#SK+ **NewSymb Task**

Purpose:

To define a task in the TMC domain.

Editor:

TMC Editor

Dialogue Task (Process):

The user enters/modifies characteristics of the task symbol.

Dialogue Items:

Name - Name - specify/modify name of the task

Number of Incarnations - specify a string or number

Description - a short description. A default template or user specified templates may be used, see WIN.INI

OK button - accepts changes

Delete button - deletes the symbol

Cancel Button - cancels changes, returns to the editor

Limitations:

-

Short Cuts:

The dialogue can be invoked by double clicking on the task symbol

#SK+ **NewSymb State**

Purpose:

To create a state symbol in a FSM diagram

Editor:

FSM Editor

Dialogue FSM Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#SK+ **NewSymb Receive**

Purpose:

To create a reception of message (or entering a called service) symbol in a FSM diagram

Editor:

FSM Editor

Dialogue FSM Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newSymbReceive
\$NewSymbol Receive
K receive;message;service
+ ZREFNS:060

#SK+ **NewSymb Send**

Purpose:

To create a send message symbol (or invoking a service) in a FSM diagram

Editor:

FSM Editor

Dialogue FSM Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newSymbSend
\$NewSymbol Send
K send;message;service
+ ZREFNS:070

#SK+ **NewSymb Synch send+receive (Ask)**

Purpose:

To create a synchronous send+receive message exchange symbol in a FSM diagram

Editor:

FSM Editor

Dialogue FSM Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newSymbAsk

\$NewSymbol Synch send+receive (Ask)

K receive;send;ask;synch;synchronous;service

+ ZREFNS:080

#SK+ **NewSymb Action**

Purpose:

To create an action/processing symbol in a FSM diagram

Editor:

FSM Editor

Dialogue FSM Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#SK+ **NewSymb Condition/Selection**

Purpose:

To create a selection symbol in a FSM diagram

Editor:

FSM Editor

Dialogue FSM Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newSymbCondition

\$NewSymbol Condition/Selection

K selection;condition

+ ZREFNS:100

#SK+ **NewSymb Iteration**

Purpose:

To create an iteration symbol in a FSM diagram

Editor:

FSM Editor

Dialogue FSM Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newSymbIteration

\$NewSymbol Iteration

K iteration;loop;repetition;label;while;until

+ ZREFNS:110

#SK+ **NewSymb Next/Label**

Purpose:

To create a next iteration or label symbol in a FSM diagram

The Next/Label symbol may supply up to 20 connectors to other symbols.

Editor:

FSM Editor

Dialogue FSM Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newSymbNext

\$NewSymbol Next/Label

K iteration;loop;repetition;label;next

+ ZREFNS:120

#SK+ **NewSymb Text**

Purpose:

To create a text symbol in a FSM diagram

Editor:

FSM Editor

Dialogue FSM Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#SK+ **NewSymb Use Case**

Purpose:

To create a text symbol in a UC diagram

Editor:

UC Editor

Dialogue UC Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newUseCase

\$NewSymb Use Case

K use case

+ ZREFNS:140

#SK+ **NewSymb Actor**

Purpose:

To create a text symbol in a UC diagram

Editor:

UC Editor

Dialogue UC Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#SK+ **NewSymb Text (in UC Diagram)**

Purpose:

To create a text symbol in a UC diagram

Editor:

UC Editor

Dialogue UC Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#SK+ Nsymb Object (in SEQ Diagram)

Purpose:

To create a text symbol in a SEQ diagram

Editor:

SEQ Editor

Dialogue UC Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newSEQobject

\$Nsymb Object (in SEQ Diagram)

K object

+ ZREFNS:170

#SK+ **NewSymb Actor (in SEQ Diagram)**

Purpose:

To create a text symbol in a SEQ diagram

Editor:

SEQ Editor

Dialogue UC Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newSEQactor

\$NewSymb Actor (in SEQ Diagram)

K actor

+ ZREFNS:180

#SK+ Nsymb Text (in SEQ Diagram)

Purpose:

To create a text symbol in a SEQ diagram

Editor:

SEQ Editor

Dialogue UC Symbol Name:

The dialogue is used to create/modify/delete the symbol.

Dialogue Items:

Enter the name string

OK button

DELETE button

Limitations:

-

Short Cuts:

The dialogue may be invoked by double clicking on the symbol.

See also:

-

#_newSEQtext

\$Nsymb Text (in SEQ Diagram)

K text;comment;string

+ ZREFNS:190

#_Utilities

Menus:

Utilities

- Task/Process Components Chart
- Finite State Machine Flow Chart
- Use Case Chart(s)
- Show Inheritance
- Statistics
- Move Object to Left
- Move Object to Right
- Delete Service
- Delete Object
- Verify Use Case
- Generate C++ Header File for this class
- Generate ALL Header Files
- Show C++ header file for this object
- List of Contents

#_utilities
\$Utilities
+ ZREFUT:000

SK+ Utilities TMC Editor

Purpose:

To switch to the TMC Editor. When selecting this menu the Task Management Component (TMC) Editor will be invoked. All objects defined in the OO-Diagram will be present in the TMC diagram's leftmost column.

Use NewSymb Processor and Task to introduce corresponding symbols, and then select a symbol by clicking and use the shift-left mouse button to connect elements to show the desired allocation structure.

Editor:

TMC can only be invoked from OO Diagram Editor.

See also:

OO-Diagram to see how to return from the FSM and TMC Editors to the OO Diagram Editor..

#SK+ Utilities FSM Editor

Purpose:

To switch to the FSM Editor. When selecting this menu the Finite State Machine (FSM) Editor will be invoked. Use NewSymb State and other NewSymb menus to create a FSM and then select a symbol by clicking and use the shift-left mouse button to connect elements to show the desired allocation structure.

Editor:

FSM can only be invoked from OO Diagram Editor.

See also:

#SK+ Utilities UC Editor

Purpose:

To switch to the Use Case editor.

Editor:

UC can only be invoked from the OO Diagram Editor.

See also:

#_utilUCeditor
\$Utilities UC Editor
K editors;UC editor
+ ZREFUT:030

Utilities Show Inheritance

Purpose:

The menu is used to present which attributes and services are available for a selected object/class directly or via the inheritance mechanism.

Editor:

OO Diagram Editor

Dialogue:

The result is presented in a menu box

Dialogue Items:

Press OK-button to return to the editor.

Limitations:

Too deep inheritance is detected by the tool, and thus the listing may be not complete. In the same way circular inheritance is detected.

Short Cuts:

shift + right mouse button

See also:

-

Utilities Statistics

Purpose:

The menu invokes presentation of number of objects and symbols in all OOther editors.
Also the size of the OO-Diagram is calculated.

Editor:

OO Diagram Editor

Dialogue:

The result is presented in a menu box

Dialogue Items:

Press OK-button to return to the editor.

Limitations:

-

Short Cuts:

-

See also:

-

#S-Utilities Verify Use Case(s)

Purpose:

The menu starts verification of consistency of the Use case diagram. If invoked from the UC Editor ALL use cases are verified. If invoked from the SEQ Editor only the current use case is checked..

The following verifications are performed:

1. if the use case co-operates wit actors they should be present in the sequence diagram
2. there should be at least one message to/from the existing actor(s)
3. objects sending messages should have services
4. objects sending messages should do it from a service box in the SEQ diagram
5. there should be a message connection between two objects who send messages to each other
6. the message connection should have right direction

If invoked under the Use Case editor, the utility will give you coverage figures showing percentage of objects and services covered by the existing use cases. The goal is to get 100% for both - it means that all classes are used at least once, and so the services are invoked at least once each. If lower there are some classes and methods that are not presented how they co-operate by the outside world.

The use cases are also good description of typical test cases for integration testing!!!

Editor:

Use Case and Sequence Diagram Editors.

Dialogue:

The result is presented in a file which is browsed by means of the MS-Windows Notepad application.

Dialogue Items:

-

Limitations:

-

Short Cuts:

-

See also:

-

#S Utilities Generate C++ Header File for this class and Generate ALL Header Files

Purpose:

It is a working demo of generation of header files. See [C++ Header Generation](#) for details. If you tag description texts properly the headers will be ready for compilation. If the utility lack any information the generated header will contain descriptive texts like "<type> function (<Arg....>);" These texts can be used to localise which tags are not defined or wrongly defined in the description texts.

The menu Generate C++ Header File for this class may be invoked when an object is selected in the OOA diagram (click on it). The generated headers are presented by means of the NotePad as file tmp.h. If you like, you can save the file under proper name.

The menu Generate ALL header files will generate all files for classes which have header file ,names defined as tags #@@+{Header File}#@@ in their description texts.

Editor:

OOA Editor

Dialogue:

-

Dialogue Items:

-

Limitations:

-

Short Cuts:

-

See also:

-

#_Utilities Show C++ Header File for this class

Purpose:

This function can be used in conjunction to the header generation function. See [C++ Header Generation](#) for details.

If you tag description texts properly, so the name of the header file is specified OOther will help You to easily pick up this file when You work with the OOA model.

The headers are presented by means of the NotePad.

The utility looks only under the directory containing Your header files, see [File, Customize](#).

Editor:

OOA Editor

Dialogue:

-

Dialogue Items:

-

Limitations:

-

Short Cuts:

-

See also:

-

#_utilShowHdr

\$Utilities Show C++ header file for this object

+ ZREFUT:080

Utilities List of Contents

Purpose:

By means of this function You can get an overall information what is present in Your model. E.g. which FSM diagram are not developed, which objects participate in use cases etc. It is also handy if you want to have a window beside OOther to see names of object, attributes, services etc.

Editor:

All editors.

Dialogue:

-

Dialogue Items:

-

Limitations:

-

Short Cuts:

-

See also:

-

Utilities Move Object to Left/Right

Purpose:

To move the selected object or actor one step to left/right in the SEQ diagram.

Editor:

SEQ Editor

Dialogue:

-

Dialogue Items:

-

Limitations:

-

Short Cuts:

-

See also:

-

#S-Utilities Delete Service

Purpose:

To delete the selected service+message from the SEQ diagram.

The function is implicitly performed in SEQ diagram if a service is deleted in the OOA diagram.

Editor:

SEQ Editor

Dialogue:

-

Dialogue Items:

-

Limitations:

-

Short Cuts:

-

See also:

-

#S-Utilities Delete Object

Purpose:

To delete the selected object and the messages/services visible in the diagram from the SEQ diagram.

The function is implicitly performed in SEQ diagram if a object is deleted in the OOA diagram.

Editor:

SEQ Editor

Dialogue:

-

Dialogue Items:

Press OK-button to return to the editor.

Limitations:

-

Short Cuts:

-

See also:

-

#SK- **OO-Diagram**

Purpose:

The menu is used to return to the OO Diagram editor.

Editor:

TMC Editor, FSM Editor and UC Editor

See also:

-

#_utilOodiagram

\$OO-Diagram

K editors;OOA editor;FSM Editor;TMC Editor;UC Editor

+ ZREFUT:120

#SK+ **UC-Diagram**

Purpose:

The menu is used to return to the UC Diagram Editor.

Editor:

SEQ Editor.

See also:

-

_utilUCdiagram

\$UC-Diagram

K editors;SEQ editor;UC Editor

+ ZREFUT:130

#SK+ **Zoom**

Purpose:

The Zoom Menu is used to switch on and off the zoom function. The zooming factor is set in View Zoom factor

Editor:

OO Diagram Editor, TMC Editor, FSM Editor, UC Editor, SEQ Editor

Short Cuts:

Right mouse button has an identical function.

See also:

View Zoom factor and mouse conventions.

_zoom

\$Zoom

K zoom

+ ZREFZA:000

#S-About

Purpose:

To inform about release and origin of the OOther program.

#S-**Help**

Purpose:

Manual pages and description of the tool.

Purpose:

The implemented function has a status of a working demo.
I.e. You can use it for fully working design if You accept some restrictions.
If you add the following tags into the description text of the OOA model, OOther will be able to generate simple C++ headers ready for compilation. In other cases you will need to edit these files by hand.
The generated header files contain all the information OOther found in the descriptions plus inheritance declarations.
You may insert space, tab and new line characters between the control strings #@@+{, keywords, values and }#@@. If the keyword is built up by more than one word there must be exactly one space character between the keywords.
If you has used the tags: *New* or *Init Value* the generated constructor will be a very simple one, just initialising the attributes. So if you don't want a so simple constructor do not enter these tags in descriptions of the class and its attributes.
In future the tags, both these described here and other uder defined, will be accessible between parts of the model by means of simple hypertext links.

Tags:

#@@+{Header File xxx }#@@

The tag is to be located in description of the object. xxx is name of the file.

Example: #@@+{Header File myobject.h }#@@.

The file name xxx is entered into a comment on the fist line and also used to create an #ifndef construction to avoid multiple defining of identifiers. The first occurrence will be used.

#@@+{Specifier xxx }#@@

The tag is used to define specifier "static" or "virtual". The tag is to be located inside the attribute and service descriptions.

Example: #@@+{Specifier static }#@@

#@@+{Type xxx }#@@

The tag is used to define types of attributes and return value type for functions (services).

The tag is to be located inside the attribute and service descriptions. The first occurrence will be used.

*Example: #@@+{Type *char }#@@*

#@@+{Call Arguments xxx }#@@

The tag defines format of arguments passed to a service. The tag may be present one or many times in the service description. Every occurrence of the tag will result in a function definition.

#_headers

\$C++ Header Generation

+ ZREFZA:200

K Header Generation; Generation of C++ headers; C++ Header Generation

Please observe that it is exact one space character between the two words Call Arguments!

Example: `###+{Call Arguments char *p, int &val, String *str }###`.

`###+{Qualifier xxx }###`

The tag is used to define qualifiers like "const" or the pure virtual "=0". Also to specify arrays, "[90]".

The tag is to be located inside the attribute and service descriptions. The first occurrence will be used.

Example: `###+{Qualifier const}###`, `###+{Qualifier [100]}###`

`###+{Init Value xxx }###`

The tag is used to define a value of attribute to be assigned when creating it. It can be constant or a variable passed from the constructor.

The tag is to be located inside the attribute descriptions. The first occurrence will be used.

Please observe that it is exact one space character between the two words Init Value!

Example: `###+{Init Value IobjectType}###` or `###+{Init Value NULL}###`

`###+{New yyy xxx }###`

The tag specifies how to create an inherited (=Gen-Spec) or contained (=Whole-Part) class. yyy is name of the class, xxx is a list of arguments to be passed. There is exactly one space between New and the class name. The first occurrence of the class name initiator will be used.

Example: `###+{New object Istatus, NULL, 801 }###`.

`###+{Container yyy xxx }###`

The tag specifies how to define the contained (= expressed by the Whole-Part association) class. yyy. yyy is name of the class, xxx is the declaration string. There is exactly one space between Container and the class name yyy. The first occurrence of the class name initiator will be used.

Example: `###+{Container objectus SortedArray }###`.

`###+{Friend xxx }###`

The tag specifies friend classes or services. The tags may be located inside object and service descriptions. Multiple occurrence is allowed.

Example: `###+{Friend int *upperName(char *s) }###`
`###+{Friend class FooZoo }###`.

#SK+ Parameters in WIN.INI

(This chapter is quite long. You may find it easier to print it - but think of the trees!.)

The OO Documentation Tool reads the windows parameter file WIN.INI (normally installed in the directory C:\WINDOWS) when the tool is started, and sets various options according to the settings in the [OoaToolFree] section.

In addition, the tool reads the following common flags:

[Intl]

iDate=2 (0 for MDY, 1 for DMY, 2 for YMD)

sDate=- (separator character, normally "-" or "/")

When you exit the OO Documentation Tool, you will be asked whether to update WIN.INI with the current settings (even if you have changed nothing). However, if you exit abnormally (i.e. you switch off your system before exiting, or your machine crashes), WIN.INI will retain the old settings.

The following tags (which are *not* case sensitive) are read. Default values are shown after the equality sign. Unspecified parameters will be set to these defaults.

All parameters except **Home** are verified when starting the OO Documentation Tool. If something fails, you will get an error message. (If you spell any parameter tags incorrectly, the tool will assume that the default value is to be used.)

[OoaToolFree]

AutoSave=C:\AUTOSAVE.OOT

Home=C:

VerifyFile=C:\OOTVERIF.TXT

PrinterZoom=3

GridX=5

GridY=5

FixedObjBoxWidth=0

TemplateObj=

TemplateAttr=

TemplateServ=

TemplateConn=

TemplateProcessor=

TemplateTask=

TemplateOemToAnsi=0

ColorObjBox=R000,G000,B255

ColorObjInstance=R255,G000,B000

ColorConnGenSpec=R050,G150,B050

ColorConnWholePart=R050,G055,B150

ColorConnInstance=R000,G000,B000

ColorConnMessage=R150,G150,B150

ColorConnTMCselected=R255,G000,B000

ColorConnTMCnormal=R050,G150,B050

ColorBoxTMCprocessor=R000,G000,B000

ColorBoxTMCtask=R250,G000,B000

ColorBoxTMCobj=R000,G000,B255

ZoomFactor=3

_winini

\$Parameters in WIN.INI

K WIN.INI;parameter;configuring the tool;autosave;template;date;international date

+ ZREFZZ:000

[OoaToolFree]

This tag identifies the section of the WIN.INI file accessed by the OO Documentation Tool. It must be enclosed within square brackets.

#Autosave=C:\path\file.oot (anypath + filename)

Here you can specify the location of your auto save file. If you are running on network use a private directory to avoid collisions.

#Home=C:\ (any path)

Here you can specify the default location of your files. Dialogues for opening and saving files use this directory as the start position. Maximum 80 characters. **NB The path must end with a back-slash.**

#PrinterZoom=3 (range: 1 to 10)

This parameter adjusts the size of your printouts. It specifies the number of printer pixels used for each screen pixel, and allows you to efficiently zoom the size of printouts (Unique feature!?).

Recommended values for 9-pin printers (e.g. Epson FX-85): 2, most laser printers: 3-5, InkJets (Brother HJ-100/Canon BJ-10e) (360DPI): 3.

This parameter may also be modified via the menu option File, Print

#GridX=5 (range: 1 to 100)

GridY=5 (range: 1 to 100)

GridX and GridY (X is horizontal, Y vertical) helps you to align object boxes in a grid of equidistant points. Using a value of 5 creates a grid meeting at every 5th screen pixel. This parameter can also be modified via the menu option File Configure.

#FixedObjBoxWidth=0 (value: 0 or 1)

Value 0 (= false) indicates that the width of object boxes automatically adjusts to the size of text strings inside the object box. Value 1 (= true) makes all box sizes equal, but the text may not fit. This parameter can also be modified via the menu option File Configure.

#KTemplateObj=path

TemplateAttr=path

TemplateServ=path

TemplateConn=path

TemplateProcessor=path

TemplateTask=path

If you specify a path and filename, the corresponding file will be read into the template used for the Descriptions of objects, attributes, services and connectors etc. The path parameter gives the location of an ASCII text file containing a maximum of 255 bytes of text (including newlines).

(1) You should specify a complete path in the format <drive>:\dir\dir...\filename or give the path relative to the **Home** parameter.

(2) The maximum path length is 80 characters.

(3) By specifying the path \dev\nul you will get an empty template!

(4) If the path is unspecified (or the entire parameter line is missing), a default, hard-coded, template is used.

#_winini_autosave

#_winini_home

#_winini_printerzoom

#_winini_grid

#_winini_fixedobjectwidth

#_winini_template

K template;description;user specified template

(5) These path names can only be specified by editing WIN.INI.

#K TemplateOemToAnsi=0 (value: 0 or 1)

Value 0 (= false) indicates that no conversion of the template files is to be performed. If 1 (= true) the files are assumed to be written using the IBM PC character set (OEM) and thus need a translation to ANSI character set.

#K ColorObjBox=R000,G000,B255

ColorObjInstance=R255,G000,B000

ColorConnGenSpec=R050,G150,B050

ColorConnWholePart=R050,G055,B150

ColorConnInstance=R000,G000,B000

ColorConnMessage=R150,G150,B150

ColorConnTMCselected=R255,G000,B000

ColorConnTMCnormal=R050,G150,B050

ColorBoxTMCprocessor=R000,G000,B000

ColorBoxTMCtask=R250,G000,B000

ColorBoxTMCobj=R000,G000,B255

The colours of graphic objects can be set by their Red/Green/Blue ratios. Each colour component must be prefixed with 'R', 'G' or 'B' (as above). Allowed values are 0-255. Black is generated by the combination R000,G000,B000; white by R255,G255,B255, Red by R255,G000,B000 etc.

#ZoomFactor=3

The parameter specifies the factor by which the diagram is to be compressed when you press the right mouse button for a zoomed image.

#FontGraph="Arial",13,0

FontDescr="Lucida Calligraphy",19,2

The selected fonts are stored as name, point size and graphic rendition (0...3: bold*2+italic*1)

#_winini_color

K color;RGB;configure

#_winini_oem

K Template;character set;OEM;ANSI

#_winini_zoom

#_winini_font

#SK+ File Format

(This chapter is quite long. You may find it easier to print it - but think of the trees!.)
You may read the file created by *File Save/Save As* into any editor, and extract the desired DESC statements.

See also Character Set, description of the OOA-part of the file, TMC part of the file, FSM part of the file, and UC and SEQ part of the file

Format:

The file is formatted as follows:

<title statement>

<OOA object model>

<TMC model>

<FSM model>

<UC+SEQ model>

Example model and the resulting file contents: A nice chart!

```
TITL A nice chart!  
OBJ: 0 330 20 Object A  
DESC 0
```

```
OBJ: 1 420 210 Zorro!  
DESC 0
```

```
OBJ: 1 60 270 obj1  
DESC 61  
+description of object 1, ends with <newline>  
+after <newline>
```

```
ATTR attr1  
DESC 20  
+description of attr1
```

```
SERV serv1  
DESC 16  
+<no description>
```

```
#_fileFormat  
$File Format  
K file format;ASCII  
+ ZREFZZ:100
```

SERV serv2
DESC 26
+some description for serv2

OBJ: 0 230 210 obj2
DESC 6
+Hello!

OBJ: 0 60 20 obj3
DESC 0

CONN m 1 3 0 0
DESC 17
+message transfer!

CONN g 3 2 0 0
DESC 16
+<no description>

CONN w 4 3 0 0
DESC 16
+<no description>

CONN i 0 4 0 0
DESC 20
+instance to instance

#S* Character Set

OOther uses ANSI character set (default within windows), i.e. the national characters may get different appearance when displayed by DOS editors.

The template files supplied by the user may be translated from IBM PC (OEM) set to ANSI when read in, see win.ini flag TemplateOemToAnsi

#_ootCharSet
\$File Format Character Set
+ ZREFZZ:200

#> OOA Section

OOA Objects are sorted in ascending *name* order. After an object there may be optional attributes, also sorted in name order. Then comes the object's services, followed by the next object etc. Finally all the connectors are listed. Note that objects, attributes, services and connectors are followed by descriptions (DESC 0 means an empty description, '+' marks a new description line.) A description line terminates when the user entered CTRL+ENTER or when all description text ends.

```
TITL titleString
OBJ: otype x y name
DESC len
+ text ...
+ text ...
<cr-nl>
SERV name
DESC len
+ ...
<cr-nl>
ATTR name
DESC len
+ ...
<cr-nl>
OBJ: otype x y name
DESC len
+ text ...
+ text ...
<cr-nl>
...
...
OBJI otype x y filename name
...
...
CONN ctype from to
DESC len
+ text ...
+ text ...
<cr-nl>
CONN ctype from to multipl1 multipl2
DESC len
+ text ...
+ text ...
<cr-nl>
CON2 ctype from to "multipl1" "multipl2"
DESC len
+ text ...
+ text ...
```

```
#_ootFileOOA
$File Format OOA Section
+ ZREFZZ:300
```


<cr-nl>

...

...

(Note in release 1.04 and later, CON2 is used to store object-object connectors, CONN can be read by 1.04 but not written).

where

titleString	arbitrary ASCII string terminated by a new line
otype	0 (class-&-object) 1 (class)
x	pixel coordinate for left edge of the object (0 is leftmost position)
y	pixel coordinate for the top edge of the object (0 is top position)
name	name of the object/attribute/service (ends with new line)
filename	name of file containing the imported object
len	length of the description text (used to allocate memory by the tool) should not exceed 5000 bytes
ctype	connector type, g (gen-spec), w (whole-part) i (instance) m (message)
from	number of the 'from' object (0, ...) in the file (note: sorted!). The 'from' object is always the whole, generic, or the object that was selected first when assigning the association.
to	number of the 'to' object (part, specific, ...)
multipl1,multipl2	multiplicity for whole-part and instance associations. The first for the 'from' object the second for the 'to' object (0=not used, 1="0,m", 2="1,m", 3="1", 4="m")
"multipl1","multipl2"	multiplicity for whole-part and instance associations. Both are specified as strings within quotes

#S-TMC Section

The TMC Section describes allocation of objects to processes/tasks and tasks to processors. Processor and task names are unique.

```
PRCR "processor" "number"
DESC len
+ text ...
+ text ...
<cr-nl>
...
TASK "task" started terminated scheduled "number"
DESC len
+ text ...
+ text ...
<cr-nl>
...
CONP "processor" "task"
...
CONT "task" "object"
```

TMC data are stored by means of PRCR, TASK and connector directives CONP and CONT

"process"	name of process
"task"	name of task
"number"	number of processes or tasks
started	how started, =0 at application start, =1 on demand, =2 by timer, =3 other
terminated	how terminated, =0 at application termination, =1 on demand, =2 by timer, =3 other
scheduled	how scheduled, =0 event, =1 by timer, =2 other

```
#_ootFileTMC
$File Format TMC Section
+ ZREFZZ:400
```

#S- FSM Section

The records do not have any order.

```
STAT "text" x y prev "cond"  
RECV "text" x y prev "cond"  
SEND "text" x y prev "cond"  
ASKM "text" x y prev "cond"  
ACTN "text" x y prev "cond"  
COND "text" x y prev "cond"  
ITER "text" x y prev "cond"  
NEXT "text" x y prev "cond"      (NEXT not written by OOther 1.06,  
only read)  
NEX2 "text" x y numprev "cond" prev1 prev2 .... prevN    (NEX2 written  
by 1.06)  
TEXT "text" x y prev ""
```

TMC data are stored by means of PRCR, TASK and connector directives CONP and CONT

"text"	string in the symbol (in quotes)
x	x coordinate of the symbol
y	x coordinate of the symbol
prev	index to previous symbol, 0,..., (-1 if not connected)
"cond"	condition string, for symbols connected to an COND symbol (i.e. prev is index to a COND symbol); (in quotes)
numprev	number of arguments prev1...prevN (0 if none)
prev1...prevN	index to previous symbol, 0,..., (N = value of numprev)

#5- UC and SEQ Section

The UC and SEQ section is structured in such a way that all the symbols of a SEQ diagram are stored after the corresponding UCAS+DESC directive. DESC is as usually a textual description. + as first character prolongs the text until a empty line is found. At the end there are all UC connectors.

UACT is identifier of an actor in UC diagram, UCAS is use case, UTXT is text. UZUU use case - use case connection, UZUA use case to actor connection and UZAA actor to actor connection.

SQOB is object or actor, SQSE is service+message, SQSR is a acknowledge message to the same service who sent the message starting the current service (I hope you figure out who is who) , SQTX is text record.

UACT "actorName" x y

DESC

+ ...

UCAS "useCaseName" x y

DESC

+ ...

SQOB objOOAnum

SQOB -1

SQSE "message" yy ysz servOOAnum sendUCobj

...

SQSE "message" yy ysz -1 sendUCobj

SQSR "message" xof yof (optional response message)

...

SQSR "message" xof yof

SQTX "text" xx yy

DESC

+ ...

UTXT "text" x y

DESC

+ ...

UZUU "name" UCnum1 UCnum2

DESC

+ ...

UZUA "name" UCnum1 actorNum1

DESC

+ ...

UZAA "name" actorNum1 actorNum2

#_ootFileUC

\$File Format UC and SEQ Section

+ ZREFZZ:600

DESC
+ ...

where

"actorName" name of the actor in UC diagram, max 32 chars

"useCaseName" name of the use case in the UC diagram, max 32 chars

x and **y** are coordinates in the UC diagram

xx and **yy** are coordinates in the SEQ diagram

xof and **yof** are offsets to locations where the response messages leave and enter services.

objOOAnum is number/index of the object in the OOA diagram, se above, or **-1** if SQOB is corresponding to an actor

"message" is name of the message in the SEQ diagram, max 32 chars

ysz is length of the service rectangle in the SEQ diagram

servOOAnum is number/index of the service in the OOA object, se above, or **-1** if the message is sent to an actor (the previous SQOB)

sendUCobj is index of the SQOB in this use case section

"text" is a short text (max 32 chars)

Limitations & Known Bugs

#_bugs
\$Limitations & Known Bugs
+ ZREFZ:900

#SK+ Postscript Printers

Flags in win.ini. Font handling has been rewritten in 1.05, so I do not think that the following flags are needed any more, however just in case... they are still documented here:

msWin31Bug01=0 (value: 0 or 1)

msWin31Bug02=0 (value: 0 or 1)

msWin31BugWarning=1 (value: 0 or 1)

The first release of MS Windows 3.1 contains a bug in the calculation of text metrics (call GetTextMetrics) when the viewport is extended (zooming) which manifests in incorrect vertical spacing of printouts. The OO Documentation Tool generates a warning message if you use this version (GetVersion() returns 0x0a03). If you set the WIN.INI flag **msWin31Bug01=1** the tool will use a by-pass. The second flag **msWin31Bug02** should be switched on (=1) when using postscript printer drivers - again the MS-Win drivers behave strange when scaling fonts and generating the gen-spec symbol.

I.e. for Postscript try to use:

msWin31Bug01=0

msWin31Bug02=1

Other printers may work correctly with:

msWin31Bug01=1

msWin31Bug02=1

or with:

msWin31Bug01=1

msWin31Bug02=0

If your win3.1 version works correctly, you may switch off the warning message by setting **msWin31BugWarning=0**. *NB the tool will not give a warning if the hex windows version is not 0x0a03 (I hope Microsoft fixes it before the next maintenance release - yes, I have reported the bug)*

#_postscript
\$PostScript Printers
K postscript;printer
+ ZREFZZ:910

Analysis - it is an inexact definition:

A systematic investigation of what is purpose of the "system", who is using it and why.

Analysis also have to create a systematic presentation/expression about the facts/knowledge, requirements, feasibility of the ideas, etc. To be unambiguous analysis need to identify some kind of entities allowing to break down the complex area into smaller pieces.

Object Oriented - it is an inexact definition:

An attempt to describe a system as a collection of independent entities (objects) communicating with each other via messages, not knowing about each other's internal structure. Each object have a well-defined information contents and behave deterministic. The objects can be classified in such a way that one can identify common properties being inherited from a more generic occurrence of the entity. One also makes a distinction between a typical object idea (class) and its real occurrence (instance).

Design - A (systematic I hope?) way to translate results of Analysis into an executable system.

_defDesign

Inheritance - A controlled way to contain properties present in a more generic kind of object classes.

#_defInheritance

Domain - an area for analysis, well defined with respect to its boundaries

#_defDomain

Object - an entity with deterministic information contents and behaviour. Class is a typical expression, instance is an occurrence of the object of that class.

#_defObject

Attribute - informational contents entity in an object.

#_defAttribute

Service, Method - an service available for outside world, operation on the internal information of the object. A service may invoke via messages services of other objects if needed.

#_defService

Message - a request directed to an object to perform a service. It can also contain some information transfer from the sending to the receiving object.

_defMessage

Association - expression of dependencies between objects. Association express how objects "know" about each other, if they inherit each other or are contained into each other, or if they in any way communicate.

Processor - a computer executing a number of tasks/processes.

#_defProcessor

Task/Process - a executing program in a processor.

#_defTask

CASE - Computer Aided System Engineering. Traditionally divided into Upper-CASE (Analysis), Lower-CASE (Programming), and Middle-CASE, i.e. somewhere between.

#_defCASE

State Machine - A technique to express behaviour of deterministic automata. The central ideas are states and messages/events. The machine knows for every state what to do when an event occurs, and which is the next state.

#_defStateMachine

Use Case - A technique to describe interwork of objects in typical scenarios. It is today, I think, the only reliable technique to understand the dynamics of co-operation of a set of objects.

Actor - An operator/user or external system co-operating with a use case, e.g. by sending stimuli (messages) initiating the use cases and receiving messages directed to the outer world.

#_defActor