WinWin
Reference Manual

A System
for
Collaboration and Negotiation
of
Requirements
This manual is compatible with WinWin release 1.0.

Copyright Notice

This document is copyrighted, and all rights are reserved by Ellis Horowitz (EH). This document may not in whole, or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent.

Copyright © 1995 Ellis Horowitz

All rights reserved. Send all comments to horowitz@usc.edu

Warranty

This manual is provided "as is" without warranty of any kind, either express or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. Moreover, EH reserves the right to revise this manual and to make changes periodically without obligation to notify any person or organization of such revision or changes.

Trademark Acknowledgement

EH has made every effort to supply trademark information about company names, products, and services mentioned in this document. Trademarks indicated below were derived from various sources.

UNIX is a registered trademark of AT&T Bell Laboratories

Motif is a trademark of Open Software Foundation, Inc.

FrameMaker and Frame Technology are registered trademarks of Frame Technology Corporation.

PostScript is a registered trademark of Adobe Systems Incorporated.

Apple, Macintosh, and LaserWriter are registered trademarks of Apple Computer, Inc.

Netscape is a registered trademark of Netscape Communications Inc.

IBM is a registered trademark of International Business Machines Corporation.

Sun Microsystems and Sun Workstation are registered trademarks, and OpenWindows, Sun-4, and SPARClstation are trademarks, of Sun Microsystems, Inc.

X Window System is a trademark of the Massachusetts Institute of Technology.

Printed in the United States of America.

This research is sponsored by the Advanced Projects Research Agency (ARPA) through Rome Laboratory under contract F30602-94-C-0195, and by the Affiliates of the USC Center for Software Engineering. The current Affiliates are

Aerospace Corp.,
Air Force Cost Analysis Agency,
AT&T Bell Laboratories,
Bellcore,
DISA,
E-Systems,
Electronic Data Systems,
Hughes Aircraft Company,
Institute for Defense Analysis,
Interactive Development Environments,
Jet Propulsion Laboratory,
Litton Data Systems,
Lockheed Martin Corp,
Loral,
Motorola,
Northrop Grumman Corp.,
Rational Inc.,
Rockwell International,
Science Applications International Corp.,
Software Engineering Institute,
Software Productivity Consortium,
Sun Microsystems,
Teledyne Inc.,
TRW,
U.S. Air Force Rome Laboratory,
U.S. Army Research Laboratory,
Xerox
# TABLE OF CONTENTS

WinWin Reference Manual 5

1.1 What is WinWin 5
1.2 Getting Started 5
1.3 A Short Tutorial 6
1.4 Understanding Artifacts 19
   1.4.1 Explanation of the fields 22
   1.4.2 Status button 23
   1.4.3 Priority button 23
   1.4.4 Artifact Relationships 23
   1.4.5 RefersTo Window 25
   1.4.6 Referencedby Window 26
   1.4.7 Taxonomy Window 29
   1.4.8 Vote Window 30
   1.4.9 Comments Window 31
   1.4.10 Attachments Window 31
   1.4.11 Artifact Set Window 33
   1.4.12 Artifact Action Buttons 33
1.5 WinWin Files and Environment Variables 33
1.6 Database Server 34
1.7 Main Menu 35
1.8 Project Menu 35
   1.8.1 Project New 36
   1.8.2 Project Open 37
   1.8.3 Project Users 37
   1.8.4 Project Update 38
   1.8.5 Project Export 38
   1.8.6 Project Options 41
   1.8.7 Project Delete 41
   1.8.8 Project Facts 41
   1.8.9 Project Exit 42
1.9 Taxonomy Menu 43
1.10 Artifacts Menu 45
   1.10.1 Artifact Copy 47
1.11 Tools Menu 47
1.12 Messages Menu 48
1.13 Analysis of the WinWin Process Model 50
1.14 Acknowledgments 52
1.1 What is WinWin

WinWin is a computer program that aids in the capture, negotiation, and coordination of requirements for a large system. It assumes that a group of people, called stakeholders, have signed on with the express purpose of discussing and refining the requirements of their proposed system. The system can be of any type. WinWin contains facilities for:

- capturing the desires (win conditions) of the stakeholders
- organizing the terminology so that stakeholders are using the same terms in the same way
- expressing disagreements or issues needing resolution
- negotiating agreements which resolve the issues
- using third party tools to enlighten or resolve issues
- producing a requirements document that summarizes the current state of the proposed system
- creating documents that support multimedia and hyperlinks
- tracing the ways by which requirements decisions were reached
- checking the completeness and consistency of requirements

1.2 Getting Started

WinWin offers a group of users a great deal of functionality, but as a result some planning is useful before getting started. In this section we offer a simple scenario for how users of WinWin might begin their work.

1. An owner of the project is identified. He identifies the other people who will participate in this negotiation.
2. The owner starts WinWin, creates a new project, and enters the names of all users. These people are called stakeholders.
3. One stakeholder is designated to define or tailor an existing set of terms for the proposed system. He enters them in WinWin.
4. One stakeholder is designated to define or tailor an existing taxonomy for the proposed system. He enters them in WinWin.
5. Stakeholders review the terms and taxonomy. They then enter Win Conditions, expressing their preferences for the project. They associate their Win Conditions with elements of the taxonomy.

6. Stakeholders may discover an issue which needs to be resolved. They enter the Issue, request stakeholders to contribute options, and various analyses may be done. Eventually an option is adopted which resolves the Issue.

7. A stakeholder drafts an Agreement, which may cover one or more Win Conditions and/or resolve one or more Issues. Stakeholders vote on the Agreement, which if ratified becomes closed.

8. Steps 5, 6, and 7 continue until there are no more uncovered Win Conditions and no unresolved Issues.

9. The stakeholders may transform the project data into either Framemaker or Mosaic for convenient viewing and distribution to others.

1.3 A Short Tutorial

After successfully installing WinWin, you should run through this short tutorial. It will both familiarize you with the features of WinWin while at the same time permit you to check out that most things are working properly.

In order to test out WinWin's collaborative capabilities, you need to simulate three users. Therefore, sit down at your UNIX workstation and in an xterm window you will start three copies of WinWin.

- setenv USER user
  start the first version of winwin
- setenv USER customer
  start winwin
- setenv USER developer
  start winwin

Notice that each copy of WinWin shows the name of the user at the top of the window. Position the windows across the top of your screen.

- Move your mouse across the top level commands while holding down the left mouse button. You will see that only the Project New, Project Open, and Project Exit commands are enabled.

- once again, working in an xterm, choose a directory in which to place the example.

- mkdir winwintest
  cd winwintest
  mkdir user
  mkdir customer
A Short Tutorial

```
mkdir developer
```

For this exercise we are going to develop and negotiate the requirements for a project that will develop a software engineering environment (or SEE) for a Satellite Ground Station product line. There are three stakeholders: user, customer and developer.

- as user, click on Project New and in the dialog box that appears enter the following:

  **Project Name:** SatGroundStation
  **Data Directory:** /usr/home/user/winwintest/user
  **Password:** test
  **Role:** user
  **Title:** Commander
  **Position:** Director
  **Organization:** US Space Systems and Operations

You are the first stakeholder. All Winwin commands should now be enabled is your copy of WinWin.

- click on Project Users and add the following two stakeholders to the project.

  **Password:** test
  **Name:** customer
  **Directory:** /usr/home/user/winwintest/customer
  **Role:** customer
  **Title:** Specialist Engineer
  **Position:** Programmer
  **Organization:** CSE Software CASE Division

  and now add another stakeholder:

  **Password:** test
  **Name:** developer
  **Directory:** /usr/home/user/winwintest/developer
  **Role:** developer
  **Title:** Senior Scientist
  **Position:** Manager
  **Organization:** California Satellite Corp.

You should now see the Project Users box looking like:
The two stakeholders that have been added to the project list can now connect to the project.

- In customer click on Project Open, select SatGroundStation and click on OK. You will get a warning message that your *.windb file does not exist, but there is no problem. Click on OK.

- Now do the same for developer.

Each stakeholder can now create any of the artifacts used by WinWin.

- For user click on Artifacts Terms. The List window that appears is empty.

- Click on the New button and a blank artifact form appears. Enter the following:
  Name: Schedule
  Body: The time duration of the SEE software development process

- Click on Apply.
The tire duration of the SEE software development process

This is the way all artifacts are created. Notice that an artifact ID has been assigned, and placed in the Terms List window as well as in the artifact ID field.

For now click on Cancel for the term and the Term List Window.

Stakeholders need to create a base-line taxonomy. This is an outline that reflects the organization of the project. Each line of the taxonomy is called a taxonomy node. The developer initially enters the taxonomy. Proposed changes are communicated to the developer. Alternate scenarios for creating the taxonomy are possible.

In FIGURE 1-3 you see a picture of a sample taxonomy for this project.

As User, click on the taxonomy button, and click on the Edit Enable button. The taxonomy commands are now enabled.

Click on Add Node, then enter "Schedule" and click on Edit.

Click on Add Node, then enter "Cost" and click on Edit.

Click on Add Node, then enter "Operational Cost", click on Edit and click on the right arrow. Now enter the other entries in the taxonomy and click on OK when done. When
you click on OK, two windows appear labeled Update Notify. These are messages from the user to the customer and developer, telling them that the user database has been updated and asking if they want a current copy.

Click on OK and once again ignore the warning messages.
Let's create two win conditions for stakeholder User.

Click on Artifacts, Win Conditions and then on the New button.

Below are the Name and Body fields for each win condition you are to create. After entering each Name and Body field click on Apply.

Name: SGS domain data reduction and reporting (DR&R)

Body: Condition: we are content to have SGS-domain specific data reduction and reporting for simulation data analysis. Rationale: this will reduce the SGS software development cost and schedule

click on Apply and then click on Cancel

Name: SGS workstation usage scenario generation (USAGE)

Body: Condition: we will accept SGS-domain specific workstation usage scenario generation. Rationale: there will be reduced SGS software development cost and schedule

click on Apply and then click on Cancel

Whenever data is entered or changed, a stakeholder must update the project, creating a database of artifacts that everyone can see, comment on, and reference. This is done by using the Project Update command. All other stakeholders currently logged in will be notified that the database has been updated.

- as user, click on Project Update

Since you are simultaneously simulating two other stakeholders, you should get two notify messages that the database has been updated.

- click on OK and update your local database.

Customer then enters the following win condition,

Name: Reduced operational costs via domain specific simulation, test, data reduction functions

Body: I believe that costs will be reduced if we do the simulation, test, and data reduction for our specific domain

Click on Apply

Customer examines the body of user-WINC-1, user-WINC-2, and since they are related to his customer-WINC-1, customer creates "RelatesTo" links from customer-WINC-1 to user-WINC-1 and user-WINC-2.

- in customer-WINC-1 move Options button to RefersTo item.
Enter user-WINC-1 and if the relation button shows "RelatesTo"

click on Apply

Do the same for user-WINC-2.

- In the Win Condition List window click on user-WINC-1 and cause it to appear.

Select the Comments Option

Enter the following comment:

"Please look at customer-WINC-1 because it calls for building domain specific versions of this functionality in the hope of reducing costs. There may be other win conditions which are affected by this strategy."

- click on Apply to record the comment

- Customer clicks on Project Update. Once again the notify messages appear. click on OK.

- the User and developer will get the Project Update message.

For the developer, click on the Messages button. A window appears that contains a set of one line messages. Each message records an action that was taken by one of the stakeholders. The messages are displayed in chronological order. Notice that the messages show that: the user created TERM-1, created WINC-1 and WINC-2, that the customer created and modified WINC-1 and that the user modified WINC-1.

- developer clicks on Artifacts Agreements and creates a new Agreement. A Name and RefersTo field MUST first be provided.

Name: SGS-domain data reduction and reporting (DR&R)

Refers-To user-WINC-1, covers and customer-WINC-1, covers

Click on Apply

Body: We agree that domain specific data reduction and reporting is the correct strategy

click on Apply

- developer selects the Vote window within the Agreement and enters the Voting Policy

Please vote this week

click on Enter Policy and Click Here

- Developer clicks on Project Update

Customer and User get the update notification and click on OK
Customer and user see in the Message Window that in developer-AGRE-1 a vote is IN_PROGRESS.

User and Customer examine developer-AGRE-1 and both vote on the Agreement. They concur and update the project.

Developer gets the update notification and sees that User and Customer have voted on developer-AGRE-1 in the message window.

Developer checks developer-AGRE-1 and sees that User and Customer have both concurred on this agreement.

Developer votes on the agreement and also concurs. Realizing the vote is completed and it has passed, he changes the State of developer-AGRE-1 to be "Passed".

Developer should click on Apply and Project Update.

At this point, all data covered by the passed agreement is frozen. Passed agreements represent requirements for which there is consensus among the stakeholders.

---

**FIGURE 1-4**

A Passed Agreement

<table>
<thead>
<tr>
<th>Agreement</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>developer-AGRE-1</td>
</tr>
<tr>
<td>CREATION DATE</td>
<td>09/23/95 15:29</td>
</tr>
<tr>
<td>REVISION DATE</td>
<td>09/23/95 15:30</td>
</tr>
<tr>
<td>ROLE</td>
<td>developer</td>
</tr>
<tr>
<td>STATUS</td>
<td>Active</td>
</tr>
<tr>
<td>PRIORITY</td>
<td>Medium</td>
</tr>
<tr>
<td>STATE</td>
<td>Passed</td>
</tr>
<tr>
<td>Name</td>
<td>SGS domain data reduction and reporti</td>
</tr>
<tr>
<td>Options</td>
<td>Vote</td>
</tr>
<tr>
<td>Policy</td>
<td>Please vote this week</td>
</tr>
<tr>
<td>Enter Policy and Click Here</td>
<td></td>
</tr>
<tr>
<td>Vote</td>
<td>Concur</td>
</tr>
<tr>
<td>voter</td>
<td>Concur</td>
</tr>
<tr>
<td>customer</td>
<td>Concur</td>
</tr>
<tr>
<td>user</td>
<td>Concur</td>
</tr>
<tr>
<td>State</td>
<td>Passed</td>
</tr>
<tr>
<td>Apply</td>
<td>Delete</td>
</tr>
</tbody>
</table>
Scenario 2 (Create an issue to cite conflicts)

Following Scenario 1, User has the following win conditions:

ID: Name:
user-WINC-1 SGS domain data reduction and reporting (DR&R)
user-WINC-2 SGS workstation usage scenario generation (US-AGE)

Customer has the following win condition:
customer-WINC-1: Reduced operational costs via simulation, test, data reduction functions

Customer now enters the following new win conditions:
Name: Ready for full use in 25 months
click on Apply and Cancel
Customer defines another Win Condition
Name: Development cost less than $7M
click on Apply and Cancel

- customer clicks on Project Update and the user and developer get the usual notify messages and click on OK.

Developer sees in the messages window that the customer has created two new win conditions. Developer examines them and then creates an issue

Developer defines an issue:
Name: Scope
Refersto user-WINC-1 and user-WINC-2 using "involves" relation. customer-WINC-1, customer-WINC-2, and customer-WINC-3 with the relation "involves'.

Click on Apply

Body: The functionalities required by the user in these win conditions: DR&R and USAGE cannot be all implemented within budget and schedule.

Click on Apply

To clarify why the cost and schedule cannot be met, the developer invokes the cocomo tool from the Tools Menu. He enters the data as shown in FIGURE 1-5. There are two modules, DR&R and USAGE. EDSI for DR&R is 67000 and for USAGE it is 72000. Both modules have three EAF factors that need to be set to HIGH. These factors are: RELY, CPLX, and SCED. Labor rate is $5,100 and development mode Embedded. Save the file under the name developer.est.
Developer causes the artifact developer-ISSU-1 to appear. He goes to the attachments window and enters:

```
cocomo, developer.est
```

He clicks on Apply and then on Project Update.

Customer and User see in the message window that developer-ISSU-1 is cited. Based upon this conflict, User, Developer and Customer each create possible solutions which address the issue, called options.

User creates a new option

```
Name: Priority - prefer data reduction
RefersTo: developer-ISSU-1, addresses
Click on Apply
```
Body: I am content to have a reduced set of usage scenarios while retaining the full data reduction and reporting features

Click on Apply

Developer creates a new option

Name: More money and time

RefersTo: developer-ISSU-1, addresses

Click on Apply

Body: We will need more money and more time to fulfill all of the requirements

Click on Apply

Customer creates a new option

Name: Top-level programmer

RefersTo: developer-ISSU-1, addresses

Click on Apply

Body: We will recover 1/2 of the extra cost and 2/3 of the extra schedule if we reassign some top developer people to this project. However, we may have to compensate the developer for the negative impact on other projects

Click on Apply

User, developer and customer update the project. All click OK in notify windows.

Developer comments on customer-OPTN-1: 'Top-level programmer requires higher salary: implying more money'

Customer comments on developer-OPTN-1: 'The budget is set; no more money can be acquired; Let's consider user's option user-OPTN-1'

Customer comments on user-OPTN-1: 'This option seems reasonable; Please let us know your priority and draft it in an agreement'.

Customer and Developer click on Project Update.

User reads the comments on his option user-OPTN-1 and creates a new model in cocomo, one that has the source code for USAGE reduced to 59,000 lines from the original 72,000. This is shown in FIGURE 1-6. He calls it model-test.est. User drafts an agreement where he specifically states that data reduction will come first and the USAGE scenario will be reduced.
Name: Scope
RefersTo: user-OPTN-1, adopts

Body: My new estimates for the sizes of the modules to build are: USAGE: 59K EDI (Effective delivered source instructions) DR&R is unchanged. The cocomo estimation shows that cost and time is within budget and schedule is in the attachments.

Attachment: cocomo, model-test.est

FIGURE 1-6
COCOMO Model Worksheet

![COCOMO Model Worksheet]

Note that model-test.est shows that the estimated cost is $6.5M and the schedule is 24.65 months when the code size of USAGE is brought down to 59,000.

User enters the voting policy: Please vote by next week.
User votes to concur, by clicking on "Concur" and Apply.

User updates the project.

Customer and Developer get the update notification and see that user-AGRE-1 has a vote IN_PROGRESS.

Customer and Developer examine user-AGRE-1 and vote to "Concur" and update.

User checks user-AGRE-1 and sees both Customer and Developer concurred on this agreement.

User changes the state of user-AGRE-1 to "Passed".

User updates the project.

Now that agreement has been reached, we might want to export all of the data into a word processor and create a final product.
- click on Project Export
- select either text, HTML or Mif and click on OK.
  the selected program will appear with all of the text of the project.

This concludes the tutorial.
1.4 Understanding Artifacts

WinWin supports the idea of an artifact. An artifact is a composite data item that is owned by a user. The fields of an artifact are fixed. Some fields in an artifact have their value set by WinWin. Other fields have their value set by the owner. Still other fields are collections of data set by the other stakeholders in a project. There are five predefined artifacts in WinWin: Win Condition, Issue, Option, Agreement, and Term. Currently there is no way to add new artifact types. The predefined artifacts are explained below:

<table>
<thead>
<tr>
<th>Predefined Artifact</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win Condition</td>
<td>contains in its body an opinion of the owner about some aspect of the project that they deem as desirable. A Win Condition may be associated with one or more nodes in the taxonomy, which helps to structure a stakeholder’s view of the win conditions.</td>
</tr>
<tr>
<td>Issue</td>
<td>contains in its body an identification of a conflict, risk or uncertainty involving one or more win conditions. An issue is unresolved, until it is resolved. This occurs by voting on a set of options that are generated as possible solutions to the issue.</td>
</tr>
<tr>
<td>Option</td>
<td>contains in its body a possible solution to an issue.</td>
</tr>
<tr>
<td>Agreement</td>
<td>contains in its body a statement about the project that is being proposed as something the stakeholders will agree to. A vote is usually expected.</td>
</tr>
</tbody>
</table>

The four artifacts above are the main ones used during the negotiation process. In addition there are two supporting artifacts:

<table>
<thead>
<tr>
<th>Predefined Artifact</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term</td>
<td>contains in its body the definition of a term used in the project. WinWin will permit the entry of terms with the same name. Duplicate terms will immediately be seen when the list of terms is displayed. Project members should determine which term meaning they prefer to use.</td>
</tr>
</tbody>
</table>

FIGURE 1-7 shows a sample Win Condition artifact. Field names are placed in the upper left hand corner of the field. Several fields are not visible in this picture, but they can be seen by clicking on the button that currently displays the word “body”. This button includes a set of options: Body, RefersTo, ReferencedBy, Attachments, Taxonomy, Vote, Comments, Artifact Set. Each option brings up a separate window within the artifact.
FIGURE 1-7  A Sample Artifact Window, for a Win Condition

Win Condition

ID  horowitz-WINC-9
CREATION DATE  '02/01/94 00:00
REVISION DATE  '07/23/94 00:00
ROLE  customer
STATUS  Active
PRIORITY  Very High
STATE  Uncovered

Name
Multimission SEE

Options=>  Body

Condition: SEE Support of multiple concurrent missions: extensions to general tools, simulation and test tools, usage scenario generators, and data reduction tools
Rationale: Result of negotiation with congress
Concerns: Likely budget and schedule conflicts, synchronization with revised SGS schedule

Apply  Delete  Cancel
TABLE 1 shows each of the fields of an artifact, who generates the value in the field, when the value is generated, and when the data is locked. The importance of these facts will become apparent after we discuss negotiation.

**Table 1: Artifact Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Generated by</th>
<th>When Data is Entered</th>
<th>When Data is Locked</th>
</tr>
</thead>
<tbody>
<tr>
<td>UniqueID</td>
<td>WinWin</td>
<td>at creation time</td>
<td>upon creation</td>
</tr>
<tr>
<td>CreationDate:Time</td>
<td>WinWin</td>
<td>at creation time</td>
<td>upon creation</td>
</tr>
<tr>
<td>RevisionDate:Time</td>
<td>WinWin</td>
<td>at Artifact APPLY time</td>
<td>upon Artifact APPLY</td>
</tr>
<tr>
<td>Role</td>
<td>WinWin</td>
<td>at creation time</td>
<td>upon creation</td>
</tr>
<tr>
<td>Name</td>
<td>owner</td>
<td>at Artifact APPLY time</td>
<td>upon Project Update</td>
</tr>
<tr>
<td>Body</td>
<td>owner</td>
<td>at Artifact APPLY time</td>
<td>upon Project Update</td>
</tr>
<tr>
<td>Status</td>
<td>owner</td>
<td>at Artifact APPLY time</td>
<td>never</td>
</tr>
<tr>
<td>Priority</td>
<td>owner</td>
<td>at Artifact APPLY time</td>
<td>never</td>
</tr>
<tr>
<td>State</td>
<td>WinWin</td>
<td>at creation time</td>
<td>Vote In-Progress</td>
</tr>
<tr>
<td>RefersTo</td>
<td>owner</td>
<td>at Artifact APPLY time</td>
<td>never</td>
</tr>
<tr>
<td>ReferencedBy</td>
<td>WinWin</td>
<td>at Artifact APPLY time</td>
<td>never</td>
</tr>
<tr>
<td>Taxonomy</td>
<td>owner</td>
<td>at Artifact APPLY time</td>
<td>Vote In-Progress</td>
</tr>
<tr>
<td>Attachments</td>
<td>owner</td>
<td>at Artifact APPLY time</td>
<td>never</td>
</tr>
<tr>
<td>Vote</td>
<td>all users</td>
<td>at Artifact APPLY time</td>
<td>never</td>
</tr>
<tr>
<td>Comments</td>
<td>all users</td>
<td>at Artifact APPLY time</td>
<td>never</td>
</tr>
<tr>
<td>Artifact Set</td>
<td>WinWin</td>
<td>at Artifact APPLY time</td>
<td>Vote In-Progress</td>
</tr>
</tbody>
</table>
1.4.1 Explanation of the fields

**UniqueID** has the form `<username>-<type>-<number>`. Right now `<type>`={TERM,OUTL,WINC,ISSU, AGRE,OPTN} which stands for term, outline, win condition, issue, agreement and option. `<num>` is a positive integer. To create a new artifact, click on the Artifact New button, and an artifact template appears. Once you click on the APPLY button, the artifact is created and a unique ID is assigned.

**CreationDate:**Time is a string containing the month/day/year hr:min that the associated artifact was created.

**Revision Date:**Time is a string containing the month/day/year hr:min that the associated artifact had a field that was altered.

**Name** is a string that describes the contents of the artifact.

**Body** is a string, that may consist of multiple paragraphs that describes a particular opinion of the owner.

**Status** is either ACTIVE or INACTIVE

**Priority** is either VERY LOW, LOW, MEDIUM, HIGH, or VERY HIGH.

**State** is a text field set by WinWin which varies for different artifact types. A Win Condition is either Uncovered or Covered. An Issue is either Unresolved or Resolved, an Option is either Unadopted or Adopted, an Agreement is either blank, InProgress, Passed or Failed. We discuss the setting of an artifact’s STATE in the section on negotiation.

**RefersTo** is a list of pairs. Each pair is a (id, relation). The set of predefined relations is described later. The list of ids represent artifacts that this artifact points to.

**ReferencedBy** is a list of pairs. Each pair contains (id, relation). The set of ids represent artifacts that point to this artifact. This field is updated by WinWin whenever a WinWin stakeholder points one of his artifacts to another.

**Taxonomy** is a list of references to the Taxonomy. The list of indices represent taxonomy nodes that this artifact points to.

**Attachments** is a list of pairs. Each pair consists of a tool name and a file name associated with that tool.

**Vote** is a policy statement plus a list of pairs. Each pair contains (user_name,user_vote) where user_vote is either Concur, Don’t Concur, or Abstain. If a user selects Concur for any artifact, he is indicating his agreement with the name and body fields of the artifact. If a WinWin user does not vote, there will be no entry by his name. If the user wishes to explain his vote, he may use the Comments field.

**Comments** is a list of three items. Each triple contains (user_name,creationtime,comment). A stakeholder at a given time provides a comment about a given artifact. The Comments button on the artifact produces a window containing all comments that have been entered on this artifact, ordered chronologically. Comments are entered via this window.

**Artifact Set** only applies to Win Conditions and Agreements. To understand the concept, consider that an Agreement may point to a Win Condition or to an Option. An Option may point to an Issue, and an Issue may point to a Win condition. The Artifact Set of an Agreement is all of the artifacts that can be reached from the
Agreement by a set of links starting at the Agreement. Similarly the Artifact Set of a Win Condition is the set of artifacts reachable to the Win Condition.

1.4.2 Status button

There are only two status options: ACTIVE and INACTIVE. Upon creation of an artifact, the status is set to ACTIVE. Once an artifact is made INACTIVE, it cannot be made ACTIVE again. However, INACTIVE artifacts may be re-created using the COPY command. Normally, only ACTIVE artifacts are displayed in List windows. By using the Project Options command one can turn on/off the display of ACTIVE and INACTIVE artifacts.

1.4.3 Priority button

The options are: VERY LOW, LOW, MEDIUM, HIGH, and VERY HIGH. The default setting when an artifact is created is MEDIUM. Whenever the user changes the priority of an artifact, a message is sent to all users informing them of the change.

1.4.4 Artifact Relationships

The purpose of relationships in the WinWin system is to group artifacts together in meaningful ways and to guide the negotiation process. A relationship is a link between a pair of artifacts. The link contains a source artifact, a destination artifact, and a name. The relationships in WinWin are predefined. They are: relates to, replaces, adopts, involves, addresses, covers, involved in, adopted by, addressed by, and covered by. Relationships are named so that users can more easily determine when a relationship should be established. All relationships in WinWin are bidirectional. Whenever a relationship, say R, is established, there is an inverse relationship, R\(^{-1}\), that is also established at the same time. The relations and their inverses are shown in pairs (R, R\(^{-1}\)): (relates to, relates to), (replaces, replaced by), (involves, involved in), (addresses, addressed by), (adopted by, adopted by), (covers, covered by).

There are six artifact types that can be used as the source or destination of a relationship. These are: Terms (TERM), Win Conditions (WINC), Issues (ISSU), Agreements (AGRE), Options (OPTN), and Taxonomy Nodes (OUTL). However, not all of these artifacts can participate as elements of every relation. The restrictions are explained below.

The relation named _relates to_ can be established between any two artifacts of any types. The _relates to_ relationship is available whenever the WinWin user wants to tie together two artifacts, but there is no existing relationship which applies. The inverse relationship of _relates to_ is itself. For example the definition of the term “Real-time system” may contain references to the terms “concurrency” and “scheduling”. The WinWin user could point the term Real-time System to the terms containing “concurrency” and “scheduling” by the _relates to_ relationship.

The relation named _replaces_ can be established between any two artifacts of the _same type_ only by the owner of the artifact. This relation designates the situation when a new artifact is meant to replace another one. For example, suppose an Agreement is created and a vote among all stakeholders is started. If the vote fails the Agreement is marked as FAILED. Perhaps at a later time a new agreement is drafted which is meant to replace the earlier agreement. In this case Agreement2 _replaces_ Agreement1 and the _replaces_ relationship is used to denote this situation. The inverse relationship to _replaces_ is _replaced by_.

Winwin REFERENCE MANUAL

23
To understand the remaining relationships, one must understand the way negotiation is assumed to take place in WinWin. Here is a simplified scenario. A user enters a Win Condition. Either the Win Condition appears to be satisfactory for all stakeholders, or else a conflict (issue or concern) appears to exist. In the former case, an Agreement is drafted and a covers link from the Agreement to the Win Condition is established. A vote is initiated which eventually passes or fails. This concludes one cycle of negotiation. In the case where a conflict exists, an Issue is drafted and an involves link from the Issue to the Win Condition is established. Examining the Issue, one or more Options may be generated. Each option has an addresses link from it to the Issue it purports to resolve. Eventually an Option is chosen. An Agreement will then be drafted which adopts the chosen Option and covers the original Win Condition. This process is graphically shown in FIGURE 1-8.

FIGURE 1-8  Simplified View of Negotiation

Table 2: Summary of Artifact Relationships

<table>
<thead>
<tr>
<th>Relation Name</th>
<th>Relation Inverse</th>
<th>Source - Destination</th>
<th>Relation Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>relatesTo</td>
<td>relatesTo</td>
<td>Any Artifact - Any Artifact</td>
<td>Two artifacts with an arbitrary coupling</td>
</tr>
<tr>
<td>replaces</td>
<td>replacedBy</td>
<td>two artifacts of same type</td>
<td>Artifact X replaces Artifact Y</td>
</tr>
<tr>
<td>involves</td>
<td>involvedIn</td>
<td>Issue - Win condition</td>
<td>An Issue which involves a WinC</td>
</tr>
<tr>
<td>addresses</td>
<td>addressedBy</td>
<td>Option - Issue</td>
<td>An Option which purports to solve an Issue</td>
</tr>
<tr>
<td>adopts</td>
<td>adoptedBy</td>
<td>Agreement - Option</td>
<td>An Agreement makes use of an Option</td>
</tr>
<tr>
<td>covers</td>
<td>coveredBy</td>
<td>Agreement - Win Condition</td>
<td>An Agreement whose resolution defines the requirement</td>
</tr>
</tbody>
</table>
1.4.5 RefersTo Window

The RefersTo portion of an artifact is the mechanism by which relationships are established. Clicking on the RefersTo button produces a window with two parts. The upper part shows all artifacts that are pointed to by the selected artifact. The bottom part is a mechanism for establishing new relationships.

FIGURE 1-9 shows the RefersTo window that appears for the Issue artifact horowitz-OPTN-4. In FIGURE 1-9 this issue is seen to refer to two other artifacts. In particular, horowitz-OPTN-4 addresses horowitz-ISSU-1. It also relates to horowitz-OPTN-3. Any of the artifacts that are referred to can be made to appear if they are double clicked.

To establish a new relationship, the user must enter the id of another artifact in the ID: field, and he must select the relation type. In the case of an OPTN, the legal relation types are: relatesTo, replaces, and addresses. The system is designed so that no illegal relationships can be established.
Under what circumstances can a relationship be deleted? Let us explore the implications. A relationship is established when the Apply button is clicked. But only the user's internal buffer will reflect the relationship. To delete this relationship, highlight it and click on Delete. When the user exits WinWin, or invokes Project Update, these links are exported to everyone else. People may examine them and enter new Win Conditions, Issues and Agreements that refer to this artifact. So we require a stakeholder to either "get-it-right the first time" or enter a new artifact that replaces the original one.

Table 5 summarizes the artifacts that an artifact can refer to.

Table 3: What an Artifact can Refer to (Other than replaces or relates to)

<table>
<thead>
<tr>
<th>Artifact</th>
<th>What it can refer to</th>
<th>Via relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win Condition</td>
<td>---------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Issue</td>
<td>Win Condition</td>
<td>involves</td>
</tr>
<tr>
<td>Option</td>
<td>Issue</td>
<td>addresses</td>
</tr>
<tr>
<td>Agreement</td>
<td>Option</td>
<td>adopts</td>
</tr>
<tr>
<td>Agreement</td>
<td>Win condition</td>
<td>covers</td>
</tr>
</tbody>
</table>

1.4.6 Referencedby Window

The ReferencedBy window has a similar appearance as the top area of the RefersTo window, but this time the selected artifact is the destination and the ReferencedBy window contains all artifacts which point to it. In the example in FIGURE 1-10 we see that bose-WINC-6 is referenced by horowitz-AGRE-3 and horowitz-WINC-11.

In the Referencedby window there is no entry that is created by the user. All entries in the ReferencedBy field are generated by WinWin.

Implementation issues: when a user first opens a project, the entire database is examined and all ReferencedBy fields are filled in. When a RefersTo link is established by the WinWin user, WinWin will immediately set the corresponding ReferencedBy field in the WinWin user's buffer. This maintains database consistency throughout the user's session. However, settings of the ReferencedBy field are not stored when a Project Update is done, because in general the person establishing the link may be pointing to an artifact he does not own, and each copy of WinWin may only alter those artifacts owned by the user running WinWin. Computing of the ReferencedBy field complicates the time to load a project, but so far the time does not appear to be noticeable.
Understanding Artifacts

FIGURE 5.10
Sample Referenced By Window

Win Condition

ID
bose-WNC-6
CREATION DATE
10/1/93 00:00
REVISION DATE
06/25/95 14:44
ROLE
User
STATUS
Active
PRIORITY
High
STATE
Uncovered

Name
Domain-specific-Tool-Extensions-1
Options: Referenced By

horowitz-AGRE-3, CoveredBy,
horowitz-WINC-10, RelatesTo

Apply Delive Cancel
Initially two Win Conditions give rise to an Issue which is resolved by an Option and an Agreement is drafted.

A new Win Condition is entered which causes the Agreement to become invalid.

Suppose we have two win conditions involved in an issue. The issue has one option which is adopted by Agreement1. Agreement1 is voted on, passes and in turn covers win condition 1 and 2. Now suppose a new win condition is entered which causes Agreement1 to become invalid. What are the actions that should result?

1. the owner of Agreement1 changes its status to INACTIVE. This causes Option1 to be unadopted, Issue1 to be unresolved, and Win conditions 1 and 2 to be uncovered.
2. A new issue is drafted, Issue2, which involves Win Conditions 1, 2, and 3. Issue2 replaces Issue1.
3. Options to resolve Issue2 are generated and Option2 is chosen to create an agreement, Agreement2. Agreement2 replaces Agreement1.
4. Agreement2 initiates a vote which eventually passes. This causes Option2 to be adopted, Issue2 resolved, and Win Conditions 1, 2, and 3 to be covered.
1.4.7 Taxonomy Window

When this option is selected a window within the artifact appears as shown in FIGURE 1-12. The top part of the window shows nodes of the taxonomy that this artifact points to. The bottom area provides a mechanism for inserting new taxonomy pointers for this artifact. To enter a new taxonomy pointer, enter the number attached to the taxonomy node in the Taxonomy Index field. Then click on Apply. The number and its associated name should appear in the top area. To delete a pointer to the taxonomy, highlight the name and number and then click on the Delete button. If you click on the Cancel button before clicking on APPLY, then you will lose the changes you have made. To save the changes to your database and to communicate it to others, click on Project Update. The importance of assigning taxonomy pointers to an artifact is that it helps to organize the artifacts into logical groupings. When you are examining the complete taxonomy, you may ask to see just those artifacts that point to a particular node. See SECTION 1.9 for details.

FIGURE 1-12 Taxonomy Window

[Image of Taxonomy Window]

Name: Multimission SEE

Options: Taxonomy

1.2.1 Satellite Coverage
1.3 Support Satellites
4.1.3.1 SOCC User Interface

Taxonomy Index: [Field]

ID: horowitz-WINC-11
CREATION DATE: 08/30/95 13:47

REVISION DATE: 09/26/95 10:21

ROLE: [Field]

STATUS: [Active]

PRIORITY: [Very High]

STATE: [Uncovered]

Apply | Delete | Cancel
1.4.8 Vote Window

When the owner of an Agreement clicks on the vote button, a window appears as shown in FIGURE 1-13. The top area of the window contains the voting policy. Below that is a record of each user's vote. The option menu showing the word Concur contains the three possible votes: Concur, Don't Concur and Abstain.

Initiating a vote: to start a vote, the owner of the artifact enters text in the Policy field, and clicks on the button immediately below this field. This causes the "Cast your vote" area to become active and a message is sent to all stakeholders requesting them to vote. Sample policy messages are: "Please cast your vote on milee-AGRE-3", or "please cast your vote on milee-AGRE-3 by 11/12/95", or "please cast your vote on milee-AGRE-3, a majority is sufficient to pass", etc. To cast a vote, any stakeholder may click on the Vote button of the artifact, choose the option they desire, and click on Apply. A user may change his vote at any time, while the state of the Agreement is InProgress. Only the most recent vote is retained.

Terminating a vote: only the owner of the Agreement can change the State of an Agreement. When a policy is initiated, every artifact in the artifact set of the Agreement is frozen. When the Agreement owner believes the vote has passed or failed, he selects the proper option and clicks on Apply. If the STATE is Passed, then all artifacts in the artifact set are frozen. The STATE of an Option is set to Adopted, the STATE of an ISSUE is set to Resolved, and the STATE of a Win Condition is set to COVERED (if there are no other Agreements or Issues pointing to it). Once a vote has passed, the only operation that can be performed on the Agreement is to change its Status from ACTIVE to INACTIVE. Doing this will cause all of the artifacts in the artifact set to be unfrozen.

FIGURE 1-13: Vote Window
1.4.9 Comments Window

When a user clicks on the Comments button of an artifact, a window appears. This window contains all of the stakeholder's comments about the artifact, displayed in chronological order. Each comment has the creator of the comment, the time it was created, followed by the comment.

![Win Condition](image)

1.4.10 Attachments Window

When a user clicks on the Attachments button, a window appears. This window contains a list of all files that are associated with this artifact, and the tool that created the file. The user may insert new (tool, filename) pairs by entering them in the Tool: and File fields and clicking on Apply. He may delete existing pairs by first highlighting them and then clicking on Delete. To invoke the tool and have it load the associated file, double click on the desired line and the tool will be started.
FIGURE 1-15 Attachments Window

FIGURE 1-16 Artifact Set Window
1.4.11 Artifact Set Window

The artifact set only applies to Win Conditions and Agreements. The artifact set of an Agreement is the set of artifacts that are reachable from the agreement. An Agreement may directly reach an Option or a Win Condition. Any Option adopted by the Agreement may in turn point to Issues, which may in turn point to Win Conditions. All such artifacts are said to be reachable from the initial Agreement, and as a result they are included in the artifact set. Similarly for a Win Condition, its artifact set includes all artifacts that are reachable by inverse relationships. A Win Condition may be pointed to by an Issue or an Agreement. These would be included in its artifact set. In addition, if the Win Condition is pointed at by an Issue, and this Issue is pointed at by Options, and the Options are pointed at by Agreements, then all such artifacts are included in the artifact set.

1.4.12 Artifact Action Buttons

**Apply** button  all unlocked fields are copied into the local buffer

**Cancel** button  causes the window to disappear. No updates to the local buffer is made

**Delete** button  when the user sets the STATUS button to INACTIVE, a message is created, but no pointers are altered. When a Delete command is invoked, the user is told:

Deletion is possible only if this artifact is not Referenced By another artifact.

Are you Sure? Yes/No All references will be deleted.

If "yes" is selected, and the ReferencedBy field is empty, WinWin will delete this artifact from the owners database and as a result all pointers from it to other artifacts are gone. Winwin will update the ReferencedBy field of all artifacts that were pointed to by the deleted artifact.

1.5 WinWin Files and Environment Variables

WinWin data files are stored in the directories specified for each stakeholder when they are registered. However there are files that are maintained by WinWin. Also, there are two environment variables in WinWin:

1. $WINWINHOME is the directory where your WinWin system has been installed.

2. $WINWINDATA = $WINWINHOME/winwin is the directory where WinWin stores all of its project data.

This latter directory contains the following subdirectories: projects, sysdata, and tmp.

The projects directory will contain subdirectories for each of the projects defined by this version of WinWin. Contained within a project subdirectory are files:

- project.users  which contains a list of users and the location of their data
- project.tax    which contains the taxonomy for this project
- spool          which contains newly generated messages
The `sysdata` directory contains the WinWin tools configuration file, various files related to exporting data to framemaker and lock files used by the `dbserver`.

The `tmp` directory is used to copy over the taxonomy and messages.

FIGURE 1-17 shows a sample set of directories and files.

![FIGURE 1-17 Sample WinWin Data Files](image)

```text
$WINWINDATA = $WINWINHOME/winwin

under $WINWINHOME/winwin directory

- projects - StrikeWare - ...
  - FireSGS - ...
  ...
  - SatGroundStation - SatGroundStation.users // users' inform.
  - SatGroundStation.users.bak
  - SatGroundStation.tax // taxonomy data
  - SatGroundStation.tax.bak
  - spool - horowitz // message file
  - junelee
  - egim

- sysdata - winwin.tools // tools configuration file
  - ArtifactsTOC.doc // .doc and .mif files used to export to
  - OutlineTOC.doc // framemaker
  - template.mif
  - *.LCK // dbserver lock files

- tmp - *.tax // temporary files used to transfer
  - *.msgin // taxonomy and message data between
  - *.msgout // winwin the dbserver process
```

1.6 Database Server

This is a daemon program whose purpose is to coordinate the actions of WinWin users that are simultaneously running WinWin and to manage the project files described before. Initially there is no dbserver running. When the first copy of WinWin is started, it determines that there is no dbserver, and WinWin starts it running. The dbserver maintains all access to the $WINWINHOME/project directory. This is the only program that can read and write to this directory. It creates the project subdirectories and the files within them. When someone updates their data, it informs the other active stakeholders. It assists in loading all stakeholder data.
1.7 Main Menu

When a user first starts WinWin, the main window shown in FIGURE 1-18 appears. If you are already a member of a project, you may connect to it by using the Project Open command. If you are creating a new project, you may use the Project New command.

![Main Menu of WinWin](image)

1.8 Project Menu

When the WinWin user clicks on Project the menu in FIGURE 1-19 appears:

![Project Menu](image)
1.8.1 Project New

Prompts the WinWin user for a project name, data directory, and password. It creates a new project and enters the user as the owner of the project, and one of the stakeholders. It stores the password for later use. The user must provide the directory where he wishes his data to be stored. WinWin will create a subdirectory called projectname and within it an empty project.windb file in the subdirectory. The dbserver creates the subdirectory with the proper protections so that the dbserver (and other copies of WinWin) can read any files that are created there. If this operation fails, an error message results. If the operation succeeds, the user is connected to the new project.

FIGURE 1-20 Window for Creating a New Project
1.8.2 Project Open

A list of all projects maintained by the database server appears. The WinWin user selects one. His name is checked against the list of legal stakeholders on the project. If illegal, an error message is given. If legal, all artifacts of all stakeholders in the project are read and placed into his WinWin program buffer. If a stakeholder's file cannot be accessed, or if a stakeholder's file does not exist, then an error message appears, but the program continues to load other stakeholder files. The name of the opened project is placed at the top banner of the main menu.

1.8.3 Project Users

This command begins by displaying a window containing all project users as shown in FIGURE 1-21. You begin by highlighting any one of the users.

FIGURE 1-21 Project Users Window

![Project Users Window](image)
If you select the Edit option, you may edit the Role, Title, Position or Organization fields. If you select the Add option, then you must provide the password for the selected project and the directory where the new user intends to place his WinWin data. The Role field is optional, but it can be used to describe whether the user is a customer, developer or user of the proposed system. WinWin tells the dbserver to enter him as a registered user and checks that the data directory exists with the proper protections. WinWin then creates a subdirectory whose name is the name of the project and then WinWin will write an empty windb file in that subdirectory.

If you select Delete, a confirmation box checks that you really want to delete a user. A user may only delete himself, except the Project owner who may delete anyone. A message appears: “Are you sure?” If he says yes, then he is delisted as a stakeholder of the selected project, but his artifacts are left intact. The owner of a project may delete any user. A user who is not an owner of the project may only delete himself.

1.8.4 Project Update

This command takes all of the WinWin user’s data, as contained in his WinWin buffer, and stores it in his windb file. It then notifies the dbserver that his windb file has been updated. The dbserver notifies all active WinWin users that an update occurs, and asks if they wish to be updated. If a user says YES to the update message, then his data buffer is written to his windb file, and then an entire new set of data for the project is loaded. All the generated messages are saved in the WinWinHome directory for every stakeholder.

1.8.5 Project Export

This command takes all data in the buffers and writes them either to a file in either text format, HTML format, or Mif format. Then the appropriate display program will be started and display the resulting output. The editor is Emacs, the browser is Netscape and the word processor is Framemaker. A window appears as shown in FIGURE 1-22.
### FIGURE 1-23 Framemaker Output of WinWin

#### Project Menu

**INCOMPATIBLE RE**

- **OS:** Incompatible REE
- **Body:** Incompatible REE Functions

**Comments:**
- **Owner:** Mike
- **Date:** 01/24/97
- **Status:** High
- **Priority:** Unapproved

<table>
<thead>
<tr>
<th>Order</th>
<th>Owner</th>
<th>Title</th>
<th>Status Date</th>
<th>Importance</th>
<th>Priority</th>
<th>Status</th>
<th>Priority</th>
<th>Status</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mike</td>
<td>1 LAST Communication</td>
<td>FSM 01/24/97</td>
<td>100%</td>
<td>High</td>
<td>Unapproved</td>
<td>Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Mike</td>
<td>2 LAST Communication</td>
<td>FSM 02/28/97</td>
<td>100%</td>
<td>High</td>
<td>Unapproved</td>
<td>Action</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Domain-Specific Eliminations:**

- **Title:** WinWin
- **Body:** Condition SGS software development and maintenance based on the model
  - Rational SGS software development and maintenance based on the model
- **Comments:**
  - **Owner:** Mike
  - **Date:** 01/24/97
  - **Status:** High
  - **Priority:** Unapproved

<table>
<thead>
<tr>
<th>Order</th>
<th>Owner</th>
<th>Title</th>
<th>Status Date</th>
<th>Importance</th>
<th>Priority</th>
<th>Status</th>
<th>Priority</th>
<th>Status</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mike</td>
<td>1 LAST Communication</td>
<td>FSM 01/24/97</td>
<td>100%</td>
<td>High</td>
<td>Unapproved</td>
<td>Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Mike</td>
<td>2 LAST Communication</td>
<td>FSM 02/28/97</td>
<td>100%</td>
<td>High</td>
<td>Unapproved</td>
<td>Action</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**References:**

- For more information, refer to WinWin Reference Manual page 39.
### FIGURE 1-24
Export to HTML, Displayed in Netscape

#### WIN_CONDITION

- **ID:** bose-WNC-1
- **Owner:** bose
- **Role:** User
- **Creation Date:** 10/11/93 00:00
- **Revision Date:** 02/09/94 00:00
- **Name:** Interoperable SEE functions
- **Body:** Interoperable SEE functions
- **Priority:** High
- **Status:** Active
- **State:** Uncovered
- **Refers To:** horowitz-AGRE-6, Relates To
- **References:** borowitz-AGRE-2, CorrectedBy

**Conditions:**
- The SEE tools operate within a software bus framework
  - Rationale: Standardized integration tools

**Comments:**
- ID: bose-WNC-2
- Owner: bose
- Role: User
- Creation Date: 10/11/93 00:00
- Revision Date: 02/01/94 00:00
- Name: Domain-specific Simulation capability
- Body: Condition: SGS workload modeling and information flow simulation based on the model.
  - Rationale: Reduced SGS software development cost and schedule, Support training
  - Priority: High
  - Status: Active
  - State: Uncovered
  - Refers To: horowitz-AGRE-7, Relates To

**Notes:**
- Document: Done.
1.8.6 Project Options

There are two options that can be set by this command. The first has to do with the display of INACTIVE artifacts. Initially, WinWin will only display ACTIVE artifacts. If an artifact is set to INACTIVE, it will disappear from all lists. However, to cause it, and all other INACTIVE artifacts to be displayed, click on the appropriate option. The second has to do with the display of artifacts and list windows. Suppose you are connected to a project, and on the screen you have the list of Win Conditions, the list of Agreements, two issues and two options displayed. But you need to leave and won’t return until tomorrow. By selecting the option “retain state”, and then updating your project, you may exit WinWin, and the next time you load the project all list windows and artifacts will be restored exactly as you had left them.

1.8.7 Project Delete

Only the owner can delete an artifact and only the owner of the project can delete it. If Project Delete is selected, a dialog box appears as shown in FIGURE 1-26. If the user selects OK, then the observer will no longer list this project. However, the owner’s data is not erased. An error message will be displayed if there is any stakeholder remaining except the owner.

1.8.8 Project Facts

A dialog box is displayed which contains a summary of the number of artifacts in the current project. The rows are organized by project user. The columns by artifact type. Artifacts are broken out according to their State.
A dialog box appears and asks if the WinWin user is sure they want to exit. If the buffer contains some data which has not been saved, the user is reminded that data will be lost if he exits, and he is given the option to update his project data.
1.9 Taxonomy Menu

WinWin supports a mechanism for organizing the artifacts. It is an outline (sometimes called a domain taxonomy). The system permits anyone to edit the outline or taxonomy as long as they know the project password. It is up to the owner of the project to coordinate among the people who know that password.

A taxonomy can be very useful, both during negotiation and in producing the Requirements document. Once a taxonomy is established, users should always associate a new artifact with one or more appropriate taxonomy nodes. This will help others to find artifacts that are related to each other. When the WinWin data is exported to Mosaic/Netscape or Framemaker, the resulting output may be organized according to the taxonomy. During a WinWin session, users may click on taxonomy nodes and display the artifacts associated with that node. This list view of the artifacts can help the user to navigate among them.

The window on the left in FIGURE 1-29 shows what appears when the user clicks on the taxonomy button on the main menu. Assuming a taxonomy has been defined, it will be placed in the main viewing area. Below the taxonomy nodes are a set of commands. Initially these commands are grayed out, so the user cannot use them. These commands are used to modify the taxonomy, edit, insert and delete nodes, or to move a node up, down, left or right within the taxonomy. To gain access to these commands, the user must click on the button labeled “Edit: Permission”. If the user knows the project password, then either WinWin will return saying that the taxonomy is currently being modified by someone else, or else the commands will become active. The user may now alter the taxonomy in any way. When done, he clicks on Edit to give up permission or simply cancels the taxonomy.

The four arrow keys control the position of a taxonomy node. For example suppose we have selected the node whose number is 2.2, and it is surrounded by nodes numbered 2.1 and 2.3. If we click the Up arrow, then it becomes 2.1, interchanging places with the node immediately above it. If we click the Down arrow, then it becomes 2.3, interchanging places with the node immediately below it. If we click on the right pointing arrow, then it becomes 2.1.1, a subpart of 2.1. If we click the left pointing arrow, then it becomes 3.

When someone makes changes to the taxonomy, these changes will not be immediately seen in artifacts. To update the Taxonomy window of an artifact so it does reflect recent changes, the user should click on the Refresh Artifacts button.

The second window in FIGURE 1-29 appears when the user clicks on Display in the taxonomy window. If the user clicks on one of the artifact type buttons in this window, and then on OK, a List window appears which gives all artifacts associated with this taxonomy node.
Sample taxonomy Window and Associated Display

1. Space Segment
   1.1 Space Segment Mission Concept
   1.2 Mission Satellites
   1.2.1 Satellite Coverage
   1.2.2 Visibility
   1.2.3 Satellite Bus
      1.2.3.1 Attitude Control Subsystem
      1.2.3.2 Computer Subsystem
      1.2.3.3 Communications Subsystem
   1.2.4 Orbit / Track
   1.2.5 Payload
   1.3 Support Satellites
2. Communication Network
3. Mission Objectives

Index: 1.2.1
Title: Satellite Coverage
1.10 **Artifacts Menu**

The Artifacts Menu has to do with the definition and manipulation of the artifacts of WinWin. There are five predefined artifacts, as discussed earlier. FIGURE 1-30 shows the menu associated with each artifact.

There are five possible artifacts that can be created or listed via this menu. The user clicks on the artifact type he wishes to examine and a List window appears. This window contains a list of all artifacts of the chosen type. From this list any artifact can be displayed, either by double clicking or selecting the item and clicking on OK. Several artifacts of any type can be displayed. In this way the user can simultaneously display whatever artifacts he needs to examine, as long as there is enough window space to accommodate them all. Initially all artifacts are listed by ID, Name and State. Using the Options button at the bottom of the list window, the fields and their order can be varied.

There are five buttons at the bottom of a List window. They are: OK, New, Delete, Options, Cancel.

When the New command is clicked, an empty artifact of the chosen type is generated. Once the APPLY button is clicked, an id assigned, the owner’s role is set, the creation date and the revision date. The user may fill in any of the editable fields. When the stakeholder clicks on Project Update, it is saved in his windb file. Cancel causes the artifact to disappear, with all changes ignored.
The OK button causes the selected item to be placed on the screen.

The Delete button causes the selected item to be deleted, as long as the artifact is owned by the person selecting the command and no other artifacts point to it.

The Cancel button causes the List window to disappear.

The Options button produces the window shown in FIGURE 1-32. There are five fields: Owner, State, Priority, Revision Date and Name. The List window can show all or any subset of the fields of the chosen artifact type. In addition the list of artifacts can be sorted by three keys, a primary, secondary and tertiary key.
1.10.1 Artifact Copy

A window appears asking for an id of an artifact. A copy of the artifact is produced, with all fields being the same except for: the owner which is the creator of the copy, the comments, and any possible vote.

Fields that are copied over exactly include the artifact’s: name, body, status, priority, refersto, taxonomy, and attachments.

Fields that are altered are: Id (a new value), Owner (the person performing the Copy command), Creation Date (the current date and time), Release Date (the current date and time), Role (of the new owner), Vote (empty), Comments (empty), and ReferencedBy (empty).

1.11 Tools Menu

When WinWin is installed a configuration file may be provided, called winwin.tools. Each line in this file is the name of a software program that is to be regarded as useful for working with WinWin. When a user starts WinWin and connects to a project, WinWin will ask the dbserver for a list of all tools. The dbserver returns the list and WinWin creates the Tools menu. We initially support COCOMO. Note that Mosaic and Framemaker...
are related to the entire project and not to a single artifact, and so they are not regarded as tools, but as programs to which WinWin exports its data.

When a user clicks on the Tools button, and then clicks on one of the menu items, see FIGURE 1-33, the chosen tool will be started. The user may work in the tool as long as he wishes. At some point the user may desire to associate some output of the tool with an artifact. To do this,

1. he displays the artifact
2. he clicks on the button labeled Attachments
3. he provides the tool name and the associated file name and clicks on Apply

1.12 Messages Menu

At the top level menu there is a button called Messages. Each stakeholder has his own Messages list. This list is added to by WinWin and managed by the owner. In its current form the Messages list is a summary of the actions that have been taken by stakeholders on their data. Table 3 is a summary of the list of messages that might appear in the Messages window. The messages are listed in chronological order.

Table 4: Summary of Messages

<table>
<thead>
<tr>
<th>Message Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artifact A has been created at time</td>
</tr>
<tr>
<td>Artifact A has been modified at time</td>
</tr>
<tr>
<td>Artifact A was deleted at time</td>
</tr>
<tr>
<td>Artifact A a vote is IN-PROGRESS</td>
</tr>
<tr>
<td>User X voted on Artifact A</td>
</tr>
<tr>
<td>Artifact A is PASSED</td>
</tr>
<tr>
<td>Artifact A has FAILED</td>
</tr>
<tr>
<td>Taxonomy was odified by User at Time</td>
</tr>
</tbody>
</table>
Messages are initiated by WinWin and require no explicit action by the stakeholder. New messages are added at the bottom of a user's Messages list when the project is opened. When the Message list is displayed the newest messages are displayed last. Messages can be deleted by the owner. By simply double clicking on one of the messages, the artifact can be brought up.

FIGURE 1-34 Sample Messages Window

<table>
<thead>
<tr>
<th>Messages</th>
<th>Double click to see artifact</th>
</tr>
</thead>
<tbody>
<tr>
<td>horowitz-WINC-10 has been created at 07/25/95 10:35</td>
<td></td>
</tr>
<tr>
<td>horowitz-ISSU-2 has been created at 07/25/95 10:36</td>
<td></td>
</tr>
<tr>
<td>horowitz-ISSU-2 has been modified at 07/25/95 10:37</td>
<td></td>
</tr>
<tr>
<td>horowitz-OPTN-5 has been created at 07/25/95 10:37</td>
<td></td>
</tr>
<tr>
<td>horowitz-OPTN-6 has been modified at 07/25/95 10:39</td>
<td></td>
</tr>
<tr>
<td>horowitz-AGRE-17 has been created at 07/25/95 10:40</td>
<td></td>
</tr>
<tr>
<td>horowitz-AGRE-17 a vote is IN_PROGRESS at 07/25/95 10:41</td>
<td></td>
</tr>
<tr>
<td>horowitz-AGRE-17 a quote voted at 08/08/95 10:02</td>
<td></td>
</tr>
<tr>
<td>curran-TERM-2 has been created at 08/29/95 10:08</td>
<td></td>
</tr>
<tr>
<td>horowitz-WINC-11 has been created at 08/30/95 13:47</td>
<td></td>
</tr>
<tr>
<td>horowitz-OPTN-6 has been modified at 08/30/95 13:50</td>
<td></td>
</tr>
<tr>
<td>horowitz-WINC-11 has been modified at 08/30/95 13:57</td>
<td></td>
</tr>
<tr>
<td>horowitz-WINC-12 has been created at 08/30/95 13:56</td>
<td></td>
</tr>
<tr>
<td>curran-TERM-3 has been created at 09/05/95 13:07</td>
<td></td>
</tr>
<tr>
<td>curran-TERM-1 has been modified at 09/19/95 13:21</td>
<td></td>
</tr>
</tbody>
</table>
1.13 Analysis of the WinWin Process Model

FIGURE 1-35 shows a state transition diagram that describes the various states of the WinWin database as negotiation proceeds. Nodes describe the possible states of the database while transitions are actions taken by the WinWin system or by the stakeholders.

<table>
<thead>
<tr>
<th>FIGURE 1-35</th>
<th>State Transition Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No unresolved Issues</td>
</tr>
<tr>
<td></td>
<td>No uncovered WinC</td>
</tr>
<tr>
<td></td>
<td>No Open Agreements</td>
</tr>
<tr>
<td></td>
<td>WinC entered</td>
</tr>
<tr>
<td></td>
<td>Uncovered WinC exists</td>
</tr>
<tr>
<td></td>
<td>stakeholders notified</td>
</tr>
<tr>
<td></td>
<td>stakeholders review</td>
</tr>
<tr>
<td></td>
<td>WinC and Agreements</td>
</tr>
<tr>
<td></td>
<td>comments imply consensus</td>
</tr>
<tr>
<td></td>
<td>Open Agreement entered</td>
</tr>
<tr>
<td></td>
<td>stakeholders notified</td>
</tr>
<tr>
<td></td>
<td>Vote successful</td>
</tr>
<tr>
<td></td>
<td>Vote Completed</td>
</tr>
<tr>
<td></td>
<td>Issue explaining WinC</td>
</tr>
<tr>
<td></td>
<td>problem is entered</td>
</tr>
<tr>
<td></td>
<td>stakeholders notified</td>
</tr>
<tr>
<td></td>
<td>Options resolving</td>
</tr>
<tr>
<td></td>
<td>Issue are entered</td>
</tr>
<tr>
<td></td>
<td>stakeholders notified</td>
</tr>
<tr>
<td></td>
<td>Option is chosen</td>
</tr>
<tr>
<td></td>
<td>to resolve the Issue</td>
</tr>
<tr>
<td></td>
<td>vote successful</td>
</tr>
<tr>
<td></td>
<td>vote unsuccessful</td>
</tr>
</tbody>
</table>
Architecture of WinWin
1.14 Acknowledgments

The design of Winwin has been a collaborative effort involving the entire faculty and staff of the Center for Software Engineering at USC. This includes: Barry Boehm, Ellis Horowitz, Prasanta Bose, and Ph.D. students Ming June Lee, Ahmed Abdalla, Brad Clark, Cristina Gacek, Yimin Bao, Hoh In, June Sup Lee and Eul Gyu Im. The developers of WinWin include: June Sup Lee, Eul Gyu Im, and Anne Curran, with some support from Hoh In, Yimin Bao.
A
ACTIVE 22, 23, 41
Add User 38
addressed by 23
addresses 23, 24
adopted 22, 28, 30
adopted by 23
adopts 23, 24
Agreement 6, 19
Analysis 50
Artifact Action Buttons 33
Artifact Copy 47
Artifact Fields 21
artifact ID 9
Artifact Relationships 23, 24
Artifact Set 19, 22, 33
artifact type 45
Artifacts 19
Artifacts Menu 45
Attachments 15, 19, 22, 31, 47, 48

B
Body 11, 19, 22, 47

C
Cancel 46
Center for Software Engineering 52
chronological order 48
COCOMO 14, 47
Comments 12, 19, 22, 31
completeness 5
collection file 47
cost 19
consistent 5, 26
COPY 23
Covered 22, 28, 30
covers 23, 24
CreationDate 22
customer 7

d
daemon 34
data directory 7, 36
Database Server 34
definition 45
Delete 26, 29, 46
Delete User 38
developer 7
Display 43
domain taxonomy 43

E
Edit 43
Emacs 38

F
Failed 22
fields 19
file name 48
Files and Environment Variables 33
Framemaker 6, 38, 47
Framemaker Output 39
frozen 30

H
HTML 18, 38

I
ID 45
INACTIVE 22, 23, 28, 41
Initiating a vote 30
InProgress 13, 22, 30
inverse relationship 23, 33
involves 23, 24, 28
Issue 6, 14, 19

K
copy 46

L
List window 45
logical grouping 29

M
Main Menu 35
manipulation 45
Messages 12
Messages Menu 48
Mif 18, 38
Mosaic 6, 47
Name 11, 22, 45, 46, 47
negotiation 5
Netsepc 38, 40
New 45

O
OK 46
Options 6, 15, 19
Options button 45, 46
Organization 38
outline 9, 43
Owner 19, 36, 46

P
Passed 13, 22, 30
password 36, 38, 43
policy 22
Position 38
predefined artifacts 19, 45
Priority 22, 23, 46, 47
Project Delete 41
Project Exit 42
Project Export 18, 38
Project Facts 41
Project Menu 35
project name 36
Project New 7, 35, 36

Winword REFERENCE MANUAL
Project Open 8, 35, 37
Project Options 23, 41
Project Update 11, 26, 29, 38, 45
Project Users 7, 37
Project Users Box 8

R
ReferencedBy 19, 22
Referencedby 26
RefersTo 19, 22, 25, 47
relates to 23
replaces 23, 28
requirements 5
resolve 28
Resolved 19, 22, 28, 30
Revision Date 22, 46
risk 19
Role 38

S
Satellite Ground Station 7
scenario 5
software engineering environment 7
sort 46
stakeholder 5, 6, 36
stakeholders 5
State 13, 22, 30, 41, 45, 46
state transition diagram 50
Status 22, 23, 47
summary 41
supporting artifacts 19

T
Taxonomy 5, 19, 22, 29, 47
Taxonomy Index 29
Taxonomy Menu 43
taxonomy node 9
Term 5, 8
Terminating a vote 30
terminology 5
Terms List window 9
text 18, 38
Title 38
Tool 31
tool name 48
Tools Menu 14, 47
Tutorial 6

U
Unadopted 22
uncertainty 19
Uncovered 22
UniqueID 22
Unresolved 19, 22
USC 52
user 7

V
Vote 19, 22, 30
Voting Policy 12, 30