APPENDIX A
INSTALLATION GUIDE

MetaEdit Personal 1.2

In MetaEdit 1.2, there are three subdirectories, as follows:

- C:/Metaedit/models
- C:/Metaedit/methdev
- C:/Metaedit/methods

The OPRR modeling file is named *.opr. This file has been saved in the directory C:/Metaedit/models.

The List Of *.Opr Files (in A:/oprr/).

- Activity.opr – for the Activity Diagram.
- Class.opr – for the Class Diagram.
- Colla.opr – for the Collaboration Diagram.
- Comp.opr – for the Component Diagram.
- Deploy.opr – for the Deployment Diagram.
- Sequence.opr – for the Sequence Diagram.
- Stateuml.opr – for the State Diagram.
- Use.opr – for the Use Case Diagram.

Install all these files in C:/Metaedit/models/

Steps:

1. To generate the files above, choose File menu and click on Open model. From the Open Model window, choose OPRR from the List of files. It will display all the *.opr files.
2. To create a new *.opr file, choose New Model from the File Menu. Find the file named oprr.mof. A new window will appear. Choose New to create a new file. This new file will be saved as *.opr in the directory, C:/Metaedit/models/ *.oprr.

3. A complete *.opr file can be generated using the Method Definition Tool. A report output will be generated. It is named as *.rpt file. In order to compile the file, we need to save the *.rpt file to *.mta file and save it in the directory, C:/Metaedit/methdev/ *.mta.

4. Install all the *.mta file in the directory C:/Metaedit/methdev. Since MetaEdit provides two separate applications, the MetaEdit and Method Compiler, a *.mta file can be viewed through the Method Compiler.

5. From the File menu, choose an option to create a new *.mta file or open the existing one. The *.mta file can be compiled through the MOFF menu. The MOFF provides 4 functions:

   - Check Syntax & Consistency
   This is used to check the consistency of *.mta files. A new generated *.mta is incomplete. It needs to be modified and the definitions such as metamodel header, shape definition and other definitions that are related to the technique need to be added.

   - Compile to MOFF File
   A consistent *.mta file can be compiled to *.mof file. A Save As window will appear. Save the file into the directory, C:/Metaedit/methods/ *.mof. Any modification to the *.mof file must be done in the *.mta file which needs to be recompiled.

   - Direct Compile from File
   The *.mta file can be compiled directly without checking its syntax and consistency.
• Show Messages

This is a message window which can be displayed after the compilation process.

List of Files ( A:/ mta/*mta).

• Aclass.mta – for the Class Diagram.
• Acomp.mta – for the Component Diagram.
• Activity.mta – for the Activity Diagram.
• Adeploy.mta – for the Deployment Diagram.
• Aseq.mta – for the Sequence Diagram.
• Astate.mta – for the State Diagram.
• Ause.mta – for the Use Case diagram.
• Colla.mta – for the Collaboration Diagram.

Install all the files into C:/Metaedit/methdev/

6. Stores all the *.mof files in the directory, C:/Metaedit/methods.

7. From the File menu, click on New Model. A Select window will appear. Choose the *.mof file.

8. Click the OK button and a new window will appear. Click on New and modeling can be started using the technique chosen.
The procedures to open the files from \texttt{A:/membw.zip}:

1. Open the Metaedit+, scroll down and click the \texttt{backup} button.
2. Enter the path : \texttt{c:\membw1\db\areas\uml11} , then press enter.
3. Enter the name : \texttt{uml} , then press enter.
4. When the confirmation dialog appear, click \texttt{`yes'}. 
5. Insert the first diskette, unzip it into the same path as, \texttt{C:\membw1} 
6. Insert the last diskette and then insert the first diskette one more time. Wait for the winzip instruction.
7. After finish unzipping the files, check the new folder through Window. It shows :

\begin{equation}
\text{C:\membw1\db} \quad (1)
\end{equation}

inside this folder there will be another directory 

\begin{equation}
\text{C:\membw1\membw\db} \quad (2)
\end{equation}

8. Copy the content of \texttt{db (2)} into \texttt{db (1)}.
9. After that, login to Metaedit+. A default login and password have been provided.

- \texttt{login}: sysadmin 
- \texttt{password}: sysadmin

10. Click on \texttt{diagram button} to view the method.
APPENDIX B
A). The OPRR Modeling For The Use Case Diagram
B) The OPRR Modeling For The Class Diagram
C). The OPRR Modeling For The Sequence Diagram
D). The OPRR Modeling For The Collaboration Diagram
E). The OPRR Modeling For The State Diagram
(F). The OPRR Modeling For The Activity Diagram
G. The OPRR Modeling For The Component Diagram
H). The OPRR Modeling For The Deployment Diagram
APPENDIX C
CASE STUDY

The XYZ Library plans to develop a suite of new applications to support the library system. In order to do that, XYZ needs to study the requirements of the library system. This includes the organizational and the management rules of the library that are currently managed by the administrative staff and the librarians. The administration staff handles the registration of new member and renews memberships. The other tasks are to handle the fee payment, send the reminder to the member who breaks the rules and suspend or exclude the member. The librarians are in charge of updating the book inventory, the loan and the reservation of books.

Since, the rules have stated that the borrowers must have a limited access to the system, they can only view the books in the library, borrow the books and they are allowed to make a request for book reservation. The borrowers are not allowed to change a certain data such as the date of a book loan. The librarians handle the updating processes. For any overdue book, a reminder will be sent to the borrower who has to pay for the fine.

The techniques that are involved in the case study are:

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- State Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram.

Notes:
The Class Diagram and the State Diagram are developed in the MetaEdit+ environment. The other diagrams are developed in the MetaEdit Personal 1.2.
A). The Use Case Diagram

![Use Case Diagram]

Description of the Use Case Diagram

There are five actors that are involved in this system and each has its own role. They are:

- **Administrative staff**
  - handles the registration of new member and renew the membership.
  - handles the fee payment.
  - sends the remainder to the member who breaks the rule.
  - has the authority to suspend or exclude the member.

- **The Librarian**
  - handles the update of the book inventory.
  - handles loan process.
  - handles the reservation process.
  - handles fine and payment for the overdue books.
- The Borrower
  - access the book reservation process.
  - generate borrowing process,
  - can view or search books.

- Inventory –
  - updates the book inventory.
  - stores the book information.

- Borrower database
  - keeps the membership information.
The Class Diagram shows the classes that are involved in the Library System. It shows that the Rule, Staff and Borrower are part of the Library class. The other classes such as Registration, Loan, Reservation and Fine inherit from the Rule class. But they also have their own attributes and operations. The Inventory class has an association relationship with the Library, since it stores all the data of books.
The Sequence Diagram above, shows the flow of the Library System. Starting with the registration process, the new member will be asked to pay the fee. Since only a member can use the facilities, directly after the registration process, the member is allowed to search for books either to borrow or to request for reservation. The rules stated that if the book is not returned on time, a reminder will be sent and the borrower needs to pay the fine. If the member has been reminded more than five times, he will be suspended. The librarians will update the book and send the reservation notice. The termination sign shows that the activity for that object has finished.
D). The Collaboration Diagram

This diagram shows a part of the Library System. The messages are numbered following their sequence. It describes how the student registers to the system and borrows the books or makes a reservation.
E). The State Diagram

The diagram above, shows the sequences of states and how the states response to the actions.
F). The Activity Diagram

The Activity Diagram above, describes the flow of the Library System.
G). The Component Diagram

The Component Diagram above, shows the dependencies of the modules contained in the Library System. The Library System provides two separate applications: the System Menu for borrowing, reservation and payment; and the administration component that handles the registration. The Library System also depends on the inventory database that stores all data of books.
H). The Deployment Diagram

This diagram shows the dependency of components in the system. The database of books and members are stored in the server. The library application will depend on the database.
A). The Method Definition Of The Use Case Diagram

metamodel "Use Case Diagram UML"
extension "USE"
shape "Rectangle"
  { shape (0@0, 200@0, 200@200, 0@200, 0@0);
    connection points (0@0, 200@0, 200@200, 0@200, 0@0);
    line type solid;
    line width 3;
    color darkgreen;
  }
shape "Line"
  { shape (100@100,107@100);
    line type solid;
    line width 2;
  }
shape "Circle"
  { shape (130@130);
    line type solid;
    line width 2;
    color black filled black;
    connection points (50@100, 50@89, 60@70, 70@59, 90@50,
                   110@50, 130@59, 139@70, 150@89, 150@100, 150@109,
                   139@129, 130@139, 110@149, 90@149, 70@139, 60@129,
                   50@109);
  }
shape "Straightline"
  { shape (100@130,100@180);
    line type solid;
    line width 2;
  }
shape "Handline"
  { shape (80@140,120@140);
    line type solid;
    line width 2;
  }
shape "Legline"
  { shape (70@190,100@180,130@190);
    line type solid;
    line width 2;
  }
shape "Invisiblrectangle"
  { shape (0@85,200@85,200@120,0@120);
    line type invisible;
    connection points (0@85,200@85,200@120,0@120);
  }
shape "Ellipse"
  { shape (200@150);
    line type solid;
    line width 2;
  }
color red;
connection points (100@150, 40@145, 5@120, 0@100, 5@80, 40@55, 100@50, 160@55, 195@80, 200@100, 195@120, 160@145);
}

shape "Arrow"
{ shape (100@100,90@95,90@105);
  line type solid;
  line width 0;
  color blue filled white;
}

symbol "Stickman"
{ shapes ("Circle","Straightline","Handline","Legline");
  scale 0.5;
  labels
  { "Name" at (200 180 180 200) centered;
  }
}

symbol "Rectangle"
{ shapes ("Rectangle");
  scale 2.0;
  labels
  { "System Name" at (0 10 190 0) centered;
  }
}

symbol "Ellipse"
{ shapes ("Ellipse");
  scale 0.4;
  labels
  { "Name" at (20 60 180 140) centered;
  }
}

symbol "Arrow"
{ shapes ("Arrow");
  scale 1.0;
  labels ["Label" at (10 30 100 100) left;]
}

symbol "Line"
{ shapes ("Line");
  scale 0.4;
  
}

symbol "Invisiblerectangle"
{ shapes ("Invisiblerectangle");
  scale 0.4;
  labels ["Note label" at (5 85 195 120) left;]
}

property type "Note label"
{ datatype String;
  values unique;
}

property type "Label"
{ datatype String;
  
}
role type "Communicate From" of "Communicates"
{ connectivity (0,M); }

role type "Communicate To" of "Communicates"
{ connectivity (0,M); }

role type "Extend From" of "Extends"
{ connectivity (0,M); }

role type "Extend To" of "Extends"
{ symbol "Arrow";
  connectivity (0,M);
  properties ("Label"); }

role type "Use From" of "Uses"
{ connectivity (0,M); }

role type "Use To" of "Uses"
{ symbol "Arrow";
  connectivity (0,M);
  properties ("Label"); }

bind relationship "Communicates"
as "Communicate From" ("Actor"),
as "Communicate To" ("Use case")
bind relationship "Extends"
as "Extend From" ("Use case"),
as "Extend To" ("Use case")
bind relationship "Uses"
as "Use From" ("Use case"),
as "Use To" ("Use case")
B) The Method Definition Of The Class Diagram

metamodel "Class Diagram Of UML"
extension "CLD"

shape "Rectangle"
{ shape (10@20, 190@20, 190@190, 10@190);
  connection points (10@20, 190@20,190@190, 10@190);
  line type solid;
  line width 2;
  color black filled white;
}
shape "2Rectangle"
{ shape (10@30, 190@30, 190@190, 10@190);
  connection points (10@30, 190@30,190@190, 10@190);
  line type solid;
  line width 2;
  color darkgreen filled white;
}
shape "Rect"
{ shape (10@50,190@50, 190@160,10@160);
  connection points (10@50,190@50,190@160,10@160);
  line type solid;
  line width 2;
  color blue filled white;
}
shape "Small Rectangle"
{ shape (10@0,100@0,100@50,10@150);
  connection points (10@0,100@0,100@50,10@150);
  line type solid;
  line width 2;
  color blue filled white;
}
shape "Half1 Diamond"
{ shape (100@100,80@90,80@110);
  line type solid;
  line width 1;
  connection points (100@100, 80@90,80@110);
  color lightgrey filled lightgrey;
}
shape "Half2 Diamond"
{ shape (60@100,80@90,80@110);
  line type solid;
  line width 1;
  connection points (60@100, 80@90,80@110, 80@110);
  color lightgrey filled lightgrey;
}
shape "Diamond1"
{ shape (100@100,80@90,80@110);
  line type solid;
  line width 1;
  connection points (100@100, 80@90,80@110);
  color black filled black;
{ datatype String;
   values unique;
}

property type "Interface Name"
{ datatype String;
   values unique;
}

property type "MetaClass Name"
{ datatype String;
   values unique;
}

property type "ObjectName"
{ datatype String;
   values unique;
}

property type "Note label"
{ datatype String;
   values unique;
}

property type "Generalization name"
{ datatype String;
}

property type "Documentation"
{ datatype Text;
}

property type "Aggregation name"
{ datatype String;
}

property type "Dependencies name"
{ datatype String;
}

property type "AttributeName"
{ datatype Text;
}

property type "Flow name"
{ datatype String;
}

property type "Label"
{ datatype String;
}

property type "MetaClassAttribute"
{ datatype Text
}

property type "Operation Name"
{ datatype Text;
}

property type "Export Control"
{ datatype editlist("Public","Protected","Private","Implementation");
}
property type "Interface Stereotype"
  { datatype editlist("Actor","Interface"); }

property type "Interface Type"
  { datatype editlist("Class","Parameterised class","Instantiated class","Parameterised class utility","Instantiated class utility","MetaClass"); }

property type "Aggregation Stereotype"
  { datatype editlist("Class","MetaClass"); }

property type "Cardinality"
  { datatype editlist("0.0","0..1","0..M","1..1","1..M","M"); }

property type "Class Stereotype"
  { datatype editlist("Actor","Interface"); }

property type "Class Type"
  { datatype editlist("Class","Parameterised class","Instantiated class","Parameterised class utility","Instantiated class utility","MetaClass"); }

property type "MetaClass Type"
  { datatype editlist("Actor","Interface"); }

property type "Package Stereotype"
  { datatype editlist("Actor","Interface","Class"); }

property type "RoleA"
  { datatype reference to "Class Name";
  }

property type "RoleB2"
  { datatype reference to "Interface Name";
  }

property type "RoleA1"
  { datatype reference to "MetaClass Name";
  }

property type "RoleB"
  { datatype reference to "Class Name";
  }

property type "MetaClass Stereotype"
  { datatype reference to "Class Name";
  }

property type "Description"
  { datatype Text;
  }

object type " Note"
  { symbol "Invisiblerectangle";
    duplicates allowed;
    properties ("Note label","Description"); }
connectivity (0,M);
properties("Label")
}

role type "Flow to" of "Aggregation"
{ symbol "Diamond";
  connectivity (0,M);
  properties("Label")
}

bind relationship "Aggregation"
  as "Flow from" ("Class"),
  as "Flow to" ("Metaclass","Class")

bind relationship "Dependencies"
  as "Flow from" ("Class"),
  as "Flow to" ("Interface","Package","Class")

bind relationship "Generalization"
  as "Flow from" ("Class"),
  as "Flow to" ("Metaclass","Class")

bind relationship "Association"
  as "Flow from" ("Class","Object"),
  as "Flow to" ("Interface","Package","Class","Object")

bind relationship "Composition"
  as "Flow from" ("Metaclass","Class"),
  as "Flow to" ("Metaclass","Class")
C). The Method Definition Of The Sequence Diagram

metamodel "Sequence Diagram Of UML"
extension "SEQ"

shape "Rectangle"
{ shape (10@20, 150@20, 150@100, 10@100);
  connection points (10@20, 150@20, 150@100, 10@100);
  line type solid;
  line width 2;
  color darkgreen filled white;
}

shape "Cross1"
{ shape (50@50,100@90);
  connection points (50@50,100@90);
  line type solid;
  line width 3;
  color black;
}

shape "Cross2"
{ shape (50@90,100@50);
  connection points (50@90,100@50);
  line type solid;
  line width 3;
  color black;
}

shape "Lifeline"
{ shape (10@20, 70@20, 70@130, 10@130);
  connection points (10@20,70@20,70@130,10@130);
  line type solid;
  line width 1;
  color black filled red;
}

shape "Invisiblerectangle"
{ shape (0@85,200@85,200@120,0@120);
  line type invisible;
  connection points (0@85,200@85,200@120,0@120);
}

shape "Arrow"
{ shape (100@100,90@95,90@105);
  line type solid;
  line width 1;
  color black filled black;
}

symbol "Cross"
{ shapes ("Cross1","Cross2");
  scale 0.4;
}

symbol "Lifeline"
{ shapes ("Lifeline");
  scale 0.4 ;
}
{ symbol "Rectangle";
 duplicates allowed;
 explodes;
 properties("Name","Description")
 }

object type "Activation" 
{ symbol "Lifeline";
 duplicates allowed;
 explodes;
 properties("ActivityID","Description")
 }

relationship type "Association"
{ directed;
 line type solid;
 line width 1;
 color black;
 }

relationship type "Interaction"
{ directed;
 line type dot;
 line width 1;
 color blue;
 }

role type "From" of "Interaction"
{ connectivity (0,M);
 }

role type "Predecessor from" of "Association"
{ connectivity (0,M);
 }

role type "Predecessor to" of "Association"
{ symbol "Arrow";
 connectivity (0,M);
 properties ("Label")
 }

role type "To" of "Interaction"
{ connectivity (0,M);
 }

bind relationship "Association"
as "Predecessor from" ("Activation"),
as "Predecessor to" ("Activation")

bind relationship "Interaction"
as "From" ("Activation","Objectbox"),
as "To" ("Activation ","Termination")
D). The Method Definition Of the Collaboration Diagram

metamodel "Collaboration Diagram Of UML"
extension "COL"

shape "Rectangle"
{ shape (10@0, 200@0, 200@160, 10@160);
    connection points (10@0, 200@0, 200@160, 10@160);
    line type solid;
    line width 2;
    color darkred filled white;
}

shape "Rectangle1"
{ shape (10@20, 150@20, 150@100, 10@100);
    connection points (10@20, 150@20, 150@100, 10@100);
    line type solid;
    line width 2;
    color darkgreen filled white;
}

shape "SmallRectangle"
{ shape (0@10, 180@10, 180@180, 0@180);
    connection points (0@10, 180@10, 180@180, 0@180);
    line type solid;
    line width 1;
    color black filled blue;
}

shape "Full Arrow"
{ shape (30@10, 60@30, 30@50);
    line type solid;
    line width 0;
    color black filled black;
}

shape "InvisibleRectangle"
{ shape (0@85, 200@85, 200@120, 0@120);
    line type invisible;
    connection points (0@85, 200@85, 200@120, 0@120);
}

shape "Line"
{ shape (100@100, 107@100);
    line type solid;
    line width 1;
}

shape "Circle"
{ shape (130@130);
    line type solid;
    line width 2;
    color black filled black;
    connection points (50@100, 50@89, 60@70, 70@59, 90@50,
                        110@50, 130@59, 139@70, 150@89, 150@100, 150@109,
                        139@129, 130@139, 110@149, 90@149, 70@139, 60@129,
                        50@109);
}
shape "Straightline"
{ shape (100@130, 100@180);
line type solid;
line width 2;
}

shape "Handline"
{
    shape (80@140,120@140);
    line type solid;
    line width 2;
}

shape "Legline"
{
    shape (70@190,100@180,130@190);
    line type solid;
    line width 2;
}

shape "Half1 Diamond"
{
    shape (100@100,80@90,80@110);
    line type solid;
    line width 1;
    connection points (100@100, 80@90,80@110);
    color black filled black;
}

shape "Half2 Diamond"
{
    shape (60@100,80@90,80@110);
    line type solid;
    line width 1;
    connection points (60@100, 80@90,80@110);
    color black filled black;
}

shape "Short line"
{
    shape (10@30,80@30);
    line type solid;
    line width 1;
}

shape "Stick arrow"
{
    shape (30@10,80@30,30@50);
    line type solid;
    line width 1;
}

shape "Half arrow"
{
    shape (80@30,30@10);
    line type solid;
    line width 1;
}

symbol "Rectangle1"
{
    shapes ("Rectangle1");
    scale 0.6;
    labels
        { "Name" at (10 60 150 60) centered;
        }
}

symbol "StickArrow"
{ shapes ("Short line","Stick arrow");
  scale 0.3;
  labels{
    "Label" at (0 30 80 90) centered;
  }
}

symbol "Half stickarrow"
{ shapes ("Short line","Half arrow");
  scale 0.3;
  labels{
    "Label" at (0 30 80 90) centered;
  }
}

symbol "Full Arrow"
{ shapes ("Short line","Full Arrow");
  scale 0.3;
  labels{
    "Label" at (0 20 80 100) centered;
  }
}

symbol "Black Diamond"
{ shapes ("Half1 Diamond","Half2 Diamond");
  scale 0.6;
  labels{
    "Label" at (0 60 35 90) centered;
  }
}

symbol "Stickman"
{ shapes ("Circle","Straightline","Handline","Legline");
  scale 0.5;
  labels{
    "Name" at (200 180 180 200) centered;
  }
}

symbol "DoubleRectangle"
{ shapes ("Rectangle","SmallRectangle");
  scale 0.5;
  labels{
    "Name" at (10 60 180 140) centered;
  }
}

symbol "Rectangle"
{ shapes ("Rectangle");
  scale 0.4;
  labels{
    "Name" at (10 60 190 140) centered;
  }
}

symbol "Line"
{ shapes ("Line");
  scale 0.4;
  labels{
    "Label" at (0 70 35 90) centered;
  }
}

symbol "Invisiblerectangle"
{ shapes ("Invisiblerectangle");
  scale 0.4;
  labels {"Note label" at (5 85 195 120) left;}
}
property type "Label"
{ datatype String; }

property type "Note label"
{ datatype String;
  values unique; }

property type "MessageID"
{ datatype String;
  values unique; }

property type "Name"
{ datatype String;
  values unique; }

property type "Operation"
{ datatype editlist("New","Destroyed","Transient"); }

property type "Message number"
{ datatype String; }

property type "Description"
{ datatype Text; }

object type "Note"
{ symbol "Invisiblerectangle";
  duplicates allowed;
  properties ("Note label","Description"); }

object type "Actor"
{ symbol "Stickman";
  duplicates allowed;
  explodes;
  properties("Name","Description") }

object type "Class Role"
{ symbol "Rectangle1";
  duplicates allowed;
  explodes;
  properties("Name","Operation","Description") }

object type "Multiobject"
{ symbol "DoubleRectangle";
  duplicates allowed;
  explodes;
  properties("Name","Operation","Description") }

relationship type "Association"
{ directed;
  line type solid; }
role type "Composition To" of "Composition"
{ connectivity (0,M);
}

role type "Control from" of "Control call"
{ connectivity (0,M);
}

role type "Control to" of "Control call"
{ symbol "StickArrow";
  connectivity (0,M);
  properties("Label")
}

role type "Flow from" of "Procedure call"
{ connectivity (0,M);
  symbol "Full Arrow";
  properties("Label")
}

role type "Flow to" of "Procedure call"
{ connectivity (0,M);
}

bind relationship "Association"
as "Association To" ("Actor","Class Role","Multiobject"),
as "Association From" ("Actor","Class Role","Multiobject")

bind relationship "Asynchronous control"
as "Asynchronous to" ("Actor","Class Role"),
as "Asynchronous from" ("Actor","Class Role","Multiobject")

bind relationship "Composition"
as "Composition Of" ("Multiobject"),
as "Composition To" ("Class Role")

bind relationship "Control call"
as "Control from" ("Actor","Class Role"),
as "Control to" ("Actor","Class Role","Multiobject")

bind relationship "Procedure call"
as "Flow to" ("Actor","Class Role"),
as "Flow from" ("Actor","Class Role","Multiobject")
E). The Method Definition Of The State Diagram

metamodel "State Diagram Of UML"
extension "STA"

shape "Rounded Rectangle"
{ shape (35@39, 35@20, 39@10, 44@5, 54@1, 146@1, 156@5, 
161@10, 165@20, 165@180, 161@190, 156@195, 
146@199, 54@199, 44@195, 39@190, 35@180,35@39, 165@39);
  line type solid;
  color darkblue;
  line width 2;
  connection points (35@1, 165@1, 165@199, 35@199);
}

shape "Invisiblerectangle"
{ shape (0@85,200@85,200@120,0@120);
  line type invisible;
  connection points (0@85,200@85,200@120,0@120);
}

shape "Stick Arrow"
{ shape (160@120,200@100,160@80);
  line type solid;
  line width 1;
  color black filled black;
}

shape "SArrow"
{
  shape (100@100,90@95,90@105);
  line type solid;
  color black filled black;
}

shape "Synchronization Bar"
{ shape (10@20, 20@20, 20@80, 10@80);
  connection points (10@20,20@20,20@80,10@80);
  line type solid;
  line width 1;
  color black filled blue;
}

shape "Smallcircle"
{
  shape (144@144);
  line type solid;
  connection points (50@100, 50@89, 60@70, 70@59, 90@50, 
110@50, 130@59, 139@70, 150@89, 150@100, 150@109, 
139@129, 130@139, 110@149, 90@149, 70@139, 60@129, 
50@109);
  line width 2;
  color black;
}

shape "Short line"
{ shape (100@100,200@100);
  line type solid;
  line width 1;
}
shape "Arrow"
{ shape (80@110,100@100,80@90);
 line type solid;
 line width 1;
}

shape "Black Circle"
{ 
 shape (122@122);
 line type solid;
 color black filled black;
 connection points (50@100, 50@89, 60@70, 70@59, 90@50, 110@50,
 130@59, 139@70, 130@89, 130@100, 130@109, 139@129,
 130@139, 110@149, 90@149, 70@139, 60@129, 50@109);
}

symbol "SArrow"
{ shapes ("SArrow");
 scale 1.0;
 labels { "Label" at (10 75 90 90) centered;}
}

symbol "Arrow"
{ shapes ("Arrow");
 scale 1.0;
 labels{
 "Label" at (10 75 90 90) centered;
 }
}

symbol "Bull eye"
{ shapes ("Smallcircle","Black Circle");
 scale 0.3;
 }

symbol "Start state"
{ shapes ("Black Circle");
 scale 0.4;
 }

symbol "Thick Line"
{ 
 shapes ("Synchronization Bar");
 scale 0.8 ;
 }

symbol "Rounded Rectangle"
{ 
 shapes ("Rounded Rectangle");
 scale 0.6;
 labels
 { "Action" at (39 59 161 170) centered;
   "State Name" at (78 5 122 34) centered;
 }
}

symbol "Invisiblerectangle"
{ shapes ("Invisiblerectangle");
 scale 0.4;
 labels ("Note label" at (5 85 195 120) left;)
}
symbol "Stick Arrow"
{ shapes ("Stick Arrow");
  scale 0.3;
  labels{
    "Name" at (50 60 80 90) centered;
  }
}
property type "Label"
{ datatype String;
  }

property type "Note label"
{ datatype String;
  values unique;
  }

property type "BarlID"
{ datatype String;
  values unique;
  }

property type "Final StateID"
{ datatype String;
  values unique;
  }

property type "Initial StateID"
{ datatype String;
  values unique;
  }

property type "Action"
{ datatype Text;
  }

property type "Transition Name"
{ datatype String;
  }

property type "State Name"
{ datatype String;
  values unique;
  }

property type "Name"
{ datatype String;
  }

property type "Description"
{ datatype Text;
  }

object type "Note"
{ symbol "Invisiblerectangle";
  duplicates allowed;
  properties ("Note label","Description");
  }

object type "Final State"
{ symbol "Bull eye";
  duplicates not allowed;
  properties("Final StateID","Description")
  }

object type "Initial State"
{ symbol "Start state";
    duplicates not allowed;
    properties("Initial StateID","Description")
}

object type "State"
{ symbol "Rounded Rectangle";
    duplicates allowed;
    explodes;
    properties("State Name","Action","Description")
}

object type "Synchronization Bar"
{ symbol "Thick Line";
    duplicates allowed;
    explodes;
    properties("BarID","Description")
}

relationship type "Transition"
{ directed;
    line type solid;
    line width 2;
    color black;
}

role type "Transition From" of "Transition"
{ connectivity (0,M);
}

role type "Transition To" of "Transition"
{ connectivity (0,M);
    symbol "SArrow";
    properties("Label");
}

bind relationship "Transition"
as "Transition From" ("Final State","Synchronization Bar","Initial State","State"),
as "Transition To" ("State","Initial State","Final State","Synchronization Bar")
metamodel "Activity Diagram Of UML"
extension "ACT"

shape "Smallcircle"
{
    shape (144@144);
    line type solid;
    connection points (50@100, 50@89, 60@70, 70@59, 90@50, 110@50, 130@59, 139@70, 150@89, 150@100, 150@109, 139@129, 130@139, 110@149, 90@149, 70@139, 60@129, 50@109);
    line width 2;
    color black;
}

shape "Black Circle"
{
    shape (122@122);
    line type solid;
    color black filled black;
    connection points (50@100, 50@89, 60@70, 70@59, 90@50, 110@50, 130@59, 139@70, 130@89, 130@100, 130@109, 139@129, 130@139, 110@149, 90@149, 70@139, 60@129, 50@109);
}

shape "Arrow"
{
    shape (80@110,100@100,80@90);
    line type solid;
    line width 1;
}

shape "Short line"
{
    shape (100@100,200@100);
    line type solid;
    line width 1;
}

shape "Invisiblerectangle"
{
    shape (0@85,200@85,200@120,0@120);
    line type invisible;
    connection points (0@85,200@85,200@120,0@120);
}

shape "Bar"
{
    shape (10@20,150@20,150@40,10@40);
    connection points (10@20,100@20,100@40,10@40);
    line type solid;
    line width 1;
    color black filled black;
}

shape "Stick Arrow"
{
    shape (160@120,200@100,160@80);
    line type solid;
    line width 1;
    color black filled black;
}

shape "Rounded Rectangle"
{
    shape (10@35,11@34,12@33,13@32,14@31,15@30,185@30,186@31,187@
symbol "Stick Arrow"
{ shapes ("Stick Arrow");
  scale 0.3;
}

symbol "Arrow"
{ shapes ("Arrow");
  scale 1.0;
  labels{
    "Label" at (10 75 90 90) centered;
  }
}
symbol "Bull Eye"
{ shapes ("SmallCircle","Black Circle");
  scale 0.3;
}

symbol "Start state"
{ shapes ("Black Circle");
  scale 0.3;
}

property type "Note label"
{ datatype String;
  values unique;
}

property type "Label"
{ datatype String;
}

property type "Name"
{ datatype String;
}

property type "BarlID"
{ datatype String;
  values unique;
}

property type "DecisionID"
{ datatype String;
  values unique;
}

property type "EndID"
{ datatype String;
  values unique;
}

property type "StartID"
{ datatype String;
  values unique;
}

property type "Action"
{ datatype String;
  values unique;
}
property type "Description"
   { datatype Text; }

object type "Activity"
   { symbol "Rounded Rectangle";
    duplicates allowed;
    explodes;
    properties("Action","Description")
   }

object type "Decision"
   { symbol "Diamond";
    duplicates allowed;
    explodes;
    properties("DecisionID")
   }

object type "End"
   { symbol "Bull Eye";
    duplicates not allowed;
    properties("EndID")
   }

object type "Start"
   { symbol "Start state";
    duplicates not allowed;
    properties("StartID")
   }

object type "Synchronization Bar"
   { symbol "Thick Line";
    duplicates allowed;
    explodes;
    properties("BariD")
   }

relationship type "Transition"
   { directed;
    line type solid;
    line width 1;
    color brown;
   }

role type "Transition From" of "Transition"
   { symbol "Arrow1";
    connectivity (0,M);
    properties ("Label");
   }

role type "Transition to" of "Transition"
   { connectivity (0,M);
   }

bind relationship "Transition"
  as "Transition to" ("Activity","Start","End","Decision","Synchronization Bar"),
  as "Transition From" ("Decision","Synchronization Bar","End","Start","Activity")
G). The Method Definition Of The Component Diagram

metamodel "Component Diagram Of UML"
extension "COD"

shape "Rectangle"
{ shape (20@20, 200@20, 200@180, 20@180);
  connection points (20@20, 200@20, 200@180, 20@180);
  line type solid;
  line width 2;
  color blue filled white;
}

shape "Rect1"
{ shape (0@50,40@50,40@70,0@70);
  line type solid;
  line width 1;
  color black filled yellow;
}

shape "Rect2"
{ shape (0@120,40@120,40@140,0@140);
  line type solid;
  line width 1;
  color black filled yellow;
}

shape "Invisiblerectangle"
{ shape (0@85,200@85,200@120,0@120);
  line type invisible;
  connection points (0@85,200@85,200@120,0@120);
}

shape "Stick Arrow"
{ shape (160@120,200@100,160@80);
  line type solid;
  line width 1;
}

shape "Circle"
{
  shape (122@122);
  line type solid;
  color black filled white;
  connection points (50@100, 50@89, 60@70, 70@59, 90@50, 110@50,
  130@59, 139@70, 130@89, 130@100, 130@109, 139@129,
  130@139, 110@149, 90@149, 70@139, 60@129, 50@109);
}

shape "Short line"
{ shape (0@100,120@100);
  line type solid;
  line width 1;
  connection points (100@100,200@100);
}

shape "Arrow"
{ shape (80@110,100@100,80@90);
  line type solid;
  line width 1;
}
symbol "Rectangle"
{ shapes ("Rectangle","Rect1","Rect2");
  scale 0.6;
  labels {"Name" at (10 60 190 140) centered;}
}

symbol "Invisiblerectangle"
{ shapes ("Invisiblerectangle");
  scale 0.4;
  labels {"Note label" at (5 85 195 120) left;}
}

symbol "Stick Arrow"
{ shapes ("Stick Arrow");
  scale 0.3;
  labels{
    "Name" at (50 60 80 90) centered;
  }
}

symbol "Arrow"
{ shapes ("Arrow");
  scale 1.0;
  labels{
    "Label" at (10 75 90 90) centered;
  }
}

symbol "Lollipop"
{ shapes ("Short line","Circle");
  scale 0.5;
  labels{
    "Name" at (200 130 100 180) centered;
  }
}

property type "Label"
{ datatype String;
}

property type "Note label"
{ datatype String;
  values unique;
}

property type "Description"
{ datatype Text;
}

property type "Name"
{ datatype String;
  values unique;
}

property type "Type"
{ datatype String;
}

object type "Note"
{ symbol "Invisiblerectangle";
  duplicates allowed;
  properties ("Note label","Description");
}
object type "Component"
{ symbol "Rectangle";
  duplicates allowed;
  explodes;
  properties("Name","Type","Description")
}

object type "Interface"
{ symbol "Lolipop";
  duplicates allowed;
  explodes;
  properties("Name","Description")
}

relationship type "Dependency"
{ directed;
  line type dot;
  line width 1;
  color blue;
}

role type "Flow From" of "Dependency"
{ connectivity (0,M);
  symbol "Arrow";
  properties ("Label")
}

role type "Flow To" of "Dependency"
{ connectivity (0,M);
}

bind relationship "Dependency"
as "Flow To" ("Component","Interface"),
as "Flow From" ("Component","Interface")
H). The Method Definition Of The Deployment Diagram

metamodel "Deployment Diagram Of UML"
extension "DEP"

shape "Rectangle"
{ shape (20@20, 200@20, 200@180, 20@180);
  connection points (20@20, 200@20,200@180, 20@180);
  line type solid;
  line width 2;
  color blue filled white;
}
shape "Rectangle1"
{ shape (15@0, 200@0, 200@160);
  connection points (10@0, 200@0,200@160, 0@200,0@10, 180@10,180@180, 0@180,20@20,0@30,200@0,180@20,200@160,180@180,0@10,15@0,200@0,180@10);
  line type solid;
  line width 3;
  color darkblue;
}
shape "SmallRectangle"
{ shape (0@10,180@10,180@180,0@180,0@10,15@0,200@0,180@10);
  connection points (0@10, 180@10,180@180, 0@180,0@10,15@0,200@0,180@10);
  line type solid;
  line width 2;
  color darkblue;
}
shape "LineRec"
{ shape (180@180,200@160);
  connection points (20@20,0@30,200@0,180@20,200@160,180@180);
  line type solid;
  line width 3;
  color darkblue;
}
shape "Rect1"
{ shape (0@50,40@50,40@70,0@70);
  connection points (0@40, 20@40,20@60, 0@60);
  line type solid;
  line width 1;
  color black filled yellow;
}
shape "Rect2"
{ shape (0@120,40@120,40@140,0@140);
  connection points (0@120, 20@20,20@140, 0@140);
  line type solid;
  line width 1;
  color black filled yellow;
}
shape "InvisibleRecangle"
{ shape (0@85,200@85,200@120,0@120);
  line type invisible;
  connection points (0@85,200@85,200@120,0@120);
}
explodes;
properties("Component Name","Type","Documentation")
}

object type "interface"
{ symbol "Lollipop";
  duplicates allowed;
  explodes;
  properties("Interface Name","Documentation")
}

object type "Node"
{ symbol "3D Cube";
  duplicates allowed;
  explodes;
  properties("Node Name","Type","Documentation")
}

object type "Object"
{ symbol "Rectangle2";
  duplicates allowed;
  explodes;
  properties("Object Name","Type","Documentation")
}

relationship type "Dependency"
{ directed;
  line type dot;
  line width 1;
  color black;
}

relationship type "Instance"
{ }

role type "From" of "Dependency"
{ connectivity (0,M);
  symbol "Arrow";
  properties ("Label");
}

role type "Part of" of "Instance"
{ connectivity (0,M);
}

role type "To" of "Dependency"
{ connectivity (0,M);
}

role type "Part from" of "Instance"
{ connectivity (0,M);
}

bind relationship "Dependency"
  as "To" ("Node","Component","Interface","Object");
as "From" ("Node","Component","Interface","Object")

bind relationship "Instance"
  as "Part from" ("Component","Object","Interface");
as "Part of" ("Node")