AFRL-HE-WP-TR-2005-0177

AIR FORCE RESEARCH LABORATORY

AIE: ACWA™ Integrated Environment Technical Survey

James Gualtieri
Scott Potter
William Elm
Jack McKee

ManTech Aegis Research Corporation
Cognitive Systems Engineering Center
501 Grant Street, Suite 475
Pittsburgh, PA 15219

June 2003

Final Report for March 2001 to June 2003

20060124 014

Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICE

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2005-0177

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

FOR THE DIRECTOR

//signed//

MARIS M. VIKMANIS
Chief, Warfighter Interface Division
Air Force Research Laboratory

This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* June 2003	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* March 2001 - June 2003

4.TITLE AND SUBTITLE
AIE: ACWA™ Integrated Environment Technical Survey

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62202F

6. AUTHOR(S)
James Gualtieri, Scott Potter, William Elm, Jack McKee

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
71841004

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)
ManTech Aegis Research Corporation
Cognitive Systems Engineering Center
501 Grant Street, Suite 475
Pittsburgh, PA 15219

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Materiel Command
Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/HECS

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-HE-WP-TR-2005-0177

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited. AFRL/PA approved 2/21/06, AFRL/WS 06-0523.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The scope of the research was to assess Integrated Development Environments (IDEs) for their impact and applicability for use within a Cognitive Systems Engineering (CSE) support tool. Testing and evaluation of the most promising extensible IDEs for the ACWA™ Integrated Environment (AIE) was completed. This report is a brief encapsulation of the technology assessment survey. The primary emphasis of this task was the detailed review and testing of products. This report was done on an 'economy of force' activity and contains four major results: 1) Technical Requirements of the AIE supporting technologies: This section describes the characteristics expected of the embedded and layered products used in the construction of AIE and is described from a system design perspective; 2) Primary Technical Results: Contains a summary of the technology assessment of the extensible IDE products that 'made the cut' for a detailed study; 3) Other Results: Other products that received a preliminary look during the first pass, but did not make the cut for a detailed study are summarized here; 4) Recommended Components: From the assessment results, an initial recommendation on products and tools to form the foundation of AIE is provided.

15. SUBJECT TERMS Integrated Development Environment (IDE), ACWA™ Integrated Environment (AIE), Cognitive Systems Engineering (CSE), Cognitive Models, Collaboration, Information Warfare

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Janet E Miller
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED	SAR	34	**19b. TELEPHONE NUMBER** *(include area code)* 937-656-4401

THIS PAGE LEFT INTENTIONALLY BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE LEFT INTENTIONALLY BLANK

A CONTENT REQUIREMENTS

A.1 Purpose of this Report

This report was developed to efficiently capture the results of Technical Survey while preserving the vast majority of the project budget for the actual survey. From the results captured in the Technical Survey and described in this report, decisions can be made about tools, products, etc. that could develop components for AIE's implementation.

A.2 Scope of this Project

The overall goal of this project was to explore tool support to aid the cognitive systems engineering effort of the IWCAT development team. Primarily, the ***focus was on finding tool support that would make the Cognitive Systems Engineering (CSE) portion of the IWCAT effort more productive and cost efficient.***

The scope of this project was to assess component technologies for their impact and applicability for use within such a CSE support tool. These assessment study results were expected to form the Technology Awareness input to follow on design activities as part of separate tasks/contracted efforts.

These survey results are complemented by a comparison requirements specifications effort to further define the functional requirements of a CSE support tool. That work is documented separately in reference 1.

A.3 Scope of this Document

Testing and evaluation of the most promising Extensible Integrated Development (IDEs) for AIE was completed. This document is a brief, efficient encapsulation of the technology assessment survey. The primary emphasis of this task was the detailed review and testing of products. The report was done on an 'economy of force' activity.

This document contains four major results:

1. Technical Requirements of the AIE supporting technologies: This section describes the characteristics expected of the embedded and layered products used in the construction of AIE. This is described from a system design perspective
2. Primary Technical Results: contains a summary of the technology assessment of the extensible IDE products that 'made the cut' for detailed study.
3. Other Results: Other products that received a preliminary look during the first pass, but did not make the cut for a detailed study are summarized here.
4. Recommended Components: From the assessment results, an initial recommendation on products and tools to form the foundation of AIE is provided.

A.4 Release/Classification Information

A.4.1 Security Classification

This report is UNCLASSIFIED.

A.4.2 Release/Distribution

Distribution Statement A: Approved for public release; distributed is unlimited.

A.5 References

1. ManTech Aegis Research Corp., Cognitive Systems Engineering Center (2002). Applied Cognitive Work Analysis (ACWA) Integrated Environment (AIE): An Initial Cognitive System Engineering Design Specification (CSEDS). (Contract Number # F33601-02-F-A219). Final Report for Air Force Research Laboratory. Wright-Patterson AFB, OH: Crew Systems Ergonomics Information Analysis Center.

2. www.netbeans.org

3. www.intellij.com/idea

4. www.borland.com

5. www.ilog.com

6. www.jfree.org/jfreechart/index.html

7. http://jgraph.sourceforge.net/

[redacted]

B TECHNICAL SURVEY APPROACH

B.1 Background

This technical survey task was performed based on two needs.

1. The technology, products, and tools available for implementing systems like AIE have evolved rapidly since the construction of the initial CACSE prototype in the late 90's, and that evolution would have a major impact on the design and implementation of the next generation tool.
2. The initial prototype fulfilled one of its major objectives by providing powerful "lessons learned" on critical capabilities needed in an actual CSE tool. One of the major lessons learned relates to the level of "multi-activity integration" that must be supported in parallel.

The technology evolution in Software layered products and libraries have expanded quickly in the last several years since the CACSE prototype was developed. Even more importantly, the vendor support for extensibility of what was formerly proprietary product source code has opened an entirely new construction avenue that is even more powerful than layered libraries. The impact of all these technology advances had to be assessed to determine an intelligent course of action for the development of AIE.

The "lessons learned" from the initial CACSE prototype clearly revealed the need to better support the capturing of opportunistic design insights for all steps in the ACWA™ process nearly simultaneously, as well as highlight the impacts on other portions of the 'design thread' as changes are made anywhere in the design. This 'tightly coupled' feel was not supported by the initial prototype. Further, a very similar functional need (for many of the same needs to support Design Insights as they occur in the mind of the developer, not as they procedurally occur in the program plan) was becoming mainstream in the object oriented software development community. This new multi-viewpoint integrated feel added considerably to the demands placed on the development platform's technology underpinnings.

Both needs have direct analogs in the software development tool support community. As a developer of leading edge, object oriented software, the project team immediately saw the parallels between what was needed for AIE, and what is demanded of a software developer's Integrated Development Environment (IDE). From that insight, the ACWA™ Integrated Environment (AIE) concept was developed. The Technology Survey then focused on technologies and platforms used in similar ways by the Software IDE community, and their applicability to supporting AIE development.

B.2 Technical Survey Approach

Initially, it was intended that the survey perform a 'standard' assessment of the various layered products available for a typical n-tiered object oriented system. Based on the insight of parallels with the software IDE community, the survey's focus shifted to exploration of extensible IDEs. These newly available, extensible environments offered a previously unavailable opportunity to leverage immediately available features shared across CSE and software, while allowing the development and integration of the CSE unique features needed to complete AIE.

Within the available IDEs, an immediate schism was recognized: some IDEs allowed (and in fact welcomed) extensions and customization; others had a more 'closed' approach was more typical of a proprietary product. Because of this and similar issues, the Technical Survey adopted a two pass strategy. The first pass made a quick initial assessment of the IDE to cut the list down to those deemed most promising. The second pass performed the actual technical assessment.

The technical assessment comprised two main activities: 1) Extensive reading of the product technical literature (primarily online) and technical knowledge bases, and 2) the actual installation and experimentation with demonstration/evaluation copies of the IDE. These ***technical assessment activities were the main focus of this project effort and consumed the large majority of the budget. The results of the technical assessment were compiled into this report as efficiently as possible, to minimize the cost/effort distraction of the reporting activity.***

[redacted]

C RECOMMENDED COMPONENTS FOR AIE

C.1 Recommended IDE and Graphical library

C.1.1 IDE: NetBeans

Justification: ManTech Aegis Research chose NetBeans as the preferred IDE to use for AIE because of two major factors.

1. The source code for NetBeans is Very accessible meaning that no agreement will be needed between ManTech, AFRL and NetBeans. (These agreements can slow the process of development a lot).
2. There is no End User License Fee.
3. NetBeans is an IDE and a development platform. Meaning that we can take the "core" of NetBeans and have a base IDE for ACWA in a short time. NetBeans is a Modular platform. Meaning that functionality can be added to the platform in "modules" to extend/increase the functionality of the ACWA IDE.

C.1.2 Graphical Library: JGraph and JFreeCharts

Justification: ManTech Aegis Research chose JGraph and JFreeCharts for the similar factors that were used in the IDE selection. Factors included were the library code being Open Source, and no End User License Fee.

C.2 Recommended AIE Development Environment

The software for AIE will be developed using several commercial-off-the-shelf libraries and tools that will improve the application and speed the development time:

C.2.1. **Borland Enterprise Studio:** This is the Java Compiler and IDE used for all Software development in the CSEC of ManTech Aegis Research.

C.2.2. **Rational Rose/TogetherSoft Object Modeler:** ManTech Aegis Research will use Rational Rose, unless we upgrade to Borland Studio 9.0, which is integrated with TogetherSoft Object Modeler. Currently, we are using Borland Studio 6.0.

C.2.3. **Sybase Power Designer 7.0:** It will be user to build the Logical and Physical Data Model that will, in turn, be used to store all the design data, documents, graphics and articles.

C.2.4. **JGraph and JFreeCharts Graphical Libraries:** These libraries will be used to create all the graphics and diagrams that are developed as part of the ACWA model/project.

C.2.5. **Xerces XML Parser:** The parser that will be used for all XML manipulation.

C.3 Recommend Application Design Specifications

Borland Enterprise Studio will be employed to develop the AIE software using our customized Rational Unified Processes (RUP). We use the reverse engineering functionality in Borland Enterprise Studio to verify the object models developed during the design phase.

C.4 Recommend Database Design Specifications

The physical data model for AIE will be created using an eXtensible Markup Language (XML) Schema for the initial version. We also will build a logical and physical data model using Sybase Database Designer.

C.5 Recommend IDE to use for Development

The Borland Enterprise Studio is an object-oriented Integrated Development Environment (IDE) used for designing, building and deploying Web-enabled applications. The Borland Enterprise Studio includes designers, managers, wizards, and other utilities that automate many of the tasks for developing Java applications. Some of the capabilities provided in the Borland Enterprise Studio are as follows:

- Build/Design object models (OMT Models)
- Reverse engineering from code to object model
- Reengineer and deploy data model information from and to supported database.
- Design application user interfaces including the forms for each application, the data to be displayed on each form, and the navigation between forms
- Customize with Java TM code
- Build, compile, and test Java applications

The AIE application will be developed in Java and use eXtensible Markup Language (XML) files to transfer all data and as the data repository. These files may be read using the latest version of Internet Explorer or any other XML compatible browser.

D Criteria Used to Evaluate IDEs

Criteria:

1. Support to 'basic' functions
   a. Discussion: Does the layered product provide the needed capabilities with little or no further development? ('basic' means things like file open/close, saving, printing, etc.)
2. Support to 'integrated views'
   a. Discussion: Does the layered product explicitly support the real-time linkage of the various views of the design repository. Is a user change to any aspect of the design automatically reflected into any/all other views impacted by that change? A CSE development must be managed as one integrated project (not a collection of Articles and Diagrams.
3. Development Licensing/Product Cost
   a. Discussion: What license are costs associated with Development of the AIE tool?
4. Fielding Licensing/Product Cost
   a. Discussion: What license are costs associated with Fielding each copy of the AIE tool?
5. Support to User Roles/Access Control
   a. Discussion: Can access be limited to the user of AIE by assigning roles to each user for each project? Example "reviewer" role: the user would only have read access to the model; the user would not be able to modify/edit any part of the model.
6. Support Save/Recover Project Design
   a. Discussion: Does the IDE have the ability to save and recover all parts of the AWCA methodology without any loss of the data or organization of the data? Are all diagrams and drawings recovered to exactly the same look as before they were previously?
7. Support Compiled Outputs (SRS, Direct SW constructs (e.g. can it produce display objects directly in JAVA?)
   a. Discussion: Does the IDE have the ability to create a java object in Java Source code, and/or SRS outlines from the AWCA project that is created and saved?
8. Multi-User Capability
   a. Does the IDE have an effective solution to multiple users working on a single project (NOT complete project locking)? One possible solution is for the IDE to have the ability to lock portions of a project for one user, while allowing another user to work on another portion of the AWCA project.
9. IDE Platform Dependencies
   a. Discussion: Does the IDE run on common development platform, or does it require exotic HW (or exotic DBMS, OS, etc.)?
10. Vendor Support

   a. Discussion: Can the provider be expected to provide high quality human and online technical support as we get the lid off and start poking around?

11. Graphics support

   a. Discussion: Does the product support the mixed media graphics/text that will be needed to do ACWA diagrams and drawing?

12. Flexible GUI "Easily Extensible Platform" - GUI that can be modified to satisfy the IDE needs of the AIE

   a. Discussion: Does the IDE allow for ManTech Aegis Research software development team to extend/modify the IDE to build a AWCA IDE? Is the code well documented, well organized? Are there APIs available? If we extend the IDE, will we be able to update to a new release of IDE easily, or will we be tied to a specific release of the IDE?

[redacted]

E EVAULATIONS OF INTEGRATED DEVELOPMENT ENVIRONMENT (IDE'S)

ManTech Aegis Research downloaded evaluation copies of each of the three IDEs that are discussed in detail. ManTech also downloaded and studied several of the IDEs in sections F.5.

E.1 Evaluation Matrix for Possible IDE Solutions

Table 1 summarizes the results for each IDE for each Criteria.

Evaluation Levels: High, Medium, Low

Criteria	JBuilder	IntelliJ IDEA	NetBeans
1. Support to 'basic' Functions	HIGH	HIGH	HIGH
2. Support to 'Integrated Views'	HIGH	HIGH	HIGH
3. Development Licensing/Product Cost	MEDIUM	MEDIUM	HIGH
4. Fielding Licensing/Product Cost	LOW	LOW	HIGH
5. Support to User Roles/Access Control	MEDIUM	MEDIUM	MEDIUM
6. Support Save/Recover Project Design	MEDIUM	MEDIUM	MEDIUM
7. Support Compiled Outputs	LOW	LOW	LOW
8. Multi-User Capability	MEDIUM	MEDIUM	MEDIUM
9. IDE Platform Dependencies	MEDIUM	HIGH	HIGH
10. Vendor Support	HIGH	HIGH	LOW
11. Graphics Support	LOW	LOW	LOW
12. Flexible GUI "Easily Extensible Platform"	UNKNOWN	HIGH	HIGH

Table 1: JBuilder Process Flow Diagram

E.2 Detailed Evaluation

E.2.1 Criteria: Support to 'basic' Functions

JBuilder: HIGH - Contains the most AIE functionality of any IDE that was evaluated.

IntelliJ IDEA: HIGH - Contains some of the functionality that is in JBuilder and NetBeans.

NetBeans: HIGH - Contains the all functionality that will be needed to create AIE.

E.2.2 Criteria: Support to 'Integrated Views'

JBuilder: HIGH - Very good support for Integrated Views.

IntelliJ IDEA: HIGH - Very good support for Integrated Views.

NetBeans: HIGH - Very good support for Integrated Views. They all support integrated views, in some fashion.

E.2.3 Criteria: Development Licensing/Product Cost

JBuilder: MEDIUM - A full product licensed per user seat cost is anywhere from $99 to $3999 per developer. Excellent support is available.

IntelliJ IDEA: MEDIUM - Cost is $459 to $699 per developer depending upon number of developers and type of maintenance/support.

NetBeans: HIGH - Is an open-source IDE and is a free download environment. It is extensible/releasable with no further fee or cost to AIE user. NOTE: Modules which extend the functionality are available for a fee.

E.2.4 Criteria: Fielding Licensing/Product Cost

JBuilder: LOW - Full product licensed per user seat needed. This would make distribution of AIE very complicated.

IntelliJ IDEA: LOW - Same as JBuilder.

NetBeans: HIGH - No product license necessary.

E.2.5 Criteria: Support to User Roles/Access Control

JBuilder: MEDIUM - No direct support available but by using a 3rd party Configuration management tool Roles/Access Control can be accomplished.

IntelliJ IDEA: MEDIUM - No direct support available but by using a 3rd party Configuration management tool Roles/Access Control can be accomplished.

NetBeans: MEDIUM - No direct support available but by using a 3rd party Configuration management tool Roles/Access Control can be accomplished.

E.2.6 Criteria: Support Save/Recover Project Design

JBuilder: MEDIUM - Does support XML manipulation but a 3rd party database or file organization system would need to have complete functionality.

IntelliJ IDEA: MEDIUM - Does support XML manipulation but a 3rd party database or file organization system would need to have complete functionality.

NetBeans: MEDIUM - Does support XML manipulation but a 3rd party database or file organization system would need to have complete functionality.

E.2.7 Criteria: Support Compiled Outputs

JBuilder: LOW - Not directly supported, functionality would have to be written or maybe we can find a third party software to "plug-in".

IntelliJ IDEA: LOW - Not directly supported, functionality would be completely custom built

NetBeans: LOW - Not directly supported, functionality would be completely custom built.

E.2.8 Criteria: Multi-User Capability

JBuilder: MEDIUM

IntelliJ IDEA: MEDIUM

NetBeans: MEDIUM

E.2.9 Criteria: IDE Platform Dependencies

JBuilder: MEDIUM - Available on Windows, Linux and Solaris.

IntelliJ IDEA: HIGH - Major platforms supported (Windows, Linux, Mac, Solaris, Generic Unix).

NetBeans: HIGH - Written in Java. Available on any platform with a Java 1.3 (or later) JRE.

E.2.10Criteria: Vendor Support

JBuilder: HIGH - Excellent support available.

IntelliJ IDEA: HIGH - Good support available.

NetBeans: LOW - Community support available but probably poor because it is open source. NetBean is not developed by a single company, but by a group of individuals.

E.2.11Criteria: Graphics Support

JBuilder: LOW - Only graphics support is for building Java displays.

IntelliJ IDEA: LOW - Only graphics support is for building Java displays.

NetBeans: LOW - Only graphics support is for building Java displays.

E.2.12Criteria: Flexible GUI "Easily Extensible Platform"

JBuilder: UNKNOWN - Need more research with Borland to correctly answer this question.

IntelliJ IDEA: HIGH - Product was build to be able to extend the IDE. Now there is a new addition that may be as easy as NetBeans to extend. We need to conduct do addition research on the latest release.

NetBeans: HIGH - Is an IDE that was built to be customized by the developer in order to be used in tool development.

[redacted]

F EVAULATED IDES

F.1 JBuilder (Borland)

F.1.1 Description

JBuilder is the leading, cross-platform environment for building industrial strength enterprise Java applications. JBuilder 8 Enterprise simplifies Web and EJB™ development with two-way visual designers and rapid deployment to leading J2EE™ platform application servers. JBuilder also provides enhanced productivity with UML™ code visualization, refactoring, code formatting, HotSwap debugging, unit testing, and version control integration.

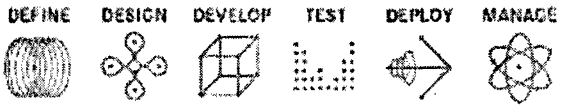
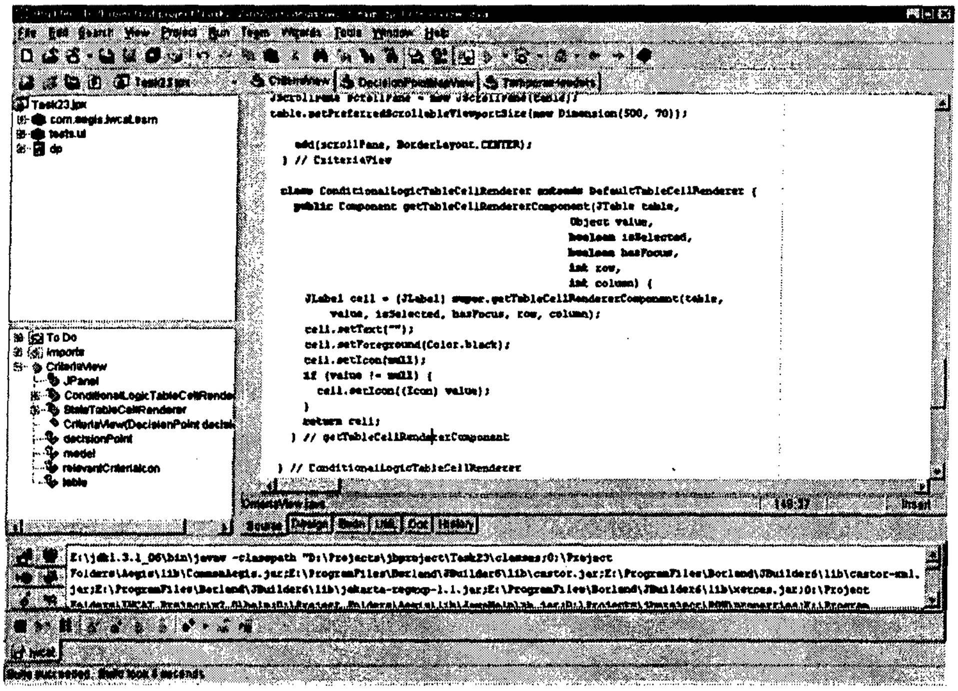


Figure 1: JBuilder Process Flow Diagram

JBuilder helps developers and large development teams become more efficient, while it meets the requirements of enterprise-scale project development. It allows for leverage of unparalleled flexibility afforded by the OpenTools architecture and import capabilities that support projects built in other environments. (See Figures 1 and 2).



```
    JScrollPane scrollPane = new JScrollPane(table);
    table.setPreferredScrollableViewportSize(new Dimension(500, 70));

      add(scrollPane, BorderLayout.CENTER);
    } // CriteriaView

    class ConditionalLogicTableCellRenderer extends DefaultTableCellRenderer {
      public Component getTableCellRendererComponent(JTable table,
                                                     Object value,
                                                     boolean isSelected,
                                                     boolean hasFocus,
                                                     int row,
                                                     int column) {
        JLabel cell = (JLabel) super.getTableCellRendererComponent(table,
            value, isSelected, hasFocus, row, column);
        cell.setText("");
        cell.setForeground(Color.black);
        cell.setIcon(null);
        if (value != null) {
          cell.setIcon((Icon) value);
        }
        return cell;
      } // getTableCellRendererComponent

    } // ConditionalLogicTableCellRenderer
```

```
E:\jdk1.3.1_06\bin\javaw -classpath "D:\Projects\jbproject\Task23\classes;O:\Project
Folders\Aegis\lib\CommonAegis.jar;E:\ProgramFiles\Borland\JBuilder6\lib\castor.jar;E:\ProgramFiles\Borland\JBuilder6\lib\castor-xml.
jar;E:\ProgramFiles\Borland\JBuilder6\lib\jakarta-regexp-1.1.jar;E:\ProgramFiles\Borland\JBuilder6\lib\xerces.jar;O:\Project
```

Build succeeded. Build took 6 seconds.

Figure 2: JBuilder IDE Sample

F.1.2 Overall Evaluation

JBuilder provides an easy to use environment but will need an agreement with Borland to as well as end-user license agreements in order to proceed.

F.2 IntelliJ IDEA - www.intellij.com/idea

F.2.1 Description

is an industry leading Java IDE power packed with leading-edge development features which includes: industry setting refactoring support, intelligent code editing assistance, a wide range of J2EE development features for rapid web-application and other enterprise development, a powerful Code Inspection tool, integrated CVS, VSS and StarTeam support, an Open API for third-party plug-in support, and a mountain of other productivity features that make Java development a pleasure.

F.2.2 Features

- **Easy to use interface:** The user interface is simple to learn, yet incredibly powerful and efficient in use. The features become available as you need them, and do not clutter up the interface. The Tips of the Day quickly teach how to take advantages of many product features.

- **Specially designed for fast work with the keyboard. 100% of features available without using the mouse:** If you only have 2 hands it may be inconvenient for you to use the mouse for some operations, switching from one input device to another. IDEA provides you ability to efficiently work with the keyboard only, greatly increasing the efficiency of your work. Sure, the mouse is usable with IDEA as well, and the mouse-wheel is supported.

- **All shortcuts are customizable:** User-defined keymaps with ability to change or set shortcuts for any menu item or editor command. Three alternative built-in keymaps are also available, emulating Mac OS X, Emacs and Visual Studio.

- **Convenient multi-pane layout with easy switching between panes, usually called tool windows:** The main frame has several panes (tool windows) and you may easily hide unnecessary panes, providing more space for the editor. You may quickly switch between panes and views using the keyboard shortcuts or the mouse.

- **Multiple projects support:** IDEA now allows you to open several projects at a time, so that each project is opened in its own frame, and you can easily switch between them. This can be useful when you need, for example, to run and debug server and client applications simultaneously

- **Various tool windows that can be now opened in floating mode (with an option to make them transparent in Windows 2000/XP):** A rich set of the IDEA tool windows helps viewing your project details in different modes, as well as browse compilation, search, inspection results, etc. Any tool window can be switched to the floating mode and positioned at any place on the screen. This is especially convenient when working with multiple monitors.

  When in floating mode, tool windows can have their transparency level adjusted, so that when they are being used on top of the editor, they conveniently fade into the background allowing you to continue working in the editor without having your screen cluttered with windows.

F.2.3 Overall Evaluation

This is a complete IDE that can be "customized" via a "plug-in" technology. The problem is removing the "Software IDE" to create a CSE IDE. The "plug-in" technology is similar to the "module" technology in NetBeans. There is also a new version of IntelliJ that has just come out that has a lot of the functionality that AIE will need and that we currently have with NetBeans.

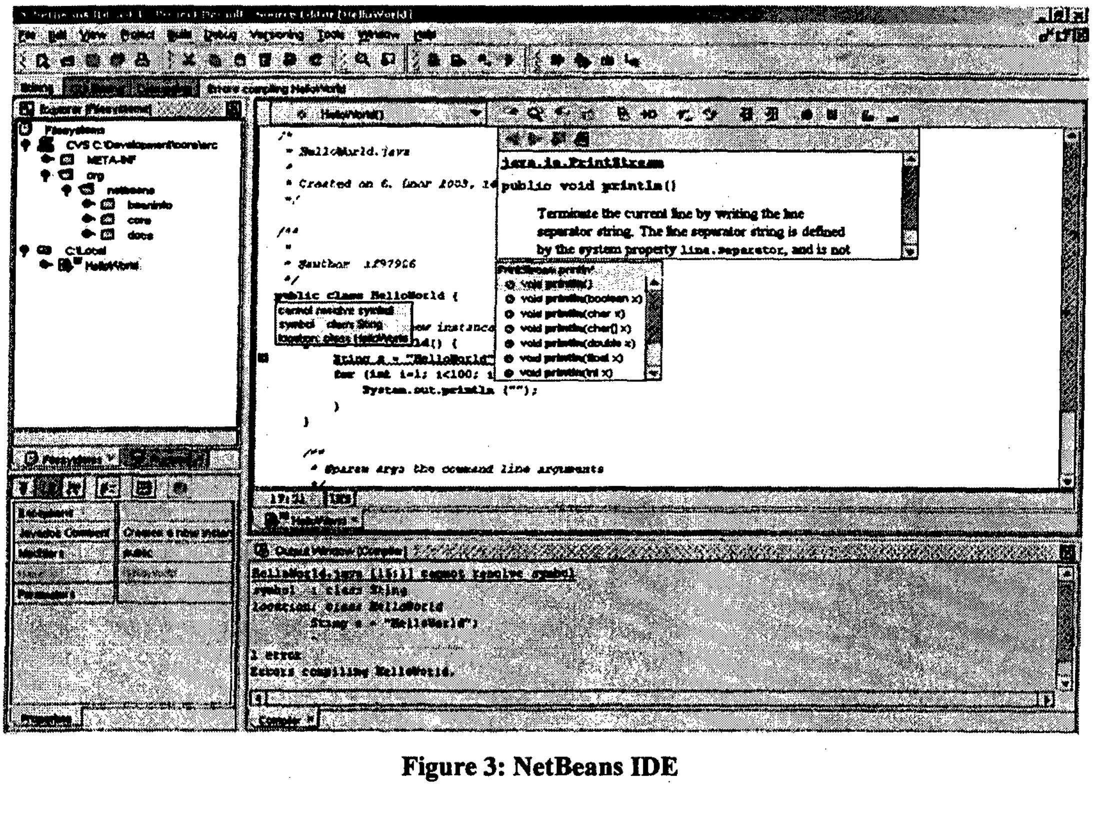
F.3 NetBeans IDE

F.3.1 Description

A world-class, professional IDE (Integrated Development Environment), the NetBeans IDE (Figure 3) has a platform (Figure 4) plus modules that include features such as an editor, tools for working with source code (Java, C++ and others), version control, and much more.

Other features include:

- Advanced syntax highlighting, error checking code editor
- Support for the Java, C, C++, XML and HTML languages
- Pluggable support for compilers, debuggers and execution services
- Support for JSP, XML, RMI, CORBA, JINI, JDBC and Servlet technologies
- Support for Ant, CVS and other version control systems
- Visual design tools
- Wizards and code generation and management tools



```
/*
 * HelloWorld.java
 *
 * Created on 6. [illegible] 2003, 1[illegible]
 */

/**
 *
 * @author [illegible]
 */
public class HelloWorld {
[illegible]
        String s = "HelloWorld";
        for (int i=1; i<100; i
            System.out.println ("");
        }
    }

    /**
     * @param args the command line arguments
     */
```

java.io.PrintStream

public void println()

Terminate the current line by writing the line separator string. The line separator string is defined by the system property line.separator, and is not

```
HelloWorld.java [15:1] cannot resolve symbol
symbol  : class Sting
location: class HelloWorld
        Sting s = "HelloWorld";
1 error
Errors compiling HelloWorld.
```

Figure 3: NetBeans IDE

F.3.2 NetBeans Platform

The NetBeans Platform is an application runtime - a "generic large desktop application." Most desktop applications have common requirements - menus, document management, settings and so forth. Nobody enjoys writing menu code or setting storage code. With the NetBeans Platform, you don't have to. Just write *modules* to implement what you need, bundle them up with the NetBeans Platform, and you have a beautiful, branded, cross-platform application. And if you need custom functionality or components, the Platform is built to be flexible.

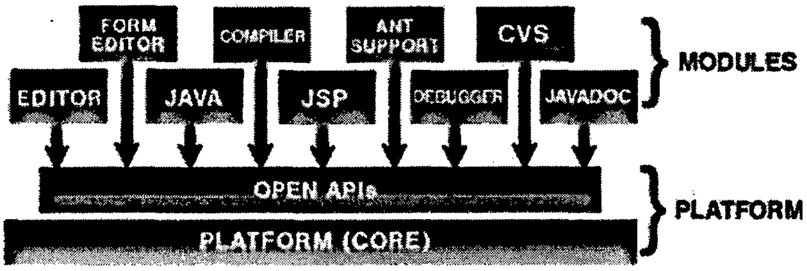


Figure 4: NetBeans Platform

With the NetBeans Platform, developers get to concentrate on the *important* parts of an application - the business logic that makes that application unique. The result is a huge savings in time and effort. Some of the features of the platform are:

- **User interface management** - Windows, menus, toolbars and other presentation components are provided by the Platform. Developers write to a set of abstractions such *actions* and *components*, saving time, and producing cleaner, more bug-free code. Custom components and behaviors can be written, but for most cases this is not needed.
- **Data and presentation management** - The NetBeans Platform contains a rich toolset for presentation data to the user and manipulating that data.
- **The Editor** - Available as an extension to the Platform, applications built on NetBeans can use the NetBeans Editor, a powerful and extensible toolset for building custom editors.
- **Setting management** - The NetBeans File systems infrastructure abstracts file-based data. Files may exist locally or remotely, on FTP or CVS servers or in a database; access to them is transparent to module code that works with files. The Platform can be extended to support new forms of storage. Applications built on NetBeans are internet-ready!
- **The Wizard framework** - a toolset for easily building extensible, user-friendly Wizards to guide users through more complex tasks.
- **Configuration management** - Rather than tediously write code to access remote data and manage and save user-configurable settings, etc., all of this is handled by the Platform. Applications consist of the platform and the logic code important to that application.
- **Storage management** - An abstraction of file-based data access. "Files" in the NetBeans paradigm may be local files, or exist remotely, for example, on an FTP server, CVS repository or in a database. Where this data is stored is completely transparent to other modules that work with this data.
- **Cross-platform** - since the Platform is written entirely in the Java language, applications based on it, by their very nature, will run on any operating system with a Java 2 compatible (1.3 or greater) JVM

See Figure 5 for examples of IDEs using NetBeans.

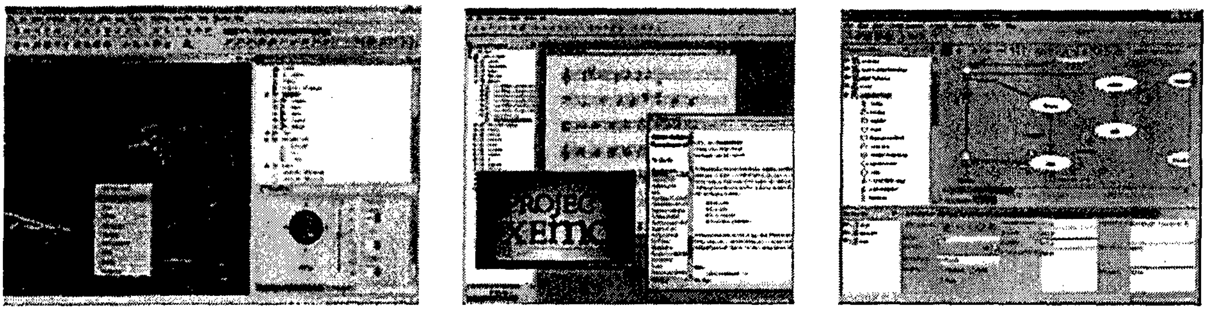


Figure 5: Examples of IDEs using NetBeans

F.3.3 NetBeans Features

- **Window management:** Rather than deal with the vagaries of window management in your own code, the NetBeans Platform handles this for you - you write component level objects that surface the user interface important to *your* functionality.
- **Action/Toolbar/Menu management:** Rather than write code to manage menus, simply write the logic that's important to you - what should *happen* when a user clicks on a menu item.
- **File/Data access:** NetBeans contains an incredibly powerful data access system. While it is file-based, it is not file-centric. Any data stored in a hierarchy can be "files" in the NetBeans paradigm. The Filesystems library provides a Service Provider Interface for plugging in new storage types. By writing to this interface, any existing module can work with that data as if it were files.
- **Wizard frameworks:** NetBeans provides a comprehensive framework for building extensible wizards, which can even be extended later by other modules added to the system. Again, the focus is on concentrating on business logic, so you don't need to reinvent the wheel.
- **Settings management and storage:** one of the most painful routine aspects of writing an application is user settings management - where and how to store them, when to persist them, how to manage multiple users of one copy of an application, etc. In NetBeans, it is enough to simply subclass the SystemOption class, and write your persisting settings as a standard JavaBeans component. Settings can be stored in a layered fashion, as global (all users), project-specific, or per-user.
- **Hierarchical data management and presentation:** The Platform contains a generic facility for the hierarchical management of data - the Nodes API - and a flexible presentation toolkit to present that data and allow users to interact with it: the Explorer API.

F.3.4 Features Available in Extension Modules

Applications using the NetBeans Platform can bundle any modules found on this site, which add additional services and functionality to the platform, such as:

- **AutoUpdate:** The AutoUpdate module allows applications based on the NetBeans Platform to query an update server (essentially an XML file served over HTTP) to determine if new versions of modules or new modules are available. Vendors of applications that use this module can deliver updates, fixes and new functionality without forcing users to download and install an entirely new distribution of the application. Applications may automatically check if there are new modules available, or this functionality can be explicitly invoked by the user.

- **Editor:** The NetBeans Editor contains powerful extensions to the Swing text APIs that make it easier to write a syntax highlighting code editor. The editor is capable of "mixed-mode" operation in which correct syntax highlighting, code-completion, formatting and macros are provided for documents that contain content in more than one language (such as a JSP page containing both HTML and Java code). Some of the features available via the editor are:
  - **Code completion** - providing a popup dialog to complete statements where those statements have a finite number of possible completions
  - **Annotations** - Lines of code can be marked with "annotations" such as debugger breakpoints or bookmarks that indicate status relating to them
  - **Abbreviations** - Abbreviations can be supplied which will be expanded to a longer string - for example, typing "sout[space]" in order to enter System.out.println("").
  - **CVS/Version control access:** The JavaCVS library is a stand-alone library that implements a Concurrent Versioning System [link to cvshome] client in Java. Also available is a module that integrates this support into Explorer.
  - **FTP support:** there is also support for files accessed via a File Transfer Protocol server. The fact that files may reside on a remote server is completely transparent to the user and to module code that interacts with them.
  - **Database support:** there are database support modules which support connecting to databases using JDBC and presenting their contents in the user interface
  - **Scripting support:** The Scripting module allows you to install arbitrary scripting languages and execute scripts in those languages within your application.
  - **Tomcat: Servlet and JSP support:** the NetBeans IDE uses this to allow users to view documentation in their web browser of choice; servlets can interact directly with the application environment and serve content from it.

F.3.5 Overall Evaluation

Flexible: NetBeans is a platform that was designed for customization and extension by developer to be used in tool development. Here is a good quote from the "NetBeans The Definitive Guide":

> ***"NetBeans is first and foremost a well-crafted open source programmer's integrated development environment (IDE). It's powerful, it's useful, it's extensible, it's open, and it's free."***

F.4 Other IDEs Evaluated

- Sun ONE Studio - Sun Microsystems (previously called Forte)
- JCreator – www.JCreator.com
- BlueJ - www.bluej.org
- RealJ - www.realj.org
- CodeGuide - www.omnicore.com

- Eclipse – www.eclipse.org (similar to NetBeans)
- JPad Pro - www.modelworks.com/editors/java.html
- Visual J++ - www.microsoft.com: Not really a good IDE since it only supports Java 1.1 and Below Plus, they are pushing C# (the Java clone)
- Microsoft .NET : Large learning curve

[redacted]

G GRAPHIC AND TABLE LIBRARIES EVALUATED

These libraries were evaluated for use in the IDE identified in Section E. AIE has several displays that will need to have the ability to create/edit graphics.

G.1 ILOG JViews Component Suite

ILOG JViews delivers unsurpassed displays, shortening Java development time, cutting costs, reducing risk -- and improving the user's experience.

Features, performance and control: Sophisticated diagrams. Map displays. Process control screens. Gantt views. 2D and 3D Charts. Whatever your display needs, JViews has the solution:

> **Functional features the others can't match:** Astound end-users with features that help them control displays and maximize decision-making. Synchronized displays, such as diagrams atop maps or Gantt charts. MVC architecture separates data model from displays. Prebuilt icons and styles. CSS-based styling engine. Both Java and thin Web clients.
>
> **Faster performance with hundreds of objects:** Architected for all problem sizes, JViews doesn't fall apart with big graphs or lots of objects. Triple buffering for lightning-fast redraws. Load-on-demand for large data sets. Binary I/O formats for speedy data loading.
>
> **Full control over look-and-feel:** Build the right user interface, adapting it to customer demands instead of predefined looks. Diagramming algorithms offer more parameters. Interactors can be subclassed to create new behaviors. Dedicated APIs for creating custom editors.

The premium suite for visualization: JViews is the premium suite for delivering high-end interfaces. Developers save time while building applications that offer unparalleled functionality -- best-of-breed interfaces that no other solution can build. Outstanding productivity lowers project risk and reduces maintenance headaches. That's why JViews has been the industry's best-selling visualization tool since 1997.

Packages for every application: JViews product packages address a broad range of visualization challenges in a single suite. Dedicated features help you build displays for monitoring, supervision, planning, analysis, designing and diagramming.

- Graph Layout package: Sophisticated diagrams.
- Gantt Chart package: Interactive scheduling and planning.
- Maps package: Asset-management displays.
- Chart packages: Rich, interactive plots.
- Graphics Framework package: Base component for graphics manipulation and rendering.
- Workflow package: Modeling and monitoring graphics

Telecommunications visualization: Build powerful telecommunications GUIs with ILOG JTGO, the first complete suite of Java graphic components for operations support systems (OSSs). ILOG JTGO enables rapid development of visualization layers for the new generation of highly flexible OSSs (see Figure 6).

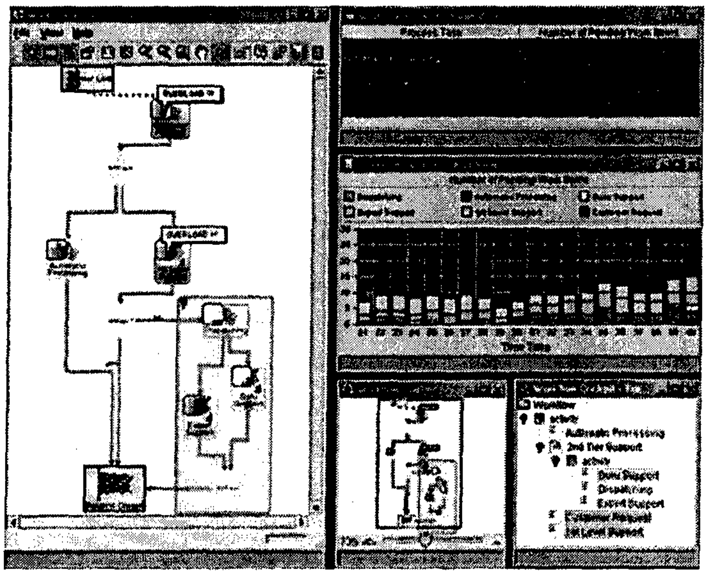


Figure 6: Example of JViews from ILog

G.1.1 Pros

- There are extensive APIs that are used to interface with the graphical libraries.
- There are six major packages, each with several different graphics.
- JViews has the most functionality of any COTS product.

G.1.2 Cons

- Not Open Source: Need an agreement with ILOG for them to open up the source code to ManTech Aegis Research.
- There would be a development and end-user License Fee.

G.2 The JGraph Swing Component

JGraph (pronounced "jay-graph") is a robust and complete graph component that is better than many of its commercial competitors. "JGraph is surprisingly small and supports drag and drop and all the selection modes and display/editing options you might expect. Enjoy!" (Sun, Swing Sightings). JGraph can also be used on the server-side, for example to read a GXL graph, apply a custom layout algorithm, and return the result as an SVG image (see Figure 7)

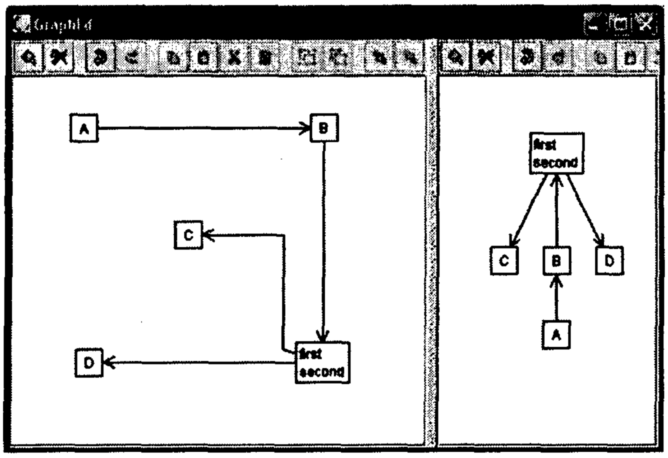


Figure 7: Example of JGraph

Features:

- 100% pure Java, fully standards-compliant developer API
- Automatic edge routing and handling of self-references
- Command history for infinite undo/redo across multiple views
- Each view can have its own set of visible cells and separate attributes
- Multiple connection points per vertex can float along the vertex boundaries
- Layers, Grouping and stepping-into groups for managing large diagrams
- Built-in marquee selection, zoom, grid, anti-aliasing, and in-place editing
- Copy and paste or drag and drop to and from Java and native applications
- Very small footprint (150 K) for embedded use
- Source code

The source code is placed under the GNU-copyright notice, and is available via anonymous CVS.

G.2.1 Pros

- Open Source, No End User License.

G.2.2 Cons

- JGraph has very limited functionality in comparison to JViews.

G.3 JFreeCharts

JFreeChart is a free Java class library for generating charts, see the example illustrated in Figure 8. It's charts include:

- pie charts
- line charts and horizontal/vertical bar charts (regular and stacked, with optional 3D-effect)
- XY plots and scatter plots

- time series, high/low/open/close charts and candle stick charts
- combination charts
- Pareto charts
- Gantt charts
- bubble charts
- wind plots, meter charts and symbol charts

JFreeChart includes many features:

- support for dual axes
- tooltips, zooming, printing and saving (PNG/JPEG/SVG/PDF)
- access data from any source that supports the defined interfaces (a similar mechanism to Swing's JTable)
- use JFreeChart in Java servlets, JSPs (thanks to Cewolf), applets or client applications
- complete source code included, under the terms of the GNU Lesser General Public License
- based on the Java 2D API, requires JDK 1.3 or later
- extensive documentation is available in Acrobat PDF format, in addition to the Javadocs

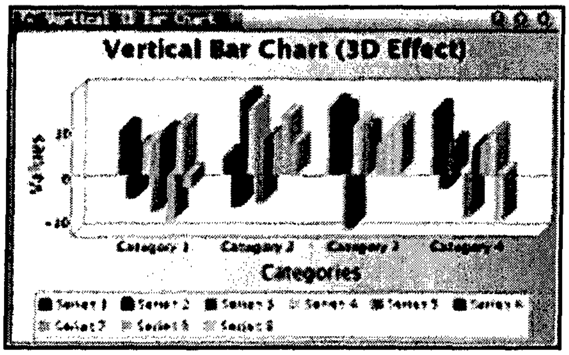


Figure 8: Example of JFreeChart

G.3.1 Pros

- Open Source, No End User License.

G.3.2 Cons

- JFreeChart is very limited in functionality in comparison to JViews.

[redacted]

H APPENDICES

These appendixes provide information for convenience in document maintenance.

H.1 System Development Process that AIE must Support

Each phase of the development effort is structured based on the CSEC's standard process for employing CSE in the development of decision support systems.

The process begins with the ACWA™ analysis, transitioning to the system design portions of ACWA™ then to software implementation, and then to test and evaluation. The second incremental spiral also is shown. This incremental spiral is a standard practice lesson learned: the revolutionary leap of an x.0 release typically misses many of the minor usability issues that become readily apparent as the release is tested. The x.1 incremental release allows 'quick response' to those obvious evaluation results to provide users a much more stable tool during the next major development spiral.

At the core of our innovative approach is the concept of Cognitive Systems Engineering (CSE): a comprehensive, robust and proven methodology to design and develop revolutionary decision support systems explicitly focused on users' decision-making processes. Beginning with the Applied Cognitive Work Analysis (ACWA™) process, a comprehensive description of the physical, as well as goal-oriented relations inherent to any specific problem domain (i.e., the cognitive reasoning process) forms the basis for the system design. A specially tailored software engineering process derived from OMT and UML then is used to implement the decision aids, either as final run-time systems, or as a partial prototype to support hands-on evaluation and testing. The software engineering process goes from design, through code/unit test, to integration testing, where conventional development typically ends. CSE adds one final step to close the design loop: Decision-Centered Testing (DCT), where the net decision-making effectiveness of the user-decision aid team is explicitly tested against the same ACWA™ analysis that formed the initial hypotheses for the system. (See Figure 9.)

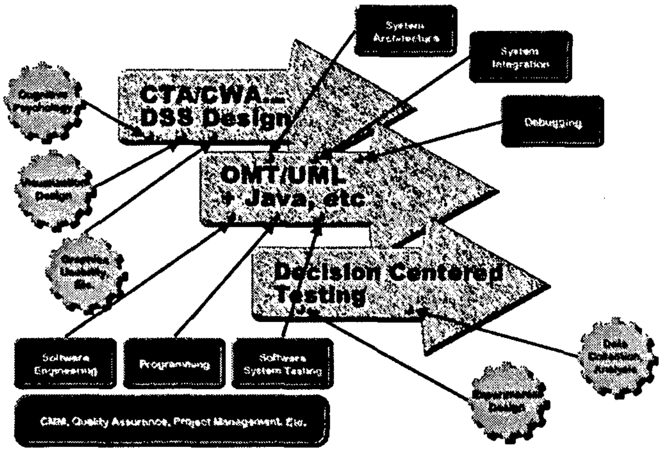


Figure 9: Innovative System Development Approach

ACWA™ was constructed as a series of relatively small manageable engineering transformations, each requiring the skilled application of the methodology's principles rather than requiring a design epiphany at any point in the process. At each intermediate point the resulting decision-centered artifacts create the spans of a design bridge that link the requirements of the domain as revealed by the cognitive analysis to the elements of the decision aid.

An integral part of the CSE process is the spiral software development model of Boehm (1988). Using this development concept, ManTech Aegis Research Corporation will develop incremental operational prototypes of an IW user interface to decompose complex application and thought processes into manageable proportions, thereby delivering increasingly robust solutions in steps, and integrating the "lessons learned" from earlier prototypes into the final solution.

This process allows the developers to return to a second cycle of design and development tasks as new design-related improvements are received from the previous spiral. In addition, it allows the developers to build continually on previous releases, while easily discarding those prototype design concepts that prove unsuccessful.

Our CSE methodology begins with a function-based, goal-means decomposition of the target domain. (See Figure 10.) This methodology has roots in the formal, analytic goal-means decomposition method pioneered by Rasmussen and his colleagues as a formalism for representing cognitive work domains as an abstraction hierarchy (Rasmussen, 1986). A work domain analysis is conducted to understand and document the goals to be achieved in the domain and the means available for achieving them. The objective of performing this analysis is to develop a structure that links the purpose(s) of individual controllable entities with the overall purpose of the system. This includes knowledge of the system's characteristics and the purposes or functions of the specific entities.

The functional model provides a framework for making explicit the goals to be achieved in the domain, as well as the alternative means available for achieving those goals. High-level goals, such as impacting a critical function, are decomposed into supporting lower-level sub-goals. Abstract "processes" are identified as the means to accomplish those goals. Together, these goals and processes form a network that provides the basis for identifying—through subsequent steps in the analysis and design process—the cognitive activities that arise in the domain and the information needed to support those decisions. This functional model provides a powerful structural representation of the decision space, forming the basis for subsequent analysis and design, as well as a definitive context for the ultimate information space presented to the users.

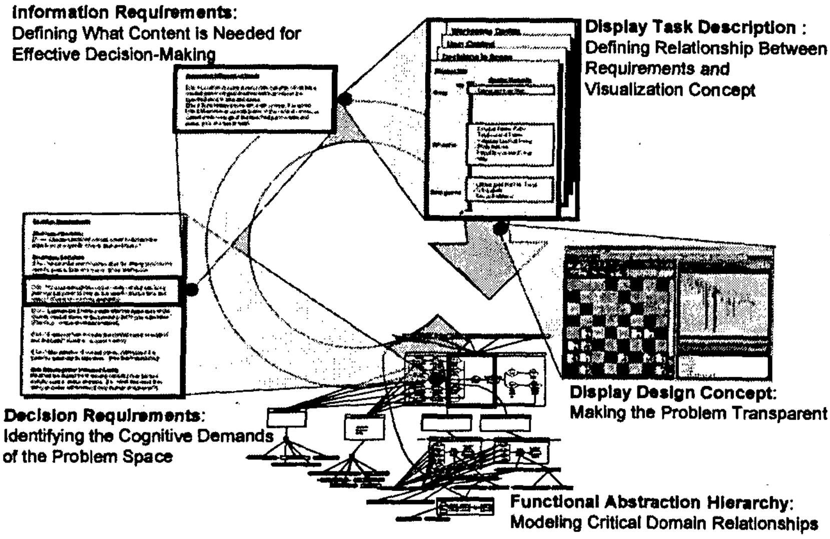

Information Requirements:
Defining What Content is Needed for Effective Decision-Making
Display Task Description :
Defining Relationship Between Requirements and Visualization Concept
Display Design Concept:
Making the Problem Transparent
Decision Requirements:
Identifying the Cognitive Demands of the Problem Space
Functional Abstraction Hierarchy:
Modeling Critical Domain Relationships

Figure 10: CSE Process

Using the functional model as the underlying framework, the next step in the CSE approach derives the cognitive demands associated with each functional node of the domain. These cognitive demands center on goal-directed behavior, such as monitoring for goal satisfaction and resource availability. By organizing the specification of cognitive demands around nodes in the functional model, rather than organizing requirements around predefined task sequences, the representation helps ensure a decision-centered perspective and not a list of the mechanical aspects of the work tasks.

Following the identification of the cognitive tasks, each "decision" is analyzed to identify and document the information required for each cognitive demand. The focus of this step in the methodology is on identifying the ideal and complete set of information necessary for effective decision-making. The information requirements identified in this step specify much more than specific data elements. These information requirements become the "challenge statement" for development of the applications layer of the system architecture, and its supporting data collection and management services, in addition to the obvious challenges to information presentation layers of the decision support environment's architecture.

From the set of information requirements, the next step is to design presentations and other aid concepts to reflect the information requirements. As a result of the linkage back to the functional model, the presentations are organized into a virtual "information space," explicitly replicating the domain structure captured in the initial analysis and reinforcing a mental model of the domain in any user of the system. From a design perspective, the goal is to reveal the critical information requirements and constraints of the decision task through the presentation space in such a way as to capitalize on the characteristics of human perception and cognition that intuitively convey answers to critical questions before they are explicitly asked.

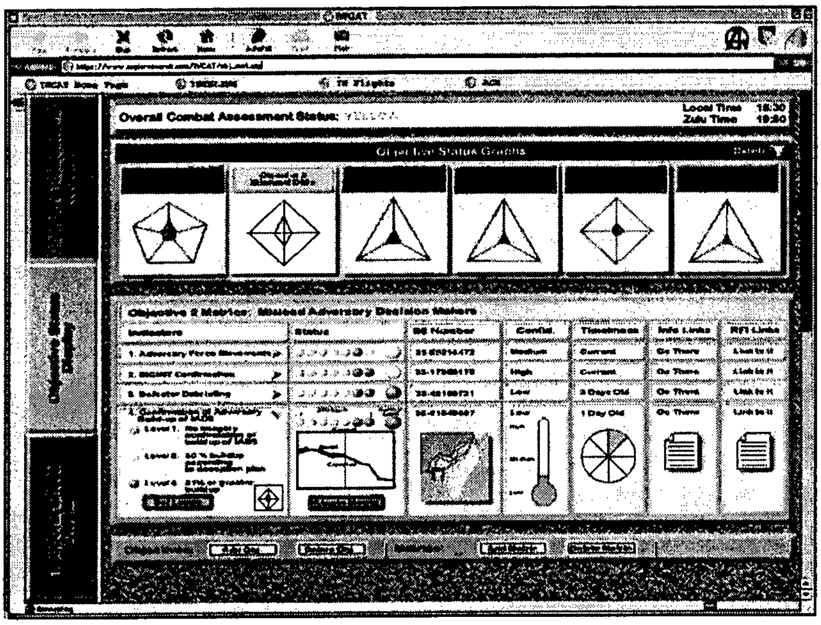


Figure 11: IWCAT Objective Status Display

The presentation concepts and how they support the cognitive demands of the domain will be captured in a Display Task Description (DTD). The DTD will define the goals and scope of a decision-aid display in terms of the cognitive demands it is intended to support (and thus its associated region of the functional model). Figure 11 is an example of a decision aid display designed during Phase II and implemented/released during Phase III of IWCAT. Explicit links are made between particular aspects of the decision-aid concepts and specific cognitive demands they are intended to support. The DTD is the key transition between the cognitive analyst, the software developer and the system tester, and can only be produced from a decision-centered CSE process.

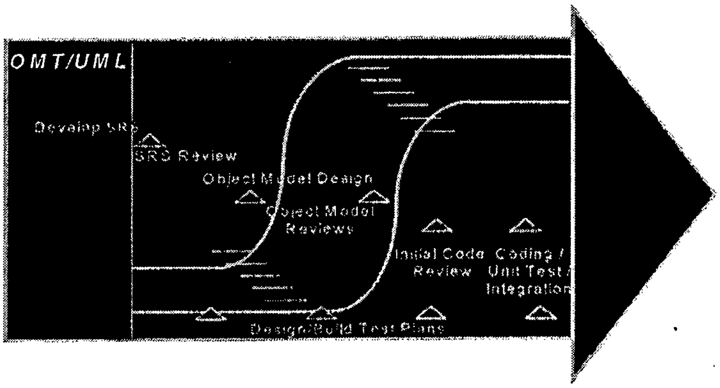


Figure 12: OMT/ULM Process

ManTech Aegis Research will employ state-of-the-practice, object-oriented software engineering methods, using characteristics of the SEI CMM Levels 2 and 3, specifically selected and tailored for their value to system development quality and productivity (see Figure 12). The software engineering team is skilled in a variety of design methodologies (e.g., OMT/UML) and implementation languages (e.g., C++ and Java) across a variety of platforms and architectures. We will employ a fully equipped suite of servers, layered third-party products, and design and development tool suites to ensure quality, efficiency, testability and maintainability of the software, as well as the program management aspects, such as configuration management. Our software developers are equipped with a variety of workstations and integration/test servers over a firewall-protected 100Mb/s, fully switched, internal development LAN. The software engineering practices of this team have been expressly tailored to work as part of the CSE process. This is a highly skilled team, trained and able to rapidly deliver even the most advanced decision- support systems.

This CSE-tailored, object- oriented approach produces a robust set of design artifacts, which support incremental design reviews throughout the process. It has been used for the development of several advanced decision support systems and is continuously revised as software engineering and associated support tools have evolved. The artifacts produced are unique in their content, and contain direct transition information from the ACWA™ analysis, which preceded the software development. The component tasks of the software development process are described in the following sections.

Decision-centered testing (DCT) is a fundamental element of the CSE approach to design of advanced decision aiding systems like IWCAT. The advanced decision aid systems design concepts developed embody hypotheses about what constitutes effective support in the domain. DCT provides feedback on the viability of the decision-aid design concepts (e.g., whether the hypothesized positive impacts of the IWCAT are realized), as well as feedback on additional unanticipated requirements for support.

DCT begins with identification of human performance issues to be supported by the system or visualization. The performance issues to be examined are probed using an understanding of individual and team cognitive factors, fundamental relationships within the domain, and known complicating factors to assess the impact of technology on human performance. The question DCT attempts to ask is, "Has a decision aiding system augmented cognition and performance?"

By using a DCT approach to evaluate decision-aiding systems like IWCAT, it is possible to develop a seamless design process that increases the understanding of the problem space, which in turn allows for improved criteria specification for bounding and optimizing the decision aid. The CSE methodology also establishes a framework for formulating hypotheses about basic work processes of the domain and points the way to further design innovation.

H.2 Acronym List/Integrated Dictionary

The following list of acronyms provides the foundation for the AIE Tool Dictionary, including the definition of all terms used. The Integrated Dictionary provides a central source for all these definitions and metadata, including those that may be provided for convenience within another product as well. At a minimum, the Integrated Dictionary is a glossary with definitions of terms used in the given architecture description. The Integrated Dictionary makes the set of architecture products stand-alone and allows it to be read and understood without reference to other documents.

ACTD	Advanced Concept Technology Demonstration
CDE	Common Desktop Environment
CIO	Cognitive Information Organizer
COE	Common Operating Environment
COTS	Commercial-off-the-Shelf
DAA	Designated Approval Authority
DII	Defense Information Infrastructure
DII COE	Defense Information Infrastructure Common Operating Environment
DoD	Department of Defense
DODD	DoD Doctrine
DoDIIS	DoD Intelligence Information System
FTP	File Transfer Protocol
GIS	Geographic Information System
GOTS	Government-off-the-Shelf
GUI	Graphical User Interface
HCI	Human Computer Interface
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
IO	Information Operations
IWPC	Information Operations Planning System
IP	Internet Protocol; Initial Point
JDK	Java Development Kit
JMTK	Joint Mapping Toolkit
LAN	Local Area Network
NTP	Network Time Protocol
SDK	Software Developer's Kit
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol
UNIX	Universal Computer Operating System
URL	Uniform Resource Locator