Framework Programmable Platform for the Advanced Software Development Workstation


Richard J. Mayer
Thomas M. Blinn
Paula S. deWitte
John W. Crump
Keith A. Ackley

Knowledge Based Systems, Inc.

April 16, 1992

Cooperative Agreement NCC 9-16
Research Activity No. SE.37

NASA Johnson Space Center
Information Systems Directorate
Information Technology Division

Research Institute for Computing and Information Systems
University of Houston-Clear Lake

TECHNICAL REPORT
Framework Programmable Platform for the Advanced Software Development Workstation

RICIS Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by Dr. Richard J. Mayer, Thomas M. Blinn, Dr. Paula S. deWitte, John W. Crump and Keith A. Ackley of Knowledge Based Systems, Inc. Dr. Charles McKay served as RICIS research coordinator.

Funding was provided by the Information Technology Division, Information Systems Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between the NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was Ernest M. Fridge, of the Software Technology Branch, Information Technology Division, Information Systems Directorate, NASA/JSC.

The views and conclusions contained in this report are those of the authors and should not be interpreted as representative of the official policies, either express or implied, of UHCL, RICIS, NASA or the United States Government.
Framework Programmable Platform for the Advanced Software Development Workstation

Demonstration Framework Document
Volume II: Framework Process Description

Produced For:
Software Technology Branch
NASA Johnson Space Center
Houston, TX 77058

Produced By:
Knowledge Based Systems, Inc.
2746 Longmire Drive
College Station, TX 77845-5424
(409) 696-7979

Dr. Paula S. deWitte, Thomas M. Blinn
Co-Principal Investigators

Under Subcontract to:
RICIS Program
University of Houston - Clear Lake
Houston, Texas 77058-1096
Subcontract Number 077:
Cooperative Agreement Number: NCC 9-16

December 14, 1991 - April 16, 1992
Framework Programmable Platform for the Advanced Software Development Workstation (FPP/ASDW)


Produced For:
Software Technology Branch
NASA Johnson Space Center
Houston, TX 77058

Authors:
Dr. Richard J. Mayer
Thomas M. Blinn
Dr. Paula S. deWitte
John W. Crump
Keith A. Ackley

Knowledge Based Systems, Inc.
2746 Longmire Drive
College Station, TX 77845-5424
(409) 696-7979

April 16, 1992
Introduction to Volume II

In this second volume of the Demonstration Framework Document, the graphical representation of the demonstration framework is given. This second document was created to facilitate the reading and comprehension of the demonstration framework. It is designed to be viewed in parallel with Section 4.2 of the first volume to help give a picture of the relationships between the UOBs of the model. The model is quite large and the design team felt that this form of presentation would make it easier for the reader to get a feel for the processes described in this document. The following pages contain the IDEF3 diagrams of the processes of an Information System Development. Volume I describes the processes and the agents involved with each process, while this volume graphically shows the precedence relationships among the processes. Figure 1 illustrates the parts of an IDEF3 Description.

![Figure 1 Reading an IDEF3 Diagram](image)

One of the primary mechanisms used for communicating information about a situation is to describe an ordered sequence of events or activities. The IDEF3 Process Flow Description Capture Method was developed to provide a mechanism for collecting and documenting processes. IDEF3 captures precedence and causality relations between situations and events in a form that is natural to domain experts.

The basic syntactic unit of IDEF3 is the unit of behavior (UOB). A UOB can be a function, activity, action, act, process, operation, event, scenario, decision, or procedure. UOBs can have decompositions and elaborations. Decompositions are associated descriptions in terms of other UOBs. As shown in Figure 1, a UOB which has a decomposition is drawn with a shadow box. Those UOBs that are drawn without a shadow have no decomposition.

UOBs are connected through the use of junctions and links. Junctions provide semantic mechanisms for representing the convergence and
divergence of process flows within a network of UOBs. The types of
junctions are 'and', 'or', and 'exclusive or', after the logical operators.
Junctions can be synchronous or asynchronous, which is delineated by the
number of vertical bars. Synchronous junctions have two vertical bars,
whereas, asynchronous junctions have one. Fan in and fan out junctions
are indicated by the location of the dot.

UOBs are numbered according to their position, reading from left to right,
top to bottom. As one goes down into decompositions, the parent's numbers
are retained and the children's numbers are appended separated by a
period. Thus, the numbering process is recursive.

This document is arranged with the diagrams ordered depth first. That is,
the top level diagram is followed by the first UOBs decomposition. This then
is followed by it's first UOBs decomposition, and so on until no more
decompositions. As with the numbering scheme, the arrangement of the
UOBs in a single decomposition is done left to right, top to bottom. After
exploring the diagrams in the following pages, the pattern should be
recognized easily.
Perform IS Concept & Initiation

Develop Info. Sys. Requirements

Design IS

Coordinate IS Implementation

Integrate & Test IS Components

SMAP Information System Life Cycle Process
Define Assurance Process Requirements
1.4.1

Define Assurance Plan
1.4.2

Document Assurance Plan
1.4.3

Decomposition of Define Assurance Strategy
Define V&V Approach

Define V&V Methods

Document in V&V Plan

2.1.6.2.1

2.1.6.2.2

2.1.6.2.3

Decomposition of Procure Independent V&V
Decomposition of Define Development Processes

1. Define New Procedures & Standards
   - 2.3.1

2. Define Sustaining Engineering Processes
   - 2.3.2

3. Identify Approach
   - 2.3.3

4. Define Methods for Activities
   - 2.3.4

5. NOE
   - 2.3.5

6. Define Incremental Development Processes
   - 2.3.6
Decomposition of Manage Design Phase
Initiate Identification Process

3.1.4.1

Initiate Evaluation Process

3.1.4.2

Initiate Selection Process

3.1.4.3
Develop Requirements Traceability

Document Requirements Traceability

Partition Design into Increments

Initiate Integration Test Procedures

Document Integration Tests

3.2.3.1

3.2.3.2

3.2.3.3

3.2.3.4

3.2.3.5

Decomposition of Allocate Requirements To Subsystems
Coordinate Interaction & Implementation of Components

Decide Whether To Proceed

&
Decomposition of Integrate Subsystems & Components

- Test Subsystems & Components
  - $\S.3.1$
- Integrate Next lower-level Component
  - $\S.3.2$
- Prepare Documentation
  - $\S.3.3$
Conduct Verification Of Product vs Design Specs

Prepare for Validation

Document V&V Activities

5.4.1

5.4.2

5.4.3
Decomposition of Deliver IS
Deliver IS process

Assign Responsibility for Change Proposals

Assign Responsibility for Deficiency Reports

Assign Responsibility for Discrepancy Reports

Generate Version Description Document

Generate User's Guide

Perform User Training

 Decomposition of Deliver IS
Decomposition of Conduct Formal Testing

- Conduct Testing of System (6.2.1)
- Analyze Acceptance Test Results (6.2.2)
- Conduct Acceptance Review (6.2.3)
- Document Acceptance Review (6.2.4)
- Perform V&V Activity (6.2.5)
Decomposition of Maintain IS

- Collect Metric Information
  - Document Metric Information
    - Document Discrepancy Reports
      - Evaluate Discrepancy Reports
      - Document Change Proposal Report
    - Evaluate Change Proposals
      - Document Change Proposal Report
Decomposition of Maintain IS

- Repeat Maintenance Activities
- Assign Resolution Responsibility 7.17
- Conduct User and Operations Training & Support 7.18