Framework Programmable Platform
for the Advanced Software
Development Workstation

Demonstration Framework
Document Volume II:
Framework Process Description

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
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 (FPP/ASDW)

Demonstration Framework Document
Volume II: Framework Process Description

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.

![Diagram of IDEF3 Description](image)

**Figure 1 Reading an IDEF3 Diagram.**

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
Define Assurance Strategy

1.4

&

Define System Concept & Scope

1.6

Document Results

1.7

Develop Management Strategy & Constraints

1.5
Decomposition of Conduct Verification Activities

- Define Validation Specifications
  - 2.1.6.1

- Develop Verification Activities
  - 2.1.6.3

- Document Expected V&V Results
  - 2.1.6.4

- Procure Independent V&V
  - 2.1.6.2
Define V&V Approach

Define V&V Methods

Document in V&V Plan
Decomposition of Define New Procedures & Standards

- Document New Procedures and Standards
  - 2.3.1.3

- &

- Develop Standards
  - 2.3.1.1

- &

- Develop Procedures
  - 2.3.1.2

- &
Define Specific Increments
2.3.6.1

Prioritize Approach
2.3.6.2

Document Incremental Processes
2.3.6.3
Partition Requirements into Increments
2.48

Review Requirements Changes
2.49
Decomposition of Conduct Engineering Design

Conduct V&V Activities

Allocate Requirements To Subsystems

Define System Architecture

Define Interface
Conduct Risk & Management Control Activities

Manage Coordination Phase

Prepare for Integration Test Activities

Review Design of Subsystems and Components

Initiate Subsystem Lifecycle

Decomposition of Coordinate IS Implementation
Coordinate Interaction & Implementation of Components

Decide Whether To Proceed

&
Collect Metric Information

Document Metric Information

Assign Resolution Responsibility

4.1.1

4.1.2

4.1.3
Decomposition of Manage Coordination Phase
Decomposition of Prepare for Integration Test Activities
Decomposition of Conduct Risk & Management Control Activities
Decomposition of Integrate Subsystems & Components

- Test Subsystems & Components
  - §3.1
- Integrate Next lower-level Component
  - §3.2
- Prepare Documentation
  - §3.3
Collect Metric Information

Document Metric Information

Evaluate Metric Information

Re-evaluate Risk Areas

Document Changes to Plans

Decomposition of Manage Deliver IS Phase
Conduct Verification Activities

Conduct Validation Testing

Document V&V Findings

6.2.5.1

6.2.5.2

6.2.5.3
Decomposition of Conduct Configuration Audits
Assign Resolution Responsibility

7.17

Conduct User and Operations Training & Support

7.18

Repeat Maintenance Activities