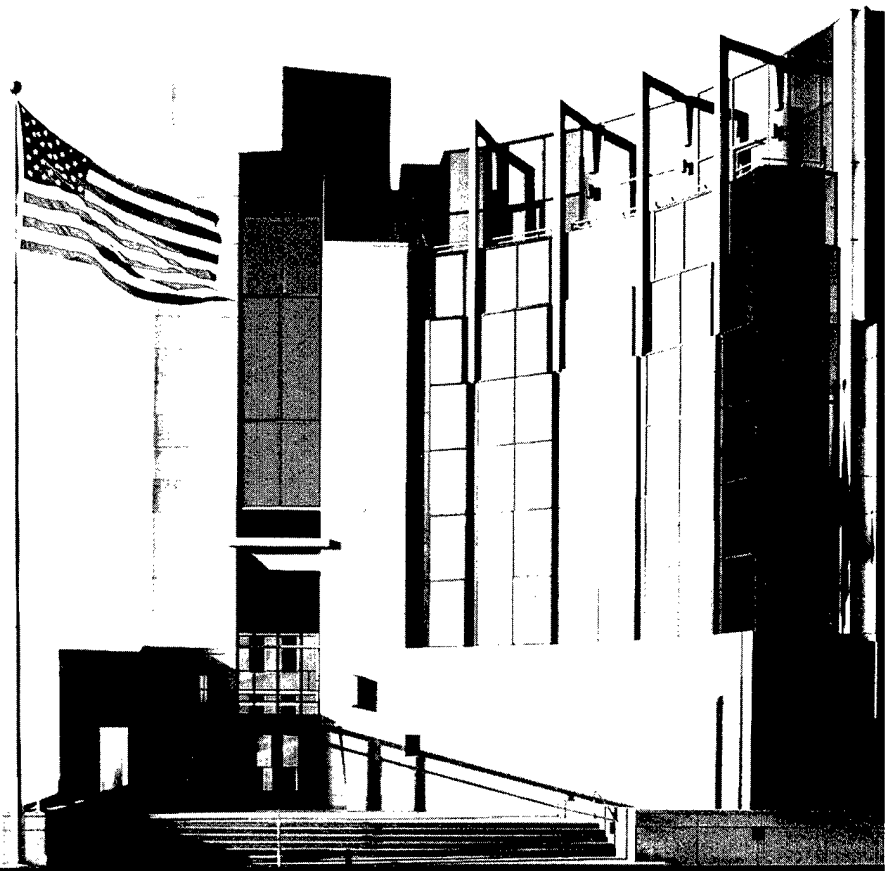
Carnegie Mellon
Software Engineering Institute

Framework Document: Model-Based Verification Pilot Study

David P. Gluch
John J. Hudak
Robert Janousek
John Walker
Charles B. Weinstock
Dave Zubrow

October 2001



CMU/SEI-2001-SR-024
SPECIAL REPORT

Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of "Don't ask, don't tell, don't pursue" excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

Framework Document: Model-Based Verification Pilot Study

CMU/SEI-2001-SR-024

David P. Gluch
John J. Hudak
Robert Janousek
John Walker
Charles B. Weinstock
Dave Zubrow

October 2001

Dependable Systems Upgrade Initiative

Table of Contents

Abstract

This Pilot Study Framework document describes the processes, activities, artifacts, and deliverables associated with an Engineering Practice Investigation that applies Model-Based Verification (MBV).

1 Description

This Engineering Practice Investigation is a structured pilot study applying model-based verification techniques to a software development project. The investigation is conducted in parallel or "shadowed" with the project's design and development efforts.

This study has two goals:

- measuring the effort involved and the benefits obtained in the application of model-based verification techniques
- identifying technical and engineering practice issues that must be addressed in order to facilitate the transition of model-based verification techniques into routine practice

There are three distinct phases to the engineering investigation:

- *Planning the pilot study.* This includes specifying goals for the pilot, developing its procedures, obtaining resources to conduct it, etc., as well as identifying the data to be collected. Data to be collected include a defined core set of measures to characterize the costs (e.g., effort) and benefits (e.g., estimated savings in effort associated with identifying defects and rework avoided) of applying MBV. Qualitative data on engineering judgments and issues must also be collected. These data will support the eventual goal of inserting model-based verification technology into the engineering practice as a component of an Independent validation and verification (IV&V) process.
- *Execution phase.* This phase will consist of the execution of the defined process including data collection and analysis.
- *Post-mortem.* This phase will consist of a review and critique of the study process, the documentation and analysis of the engineering results, technical problems encountered, and research issues that have been identified as a result of the investigation.

The key issues to be evaluated in this study and the data needed to address them include

1. transition and adoption costs. Address skill level, time, resources, and training time of engineering personnel.
2. discovered defects and their classification. Review and analyze the defect data relative to the phases of the software lifecycle.
3. programmatic return on investment (ROI). Provide a cost benefit analysis of transition and adoption costs and benefits including estimated rework costs avoided.
4. software engineering practice improvement. Based on evidence provided from this study, compose guidelines and recommendations to enhance the development process, as well as for the use of MBV in the development lifecycle, specifically as an (IV&V) activity.

Defect identification will be performed by applying MBV to an existing set of software requirements and related design specifications. A team of engineers will

- read the requirements
- construct state representations of the requirements at various levels of abstraction
- transform the state models into a mathematical representation appropriate for model checking
- analyze the models using automated model checking tools and claims about expected system behavior

Defects are discovered when the results of building and analyzing models do not support the stated requirements. Correlation and valuation of defects is accomplished by taking the defect set uncovered in the study and comparing it to defects found in the transformation of the requirements through the design, implementation, and testing phases of the software development lifecycle. Effort to fix the defects can be used to determine the value of finding them earlier using MBV. If actual effort data are not available, valuation of the earlier discovery of defects can be estimated by noting the differences in the phase of discovery.

It is important in this investigation to define a process that will produce the data to support the metrics. A description of the process and metrics to be used in this study are outlined in the following sections.

2 Study Plan and Activities

This section provides a high-level view of the overall study. The following are key activities with associated milestones or goals:

A team will be selected to conduct the study. We expect the team to consist of three SEI engineers.

The team will perform a top-level review of the system and software specifications and associated material in order to become familiar with the domain.

A briefing by development project personnel will be given to the SEI team to help in understanding the problem domain and provide insight into areas that have been most troublesome.

The SEI engineers will hold weekly status meetings. In general, the goals of the meeting will be to provide a status with respect to progress; clarification of team or individual understandings of technical or procedural issues; problem identification and rectification; and ensuring compliance of all team members to the pilot study process.

Impromptu meetings will be held as needed. Minutes will be kept for all team meetings.

The Pilot Study will be carried out in three "cycles." A cycle is composed of a) reviewing and understanding the assigned specification b) building the models c) analyzing the models, and d) assessing and documenting the results.

In every cycle, each engineer will be assigned a unique section of the specification to analyze for defects.

All defects found by each engineer will be presented at the conclusion of each cycle. This information may be used to revise the methodology for choosing which sections of the specification will be studied in the following cycle.

Each team member will keep individual Activity, Defect, and Project Logs that will be submitted to the metrics engineer on a weekly basis.

Each engineer will also keep an Observation Log to capture relevant issues, insights, etc., associated with the procedure or technical activities.

A final report will be generated.

3 Metrics Summary

Metrics for this effort are obtained through the daily completion of logs. Four types of logs will be kept and used in analysis:

1. Activities log –a record of the duration of time spent on a specific pilot activity. This information will be used to characterize the amount of effort associated with the activities used to implement MBV.
2. Defect log –a description of the defect and the activity during which it was discovered. This information will be used to estimate the benefits of implementing MBV and the defect profile associated with the various defect discovery activities.
3. Observation log –a repository of observations related to both the pilot study process as well as the MBV technical issues. This will be used to construct implementation guidelines and lessons learned for transition into established verification practices, as well as refinement of the engineering process of MBV.
4. Project log –information that will be used to capture attributes of the context within which the pilot study was carried out.

The metrics will be analyzed periodically throughout the study and summary results will be made available to the sponsor at the end of the project.

4 Deliverables

A final report will be submitted to the sponsor focusing on the following key areas: return on investment, procedural and technical issues surrounding the use of MBV associated with this pilot study; and recommendations to implement MBV.

Appendix (Procedure Manual)

As referenced earlier in this document, a draft version of the "Pilot Study Procedure Manual" is included in the Appendix. This is envisioned as a living document throughout the study, with reviews and updates being made as needed.

Pilot Study Procedure Manual

Version 1.1 10/25/00

Revision Log

This is the Revision Log associated with this document

Date	By	Revision	Changes
Aug. 15, 2000	dpg/jjh	v3.0	Initial release
Oct. 4, 2000	jjh	v3.1	Added 'Type' category to Observation Log, clarified Time Log procedure, misc. edits.
Oct 10, 2000	jjh	v3.2	Activities Log categories changed per DZ comments
Oct 16, 2000	jjh	v3.3	Modified Observation Log categories based on insight from previous study. Team misc. edits.
October 25, 2000	jjh/cd	v3.4	Refined key issues and Study Plan and Activities. Level 1 edit for release to customer.
January 8, 2001	cd	v3.5	Performed Level 3 edit and refined formatting as required for unlimited distribution.
February 15, 2001	jjh/cd	v3.6	Final review and edits by team
March 30, 2001	cd	v3.7	Final edit

Introduction

Overview

This manual provides project operational guidance and materials for a software engineer who is participating in the model-based verification pilot study. It is a procedural document addressing data recording and routine actions to be carried out within the pilot study.

Objectives

The objective of a model-based verification pilot study is to gain insight into the costs, efficacy, and problems associated with the MBV practice.

Responsibilities of a Participant

The responsibilities of a participant in the pilot include

- participating as a team member to help define and improve the MBV process
- identifying defects in software artifacts
- capturing defect activity and time data
- recording engineering and process observations
- providing other support as needed

Global Procedures

Description

The team of engineers conducting the study at the SEI will have a wide range of expertise in software engineering. A subset of engineers on the team will actually perform the MBV technical work.

The team will perform a quick review of the materials supplied by the developer of the system.

The engineers will be given specific areas of the artifacts to focus on, guided by suggestions from the system developers as to which areas have been most troublesome and warrant additional review.

Note: In this document, the term "specification" will be used to represent all documents in the software creation process that are made available to the team. Typically this includes the software requirements specification (SRS) and various design documents.

1. **Project activities will entail the following:** Weekly status meetings will be held to discuss progress and problems that are uncovered in the process used in the Pilot Study or in MBV itself. These meetings will include a walk-through of issues uncovered and suggestions for their resolution. Questions about the specification or general readability errors should be discussed among team members. Engineers will share their knowledge about acronyms and in general help each other acquire the domain knowledge and learn the meaning of the specification. This will enable the team to choose which area of the specification should receive the most attention. Once the specification has been divided into sections, team members may discuss the specifics of their section (including any information about defects found) with other team members active in the defect search process. A review of each log category and types will also be included to help ensure the correct artifacts, related to the project measurement goals, are being recorded.
2. Impromptu meetings will be held as needed. Minutes will be kept for all team meetings. These minutes will be included in the project report at the end of the Pilot Study.
3. The Pilot Study will be carried out in three "cycles." A cycle consists of these activities: a) reviewing and understanding the assigned specification b) building the verification models c) exercising the model, and d) analyzing and documenting the results. In every cycle, each engineer will be assigned a unique section of the specification to analyze. The document section should be sized so that a complete modeling and analysis activity can be completed in approximately two to three months.
4. Available defect information about the system will be maintained by a metrics engineer at the SEI. This person is also a member of the pilot study team. He will not disclose to any other members of this Pilot Study any information about, or in reference to 1) the number of defects previously found

2) their location 3) their type, or 4) the manner in which they were found. Only after the last cycle has been completed will this information be shared among the team members.

5. On a weekly basis engineers will provide the metrics engineer with a copy of their logs (Activity, Defect, Observation, and Project) via email. The metrics engineer will use this information to prepare status reports as to the efficacy of the Pilot Study.

6. At the conclusion of each cycle, each engineer will present to the SEI-MBV team a review of their logs (all) in the fashion of a walk-through. The knowledge shared during these meetings may be used to revise the methodology in the appropriate areas (i.e., modeling methods, tools, specification partitioning, etc.) for the next cycle.

   For the first cycle, all engineers will use either Symbolic Model Verifier (SMV) or Software Cost Reduction (SCR) as a tool to support their modeling and analysis activities.

   These procedures shall be reviewed and may be changed at the end of each cycle.

Individual Procedures

Description

Individuals will be responsible for particular portions of the specification. They will learn as much as they can about the portion assigned to them. Using that knowledge and information obtained from the briefing by the system designers, they will develop their own models of the essential properties of the subsystems for which they are responsible. The models will be exercised with various claims, and the results noted and analyzed.

Each individual will be responsible for presenting status, and/or discussing issues at the weekly status meetings.

Each individual will be responsible for keeping an engineering log of all activities in soft copy form (i.e., spreadsheet file). He or she will also track the information required by the activity log as discussed in the next section, including any engineering observations, etc.

Logs

Description

There are four logs that must be maintained:

- Activity (time) Log
- Defect Log
- Observation Log
- Project log

Activity Log

The activity log is the place for recording the activities performed and the time spent on each activity. This should be kept as a sequential log of an individual's activities on the project. The individual's activity log will be submitted on a weekly basis. Periodically, the project manager will review the activity logging procedures to identify problems and discuss improvements. Any questions regarding the filling out of the activity log should be discussed with the local project manager as promptly as possible. All sections of the activity log should be filled out for each entry. The activity log can be either an electronic activity log (i.e., via Excel) or a hard copy version.

Recording Time

As an individual begins to work on an activity, the start time should be noted. If an activity changes, e.g., goes from learning the system to modeling the system, the termination time for the earlier activity should be logged. The new activity is then recorded on a separate line of the activity log. If a person forgets to note the start or end time for an activity, an estimate of its duration should be entered. There should never be more than one activity code in each of the log entries. Only one activity code per entry should be entered. If multiple activities were involved, the logger should either determine which one was dominant and use that code, or prorate the allocated time and make two entries. Also, it should be noted in the Comment section as to whether the work performed is unique to the project (e.g., learning the domain), generic for the MBV activity (e.g., building models), or work intended to define or enhance the MBV practice (e.g., developing MBV guidance or tools).

Activity Types and Codes

The activity types are listed below.

*** Producing Project Documents
PD – 1 (DR) Create and Maintain Domain Knowledge Repository
PD – 2 (SS) Write Statement of Scope and Formalism
PD – 3 (SP) Write Statement of Perspective

*** Model Development and Analysis
MD – 1 (SF0) System Familiarization
MD – 2 (SF) Detailed System Familiarization
MD – 3 (CM) Create Model
MD – 4 (RM) Revise Model
MD – 5 (CC) Create Claims
MD – 6 (RC) Revise Claims
MD – 7 (AIR) Analyzing and Interpreting Results
MD – 8 (DAG) Defect Analysis Group Meeting
MD – 9 (MT) Project Meetings

*** Indirect and Other
IO – 1 (MA) Learning the Modeling Approach
IO – 2 (LT) Learning a new additional tool
IO – 3 (DT) Develop Tools and Techniques
IO – 4 (WM) Writing MBV Documentation
IO – 5 (PU) Plan/Create/Update Forms
IO – 6 (SME) Setup/maintain equipment
IO – 7 (OH) Overhead
IO – 8 (OT) Other
IO – 9 (MR) Misc. Reading
IO – 10 (OM) Other Meetings

Activity Log Example

Activity	Date	Start	End	Duration	Domain Expert	Comment/ Description
MD-2 SF	8-20-99	9:30	11:00	1:30	No	Review of communications subsystem spec
MD-2 SF	8-25-99	1:00	4:00	3:00	Yes	Discussing system architecture with the DE to clarify understanding

Defect Log

The defect log used in this study is a basic one. It will only be used to track defects found in the specification. It requires the recording of all of the following:

- defect ID
- date
- activity (only one per defect, the dominant activity for that discovery)
- type
- location in the specification
- comments and description

Every section of the log should be filled out for each defect.

Defect ID

Defects will beidentified in the following manner:

Initials of engineer - cycle defect was found - defect number

Each engineer will assign the defect numbers sequentially, starting with one and continuing up to the number of defects found in that cycle.

For example **<dpg-2-1>** would be interpreted as "Dave Gluch, during the second cycle of the Pilot Study, found defect one."

Defect Types

The list of Defect Types (for specifications) and their associated descriptions can be found in the Log Template section of this guide.

Observation Log

The goal of the observation log activity is to help define and improve the MBV practice. This is a place to record engineering and process observations and issues. It is intended to help capture activity rationale, engineering choices and decisions, and also any difficulties encountered/errors made while doing model-based verification.

The Observation Log contains the following:

- observation – a description of any insights, anomalies, issues, etc., that the engineer has observed in this particular activity
- date – date of the observation
- activity – the dominant activity for that discovery (only one per observation)
- type – area to which the observation is related. See list below.
- comments – recommendations, solutions, ideas, etc., relevant to the observation

Users should make every possible effort to note their observations during their tenure with the project. If it is unclear what type of observation is being entered, the Type should be left empty. The user should try to be as descriptive as possible with the observations, especially where the Type is left blank. This will help in future assessments of the observations log.

Observation Type Codes

The observation types are

Tool implementation (TI)	(i.e., NuSMV, SCR, etc.) cryptic, ease of use, available doc, slow, memory hog, documentation quality, etc.
Tool paradigm (TP)	how appropriate the tool model paradigm is with respect to the modeling paradigm being used in the project
Modeling paradigm (MP)	how the modeling paradigm (i.e., state charts, etc) fits with the application area
Claim development (CD)	level of difficulty to construct, insights gained
Project administration (PA)	usefulness of meetings, coordination, information exchange, etc.
Domain knowledge (DK)	how much is needed, effects in construction model (example of point of confusion), methods used to understand system
Documentation (D)	quality of a specific doc (e.g., specs), information content, testable (spec related), verifiable (spec related), etc.

Project Log

There is one project log that will consist of three parts:

1. a description of the salient characteristics of the artifact under review
2. a summary of the software development process that was used by the original system designers
3. a summary of major results and comments gleaned from the logs kept by the individual participants

The salient characteristics of the artifact under review will include

- a prose description of the artifact (including type, i.e., requirements, design, architecture, code, etc.)
- the size of the artifact, measured in pages
- the density of information on the pages (informally characterized)
- sample pages as appropriate
- the application domain
- the development technology used in the artifact, including tools, methodology (e.g., object-oriented design), etc.
- any other characteristics of the artifact deemed to be of interest

The summary of the development process will include minutes from all of the team and joint meetings held, in order to help track the decision making process.

Log Templates

Activity Log

Name: ______________________________

Week: ______________________________

Pilot Study ___________ Other_________

Activity	Date	Start	End	Duration	Domain Expert	Comment/Description

Defect Recording Log

Name: _____

Week: _____

Reference: _____

Activity	Date	ID	Type	Page	Section	Line

Description: _____

Activity	Date	ID	Type	Page	Section	Line

Description: _____

Activity	Date	ID	Type	Page	Section	Line

Description: _____

Activity	Date	ID	Type	Page	Section	Line

Description: _____

Defect Recording Log Instructions [1]

Purpose	- Use this form to hold data on the defects you find and correct. - Keep a separate log for each project cycle.
General	- Record each defect separately and completely. - If you need additional space, use another copy of the form.
Header	Enter - your name - the Monday date of the week - reference the pilot study as source of defect information
Activity	Enter the activity type you were engaged in when you found the defect. Use the same abbreviations used on the Activity log.
Date	Enter the date the defect was discovered.
ID	Enter the defect identifier, in the format below: *Initials of engineer - cycle defect was found - defect number*
Type	Enter the defect type from the defect type list summarized in the top left corner of the log form. Use your best judgment in selecting which type applies.
Page	Enter the page on which the defect is located.
Section	Enter the section in which the defect is located.
Line	Enter the line on which the defect is located.
Description	Write a succinct description of the defect that is clear enough to later remind you about the error.

[1] Adapted from *Introduction to the Team Software Process*. Watts Humphrey with support by James W. Over. CopyrightWatts Humphrey, 1998.

Defect Types for Specifications[2]

1x Logic

10 Forgotten cases or steps
11 Duplicate logic
12 Extreme conditions neglected
13 Unnecessary function
14 Misinterpretation
15 Missing condition test
16 Checking wrong variable
17 Iterating loop incorrectly

2x Computational problem

20 Equation insufficient or incorrect
21 Precision loss
22 Sign convention fault

3x Interface/Timing problem

30 Interrupts handled incorrectly
31 I/O timing incorrect
32 Subroutine/module mismatch
(wrong or nonexistent subroutine called, or call is formatted incorrectly)

4x Data Handling problem

40 Initialized data incorrectly
41 Accessed or stored data incorrectly (wrong flag/index, (un)packed incorrectly, reference out of bounds or wrong variable)
42 Scaling or units of data incorrect
43 Dimensioned data incorrectly (wrong variable type)
44 Scope of data incorrect

5x Data problem

50 Sensor data incorrect or missing
51 Operator data incorrect or missing
52 Embedded data in tables incorrect or missing
53 External data incorrect or missing
54 Output data incorrect or missing
55 Input data incorrect or missing

6x Documentation Problem

60 Ambiguous statement
61 Incomplete item
62 Incorrect item
63 Missing item
64 Conflicting items

---

[2] Based on fault types in *Software Metrics: a rigorous and practical approach*. Norman E. Fenton and Shari Lawrence Pfleeger. 2nd ed., London: International Thomson Computer Press; Boston: PWS Pub., 1997.

65 Redundant items
66 Confusing items
67 Illogical item
68 Non-verifiable item
69 Unachievable item

7x Document Quality Problems

70 Applicable standards not met
71 Not traceable
72 Not current
73 Inconsistencies
74 Incomplete
75 No identification

8x Enhancement

80 Change in program requirements (add new, remove unnecessary, update current capability)
81 Improve comments
82 Improve Code efficiency
83 Implement editorial changes
84 Improve usability
85 Software fix of a hardware problem
86 Other enhancement

9x Failure caused by a previous fix

90 new failure due to a previous fix

0x Other

00 other

Defect Type Descriptions

1x Logic: Defects that have to do with the correctness (logical structure) of loop control, execution flow, condition testing in general, and misinterpretation of logical statements.
Example: specifying "x<10" as a loop condition when it should be "x<=10."

2x Computational Problem: Problems with numerical arithmetic, including insufficient precision, sign errors, incorrect mathematical formula used.
Example: "-x" instead of "x"

3x Interface/Timing Problem: Problems with the communications between two software systems. These can include use of interrupts, message passing, and function calls.
Example: incorrect parameters passed to a function, correct parameters passed in the wrong order, or impossible timing condition.

4x Data Handling Problem: Data that is improperly initialized, stored in or retrieved from a data structure, given incorrect units, or given incorrect data type.
Example: "int x = 5.55."

5x Data Problem: Improper or incorrect data in or from a data store. Could be improper entries in a lookup table or other resources.
Example: addressing incorrect value in a data store or addressing the correct memory location only to find the value there is incorrect. (i.e., pi = 17)

6x Documentation Problem: Statements that are ambiguous, do not make sense, conflict with other requirements, are illogical, are impossible, or are simply incorrect. They may not prevent the system from running but they could cause behavior that deviates from the expected.
Example: "Window 17 shall never be blanked. ... When the DTS is in DBTC mode, Window 17 is blanked."

7x Document Quality Problems: Failure to meet documentation standards, inclusion of outdated or unidentified information. These are errors of presentation or verification errors.
Example: "This specification shall not contain hexadecimal numbers. All numbers will be given in decimal or binary format. For the rationale, see page 0x17AF."

8x Enhancement: Defects caused by changes in the product meant to add functionality or capability.
Example: In order to increase the resolution, the transmission rate of a signal is doubled. Unfortunately, there is not enough processor/bus/whatever capacity to handle the increased messaging.

9x Failure caused by a previous fix: The system wasn't broken until we tried to fix it.
Example: We won't be changing the system, so this isn't relevant.

0x Other: None of the above, miscellaneous, etc.
Example: Something not otherwise mentioned.

Observation Log

Name : _____

Week : _____

Observation	Date	Activity	Type	Comment

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	October 2001	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Framework Document: Model-Based Verification Pilot Study	F19628-00-C-0003

6. AUTHOR(S)

David P. Gluch, John J. Hudak, Robert Janousek, John Walker, Charles B. Weinstock, Dave Zubrow

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2001-SR-024

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

This Pilot Study Framework document describes the processes, activities, artifacts, and deliverables associated with an Engineering Practice Investigation of Model-Based Verification (MBV).

14. SUBJECT TERMS	15. NUMBER OF PAGES
pilot study, modeling, procedures, model-based verification (MBV)	35

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18 298-102