Improving Product Development Capability Maturity through Object Process Methodology

by

Hsuehyung Benjamin Koo

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

September 2001

The author hereby grants MIT permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part.

Signature redacted

Hsuehyung Benjamin Koo
System Design and Management Program

Signature redacted

Dov Dori
Thesis Supervisor
Associate Professor of System Design and Management

Signature redacted

Steven D. Eppinger
Co-Director, LFM/SDM
GM, FM Professor of Management Science and Engineering Systems

Signature redacted

Paul A. Lagace
Co-Director, LFM/SDM
Professor of Aeronautics & Astronautics and Engineering Systems
Abstract

A comprehensive process-modeling language framework is a valuable instrument for managing evolving organizations. An OPM-based formal framework has been developed and applied to the modeling and assessment of organizational capability maturity. Specifically, this paper applies OPM as a practical instrument for supporting processes within evolving organizations. To this end, OPM has been enhanced to reflect the nature of decision-making processes and to effectively manage vague and uncertain information. To adequately represent organizational decision-making, game theory and evolutionary theory are integrated to provide a formal quantitative analysis framework, with CMM concepts serving as the ontological infrastructure. Incorporating fuzzy logic and probability theory into the OPM language kernel enables consistent treatment of vague and uncertain information. With the extensions developed and illustrated in this paper, OPM becomes an effective instrument for organizational process knowledge management. Equipped with such knowledge management infrastructure, evolving organizations will be able to make prudent process migration decisions along the lines of CMM while minimizing unexpected negative impacts.
Acknowledgement

The author wants to thank the following people for their generous help and encouragement for completing this thesis. First, I need to present my deepest gratitude to my thesis advisor, Dr. Dov Dori, for his guidance and brilliance in supporting my research as well as giving invaluable advises throughout this research project. I also need to mention various System Design and Management (SDM) Fellows who spent countless hours discussing ideas. These SDM fellows are Christine Miyachi, Peter Panetta, John Woods, David Sharman and Ray Chou. I need to point out that my research was particularly influenced by Christine’s prior research results.

During the writing of this paper, I also had my friends and colleagues providing their critical views and editorial changes to my writing. Their inputs made this paper much more presentable. Carl Steadman, who spent a significant amount of time to proof read and point out structural flaws in this paper. Penju Kang, who also shed his knowledge on control theory and process management with me. He also spent a lot of time prove read this paper.

Finally, I need to thank my parents for their support and caring. Their love is the ultimate motivating force that brings this research project to life.
# TABLE OF CONTENT

1 INTRODUCTION .............................................................................................................................................. 4
   1.1 WHERE DOES Capability Maturity Model COME FROM? ................................................................. 4
   1.2 CMM REACHING BEYOND THE SOFTWARE INDUSTRY ............................................................. 5
   1.3 ENHANCING CMM WITH OPM ........................................................................................................ 6
      1.3.1 Object Process Methodology ................................................................................................. 6
   1.4 MANAGING INHERENT VAGUENESS AND UNCERTAINTY IN ORGANIZATIONAL PROCESSES ... 7
   1.5 ESTABLISHING A CONSISTENT MENTAL MODEL FOR CMM PARTICIPANTS ........................... 8

2 RESEARCH APPROACH ............................................................................................................................... 10
   2.1 OPM AS A GENERAL SYSTEM DESCRIPTION LANGUAGE ..................................................... 10
   2.2 CMM AS A GENERALIZED MODEL OF PROCESS MATURITY EVOLUTION ............................ 11
   2.3 PROJECT MODEL AS A FRAMEWORK FOR PROJECT PERFORMANCE EVALUATION ............. 11
   2.4 A CMM-BASED PROCESS KNOWLEDGE MANAGEMENT SYSTEM ........................................ 13

3 OBJECT PROCESS METHODOLOGY ........................................................................................................ 14
   3.1 LANGUAGE AS A MEDIUM FOR INFORMATION SHARING ........................................................... 15
      3.1.1 Using OPM to Improve Communication .................................................................................. 15
   3.2 LANGUAGE AS AN INSTRUMENT FOR PROCESS PLANNING .................................................... 17
      3.2.1 Representing Process Dynamics and Organizational Structures in OPM ............................... 18
      3.2.2 Visualizing the Interactions between Structural and Behavioral Patterns .......................... 19
   3.3 LANGUAGE AS A MEDIUM FOR KNOWLEDGE ACCUMULATION .......................................... 20
      3.3.1 OPM as a Tool for Knowledge Management ....................................................................... 21
   3.4 WHY NOT UML? .............................................................................................................................. 21
      3.4.1 Model Multiplicity .................................................................................................................... 21
      3.4.2 Inflexible Definition of Language Constructs ....................................................................... 22

4 CAPABILITY MATURITY MODEL .............................................................................................................. 23
   4.1 CMM AS A MODEL FOR ORGANIZATIONAL EVOLUTION ....................................................... 23
      4.1.1 The Optimized Maturity Level may not be Optimal ............................................................... 25
      4.1.2 An Evolution Model Governed by Business Objectives ........................................................ 26
   4.2 THE STRUCTURE OF PROCESS AREAS ........................................................................................ 27
      4.2.1 The Interactions between Process Areas ................................................................................. 29
      4.2.2 CMM provides Key Measurements for Each Process Area ................................................ 30
   4.3 SUMMARY FOR CMM'S CONTRIBUTION TO FORMAL ORGANIZATIONAL MODELING ........... 30

5 SIMULATING PRODUCT DEVELOPMENT PROJECTS BASED ON FORMAL CMM MODELS .................. 32
   5.1 REWORK GENERATION IS THE ROOT CAUSE OF PROJECT DELAY AND FAILURE ............... 32
      5.1.1 Expressing Differential Equations in Stock and Flow Diagrams ......................................... 34
   5.2 USING OPD TO VISUALIZE PROJECT DYNAMICS .................................................................... 35
   5.3 MANAGING MODEL REUSE WITH OPM ..................................................................................... 36
      5.3.1 Model Partitioning: Composition and Decomposition ......................................................... 37
      5.3.2 Model Reuse: Generalization-Specialization ..................................................................... 38
   5.4 ORGANIZE SIMULATION MODEL BASED ON CMM PROCESS ARCHITECTURE ..................... 39
   5.5 CMM-OPM AS A HOLISTIC APPROACH TO REDUCE REWORK GENERATION ..................... 40

6 MANAGING VAGUENESS AND UNCERTAINTY ..................................................................................... 42
   6.1 SOURCES OF IMPRECISE MODEL INFORMATION .................................................................... 42
   6.2 IMPROVING THE SEMANTIC FIDELITY OF DYNAMICAL PROCESS MODELS .......................... 43
   6.3 VAGUENESS AND UNCERTAINTY AS TWO DISTINCT SOURCES OF INACCURACY ............... 43
LIST OF FIGURES

FIGURE 3-1 OPD OF CMM LEVEL-2 ORGANIZATION ................................................................. 16
FIGURE 3-2 ENTIRE COLLECTION OF BASELINE OPD SYMBOLS ........................................... 17
FIGURE 3-3 A SAMPLE OPD REPRESENTING HIGH LEVEL CMM PROCESSES .................. 19
FIGURE 3-4 AN EXAMPLE OF OPD/OPL PAIR ........................................................................ 20
FIGURE 4-1 OPD OF AN EVOLVING ORGANIZATION .......................................................... 24
FIGURE 4-2 CMM's VIEW OF AN EVOLVING ORGANIZATION ........................................... 24
FIGURE 4-3 PROPOSED MODEL OF CMM-BASED EVOLVING ORGANIZATION ............... 25
FIGURE 4-4 EVOLUTION DRIVEN BY BUSINESS OBJECTIVES ......................................... 27
FIGURE 4-5 PROCESS AREA DATA STRUCTURES IN OPD ................................................... 28
FIGURE 5-1 STOCK AND FLOW DIAGRAM OF REWORK GENERATION ............................... 33
FIGURE 5-2 VISUALIZING POLICY CONSEQUENCES (COURTESY OF J. M. LYNEIS) ............ 34
FIGURE 5-3 DIFFERENTIAL EQUATIONS IN STOCK AND FLOW DIAGRAM ...................... 35
FIGURE 5-4 REWORK GENERATION IN OPD ........................................................................ 36
FIGURE 5-5 COMPOSITION/DECOMPOSITION OF THINGS ................................................. 38
FIGURE 5-6 PROCESS SPECIALIZATION/GENERALIZATION ............................................... 39
FIGURE 5-7 PROPOSED CMM PROCESS ASSETS ................................................................. 41
FIGURE 6-1 DETERMINISTIC VARIABLE IN OPD ................................................................. 44
FIGURE 6-2 PROBABILISTIC VARIABLES IN OPD ................................................................. 46
FIGURE 6-3 FUZZY VARIABLE IN OPD .................................................................................. 47
FIGURE 6-4 TWO DIMENSIONAL LOOKUP TABLE ............................................................... 49
FIGURE 6-5 OPD OF FUZZY VARIABLE RELATIONSHIPS .................................................. 50
FIGURE 6-6 THREE DIMENSIONAL LOOKUP TABLE ......................................................... 51
FIGURE 6-7 MEMBERSHIP FUNCTION EDITOR IN MATLAB'S FUZZY LOGIC TOOLBOX ....... 52
FIGURE 6-8 VARIABLE TYPES IN OPD .................................................................................. 53
FIGURE 7-1 HIGH LEVEL ARCHITECTURE OF OPM-BASED KNOWLEDGE MANAGEMENT SYSTEM .................................................................................................................. 58
1 Introduction

The ever-increasing speed of technological advancement increases complexity and introduces additional uncertainty in the process of new product development. However, the general practices of organizational management have not evolved at a similar rate. To increase the evolving speed of management practices, the conceptual frameworks of process architecture and related knowledge management instruments must be capable of dealing with accelerating complexity and uncertainty. An integrated approach to managing process knowledge for fast-evolving organizations is presented. It combines two complementary methodologies, Capability Maturity Model (CMM) and Object Process Methodology (OPM). OPM provides a high-level formal language framework to retain, communicate and organize process knowledge for CMM practitioners. OPM can function as a knowledge representation medium to bridge the cognitive gaps between quantitative and qualitative management techniques.

This introduction explains the origin and application scope of CMM. Then, it outlines OPM's possible roles in CMM implementation. OPM's formalism is establishes a consistent mental model for CMM participants to access process knowledge.

1.1 Where does Capability Maturity Model come from?

The lack of scientific methods for project management is an issue particularly visible to the software industry. According to a study performed by the Software Engineering Institute (SEI), over 80 percent of large-scale software projects are canceled or delayed [Humphrey’s lecture notes, 2000]. Based on the observation of industry veterans such as Watts Humphrey, this
widespread failure to meet expectations can be primarily attributed to inadequate management practice, and the lack of industry-wide knowledge about complex product development.

Seeking to improve project management in the software industry, the Software Engineering Institute (SEI) developed a set of comprehensive project management guidelines, introduced in the 1990s as the "Capability Maturity Model" (CMM). A process-oriented model, CMM describes a generic organizational architecture for software development. Since then, CMM has been extensively employed within the software development groups of government agencies and large corporations. Various studies have confirmed the positive impact of CMM on the organization, showing measurable increases in both development speed and product quality. The widespread adoption of CMM has helped introduce and establish quantitative management techniques into the field of software development. In fact, some government agencies now require contractors to demonstrate specific levels of CMM maturity in order to remain eligible for contract-bidding activities. From a pragmatic standpoint, CMM has established its business value and political significance in the software industry.

1.2 CMM Reaching Beyond the Software Industry
To ensure the success of software development projects, CMM has been looked upon as being able to address management practices and organizational issues beyond those strictly related to the production of software. For example, an organization certified at the highest CMM maturity level must be prepared to cope with the infusion of innovative technologies. To minimize the risks induced by technological infusion, mature organizations are expected to proactively manage structural and process changes. Since managerial challenges such as these are common to all types of organizations, the Software Engineering Institute has expanded and further
generalized CMM for use outside the software industry. Specifically, it created CMMI, or “Capability Maturity Model-Integrated,” for all types of product development. The SEI also publishes a set of recommended practices of human resource management, the “People Capability Maturity Model” (P-CMM). Over the last few years, the CMM community has increasingly made the case for applying scientific management techniques to large-scale product development projects.

One of the most notable contributions of CMM to management practice is its highly generalized and well-articulated process architecture. Almost all product development organizations can apply CMM principles to categorize their internal organizations according to the process architecture employed by the methodology. This generalized process architecture allows CMM to establish a cross-industry mental model that can be used to conceptualize and assess the maturity of an organization. Once in place, this architecture also provides a standard auditing procedure to compare and analyze organizations, based on the efficacy of process execution. In short, CMM provides a popular and convincing approach to understanding product development organizations.

1.3 Enhancing CMM with OPM

CMM does have its limitations. Since the description of the CMM process model is written in natural language (English), process-related decisions still rely on human intuition, and are subject to debate. This lack of formalism in CMM makes it difficult to objectively and quantitatively study the consequences of specific practices and process policies. For the purpose of validation, decision-makers should be able to apply a quantitative approach to estimate the impact of new process introduction or migration. Though attempts have been made at a more
rigorous formulation of the CMM principles, most process models that have attempted to estimate policy impacts are only intended for a unique instance of the CMM organization. Because of this, the models thus derived are not immediately applicable and available to the general CMM community. To resolve this issue, the author presents a systematic approach to extend CMM’s conceptual framework into a formal process model. This formal model can then be shared across all types of CMM-implementing organizations.

1.3.1 **Object Process Methodology**
Applying a formal language to describe the process architecture of CMM has many advantages. It reduces communication errors due to misinterpretation or misunderstanding. Using a formal language also encourages knowledge sharing; CMM-practicing organizations can share and evaluate process models and their results through a common language. We apply Object Process Methodology is applied as a high-level, user-friendly formal language for describing the structure and dynamics of complex systems using formal semantics and syntax. OPM enables the creation of models that can illustrate cause-and-effect relationships within organizational processes, allowing otherwise vague and uncertain data to be consistently interpreted by both machine and human.

1.4 **Managing inherent vagueness and uncertainty in organizational processes**
A formal methodology should adequately address the inherent vagueness and uncertain nature embedded in the processes of product development. It is unrealistic to expect a formal model of organizational processes to accurately predict the future performance of organizations. By introducing specific classes of fuzzy and probabilistic variables one can explicitly express the notion of vagueness and uncertainty within the context of a formalized process model. To this
end, OPM’s notation is extended to enable convenient and rational expression of fuzzy and probabilistic variables, allowing modelers and process analysts to consistently manage areas of vagueness and uncertainty.

1.5 Establishing a consistent mental model for CMM participants

According to the three lenses organizational process analysis technique [Ancona 1999], a truly robust methodology should be able to address an organization’s political, cultural and strategic needs. Representing CMM in a high-level formal language framework can be an instrument for creating cultural, political and strategic benefits for the CMM-enabled organization. Culturally, a formal and intuitive high-level language reduces ambiguity within communications, thereby increasing the shared knowledge within an organization. Politically, using standard and formal process methodologies such as CMM and OPM strengthen process-oriented initiatives that might otherwise be hampered by the regulatory concerns of government and industry conventions. Strategically, OPM-based CMM models help formalize the representation of process knowledge, thereby enabling streamlined sharing of management expertise and allowing for the systematic accumulation of process management knowledge.

Grounding the knowledge structure of CMM in a firm linguistic foundation has a positive impact on current process-management practices. A formal and intuitive language for CMM specification allows for more effective communication across the participants in CMM processes and provide an objective way to describe and discuss the consequences of management policies.

Finally, a formal language framework streamlines the translation of process models into simulation and other quantitative models. Ideally, a formal language framework for CMM, such as OPM, can act as a platform for knowledge management that seamlessly integrates scientific
management methods while directly and systematically influencing the behavior of CMM-implementing organizations.
2 Research Approach
This paper takes an analytical approach to show that a highly expressive and formal language is a necessary instrument for an organization to systematically accumulate process knowledge and therefore methodically improve capability maturity. More specifically, OPM will be examined and expanded to reflect the needs of managing system complexity and identifying the sources of inaccuracy in evolving organizations. In a related research effort, Miyachi [2000] presented OPM as a modeling language to describe CMM’s process architecture. This paper extends the use of OPM to represent the multiple dynamic models of CMM-implementing organizations. In order to robustly evaluate the consequences of process policies, CMM practitioners need to seamlessly integrate quantitative and qualitative analytical methods. Using OPM to describe the organization in quantitative and qualitative terms improves the communication and coordination effectiveness of CMM implementing organizations.

2.1 OPM as a General System Description Language
OPM was designed as a high-level general-purpose language that can effectively represent the structural and dynamical aspects of a system in arbitrary forms of scales. The generality of a language framework is critical for its successful use and growth in actual practice. The expressive power of OPM makes it adequate to describe arbitrary complex systems, such as a product-developing organization. In the context of CMM, OPM can conveniently illustrate the process architecture with a set of Object Process Diagram (OPD set) that are immediately translated to Object Process Language (OPL). OPM manages system modularity by abstracting complexity and defining manageable subsystems and presents the interrelationships between structure and processes. These unique features of OPM help process modelers and CMM
participants to share and communicate their process with the rest of the organization as well as the entire CMM community.

2.2 CMM as a Generalized Model of Process Maturity Evolution
One of the trademarks of CMM is its serialized maturity grading system. The definition of maturity levels reveals a generalized evolutionary pattern of product development organizations. Based on these evolutionary stages, decision-makers can prioritize and anticipate the needs in respective process areas and therefore allocate resources accordingly. To rigorously manage the evolutionary path of an organization, OPM is used to represent the CMM process architecture. We use the well-defined CMM vocabulary as a naming standard for all performance measurement variables and the relevant nomenclature of related activities, instruments and organizational entities. Using CMM-defined vocabulary in a formal model should improve its acceptance and ease of understanding across the entire CMM community. A consistent naming convention encourages systematic model reuse and sharing. We integrate the OPM framework into CMM as its formal specification language as a means to formalize it and reduce model translation and misinterpretation labor and costs.

2.3 Project Model as a Framework for Project Performance Evaluation
Timing, sequencing and the degree of process changes are primary management issues in CMM implementation. These issues need to be addressed by carefully studying the dynamic consequences of process changes and resource allocation. To systematically estimate the performance of CMM practices, many organizations have applied System Dynamics and discrete-event simulation to help analyze the dynamic consequences of policies. Prior to using OPM, continuous and discrete dynamics are usually separately captured by System Dynamics
models and discrete simulation models, respectively. Due to the nature of the modeling tools, these two analytical techniques are rarely integrated to complement each other. This separation between discrete and continuous models makes integrated analysis difficult and error prone. As we show, OPM-based models can uniformly represent the dynamics in both discrete and continuous domains. Specifically, rework generation can be measured quantitatively and provide a tool for evaluating project performance. Project Model [Lyneis, 2000] is used as a foundation to discuss the impact of rework generation in the context of CMM.

To reduce the complexity of the Project Models, we use Process Areas based on CMM’s process architecture. To improve the presentation format and expressiveness of the dynamics model, OPM notation replaces the stock-and-flow notation set commonly used by system dynamists. Managing the Sources of Vagueness and Uncertainty

Besides infusing the nomenclature of CMM into the Project Model, one also needs to consider the precision and validity aspects of organizational behavior modeling. A useful model should adequately reflect the vagueness and uncertainty information in its model presentation. New OPM language constructs provide formal distinction amongst deterministic, vague and uncertain variable types. This enables model developers to use adequate notation to represent areas of uncertainty and vagueness within the model. Formally classified variable types provide process modelers with a systematic way of managing the sources of vagueness and uncertainty. Unique OPM symbols and semantics increase the cognitive distinction between deterministic, vague and uncertain variables, improving human understanding of the source of vagueness and uncertainty in CMM-based organizations.
2.4  **A CMM-based Process Knowledge Management System**

To apply simulation and formal knowledge to process management, the participants of CMM activities and simulation modelers need an integrated tool to perform simulation and derive analytical results. To this end, we have designed an architecture of such simulation/knowledge management tool. The speed of evolution toward higher-level maturity is dependent upon the quality of the model description language as well as its information architecture. This system can be employed to model the dynamics of a CMM-based organization and share the process knowledge freely with the entire CMM community.
3 **Object Process Methodology**

CMM establishes a standard definition of process management related vocabulary. It describes the structural relationships between process areas and required resources based on this vocabulary. The vocabulary and structural relationships in CMM can be viewed as the words and rules of a communication protocol. Instilling a common protocol in an organization increases communication efficiency and improves the efficacy of collaboration across people and departments. If one conceptualizes the field of product development management as a community activity, CMM is a process management protocol designed to streamline knowledge sharing within that community. To further propel the quality of CMM practices, we therefore need to strengthen this communication protocol.

The official CMM documents do not endorse a specific formal language to CMM practitioners. The lack of formalism allows CMM practitioners to implement organizational practices with significant latitude. However, loose definitions and process guidelines reduce the robustness and accuracy of CMM practice. In the book, the “Human Use of Human Beings”, Norbert Wiener stated that “to assure a fair and productive society, the law must be unambiguous and made aware to all participants in advance” [Wiener, 1970]. Similarly, a product development organization must establish clear and repeatable interpretation of its process guidelines. One way of achieving it is employing a formal language.

OPM can serve as a foundation to define and capture CMM concepts with a consistent linguistic framework. Using a formal language framework to describe organizational activities has important operational advantages, discussed below.
3.1 Language as a Medium for Information Sharing

Since the general purpose of CMM is to help an organization robustly achieve business objectives and minimize uncertainty in project execution. All areas of the organization should coherently approach this goal by sharing relevant information with each other. However, the format and syntax of information shared must adhere to a common protocol. This protocol, namely, a language schema, should be able to dynamically address the evolving communication needs of the organization. By adopting a standard language schema, information can be transmitted across organization boundaries and between individuals. By sharing a common language schema, all participants should be able to interpret share information precisely and consistently. The embedded semantics of the standard language eliminates the need for participants to negotiate the meaning of each piece of information, therefore, improve its communication efficiency. Ideally, an organization with high communication efficiency should also have better capacity to handle arbitrary events, henceforth becoming a more adaptive organization.

3.1.1 Using OPM to Improve Communication

As a general system description language, OPM is a formal language framework that is ergonomically designed to specify and present information of arbitrary systems. OPM's textual element, Object Process Language (OPL), adopts formal English grammar that allows English-speaking people to interpret the intended textual messages without additional grammatical knowledge. The graphical form of OPM, Object Process Diagram (OPD), illustrates the same message with nodes and arcs that represent things and relations.

The following diagram shows an OPD that describes a CMM level-2 organization.
This OPD shows that each process area can be modeled as an agent (objects that autonomously carry out activities). In practice, process areas are likely to be organized as departments that carryout associated activities autonomously. Each “process” represents a commitment, ability or activity of the corresponding process area. For example, within an organization, all process areas must share a single “Resource Pool”. The primary contention between process areas or their corresponding departments can be easily visualized as a competition of resources. This presentation format concisely reveals the interaction amongst process areas that cannot be easily identified with textual description.

Notice that ellipses represent processes and rectangles represent objects or agents. There are also multiple types of links. A well-refined symbol set should be rich enough to cover meaningful
expressions while simple enough for average users to master easily. The following diagram shows the graphical notation set of OPM.

![Diagram of OPM graphical notation set]

This graphical notation set shares the same symbols of popular graphical modeling languages, such as UML and Petri-Net. Sharing the same notation set not only leverage the experience of the modeling methods, it also selectively exploit the best-of-breed visual properties of other graphical languages. OPM's graphical notation and textual format are optimized for human consumption, therefore, making OPM a good candidate for organizational process modeling and communication.

3.2 **Language as an Instrument for Process Planning**

According to Lakoff et al. [Metaphors we live by. Lakoff 1980], language structures the actions that mind and body performs. In software development activities, programming languages not
only instructs computers to perform tasks, it also helps software developers to structure their thoughts. To software developers, selecting an adequate programming language is equivalent to doctors selecting a proper instrument during a surgical operation. Poor choice of an instrument may cause critical delay or bring catastrophe to the activity at hand. Similarly, the working language of an organization is likely to influence its performance in terms of product quality and process maturity.

3.2.1 Representing Process Dynamics and Organizational Structures in OPM

OPM is a language framework that possesses the expressive power of multiple languages. OPM is a superset of many formal language systems. It has the combined expressive power of predicate logic language, propositional logic, type calculus, relational entity diagram and state-transition diagram (Petri-Net). The language kernel also allows designated language schema administrators to add new language constructs. OPM is highly flexible because it is a meta-language that can be redefined to covers all categories of formal expressions.

The detailed explanation of OPM’s language features can be found in the paper, Meta Reflexive Modeling using Object-Process Methodology [Dori 1999].

This rich expressive power of OPM is highly desirable for the modeling of human organizations. Some of the most common characteristic of system descriptions can be easily specified using OPM. For example, OPM is capable of specifying the workflow (sequence of events), expressing all possible states of each object, and visualizing the relationships between objects and processes, all in a single diagram. The following OPD shows an example of such system.
3.2.2 Visualizing the Interactions between Structural and Behavioral Patterns

To better explore a modeler's cognitive potential for process modeling, Object Process Diagram (OPD) combines the symbols of class diagram, state-transition diagram (Petri Net) and entity-relational diagram into a single symbol system. The comprehensive notation set enables OPD users to visualize the interrelationships between structural and behavioral patterns of a system. The ability to visualize the connections between structural and dynamical models is a crucial advantage of OPD. It allows people who are more concerned about structural details to visually perceive the dynamic consequences and vice versa. In another word, OPD has the flexibility to show the holistic picture of the system, therefore improves the quality of process planning and structural modeling.

The textual element of OPM, Object Process Language (OPL), represents the exact information encoded in its graphical counterpart, OPD. Given that OPL is based on formal English grammar, it provides a cross reference for OPD users to verify the meaning of the graphical model. Based
on personal observation, this textual-visual bi-directional model verification is an important psychological aid for modelers and model readers. The cross reference between two representations reduces model ambiguity throughout the entire life cycle of its usage. This translates into higher quality model and indirectly improves the performance of the organization that follows the model. The following diagram shows a simple example of multiple OPD/OPL pairs.

![Figure 3-4 An Example of OPD/OPL Pair](image)

3.3 **Language as a Medium for Knowledge Accumulation**

In this paper, the author narrowly defines process knowledge as recorded information that associates process policies with their relevant consequences. Based on this definition, we can discuss the relationships between the properties of a formal language and its relative performance when applied to knowledge management.
3.3.1 OPM as a Tool for Knowledge Management
As mentioned earlier, OPM is a combination of many formal languages. It includes all the relational operators of all languages that it desires to include. Each relational operator provides a specific syntactical structure that helps human or machine to organize the associated knowledge content. Having a variety of operators allows users of OPM to combine multiple strategies to explore the entire process knowledge base. Without explicitly labeling the relational information amongst objects and processes, knowledge management usually resorted to statistically based data mining in a rather amorphous data storage. Storing CMM process models in OPM enables knowledge management software to mechanically organize information based on relational operators amongst objects and processes. Organizing process models with OPM’s relational operators preserves the semantic content of model information.

3.4 Why not UML?
UML and OPM share many common goals. However, UML does present the following specific shortcomings when applied to model organizational activities. They can be categorized into a few main sources of problems. They are: the issue of model multiplicity, and its inflexibility of language constructs.

3.4.1 Model Multiplicity
Traditionally, each diagramming language is designed to tackle one aspect of formal system property. For example, state diagram only represents the dynamic aspect of a system. Relational diagrams only represent the structural aspect of the system. To express multiple aspects of a system, the modelers and model users must manage multiple models. UML is designed as a collection of nine diagramming languages. Its multi-diagram approach makes it challenging for human to master all nine symbol sets. It is also difficult to build a modeling tool that robustly
ensures the integrity of all nine sub-models. In contrast to UML, OPD expresses all formal properties of a system in a single notation set. OPD’s “unified” notation set eliminates the mental burden and technical complexity of managing multiple modeling languages. Dori et al. had extensively discussed this issue in their Model Multiplicity paper.

3.4.2 Inflexible Definition of Language Constructs

UML’s language constructs are defined by a non-profit organization, Object Management Group (OMG). Since UML was originally designed for formalizing software system descriptions, its later development focused on the needs of software engineers. Its language constructs have not addressed issues related to organizational modeling. In contrast, OPM is an extendable language that was originally designed to describe arbitrary systems, therefore the language kernel of OPM allows users to define new language constructs as needed. These inherent limitations of UML depreciate its technical value to perform CMM-based process modeling.
4 Capability Maturity Model

CMM offers a conceptual framework to characterize a product development organization. The conceptual framework modularizes an organization into Key Process Areas (KPAs) and stages the organization's evolution based on the performance of KPAs. This modularized process architecture and its corresponding evolutionary path are derived from immense amount of process management knowledge. However, SEI has not officially adopted any formal language framework to describe CMM process architecture or organizational evolutionary dynamics. In this chapter, OPM will be used to formally describe the three essential aspects of an organization based on CMM's conceptual framework. The three aspects are: the driving forces behind organizational evolution processes, the interactions between two process areas, and the key measurements associated with each process area.

4.1 CMM as a Model for Organizational Evolution

CMM categorizes organizations' evolutionary stages into five levels of capability maturity. Each maturity level specifies a minimal set of required process areas. Estimating the scale of organizational maturity based on the collection of active process areas provides a consistent mechanism to model organizational evolution. For example, the advancement of capability maturity involves adding new process areas to the existing collection. Similarly, the degradation of maturity implies a reduction of active process areas. Using process area and maturity level to characterize evolution, the dynamic model of an organization can be illustrated in the following OPD.
One may consider the concept of maturity advancement as a special case for organizational evolution. The following OPD illustrates CMM's concept of organizational evolution.

Figure 4-1 OPD of an Evolving Organization

Figure 4-2 CMM's View of an Evolving Organization
This simple OPD reveals the incompleteness of CMM as a model for organizational evolution. For the purpose of advancing organizational maturity, CMM only provides guidelines on introducing additional process areas. In practical settings, an organization should also prepare to regress back to lower maturity levels due to the lack of resources and change of product requirements. To offer robust management guidelines for evolving organizations, CMM should include explicit information on how to deactivating existing process areas. The evolutionary model for CMM implementing organizations can be illustrated in the following OPD.

![Figure 4-3 Proposed Model of CMM-based Evolving Organization](image)

**Figure 4-3 Proposed Model of CMM-based Evolving Organization**

4.1.1 *The Optimized Maturity Level may not be Optimal*

According to CMM, a higher maturity certification must execute all process areas specified for lower levels. For example, if a programming shop is particularly good at adopting new technologies into the organization while its ability to manage sub contractors is not successful, then, based on the most stringent definition of CMM classification, this organization would remain at the Initial maturity level.
Based on this example, decision makers who are strongly motivated by the status of CMM certification level is likely to reduce resource allocation on important but expensive high-level maturity process areas and focus on improving less important and lower level process areas that have not been certified. This approach may present conflict of interests in the organization.

Pursuing the highest maturity-level certification is usually not the primary goal of an organization. In extreme cases, decision-makers whose sole purpose is to pursue higher maturity level may be fixated on the status of maturity while ignoring the true strength of the organization. To ensure that all participants of CMM practices do not lose sight of the cohesive goal of the organization, each process area has explicit goals and each organization should have explicitly document business objectives. Its business objectives and realistic needs should determine the ultimate maturity level of an organization.

4.1.2 An Evolution model Governed by Business Objectives

With few exceptions, the process areas in lower maturity levels are easier and less expensive to implement than the process areas in higher maturity level. Furthermore, once the process areas in lower maturity levels are in place, they make implementing additional process areas easier and more economical. Given a wide variety of realistic situations, the combination of process areas and their activating sequence should have sizeable impact on CMM implementation. Moreover, the specific business objectives of an organization should be used to determine the sequence and combination of process area activation.

OPM can be used to formally illustrate the relationships between business objectives and the selection of process area activation sequence. The following OPD shows the roles of Business Objectives and process area selection.
Figure 4-4 Evolution Driven by Business Objectives

This formal evolutionary model can be used as a decision-making model for prioritizing the investment of process improvement. This evolutionary model is based on the value-seeking and value-seeing principles described by Kim Clark et al. [Design Rules, 2000] With integration of process simulation tools; decision-makers can perform what-if scenario analysis to estimate their organizational transformation strategy.

4.2 The Structure of Process Areas

CMM describes an organization’s modularized functionality in terms of process areas. This section describes the internal structure of process areas. To ensure consistent structure and phrasing, the guidelines published by SEI include a template to outline the basic elements of a process area. We will present the internal structure of a process area based on this template.
On the top level, a process area can be illustrated in the following OPL sentence:


The objects listed in the OPL sentence above can be used to develop a generic representation for analyzing process areas. For example, Collection of Goals to and Collection of Commitment can be further decomposed into a set of conditional statements. The status of a given process area can be associated with the evaluation results of conditional statements. The Collection of Activities and Implementation can be decomposed into “Collection of Subpractices”. Each sub-practice can be modeled as a process that formally characterizes the dynamic properties. Having the ability to
describe all process areas in terms of conditional statements and formalized process models establishes a repeatable and explicit approach to analyze organizational performance.

Defining the template of process area in OPM is helpful in terms of formal model development. This template indicates that a process area must have its internal goal(s), it should have the ability and resources (commitment) to perform certain tasks. There should be some specific activities that can be defined and performed with the process area. It also includes certain measurements to help quantify the performance of the process area. Based on these features, one can model each process area as an autonomous organization and create process models for each one of them. On the other hand, CMM provided a good outline to help organizational modelers to modularize the first level building block of an organization is to divide them into process areas. This also suggests that each process area should be formally modeled as an autonomous object (agent) instead of a static process or object.

4.2.1 The Interactions between Process Areas
To estimate performance of the organization as a whole, one must be able to clearly define the possible interactions across all process areas. CMM guidelines require the organization to document the interactions across process area boundary. The detailed interaction information can be documented in an Activity Performed entry. By explicitly defining which activity causes interaction between process areas, it makes the interference between process areas to be traceable.

OPM modelers can represent the interactions as objects that are shared across two or more process models. Having a standard representation to handle interactions will reduce model complexity and system model coupling in the long run.
4.2.2 **CMM provides Key Measurements for Each Process Area**

Within each process area, CMM also presented a number of typical quantitative measurements. These measurements should indicate the performance of each process area. Having this information is invaluable to quantitative process modeling and statistics collection, since that choosing proper variables to represent the performance of an organization is one of the most challenging tasks in a modeling session. Knowing the key measurements of each process area can provide significant insight to reduce model complexity. For example, dollar amounts and person hours expended on a project are the most commonly used metrics to measure the performance of a process area. These quantitative measurements can be easily adopted as modeling variables in simulation models or process statistics collection. In a later chapter, a simulation model based on CMM-defined measurements will be presented.

4.3 **Summary for CMM’s Contribution to Formal Organizational Modeling**

Although CMM is not written in a formalized language, it does provide the conceptual foundation to specify a formal model for an evolving organization. Distinguishing the levels of maturity based on collections of process area is a novel approach to characterize the evolutionary nature of organization. This chapter shows that the high-level decision dynamics of activation and deactivation of process areas can be formally illustrated in an OPD. This OPD demonstrates that organizational evolution is an iterative structure. A computational model that emulates the evolutionary nature of CMM-implementing organizations can therefore be created based on this OPD.

CMM also provides a comprehensive coverage of process areas that are required to perform product development. Without the guidelines presented by CMM, organizational modelers must
study domain specific practices and then create an isolated instance of organizational model from scratch. This is likely to create a lot of model complexity and requires additional labor to verify model validity. If the modelers borrow this generic model presented by CMM, it will dramatically reduce the required preparation work done within individual modeling assignments. CMM also provides the documentation to illustrate the interaction and relationships between process areas. The process area template provides a common schema to describe existing and future process areas. This template can be viewed as a meta-model to specify the features of future and existing organizational processes. A process area template prompts process modelers to manage modeling information and to acquire process areas specific knowledge in a standard format. This standard information format facilitates knowledge sharing across the entire CMM community. The process area template also implicitly instructs modelers to standardize on a common interface for process area interaction modeling. Moreover, each process area is associated with key measurements. The standard association of process areas measurements is particularly useful for modelers that are interested in estimating the performance of the organization. Having a standard set of measurements enables the modelers to effectively share and collect statistical data. Having well-defined measurements also allow CMM-implementers to the compare and contrast organizational performance more easily.
5 Simulating Product Development Projects based on Formal CMM Models

For complex and expensive product development projects, a growing number of organizations are using process simulation models to estimate project performance and resource allocation scenarios. In many product development projects, process simulation techniques have derived useful predictions from computer simulation that is not easily obtainable by sheer human intuition or other manual analytical methods [Sterman, 2000]. Computer-based process simulation is an attractive management technique because it allows managers to interactively perform what-if scenario analysis before or after the real project has begun.

According to Jay Forrester [1961], the goal of process simulation is to validate management assumptions. However, developing simulation models for practical problems can become complicated tasks by themselves. Creating and organizing reusable process simulation models have become a strong interest for project managers and researchers [Hines, Molecules, Malone, Process Handbook]. In this chapter, the author will discuss how to apply CMM and OPM to improve the management and reuse of process simulation models.

5.1 Rework Generation is the Root Cause of Project Delay and Failure

Applying simulation techniques to estimate organizational performance has been extensively practiced in the System dynamics community [Sterman 2000]. System Dynamists have created a persuasive model that reflects many aspects of project management. This model is commonly known as the “Project Model”. From the highest level, Project Model helps quantify the effects of management policies such as: mission consistency, requirement changes, labor availability and work quality. All these managerial issues are linked with a quantifiable property, the amount
of rework generated during project execution. [Lyneis 2000] The concept of rework generation can be illustrated as a stock and flow diagram.

This stock and flow diagram captures the essential dynamic relationships amongst a number of key quantitative variables. At the initial state of the simulation, “Work to do” starts out with a fixed number of tasks specified by the “Initial work requirement”. Based on the value of “Work rate”, it depletes the value of “Work to do” over a period of time. If work performed is not perfect, the value of “Defective work rate” will be assigned a positive value and hence accumulate “Undiscovered Rework” over time. Based on the rate of rework discovery, the total amount of Discovered Rework will change over time. Similarly, if requirements are changed after the project had already started, the value of “Rate of requirement change” will also affect the total amount of “Discovered Rework”. Once “Discovered Rework” becomes available, they will be contributing to the amount of “Work to Do”. By setting the initial value for variables such as “Rate of requirement change” and “Defective work rate”, simulation analysts can quickly
experiment with alternative project management policies and observe the dynamic consequences of these policies without committing to any premature decisions.

A typical way to visualize simulation results can be shown in the following diagram. By visually comparing the time-series plot of various simulated policies, it is an excellent way to present alternative policies and their consequences.

![Graph for Staff Level](image)

**Figure 5-2 Visualizing Policy Consequences (Courtesy of J. M. Lyneis)**

5.1.1 **Expressing Differential Equations in Stock and Flow Diagrams**

System dynamics is a quantitative method for studying the evolving states of multiple interactive variables. To make System Dynamics more accessible to non-technical people, stock and flow diagram was invented to help model readers to visualize the underlying system of differential equation. This notation set is grounded on the metaphor of a hydraulic system. A stock is represented as a rectangle fluid container, and a flow is represented as a directed pipe with a
control valve. This visual metaphor is adequate for illustrating the dynamic behavior of continuous variables. The three symbols, stock, flow and valve are sufficient to represent simple differential equations. The following equation pair is an alternative presentation of the associated stock and flow diagram.

\[
\text{Work To Do} = \int (\text{Additional work} - \text{Work rate} - \text{Defective work rate}) \, dt + \text{initial work requirement}
\]

\[
\text{Completed work} = \int (\text{Work rate} - \text{Rate of requirement change}) \, dt + 0
\]

![Figure 5-3 Differential Equations in Stock and Flow Diagram](image)

The stock and flow diagram is a convenient way to model linear and non-linear dynamics of evolving systems. Alternatively, the stock and flow symbols can be directly mapped onto objects and processes in the OPD symbol set. The following section will present an alternative notation set to present the dynamics of rework generation.

5.2 Using OPD to Visualize Project Dynamics

The following OPD is simply a transcription of the project model from the stock and flow diagram notation. Figure 5-4 is a static image that demonstrates the initial state of the process.
simulation. Notice that OPD is capable of displaying the values of objects during runtime. Each effective link between a Process and an Object can be associated with an algebraic equation specified by the modeler. The algebraic relationship between two objects can be specified by this equation. For example, the object “Initial work order” provides the initial value for the object: “Remaining Work”. This simple example demonstrates that OPM as a language can be extended to fully capture the information of the Stock and Flow diagram. Additionally, it can provide runtime state visualization by displaying the states explicitly.

5.3 Managing Model Reuse with OPM

Since the original Project Model can be presented as a collection of interrelated variables, it is difficult to carve out a portion of the model without affecting the technical integrity of the model.
To make matters worse, the stock and flow notation set does not offer specific language constructs to facilitate the partitioning or specialization of complex models. Managing model complexity and increasing ease of model reuse has become a crucial technical challenge for process simulation experts. To systematically reduce model complexity and establish a formal mechanism for model reuse, OPM includes language constructs that are designed to accomplish these tasks. OPM employs the notion of composition/decomposition to minimize model complexity. It also uses generalization/specialization as a structural transformation construct to facilitate model reuse. These two constructs are explained below.

5.3.1 Model Partitioning: Composition and Decomposition
In the language of OPM, the root class is Thing. It can be categorized as two operand types, Object and Process. OPM allows a Thing to be composed of other Things. The concept of composition and decomposition allows modeler to selectively expose and suppress complexity during any period of modeling session. Whenever appropriate, modelers can choose to work with the internal composition of a decomposed Thing. Otherwise, a modeler can work with a collection of composite Things. This language feature directly addresses the issue of model partitioning.
5.3.2 Model Reuse: Generalization-Specialization

The notion of specialization and generalization of typed objects is a popular concept known as object inheritance in Object-Oriented methods. In OPM, the concept of generalization and specialization can be applied to both Object and Process. The notion of generalization/specialization is a necessary mechanism for systematic model reuse. For example, marking an Object B to be a specialized case of Object A implies that Object B will automatically derive all the properties, values and processes defined within Object A with the additional information specified in Object B locally. In terms of model reuse, if a simulation model for Level-4 organization has been created as a composite Object, to develop a simulation model for level-5, the modeler only need to specify the unique properties and behavior that is specific to level-5. The syntax of inheritance is a potent information management mechanism to remove unwanted complexity by hiding them in generalized cases and focus modeler’s attention on specialized cases. For example, the process of each higher CMM level can be modeled as a specialized case of a lower CMM maturity level. This specialization/generalization mechanism is
an essential language construct in object-oriented programming. It is also a popular and proven technique to systematically approach model reuse and complexity management.

![Diagram of Software Development Life Cycle and CMM Levels]

**Figure 5-6 Process Specialization/Generalization**

5.4 **Organize Simulation Model based on CMM Process Architecture**

Although the Project Model shed valuable insights on the dynamic relationships between certain key performance variables, the decision-maker still needs to identify a proper approach to interpret simulation results. For example, if labor shortage is causing schedule slippage, should the organization hire more workers, employ more recruiters, or both? This type of questions may have been clearly explained and documented in CMM documents. Describing the simulation models and process area definitions in the same language framework facilitates cross-referencing between the two distinct knowledge domains. Modularizing simulation model based on CMM process areas also provides contextual information to simulation results. Describing simulation models and CMM process architecture in OPM allows OPM users to cross reference quantitative...
and qualitative information in an integrated ontological structure. This integration should enhance the systematic accumulation and organization of process knowledge.

5.5 **CMM-OPM as a Holistic Approach to Reduce Rework Generation**

Most CMM process areas are explicitly designed to address problems related to rework generation. For example, Requirement Management and Contractor Management process areas are defined to improve the performance of mission consistency and work quality, respectively. However, knowing that all process areas are directly or indirectly related with each other, the performance of other process areas are likely to influence the amount of rework generation.

Making conceptual linkages between the description model of CMM toward the quantitative System Dynamics model requires knowledge in both subjects. The ideal scenario is to include the Project Model as a part of standard CMM process asset. By absorbing the concept of rework generation as a part of CMM standard process asset, it will encourage more CMM implementations to utilize process simulation as a part of their project planning practice. The following OPD shows a proposed organization of CMM-based process asset.
Figure 5-7 Proposed CMM Process Assets
6 Managing Vagueness and Uncertainty

Applying quantitative methods to evaluate process performance is an important aspect of CMM practice. Realistically, quantitative methods must deal with vague, incomplete and uncertain information while still produce meaningful analytical results. In practice, the ability to extract important system-level properties under the influence of vague and uncertain information is highly appreciated. However, incomplete and uncertain information sources introduce significant variability in process models and in term devalue the result validity. The wide range of variability caused by vague and uncertain model information may ultimately cause confusion and doubt for their analytical results. To methodically manage the effects of vague and uncertain variables, this chapter introduces new OPM language constructs to explicitly classify the types and nature of variability.

6.1 Sources of Imprecise Model Information

In a process development organization, many of the modeling variables do not have precise values. For example, the absent rate for worker is an uncertain value, it may only be calculated from the statistics of past events. The numeric value can be influenced by random events throughout the entire project life cycle. On the contrary, the dollar amount of initial budget is usually modeled as a known quantity. It generally does not possess any variability over the project life cycle. The nature of each quantitative variable conveys important process information and should be distinguished.

There is another source of variability in process modeling. For example, in a large organization, employee morale may be modeled as a quantitative variable, namely employee morale index.
However, its relationship with the number of vacation days may only be approximated with some forms of algebraic formula. When this algebraic formula does not precisely reflect the actual situation, it creates additional confusion and variability in process models.

6.2 Improving the Semantic Fidelity of Dynamical Process Models

In the language schema of System Dynamics, there is no formal distinction between deterministic and non-deterministic variables. (Depends on the modeling software, some notion of fuzzy and probabilistic variables can be expressed in embedded equations. Business Dynamics, P.529) Modeling experts must keep track of variable types in their mind and perform professional judgement on every single model change. The lack of formal distinctions to separate variable types hides many crucial assumptions made during the construction of simulation model. The hidden assumptions may eventually get completely lost and there translate into loss of process knowledge. The semantic framework of quantitative models should prevent this type of knowledge loss. This chapter proposes a formal mechanism to categorize the types of variables. The author argues that by classifying the variables into three distinct types, it will provide a systematic mechanism to identify vagueness and uncertainty in a model. The author will also discuss how to utilize the language features of OPM to represent the different types of variables.

6.3 Vagueness and Uncertainty as Two Distinct Sources of Inaccuracy

In a given model, if all variables are deterministic, then, the model should perfectly predict the behavior of the system being modeled. For example, calculating the interests earned in a fixed rate money market account can be modeled in a dynamic model, then, the calculated future value should perfectly match the final outcome. However, if the interest earned is based on certain
stock performance index, then, the chances of exactly predicting the outcome will be slim. In the later case, we can identify the source of uncertainty comes from a particular variable, the stock performance index. Assuming a formal process model under investigation is sufficiently complete and technically sound, the sources of error should be coming from the uncertain and vague variables. However, without assigning a formal mechanism to distinguish them from each other, it is difficult to identify the source of errors.

6.3.1 **Deterministic Variables**

Quantitative and qualitative variables can be modeled as named states (values) in OPM. If the issues of vagueness and uncertainty are ignored, at any given time point, an Object can only be at one particular state. When the number of possible states is small, each state can be assigned with a specific name, otherwise, the state can be represented as an integer or real number. The following OPM representation shows an object with two possible values.

![Process Area](image)

**Figure 6-1 Deterministic Variable in OPD**

Process Area can be **activated** or **deactivated**.

Process Implementing changes Process Area from **deactivated** to **activated**.
This example implies that these two states have distinct properties, a Process Area is either activated or deactivated. It also shows that “Process Implementing” will deterministically change the state of the object. Although, this variable type is commonly accepted in both technical language and natural language, the implication of deterministic variable is extremely strict. For example, after the execution of “Process Implementing”, the Process Area must be activated without any room for failure or incompleteness. This strict statement is rarely true for arbitrary processes. While modeling a human process, using pure deterministic variables cannot practically express the uncertainty issues that exist in every instance of process execution.

6.3.2 Probabilistic Variables
As mentioned earlier, in the process of modeling human organization, the notion of distinct properties and deterministic state changes are not always guaranteed. For example, before a Process Area undergoes the Process Inspection process, the outcomes can not be determined ahead of time. Although the certified and not-certified states are clearly distinguishable, the outcome is dependent on the probability distribution of these two states. The following OPD/OPL pair shows an attempt to represent this type of uncertainty.
Process Area can **probably** be **certified** or **not-certified**.

**Process Inspecting** **potentially** changes Process Area from **not-certified** to **certified**.

Notice that two keywords are added to OPL. The term, "probably", labels "certified" and "not-certified" as two uncertain states. They need to be accompanied with probability information. The term, "potentially", is a modifier for the "Process Inspecting" process. It labels this process to have uncertain outcome.

This variable type provides an important language construct to define uncertainty. However, it assumes that the outcomes can only be one of many possible values. In the example above, the final state of Process Area is either certified or not-certified. There is no state that possesses a joint membership of the two distinct states. This probability variable language construct only specifies discrete membership for each possible state. For organizational modelers, who deal with vague information constantly, probability variables are not semantically accurate to represent the vagueness across discrete membership states. To further enrich the semantics of OPM, the following section describes a new language construct in OPM, fuzzy variable.
6.3.3 Fuzzy Variables

Another source of model inaccuracy comes from the vagueness of state classification. For the purpose of human communication, graded values are usually given distinct names. These distinct classes of variables are also called “linguistic variables”. [Mathematical Principles of Fuzzy Logic] For example, a Process Area can be classified into two modes, efficient and inefficient. These two modes are distinguished by their degree of efficiency. The degree of membership for an operation to be classified in either class is specified by a membership distribution function. This variable type can be modeled in OPM as follow:

![Diagram of Fuzzy Variables in OPD](image)

**Figure 6-3 Fuzzy Variable in OPD**

*Process Area* can be *fuzzily efficient* or *inefficient*.

*Process Improving* *incrementally* changes Process Area from *inefficient* to *efficient*.

Notice that a new OPD notation is needed to distinguish fuzzy variable from deterministic variables. To provide a good grounding metaphor of fuzzy values, the overlapping Venn diagram is used to visualize the joint membership across multiple states. (This new notation is based on in-person discussion with Dr. Dori.) Two key words are introduced to OPL to distinguish fuzzy variables from non-fuzzy variables. The term, “fuzzily”, is used as a modifier to label the
following named states as fuzzy variables. The term, “incrementally”, is used to label the process as a graded attempt to change the fuzzy states.

One must notice that the fuzzy variable construct is a super set of all variable types. Probability and deterministic variables are special cases of fuzzy variables. For instance, probability variables can be represented in fuzzy variables with disjoint membership across value classes. Similarly, deterministic variables can be modeled as fuzzy variables with a single value class. Having a shared presentation of variable types resolves a crucial issue in knowledge representation.

6.4 Managing Imprecise Quantitative Relationships
Whenever a precise algebraic equation cannot be found between two quantitative variables, process modelers may desire to create a manually constructed function (lookup table) to approximate the quantitative relationships. For example, in the Project Model created by James Lyneis, he uses an S-shaped function to approximate the relationships between relationships between two variables, “Elimination of Uncertainty Based on Progress” and “Fraction Perceived to be Complete”. The value for each point on this lookup table is manually assigned to accommodate the overall model creates reasonable simulation results.
Using lookup tables to approximate functional relationships is adequate for simulation models that do not require frequent changes. As the number of manually constructed lookup tables increases, it becomes increasingly difficult to interpret the effects on the entire simulation model. Additionally, when the functional relationships involve multiple variables, assigning multidimensional numerical values or graphically editing a multidimensional functional graph should prove to be challenging and avoided if possible.

6.4.1 Use Linguistic Variables to Encode Imprecise Functional Relationships
Fuzzy logic is perfectly suited to handle imprecise functional relationships. The quantity of each variable can be encoded as linguistic variables that maps to meaningful concepts based on the application context. Using the computational rules of fuzzy logic, the variables can be translated
between continuous domain and the linguistic variable domain. The following example shows an output variable driven by two fuzzy inputs.

![Diagram of Fuzzy Variable Relationships]

**Figure 6-5 OPD of Fuzzy Variable Relationships**

This example shows that “Product Quality” is dependent on two fuzzy variables, “Employee Morale” and “Quality of Requirement Document”. For the purpose of demonstration, Product Quality is set to increase from “low” to “high” with the improvement of “Employee Morale” and the “Quality of Requirement Documentation”. The following fuzzy rules, written in OPL should encode this quantitative relationship.

- **If Employee Morale is up and Quality of Requirement Documentation is good, then Product is high.**
- **If Employee Morale is up and Quality of Requirement Documentation is ok, Product Quality is high.**
- **If Employee Morale is down and Quality of Requirement Documentation is bad, Product Quality is low.**

These rules can be used as inputs to a typical fuzzy logic processor that translates linguistic variables into numeric quantities. (The following diagrams are produced with MATLAB’s Fuzzy Logic Toolbox.)
This example shows that using linguistic variables to encode the lookup table has multiple advantages. It allows modelers to represent the quantitative relationships in terms of fuzzy rules. These rules provide meaningful contextual information that may help future users of the process model to interpret lookup tables properly. Using linguistic variables and membership functions to edit lookup table is a systematic way to parameterize complex functional relationships. Representing imprecise functional relationships with a set of fuzzy logic parameters simplifies the reproduction and modification of the functional relationships. This simplified mechanism
should also contribute to modeling efficiency. For example, the “Quality of Requirement Document” can be edited in a standard input form shown as below:

![Membership Function Editor in Matlab’s Fuzzy Logic Toolbox](image)

Figure 6-7 Membership Function Editor in Matlab’s Fuzzy Logic Toolbox

This three-variable fuzzy logic example also demonstrated that the new OPM language constructs can adequately represent these fuzzy rules. Having shown that OPM can adequately display the fuzzy rules in both diagrammatic and textual forms, process modelers may directly use OPD or OPL to specify fuzzy rules without learning additional programming languages.

6.5 Separation of Quantity and Quality

Once variables can be classified as deterministic, fuzzy and uncertain types, each type also needs to be characterized with additional information. For example, an uncertain variable, Mean Time Between Failure (MTBF), of a certain computer is 10,000 hours. The standard deviation of this variable may be 300 hours. The probability distribution pattern of this variable is a normal distribution. In this case, the number, “10,000” only specifies partial information of the variable. The type of random distribution and its standard deviation further describes the quality of this
variable. This example shows that uncertain variables require more than one number to specify its quantity and quality. By applying a variable typing mechanism, OPM leverages its language infrastructure to systematically organize the representation of non-deterministic variables. The following OPD shows the OPM model of these three variable types.

![Variable Types in OPD](image)

**Figure 6-8 Variable Types in OPD**

### 6.6 Variable Typing Enhances Model Completeness and Consistency

This chapter presented a systematic way to characterize the nature of quantitative variables. It also presented the OPM schema to define the properties of each variable type. To make each variable type to be visually distinguishable in OPDs, specific graphical notations are introduced to make the best attempt for visual clarity and meaningfulness. Making distinctions of variable types on the language construct level provides a traceable mechanism to isolate the sources of inaccuracy. The typing information of each variable can be mechanically searched to provide automated classification of error sources. Once the sources of inaccuracy are mechanically identified, it will be relatively easier to manipulate models to produce expected results. Having
the ability to isolate inaccuracy shall significantly reduce the model validation effort. Another benefit of classifying variable types is that modelers can provide a standard modeling interface to parameterize each category of variables. It allows modelers to parametrically adjust all families of variables in a consistent manner. This approach will streamline the management of model parameters and therefore improve model reusability.
7 OPM-based Knowledge Management Infrastructure

One can broadly categorize CMM as a guideline to systematically manage process knowledge in product development environments. However, to conduct effective knowledge management, the conceptual model must be supported by tangible infrastructures. In this chapter, the author will describe an OPM-based information architecture that is specifically designed to help manage process knowledge for the CMM community.

7.1 Information System as a Necessary Instrument for Knowledge Management

There are many information systems that serve the purpose of knowledge management for large and small organizations. For example, the news groups extensively employed by the open source community have proven to be invaluable for knowledge sharing and retention. The success of these grassroots knowledge management systems indicates the potential of employing similar systems to help manage the knowledge of CMM practices. In ideal cases, properly designed information systems are an indispensable element of organizational learning. It can serve as an instrument to transform the regular activities of an organization. If executed properly, a good knowledge management system can help expedite the evolution speed of an organization and attain higher process maturity level with less time and resources. However, building a knowledge management system involves many aspects of information system design and implementation challenges. This chapter will show that OPM can be used as a key ingredient to reduce obstacles in the implementation of a knowledge management system.

7.2 Designing a CMM Knowledge Management System based on OPM

Most Knowledge Management (KM) approaches are not suited for process knowledge management because their software architectures are tightly coupled with specific use-case
scenarios. For example, managing a portal-like website that help users navigate relevant process document is one way of disseminating process knowledge. However, this approach narrowly defines knowledge as a collection of process document and bound the structure of knowledge to the content navigation mechanism. If the navigation system is designed poorly or statically, the knowledge presentation becomes stale to its users. Ultimately, knowledge management systems that do not flexibly reflect users’ needs are used as automated document management systems. To fully explore the potential of knowledge management systems, knowledge management systems must have a more effective and flexible way to interact with users.

7.2.1 **OPM as an Efficient User Interface for Knowledge Navigation**
OPM is a language framework that provides a consistent syntax and semantics to capture knowledge content of an organization in terms of its process structures and dynamics. The knowledge content can be graphically presented in OPD and textually displayed as OPL. OPD is a simple diagram format that consists of boxes, circles and lines, it does not pose a demanding graphical display requirement. OPL is a textual language, which also presents very little demand on the computing resources of the end user. The combination of OPD and OPL is highly expressive and can be used as a user interface to convey navigational information. Ideally, users can construct query statements using standard OPD or OPL as the query input mechanism. Given the current state of web-based technologies, it is trivial to implement network-deployable applets that displays OPD and OPL in popular web browsers. Deploying OPD and OPL display/editing applets in browser-based technologies has a distinct advantage. Browser based network applications leverages the resources of popular network infrastructure that reduces the overhead
of machine-specific installation and maintenance. A more detailed implementation plan on OPL/OPD interface can be found on Christine Miyachi's master thesis [Miyachi 2000].

7.2.2 Related Implementation Efforts
Displaying process models graphically and use the visual presentation of the process model as a navigation mechanism has been implemented by John Quimby at MIT's Process Handbook research group [Quimby 2000]. The process handbook is a knowledge base of dynamic process models and associated process documentation. It includes the intelligence to traverse related models based on the object types. It also has the ability to manage continuous and discrete process dynamic models. An effort to integrate user interface, simulation engine and process model storage (Process Handbook) is currently underway. Based on personal communication with Quimby, using OPM as a standard language framework to navigate a large collection of process model is technically feasible.

7.3 The Architecture of a CMM Knowledge Management System
To ensure a seamless application of CMM-based process knowledge to the daily operation of the underlying organization, the formal process model should be tightly integrated with the organization’s Enterprise Resource Planning (ERP) system. Ideally, given that CMM organizational architecture and process dynamics can be formally captured in OPM, it should be used as the modeling language for ERP systems. If an organization uses OPM to capture the procedural information of organizational activities, it would be possible to automatically translate these OPM-based procedure descriptions into ERP system specifications or process flow configurations. More importantly, having a single language that describes the organization from its high-level business objectives to the lower level automated activities, should improve its
internal communication efficiencies and therefore promotes the sharing and verification of organizational knowledge.

The following diagram shows the overall architecture of an OPM-based knowledge management system.

![Figure 7-1 High Level Architecture of OPM-based Knowledge Management System](image)

7.4 **A New Way to Study Evolving Organizations**

According to CMM, organizations classified with higher maturity levels are more capable of dealing with continuous improvements and adapt to constant structural and process changes. Instead of focusing on the management recipe for each change scenario, this paper argues that an efficient process modeling language is a required cognitive instrument to enable organizations better understand the cause and effect of changes. It further demonstrates that a flexible and expressive language such as OPM can serve multiple purposes in evolving organizations.
paper identifies the essential aspects of organizational evolution and finds appropriate OPM constructs to formally represent their properties. Since OPM can simultaneously represent multiple aspects of a system, it allows organizations to describe and examine the cause and effect of evolution from a holistic viewpoint. Using OPM to describe organizational processes and structures establishes a common linguistic foundation throughout the entire organization. Properly deployed language standard should enhance the communication efficiency and efficacy of an organization. Conversely, the quality of organizational processes may be limited by the quality of its language framework and its delivery mechanism in an organization. A possible future research direction is to apply cognitive sciences to assess the evolutionary potential of an organization based on its working language framework.
8 Conclusion

In this paper, the author introduced four independent modeling concepts in an integrative manner. It first illustrated OPM as a linguistic foundation of organizational modeling. The main advantage of OPM is that it possesses a concise and sufficient collection of operators and operands in its language framework, which is required to express the essential properties of an evolving organization. To complement the expressive power of OPM, CMM offers a set of well-defined vocabulary and process architecture that has been refined to cover a large class of product development issues and conceptual frameworks. CMM’s vocabulary and its process architecture reduce the need for organizational modelers to seek process management related domain knowledge from the ground up. The concept of Rework Generation and its originating modeling technique, System Dynamics, fills a gap in providing a quantitative assessment methodology to systematically evaluate organizational performance. This paper also showed that the dynamic structure of Rework Generation and the semantics of System Dynamics in general can be fully expressed in the OPM language framework for consistent and effective simulation model management. To further improve the management and manipulation of organizational performance models, categorizing quantitative variables into crisp, fuzzy and uncertain types is also presented as new constructs in the OPM language. This paper further demonstrates that a formal language framework can be used as an underlying infrastructure for knowledge management on an organizational level. It also discussed why OPM’s language features are flexible and expressive enough to take on such technical and organizational challenge.
In summary, this paper points out that the use and integration of language frameworks directly reflects an organization’s ability to capture organizational experience, conduct interactions and assess future states. Therefore, without adopting a language framework that possesses the qualities of OPM and its extensions presented in this paper, migrating an organization toward higher levels of capability maturity will likely to remain as a form of art.
9 Bibliography