

December 1998

# The Representation and Use of Process Knowledge in the Design of Information Systems

Kishore Sengupta  
*Naval Postgraduate School*

Follow this and additional works at: <http://aisel.aisnet.org/amcis1998>

---

## Recommended Citation

Sengupta, Kishore, "The Representation and Use of Process Knowledge in the Design of Information Systems" (1998). *AMCIS 1998 Proceedings*. 397.  
<http://aisel.aisnet.org/amcis1998/397>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1998 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# The Representation and Use of Process Knowledge in the Design of Information Systems

Kishore Sengupta  
Naval Postgraduate School

## Abstract

*The design of an information system entails making a series of interdependent decisions. These decisions result in the creation of design artifacts such as drawings, prototypes, and design documentation. Current approaches toward system design focus on the creation of the artifacts, rather than capturing the accompanying process knowledge, such as the rationale underlying their design. The loss of process knowledge can frequently cause problems downstream, e.g., in subsequent use or modification of the artifact. In recognition of the value of process knowledge in the system development lifecycle, much work has been done in the recent past on developing methods for the capture and use of process knowledge. Many of these methods have been incorporated in recent versions of CASE tools.*

*This tutorial will provide an overview of the theories underlying the representation and use of process knowledge, the principal methods employed, and the implementation of the methods in CASE tools. We will also discuss factors that constrain the capture and use of process knowledge in organizational settings. The material covered in the tutorial will be illustrated through a stylized example, abstracted from an actual design situation.*

## Motivation for the Tutorial

The design of systems is a deliberative process, resulting in a series of interdependent decisions. The results of these decisions are visible through artifacts such as drawings, prototypes, or design documentation. These deliberations result in a great deal of *process knowledge*, i.e., knowledge underlying the creation of the artifacts. Process knowledge can consist of multiple components, e.g., the rationale behind the decisions, the task/sub-task structures that lead to artifacts, and relationships among participants.

The capture and maintenance of process knowledge is important in the development and maintenance of large scale systems, for several reasons. First, the availability of the rationale behind the design of artifacts can make an important difference in the subsequent use and/or modification of the artifact. Second, the availability of reliable records of the rationale behind design decisions enables work groups to avoid revisiting the same issues. Third, large projects typically entail changes in development teams over time and across phases. Process knowledge serves to facilitate communication and retain the context in which design decisions are made. Finally, the retention of a corporate history of rationale enables design teams to review past decisions for identifying past errors and avoiding repetitions.

However, current approaches to system design stress the representation of artifacts, rather than the process knowledge underlying their creation. As a result, much of the knowledge involving deliberations on alternative requirements and design decisions is typically lost in the course of designing and maintaining systems. From an *efficiency* perspective, this can result in repetitions of previous design deliberations, causing much process loss. From an *effectiveness* perspective, a lack of access to process knowledge can result in prior design decisions being misunderstood or misinterpreted. In consequence, the team may commit design errors that could have been otherwise avoided.

A recognition of the importance of retaining process knowledge has led researchers and practitioners to seek methods and tools for the capture and use of process knowledge. There have also been studies of the use of these tools in system development. In addition, new versions of software development tools increasingly contain facilities for capturing process knowledge.

The objective of this tutorial is to provide an overview of the current state of research and practice in the capture and use of process knowledge for developing information systems. In addition, I will also seek to provide a perspective on the future evolution of research and practice in this area, as well as highlight the important issues for IS researchers.

## Audience for the Tutorial

The topic is germane to system development in general, and can be profitably attended by anyone with a strong interest in this area. The audience to whom the topic is of potentially the greatest interest includes researchers and practitioners seeking better methods for system development.

## Outline of Tutorial

1. Introduction: What is process knowledge?
  - Definition/s
  - Examples (throughout the tutorial, examples will be interleaved with the material, rather than being presented separately).
2. Why capture process knowledge?
  - Improve the effectiveness of the system design process
  - Improve the efficiency of the system design process
  - Examples
3. Capturing process knowledge: components, representation, methods
  - Components of process knowledge; examples
  - Representation of process knowledge: choices and trade-offs entailed in formal and informal methods
  - Methods of capturing process knowledge:
    - Argumentation methods, e.g., REMAP, IBE; examples
    - Questions, Options, and Criteria, e.g., QOC; examples
    - Decision theoretic methods, e.g., SIBYL; examples
4. Implementing process knowledge: tools
  - Brief descriptions of available tools: SLATE, RDD-100, Teamwork/RQT
  - Demonstration of RDD-100
5. Constraints: technical, organizational
  - Technical constraints:
    - Limits to the amount of context information that can be reasonably captured
    - Ease of capture of process knowledge, versus computational tractability
  - Organizational constraints:
    - Retention of process knowledge is an upstream activity, whereas the benefits occur downstream
    - Design of suitable incentive schemes for encouraging the capture and use of process knowledge
6. Future Directions in research and practice

### *Biosketch of Presenter*

The tutorial will be conducted by **Kishore Sengupta**, associate professor of Information Systems, Naval Postgraduate School, Monterey, California. In 1996-1997, he was a Visiting Scholar at the Hong Kong University of Science and Technology. Among his research interests is the use of process knowledge in organizations. Dr. Sengupta's research on process knowledge has been conducted in the context of the design of multimedia systems, conflict resolution in concurrent engineering, and product development in virtual teams. He has consulted with IT organizations on the use of process knowledge in system development. Dr. Sengupta has published in leading journals on management and information systems, including *Management Science*, *MIS Quarterly*, *IEEE Transactions on Software Engineering*, *IEEE Transactions on Systems, Man, and Cybernetics*, *IEEE Transactions on Engineering Management*, *Omega*, *Decision Support Systems*, *Accounting, Management, and Information Technology*, and *Concurrent Engineering Research and Applications*.