Chapter 6
Knowledge Management as the Basis of Crosscutting Problem-Solving Approaches

6.1 Introduction

In Chap. 2, we argue that supply chain configuration is one of the principal supply chain management decisions and that it has a profound impact on other subsequent managerial decisions. As described therein, the supply chain configuration problem is a complex problem, which is composed of several sub-problems. It is also emphasized that the solutions to these problems require design, modeling, and problem-solving techniques based on knowledge from various fields such as systems science, systems engineering, operations research, industrial engineering, decision sciences, management science, statistics, information sciences, computer science, and artificial intelligence. Some of the prominent techniques utilized from these fields are information modeling, process modeling, simulation modeling, data mining, and optimization. We build on this proposition by adopting a key problem of information integration in the supply chain, which has an embedded structure representing various sub-problems, and how its management relates many of the concepts espoused in this book about supply chain configuration. Also, this problem serves as a prime example of how crosscutting approaches drawn from various disciplines highlighted above may be adopted in devising solutions for the complex supply chain configuration problem. Before we proceed further, let us first develop a clear understanding of the information integration problem in the supply chain.

The Supply Chain and the Information Integration Problem. One way to look at a supply chain is as an alignment of firms that bring products or services to the market (Lambert et al. 1998). This alignment is in the form of an extended enterprise, where firms collectively organize the supply, production, and distribution of products and services.

The management of such a complex organization can be brought to the integration of its business processes. Process-oriented management vs. function-oriented management is an important feature that makes the supply chain a distinct enterprise system class. Another facet of supply chain system complexity is its

C. Chandra, J. Grabis, *Supply Chain Configuration*,
DOI 10.1007/978-1-4939-3557-4_6

organizational dynamics and operational specifics. Organizational dynamics assumes frequent changes in organizational structures such as control hierarchy, goal structure, members' network, and so on. Operational specifics are mainly related to the uncertainty in which a supply chain organization operates. Integration of supply chain processes assumes additional complexity when the decision-making mode (i.e., centralized vs. decentralized) is considered in the mix.

One of the key issues in managing a supply chain process is information integration among its constituents. To facilitate this integration, supply chain information resources ought to be effectively organized and shared. Information integration provides channels that convey information from one supply chain constituent to another. One form of this problem involves the integration of existing implementations that have been built in heterogeneous infrastructures, such as different hardware platforms, operating systems, and database management systems. Presenting the data on which applications perform in a uniform, self-consistent way ensures that they share the same view of the supply chain. Another form of integration is concerned with working collectively on common problems by sharing an understanding of the problems' reasoning logic and applying best practices. This provides a common architecture in information sharing so that supply chain members' collaborative activities provide performance improvement to each member and to the entire supply chain.

The problem of process integration, and its surrogate information integration described above, needs an appropriate solution. In this context, we advocate the necessity of applying system principles and knowledge management methodologies based on the following reasons: (1) the extent of knowledge becomes intractably large, (2) business units are geographically decentralized but more closely networked, (3) collaboration among individual workers is important, and (4) challenges are faced in eliciting requirements when user partners are large, decentralized, and unknown.

In this chapter, we propose a framework and implementation mechanisms for designing a knowledge management system capable of supporting organizational dynamics and operational uncertainty, as well as facilitating process integration in a supply chain. Taxonomies and ontologies are viewed as a means for conceptualizing the knowledge to share and utilize in decision-modeling applications. They bring formalism into the knowledge management system, thus offering standards for communication, which is necessary for collaborative problem solving.

The general trend in process integration is to develop information models that system users can share, thereby sharing the same view of the world (CIMOSA (Kosanke 1995), TOVE (Gruninger et al. 2000), Supply chain ontology (Grubic et al. 2011). The gap seen in these and other research efforts is the absence of a system reference model that can identify information model components and define mechanisms for their design and implementation. This reference model formally represents the source system, such as supply chain, its informational needs, and constructs that need to be built to support system processes. The ultimate target and value of the proposed approach, and hence the reference model is taxonomy and ontology development as a platform for integrated supply chain knowledge management. That is particularly important in the era of web-based supply chain collaboration and cloud computing, where the reference models and integration standards provide the basis of knowledge sharing and supply chain integration (Huang and Lin 2010).

Further, the proposed reference model enables fulfilling the purpose of this chapter in laying the ground work for integrated solutions proposed in Chaps. 7–9. The importance of this chapter is to highlight that solutions to supply chain configuration problems must integrate complex modeling and analysis techniques drawn from a host of disciplines.

The chapter is organized as follows. Section 6.2 describes the motivation, focus, and significance of crosscutting approaches. Section 6.3 discusses the notion of taxonomy and ontology and how it contributes to system integration. Section 6.4 introduces the knowledge management system development framework, which starts from the source system, goes through system component decomposition, and presents knowledge-modules design for these components. Section 6.5 formally presents the knowledge management system reference model, relating elements in the proposed framework and describing their meaning. Section 6.6 presents four stages of the knowledge management system development life cycle, as well as describes how the reference model can be implemented for each stage.

6.2 Crosscutting Approaches: Motivation, Focus, and Significance

Supply chain configuration draws from an array of fields as far as framework, models, and methodologies are concerned. This is primarily owing to the impact of any configuration effected on a supply chain, on its strategic, tactical, and operational decision-making environments. In this section, we discuss the motivation behind developing an integrated supply chain configuration framework and a reference model for designing knowledge and its management, with the aim of improving supply chain management.

6.2.1 Motivation and Focus

The motivation and focus of the research methodology proposed in this chapter is to integrate various problem-solving approaches from a host of fields in the design of proposed supply chain configuration problem-solving methodologies. It is characterized by two main purposes: general and specific.

The *general* purpose is to develop a common body of interdisciplinary knowledge to understand issues and problems related to reconfigurable systems.

The *specific* purpose is to (a) develop methodology and tools for supply chain reconfiguration, (b) elaborate framework for knowledge-based problem analysis and model building, and (c) quantify factors influencing supply chain reconfiguration.

The general problems in reconfigurable systems can be classified as related to the system's environment, availability of appropriate modeling tools, interconnectedness of decisions at various levels of supply chain, and availability of common knowledge throughout the system. These can be listed as follows:

- Increasing competitive pressures and consumer focus requires innovative supply chain modeling and management tools.
- Supply chain modeling tools must capture complex interactions within the supply chain.
- Supply chain configuration decisions have significant impact on other decisions at all levels.
- Knowledge assumes a critical role in a firm's success, and, therefore must be captured, organized and utilized effectively.

Problem-solving strategies applied to reconfigurable manufacturing systems entail developing (a) domain independent solution(s) templates at the macro level, (b) capability models for application specific domain dependent problems at the micro level, and (c) coordination models to integrate models developed in (a) and (b).

6.2.2 Problem Solving for Configurable Systems

To provide an integrated overview of interconnectedness of crosscutting research areas for configurable systems, three problem-solving approaches are proposed: systemic, reductionist, and analytic. These are defined as follows:

- *Systemic Approach* This incorporates the abstract level. This *level of inquiry* deals with issues of scalability of system, meta-modeling of systems, and defining the dynamic knowledge problem domain model.
- *Reductionist Approach.* This incorporates the activity level. This *level of inquiry* consists of dynamic knowledge problem domain model, internal state, and goals and objectives of the enterprise *units,* (producer, plant, department, supplier, vendor, etc.), and strategic management models.
- *Analytic Approach.* This incorporates the implementation level. This *level of inquiry* consists of internal state, goals and objectives of the enterprise units, strategic management models, and shared goals and objectives of the enterprise.

We discuss below related research in direct comparison to these three problem-solving approaches.

At the systemic level, a supply chain is a general class of system that exhibits a cooperative behavior within its business and market environment (Klir 1991).

The foundation of this system is built on a network architecture that has various demand and supply nodes as it provides, as well as receives goods and services to and from its customers and suppliers, respectively (Chandra 1997; Lee and Billington 1993; Swaminathan et al. 1998; Bellamy and Basole 2013). Supply chain system frameworks describe general foundational elements of integration between its marketing and production functions. These are in the form of general theories, hypotheses, standards, procedures, and models that are based on well-founded principles in these disciplines (Cohen and Lee 1989; Deleersnyder et al. 1992; Drew 1975; Graves 1982; Hackman and Leachman 1989; Lee 1993; NIST 1999; Tzafestas and Kapsiotis 1994; Younis and Mahmoud 1986). Systems modeling deals with general modeling issues of this class of systems, such as how to represent, quantify, and measure cooperation, coordination, synchronization, and integration (Little 1992; Morris 1967; Pritsker 1997). Systems engineering describes methodologies for structuring systems as these are implemented in various application domains (Blanchard and Fabrycky 1990). System integration deals with achieving common interface within and between different components at various levels of hierarchy in an enterprise (Shaw et al. 1992), as well as different architectures and methodologies (ISO TC 184/SC 5/WG 1 1997; IMTR 1999; Hirsch 1995), using distributed artificial intelligence and intelligent agents (Gruber 1995; Stumptner 1997; Wooldridge and Jennings 1995).

At the reductionist level, a supply chain configuration must be based on its local, as well as global, environmental constraints. These constraints are partly imposed as the supply chain negotiates and compromises to adapt to its cooperative behavior (Jennings 1994). Enterprise modeling as a technique has been used effectively in decomposing complex enterprises, such as a supply chain. Ontologies are defined to describe unique system descriptions of supply chains that are relevant to specific application domains (Gruninger 1997). The classic problem for a supply chain is an inventory management problem requiring coordination of product and information flows through a multi-echelon supply chain. This class of problem has been solved by integration of the front and back ends of the supply chain with costs and lead times as key measures of its performance (Clark 1972; Clark and Scarf 1960; Diks et al. 1996; Diks and De Kok 1998; Hariharan and Zipkin 1995; Pyke and Cohen 1990).

The analytic approach for the general class of supply chains has its origins in economic models of supply and demand coordination. Game Theory principles for payoffs among market competitors have been used effectively to design competitive strategies for supply chains (Gupta and Loulou 1998; Masahiko 1984). Coordination and cooperation—dealing with interfaces between strategies, objectives, and policies for various functions of an enterprise, has received much attention in optimizing the performance of a supply chain (Malone and Crowston 1994; Thomas and Griffin 1996; Whang 1995). Various aspects of cooperation have been prescribed for effective management of supply chains (Sousa et al. 1999).

Starting from the evaluation of existing enterprise integration architectures (CIMOSA, GRAI/GIM, and PERA), the IFAC/IFIP Task Force on Architectures for Enterprise Integration has developed an overall definition of a generalized architecture framework called GERAM or Generalized Enterprise Reference Architecture and Methodology (ISO TC 184/SC 5/WG 1 1997).

6.2.3 Significance of This Approach

Supply chain management strategies have the potential of enabling smart manufacturing and services

- Adaptable, and integrated equipment, processes, and systems that can be readily reconfigured.
- Manufacturing processes that minimize waste.
- System synthesis, modeling, and simulation for all manufacturing operations.
- Technologies to convert information into knowledge for effective decision-making.
- Software for intelligent collaboration systems.
- New educational and training methods that enable the rapid assimilation of knowledge.

The common thread in the deployment of these technologies is achieving (a) reconfigurability, (b) efficiency, and (c) complex modeling and analysis in decision-making related to managing advanced manufacturing systems.

This emphasis on developing enhanced manufacturing capabilities and technologies to support infrastructure mandates research in following crosscutting areas

- Adaptable and reconfigurable manufacturing systems.
- Information and communication technologies.
- Processes for capturing and using knowledge for manufacturing.
- Adopting and incorporating IT into collaboration systems and models focused on improving methods for people to make decisions, individually and as a group.
- Enterprise modeling and simulation.
- Analytical tools for modeling and assessment.
- Managing and using information to make intelligent decisions among a vast array of alternatives.
- Adapting and reconfiguring manufacturing enterprises to enable formation of complex alliances with other organizations.

The objective of the research presented in this chapter is to formalize the capture and management of supply chain management knowledge accumulated in various domains of science, engineering, and technology, and using various problem-solving techniques.

6.3 Taxonomy, Ontology, and System Integration

To see linkages between problems and decision-making models utilized in a complex enterprise such as a supply chain, it is imperative that these components be formally represented. Taxonomy and ontology provide the means to classify the supply chain problems and represent formal knowledge, which is used in decision-making. We take up discussion on this topic next.

6.3.1 Taxonomy

According to the American Heritage® Dictionary of the English Language, Fourth Ed. (2000), taxonomy is the classification of organisms in an ordered system that indicates natural relationships. It is the science, laws, or principles of classification. Further, it is an arrangement by which systems may be divided into ordered groups or categories according to common characteristics.

System taxonomy reflects information about relationships both inside the system, and with its surrounding environment. Supply chain system taxonomy aims to provide a multidisciplinary representation of supply chain activities and characteristics. The review of research in the field of supply chain taxonomy development reveals that most of them are based on single case studies, providing taxonomy for a subset of information. System taxonomy is organized for the entire system. Organizing information representation for a part of a system or for one problem jeopardizes decision-making because it may miss some key aspects. The supply chain is an organization whose components are interrelated to each other. This cohesion makes the system unmanageable if it is considered as one unbreakable unit. Based on biological classification, system taxonomy provides mechanisms for dividing a supply chain system into relatively independent units, providing as minimal a coupling between units as possible by collecting characteristics in groupings by their similarity. Further, iterative decomposition of groupings and creating new groupings can build a robust hierarchy of describing system characteristics.

System taxonomy serves two purposes: (1) standardization of terms and definition, and (2) unification of information representation. This brings out reusability of developed information models, as well as organization and structure, to knowledge management. Scalability and traceability are the most important features that system taxonomy provides, and thus, new features can be added and existing ones easily found.

6.3.2 Ontology

In the Artificial Intelligence (AI) literature, ontology is defined as the study of the kinds of things that exist (Sowa 2000). In AI, programs and logic deal with various kinds of objects, and we study what these kinds are and their basic properties (McCarthy 2003). Over the years, ontology has become more than an abstract representation of objects and their properties and is becoming a part of the software application domain with application to other branches of AI, such as heuristics and epistemology. The latter is a study of the kinds of knowledge that are required for solving problems in the world, and the former is a way of trying to discover something, or an idea embedded in a program. Along with shaping its pragmatic purpose, ontology has found its application in many fields, such as knowledge representation, system integration, enterprise modeling, conceptual modeling, and Semantic Web.

The above definition of ontology by Sowa (2000), as a study of the kinds of things that exist, is very generic. However, during the last two decades, several features of ontology have evolved that define its broader and more diverse scope and purpose in designing information support for decision-making. A review of the pertinent literature offers the following contrasting definitions and interpretations of ontology to validate our above assertions:

- Ontology is an explicit specification of conceptualization (Gruber 1993), meaning that ontology defines kinds of things, their possible relationships, and plausible implementation.
- Ontology is a catalog of types of things that are assumed to exist in a domain of interest, D, from the perspective of a person who uses a language, L, for the purpose of talking about D (Sowa 2000). This feature of ontology assumes the existence of a language with enough expressiveness for representing the domain of interest.
- Ontology refers to an engineering artifact, constituted by a specific vocabulary that is used to describe a certain reality and by a set of explicit assumptions regarding the intended meaning of words in the vocabulary (Guarino 1995). This definition of ontology adds a new feature requiring that it must have mechanisms and terminology for describing the meaning of words and vocabulary as well as their interpretations.

Ontology as a tool for information modeling has been adopted for a large body of research initiatives. As part of the research described in this chapter, a number of ontology tools, languages, and research projects have been studied to understand the role of ontology in information support systems, particularly for information integration. Some of them, such as Ontolingua (Farquhar et al. 1997) and OntoBroker (Fensel et al. 2001), investigate ontology narrowly as a standalone discipline. Other projects, such as TOVE (Fox and Gruninger 1999; Fox et al. 2000) and DOGMA (Meersman 2001), combine knowledge organization with specific domains, investigating agents for which knowledge is organized. Others look at the problem more widely, including development of ontologies in enterprise modeling systems, such as Enterprise (Stader 1996) and Process Handbook (Malone et al. 1999), aiming to support organizations effectively in change management. What is common in all of these projects, however, is that ontology explicitly defines the vocabulary presented with a language in which queries and assertions are exchanged among users (Grubic and Fan 2010). The next section describes the knowledge management system development stages in a framework.

6.4 Knowledge Management System Development: A Proposed Framework

The framework for knowledge management system conceptualization is depicted in Fig. 6.1. Based on the theoretical background developed in Sect. 6.3, a technique is proposed for conceptualizing supply chain organization and problem knowledge. The proposed advances offer integration of knowledge components with decision support systems and their consumption by software applications or agents. Knowledge components encompass ontology models and the infrastructure supporting their creation, storage, and use.

To serve the needs of a knowledge management system in formalizing and delivering knowledge to decision-modeling applications, several requirements are imposed on ontology conceptualization, such as (1) systematic principles for knowledge conceptualization, (2) the problem-specific nature of ontology constructs, (3) the modularity and object nature of formed knowledge, (4) reusability of created knowledge, (5) integration of distributed data, and (6) machine-readable format of delivered knowledge.

The genesis of the proposed framework is taxonomy and its amplifications to problems and problem-solving techniques, particularly when applied to supply chain management.

6.4.1 Taxonomy Development

Taxonomy is a systematic representation of a system's existence (McKelvey 1982). Accordingly, taxonomy is built based on principles of system theory. It is a mechanism for structuring the knowledge about a certain system domain. The process of taxonomy development consists of information collection, systematic analysis, and classification of system attributes.

Problem taxonomy provides the overall framework under which problem-oriented information system components can be designed and implemented.

Supply chain problem taxonomy comprises: (a) classification of supply chain problems, (b) classification of problem solving methodologies for supply chain

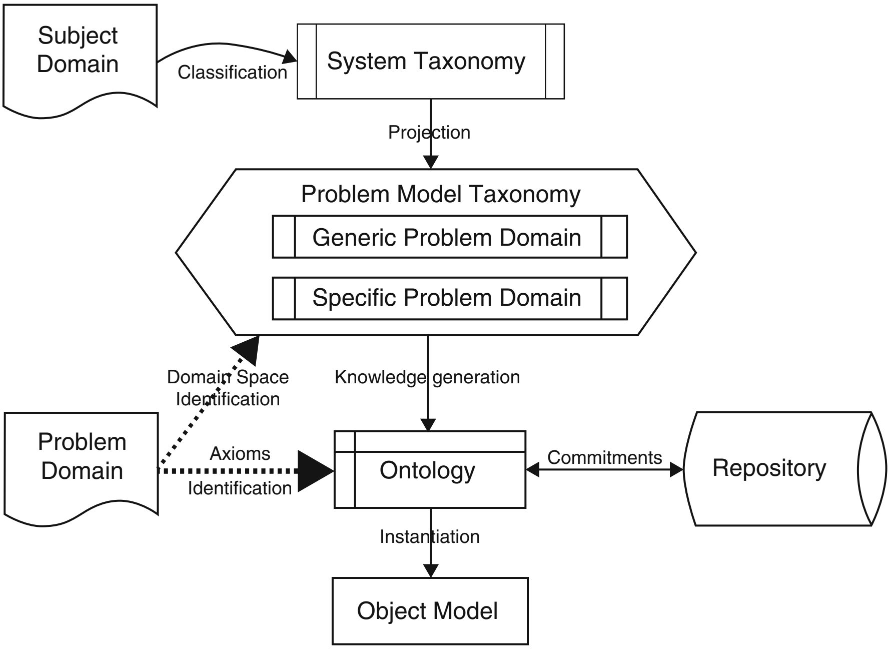


Fig. 6.1 The conceptual framework of a knowledge management system

management, and (c) hierarchical classification of variables or factors necessary for dealing with such problems. We explain this concept with the help of one of the fundamental problems in the supply chain management literature—the bullwhip effect.

The Bullwhip Effect
The most downstream supply chain unit observes an external demand, transmitted up on a supply chain as inventory replenishment orders move from one unit to another. It has been observed that substantial information distortion may occur during this transmission. This information distortion, known as the bullwhip effect, appears as an order variance increase as one moves up the supply chain.

Classification of the Bullwhip Effect Problem
The bullwhip effect is a prime example of problems encountered in a complex system, such as the supply chain. In these systems, problems are multifaceted with a primary problem and many related sub-problems. For instance, the bullwhip effect, one of the fundamental problems in supply chain management literature (Lee et al. 1997a), has several secondary problems, such as order management, demand forecasting management, inventory management, and shipment consolidation. Further, Lee et al. (1997a) formally identify the main causes of the bullwhip effect, while Lee et al. (1997b) discuss their managerial implications. They state that if the following conditions hold—(1) demand is mean stationary and no signal processing

is used, (2) lead time is zero, (3) fixed ordering cost is zero, and (4) no price variation occurs—then the order variance increase does not occur. However, if some of these conditions are relaxed, the bullwhip effect may be observed.

Classification of Techniques
We need to discuss some of the published techniques utilized in managing the bullwhip effect to highlight their classification. Chen et al. (2000a) use the simple moving average forecasting technique to obtain forecasts and investigate the bullwhip effect according to lead time and information sharing. Chen et al. (2000b) and Xu et al. (2001) use the exponential smoothing technique in forecasting. Chen et al. (2000b) also show that, if a smoothing parameter in exponential smoothing is set to have equal forecasting accuracy for both exponential smoothing and moving average methods, then exponential smoothing gives larger order variance. Graves (1999) demonstrates the presence of the bullwhip effect, if external demand, which is the first-order integrated moving average process, is forecasted using exponential smoothing with an optimally set smoothing parameter. Metters (1997) measures the impact of the bullwhip effect by comparing results obtained for highly variable and seasonal demand against the case with low demand variability and weak seasonality. Cachon (1999) proposes methods to reduce the bullwhip effect using balanced ordering.

Problem model taxonomy is a projection of system taxonomy, and thus inherits system structure and vocabulary. A problem domain is presented at two levels—generic problem domain and specific problem domain. Generic problem domain taxonomy is a class of problems that can occur in a supply chain, such as *coordination of production activities.* It is a highly generic problem that comprises several tasks, such as *scheduling of production or inventory replenishment*, which are problems describing more specific issues. Usually, specific problem domain taxonomy is represented by domain-dependent (or specialized) model(s). Splitting problem representation modeling into the above defined two parts provides the means for developing generic and specific problem models.

The process of problem model taxonomy development starts with problem domain space identification. This involves analysis and design of functional requirements for the problem and proposing a structured representation of relevant information. For example, for *a scheduling of production problem*, the model comprises its input and output variables, underlying sub-tasks or activities, tools and mechanisms for solving the problem, problem-oriented goals, roles and agents involved in performing them in accomplishing tasks to achieve identified goals, and external environmental issues. The purpose of problem taxonomy (PT) is the systematic representation of supply chain domain constituents, such as problems and their content.

Different problem models have the same representation format and characteristics vocabulary, thus providing standardization of information representation in the supply chain domain. Problem model taxonomy serves as a meta-model for knowledge model generation and ontology engineering. Ontology inherits concepts, subsumption relationships, and characteristics from the problem model, thus providing consistency in representing various problems. Ontology development components enrich the problem model with constructs, thus turning it from an abstract problem representation into a knowledge model by formulating rules and regulations related to the problem domain. These constructs are: (1) axioms, defining rules specific to the problem domain; (2) algorithms, providing step-by-step procedures for approaching the problem and solving it; and (3) commitments, linking characteristics to data and assigning variables with values. The first two components are modeled through a comprehensive analysis of the problem. System analysis and design techniques, such as process modeling and object-oriented design, are applied for this purpose. The identification of the first two components is the most important part of ontology development. Ontology by itself is a vocabulary with rules on its use. Real world applications require data to operate. Ontological commitments provide these data.

Object model generation is a software engineering practice. If parallels are drawn with software engineering, ontologies can be considered as classes, while object models are their instances encapsulated into software entities. Object models are tangible software constructs, where problem-specific data are represented in a common programming language, encapsulated in a formal model, and accompanied with descriptions of what to do with the data and how to do it.

The next section formalizes the proposed framework with the help of a knowledge management system reference model.

6.5 Knowledge Management System Reference Model

The knowledge management system reference model is proposed as a theoretical foundation for building knowledge-based information systems. It follows various stages in the above-described framework and formally represents its component types, their meaning, and functions. The reference model is divided into three parts: source system representation (system taxonomy), supply chain functional requirements representation (problem taxonomy), and formal knowledge representation (ontology). First, notation to represent the reference model is presented. Next, the reference model is formally enumerated in the form of a set of equations.

Notations related to general problem representation	
S	System
T	Thing symbolizing the elements of a system
R	Relationships among things of a system defined on T
GP	Generic problem model
at_i	Attribute (the index i here and afterwards signifies the ith attribute in the set of attributes)
At_i	Set of instances of at_i attribute
vv_i	Variable that can be assigned to attribute at_i for generic problems
VV_i	Set of possible values that variable vv_i may possess

(continued)

ww_i	System generic state for vv_i, respectively
WW_i	Set of possible states of ww_i
Notations related to specific problem representation	
Ob	Object model
b_i	Observation channel
B_i	Set of possible b_i states
SP	Specific problem model
v_i	Variable that can be assigned to attribute at_i for specific problems
V_i	Set of possible values that variable v_i may possess
w_i	System specific state
W_i	Set of possible states w_i
o_i	Observation channel for attributes at_i
$\tilde{O}$	Relationship between object system and problem system
W	Class instances of S for supply chain domain (general representation of W_i)
Notations common for specific and general problem representations	
$\hat{E}$	Relationship between specific and generic systems
e_i	Relationship between V_i, VV_i
k_j	Relationship between W_j, WW_j (the index j signifies the jth relationship between general WW_j and specific W_j system states as well as between B_j and W_j)
S_w	Specific system for supply chain domain (an instance of S)
T_w	Things specific to supply chain domain (an instance of T)
R_w	Set of relationships held on T_W
Notations for ontology	
M	Data model for a supply chain domain
I	Ontological commitments
V	Set of variables (General representation of V_i)
B_w	Observation channels for defining variables
B_C	Observation channels for defining constraints
B_H	Observation channels for defining algorithms
J	Set of Interpretation functions I
M_w	Data model for a supply chain problem
C	Constraints on data
O	Ontology model
A	Set of axioms
H	Algorithm or heuristics
G	Set of equations

The system consists of interrelated elements. Ackoff (1971) identifies system characteristics, such as an abstract and a concrete system, system state, its changes, and so on. These characteristics guided us during the development of the reference model. System taxonomy is an abstract system whose elements are concepts. Problem taxonomy has two system representation forms—abstract system representation, where problem-relevant elements are presented as concepts: and concrete system representation, where these elements are presented as objects. Ontology presents the states of the system from both static and dynamic perspectives. Ontology also presents system behavior such as response or reaction.

The approach for source abstract system representation is adopted from Klir (1984) and can be formulated as follows:

$$S = (T, R) \tag{6.1}$$

Formally, a supply chain system can be represented as a collection of all possible instances of a generic system applied to a supply chain, with corresponding relationships

$$S = (T, W, R) \tag{6.2}$$

Equation (6.1) is a highly generic and domain-independent system representation. Equation (6.2) is still generic, but is a domain-dependent representation. Only those things and their relationships are considered to exist in system instances W. Equation (6.2) is the system taxonomy formalism, where W is the supply chain domain. A detailed description of the taxonomy of a system in general can be found in Chandra et al. (2007). For each possible system instance $w \in W$, the intended structure of w according to S is the structure (problem classification in problem taxonomy)

$$S_w = (T_w, R_w) \tag{6.3}$$

R_w is the set of extensions (relative to w) of elements of T_w

$$R = \{R_w | w \in W\},\ T = \{T_w | w \in W\}| \tag{6.4}$$

We denote with S the set of all the intended system instance structures of the system

$$S = \{S_w | w \in W\} \tag{6.5}$$

Equations (6.4) and (6.5) reveal that for each system instance w, there is only one system structure S_w with one set of things T_w and one set of relationships R_w. Each S_w is a description of a problem (model) defined as a part of problem taxonomy for which a solution is to be found. S_w contains the names of parameters identified in the problem description, with corresponding relationships organized in a structured hierarchy. Problem model development based on this formalism offers two sublevels of the problem-modeling layer: problem object model and problem formal model.

Problem Object Model and Problem Formal Model

The notion of thing is abstract. To investigate a single thing, we separate it from the outside world, and examine it as an object.

$$Ob = \left(\left\{\left(at_i, At_i\right)\middle| i \in N_n\right\}, \left\{\left(b_j, B_j\right)\middle| j \in N_m\right\}\right) \tag{6.6}$$

where $N_n = \{1,2,\ldots,n\}$ is the number of attributes that the object *Ob* possesses; and $N_m = \{1,2,\ldots,m\}$ is the number of observation channels where attributes are examined and collected. N_n and N_m are the rows and columns, respectively, of a two-dimensional matrix with n rows and m columns. Observation channels are situations, circumstances, processes, narrative descriptions, or any other sources where the problem can be investigated. The set of possible observation channels is denoted by B_j. Different observations where attributes are examined are called backdrops. When investigating a more specific system, backdrops can be considered as situations, where we examine the same attribute. These situations can be subdomains or problems. (at_i, At_i) denotes an attribute and a set of its appearances (possible values that the attribute can possess), respectively. (b_j, B_j) denotes an observation channel and a set of its states, respectively.

Ob is the object (an instance of a thing). Variables are used for an operational representation of an attribute. Each attribute has a name, which is taken from the set of possible values (At_i).

Attributes define two types of variables general and specific for use in general and specific models, respectively. General and specific variables are components of three primitive systems: object system, specific problem system, and general problem system. The last two primitive system representations connect observed domain attributes to real world variables, which this book classifies as ontological commitments. Separation of generic and specific objects is comparative. In some situations, only one problem model is required, while in other cases two or more problem models are necessary to alleviate the complexity by separating problem domains into information models with various levels of abstractions. Two levels of abstraction are discussed: generic Eq. (6.8) and specific Eq. (6.7).

$$SP = \left(\left\{\left(v_i, V_i\right)\middle| i \in N_n\right\}, \left\{\left(w_j, W_j\right)\middle| j \in N_m\right\}\right) \tag{6.7}$$

$$GP = \left(\left\{\left(vv_i, VV_i\right)\middle| i \in N_n\right\}, \left\{\left(ww_j, WW_j\right)\middle| j \in N_m\right\}\right) \tag{6.8}$$

SP contains variables v_i related to a specific problem, and *GP* contains variables related to a general problem vv_i. Both specific and general variables may have sets of states (values vv_i; VV_i) and participate in a set of system states (situations ww_j; WW_j). A specific problem model contains variables related to a set of abstractions (one for each variable), expressing the relationships between specific and general problem systems. It can be called an abstraction channel, which formally can be represented in Eq. (6.9) as

$$\hat{E} = \left(\left\{\left(VV_i, V_i, e_i\right)\middle| i \in N_n\right\}, \left\{\left(WW_j, W_j, k_j\right)\middle| j \in N_m\right\}\right) \tag{6.9}$$

The relationship between the object system and the problem system Eq. (6.10) is expressed by an observation channel consisting of individual observation channels for each attribute in the examined system.

$$\widetilde{\mathrm{O}} = \left(\left\{(At_i, V_i, o_i)\middle| i \in N_n\right\}, \left\{\left(B_j, W_j, w_j\right)\middle| j \in N_m\right\}\right) \tag{6.10}$$

The notion of thing about a particular problem can be formulated as

$$T_w = (Ob, GP, SP) \tag{6.11}$$

Relationships among things can be formulated as

$$R_w \cup \widetilde{\mathrm{O}} \cup \hat{\mathrm{E}} \tag{6.12}$$

Equation (6.12) comprises all possible relationships that may exist in system instance w. To keep the model simple, we will refer to the problem model Eq. (6.3) as the object model and to the set of relationships as R_w. A problem model S_w is an abstract representation of a problem domain—a meta-model. Ontological commitments are for developing a data model out of this meta-model. These commitments are interfaces between abstract problem representation and real world data storage. Rearranging the standard definition, we can define a model M as a structure (S, I), where $S = (T, R)$ is a global structure (standard system definition) and I is an interpretation function assigning elements of T to constant symbols (variables) of V.

$$M = (S, I) \tag{6.13}$$

$$I = (V \to T_w \cup B_w) \tag{6.14}$$

I in Eqs. (6.13) and (6.14) is a function that through observation channel B assigns attributes of T to variables of V. This intentional interpretation can be classified as the first ontological commitment. Observation channels are situations where the system state is captured to observe variables V. These can be process models, where required variables participate. Studying the documented process model may reveal the meaning of variables and where they can be taken from. Another example of an observation channel can be database schema, precisely describing how variables can be queried and stored. The model representation can be used for more general cases:

$$M = (T, W,\ R,\ J) \tag{6.15}$$

These are data models for a system in general, including all of its instances and possible interpretations. Equation (6.15) is not practical, because it will never be implemented for presenting actual system data models. Rather, data model presentations for specific system instances are more practical. If we assume $S_w \in S$, for each instance $w \in W$

$$M_w = (T_w, R_\mathrm{w}, I) \tag{6.16}$$

M_w is a projection of M, but we refer to it as a model, not a model instance, because model M in reality will never be implemented. A model can describe a situation common to many states. The second ontological commitment is the application of logical axioms designed to account for the intended meaning of vocabulary, and assigning constraints to system variables. Ontology can be represented as a continuation of problem representation by adding new features to it.

$$O = (M, A,\ H) \tag{6.17}$$

A is a set of axioms for assigning constraint C to variables through the B_C observation channel.

$$A = (C \rightarrow V \cup \mathrm{B}_C) \tag{6.18}$$

H is a set of algorithms for assigning mechanisms (G) to data model processing through the B_H observation channel.

$$H = (G \rightarrow M \cup \mathrm{B}_H) \tag{6.19}$$

6.6 Development of Components of Knowledge Management System

Ontology development is the implementation of the reference model described in the previous section in capturing its elements, assembling them using a computational language, and storing them in an environment that would facilitate dissemination and usage. Particularly, software tools and techniques will use the developed ontology as part of supply chain decision modeling, the other significant part of supply chain configuration, which is taken up for discussion in Chaps. 7–11 of this book. Various stages of ontology development are described next.

6.6.1 Capture

This stage involves the following activities: (1) identification of key concepts and relationships in the domain of interest, (2) production of unambiguous text definitions for such concepts and relationships, and (3) identification of terms to refer to such concepts and relationships (Uschold and Gruninger 1996). Development of a system taxonomy aims to achieve the first two activities for the supply chain domain in general. Identification of concepts for a specific purpose and scope (i.e., the third activity), is the task for the ontology capture activity. The difference between ontology development from scratch and using system taxonomy is that the latter uses search and navigation in the taxonomy hierarchy to find relevant concepts. Once concepts are chosen, an instance of system taxonomy is created that captures only selected concepts, which is T_w and specified by Eq. (6.11). As was mentioned earlier, for the sake of simplicity, thing T_w will be regarded as the object model Eq. (6.6). Relationships R_w are captured automatically in the form of a taxonomy structure that defines how concepts relate to each other.

The knowledge management system conceptualization framework, in addition to the data model (M) identified in the previous section, defines two other components—axioms (A in Eq. (6.18)) and algorithms (H in Eq. (6.19)). Axioms and algorithms capture a process for a search of rules held in the domain of interest for which an ontology is to be built. The theory for axiom representation is based on situation calculus and predicate calculus for representing a dynamically changing supply chain environment. Situation theory (Lesperance et al. 1995) views a domain as having a state (or situation). When the state is changed, there is a necessity for an action. Predicate theory defines conditions on which specific actions can be taken. Based on these two theories, ontology calculus for a supply chain is planned to be built. It will be based on extending both predicate and situation calculus with new terminology specific to the supply chain domain. The term *do(x, s)* represents the state after an agent performs an action *x* in state *s*. A more supply chain-specific example can be the statement that each product should have demand. This can be formulated as *Exist (demand, Product*). Another example of an axiom is the inventory constraint: Maximum Inventory ≥ Current Inventory Level, which can be formulated as *Less(MaxInventory, CurrInventory).*

An example of a portion of an algorithm is the formula according to which order size is calculated as $s = L \times AVG + z \times STD$; IF IL $< s$ THEN Order = s-IL, where s is the reorder level, L is lead time; AVG, STD are forecasted demand means and standard deviation, respectively, and z is a customer service indicator. If the inventory level (IL) is less than the calculated reorder level, an order is placed (Order), which is equal to the difference of reorder and inventory levels. This axiom can be formulated through situation calculus as $\mathrm{Poss}(\mathrm{do}((L \times \mathrm{AVG} + z \times \mathrm{STD}) = s) > \mathrm{Il}) = \mathrm{MakeOrder}(s - \mathrm{Il})$.

6.6.2 *Assembly*

Assembly is an explicit representation in some formal language of the conceptualization captured in the preceding stage. This involves (1) committing to the basic terms that will be used to specify the ontology, (2) choosing a representation language, and (3) writing the code. It simply has to do with writing down, in some language or communicative medium, descriptions or pictures that correspond in some salient way to the world, or a state of the world, of structured data.

For ontology representation, different programming languages and standards have been utilized. Ontolingua (Farquhar et al. 1997) adds primitives to defined classes, functions, and instances. Ontolingua is not a representation system, but rather a mechanism for translating from standard syntax to multiple representation systems. OIL (Ontology Interchange Language) (Fensel et al. 2002) fuses two paradigms—frame-based modeling with semantics based on description logic, and syntax based on web standards, such as extensible markup language (XML) schema and resource description framework (RDF) schema. Both Ontolingua and OIL are frame-based languages that do not provide formalism for first-order logic. With the latter, we intend to represent process logic the same way as frame logic.

XML has become a standard for communication between heterogeneous systems and is widely used on the Internet (Staab et al. 2001). This presents new opportunities for knowledge representation and acquisition and has two aspects. First, XML documents can easily be translated into knowledge representation format and parsed by problem-solving environments or domains. Second, XML can directly connect with data storage repositories (RDBMS or ERP systems), thus enabling database queries to be more expressive, accurate, and powerful. The two objectives can be achieved by enhancing the semantic expressiveness of XML, especially XML data schemas (XSD). We propose a new language, Supply Chain Markup Language (SCML), for presenting knowledge about supply chains. The specification of SCML is formulated as a XSD data schema, depicted in Fig. 6.2. It reflects system representation formalism presented in system taxonomy. At the top level there are seven groupings: input, output, functions, environment, processes, and mechanisms. Each grouping is a container, which consists of subclasses.

A representative sample of SCML is depicted in Fig. 6.3. It defines the entity Axioms, any elements it may have, and entities it may contain. An Axioms entity class may have one or many Rules (unbounded) entities, which may have Attributes entities (0 or many). An Argument entity may have two attributes: Name and Description. The entity Rule may have one and only one Body entity and two attributes.

The SCML XSD specification defines the format of knowledge representation and can be used for developing ontology models and verifying their correctness.

The assembly process, as viewed in this chapter, is the representation of captured knowledge with XML formalism. Three components of an ontology model can be represented. Data assembly is concerned with developing software programs for connecting to data storage facilities and building XML data files based on schema described earlier. A data model example is represented in Fig. 6.4. Demand for a part (Number 295) produced for Customer Number 21 is demonstrated with four attributes. The Demand Net attribute can be used if the demand is stationary. In case it is dynamic, the DemandMeans and DemandDeviation pair of attributes can be used to present the demand distribution function (it is assumed that demand has a normal distribution). The numberofRegression attribute presents the number of observations when calculating its mean and deviation.

Axiom assembly is a manual process consisting of manually entering rules captured with ontology calculus into an XML data file based on SCML schema.

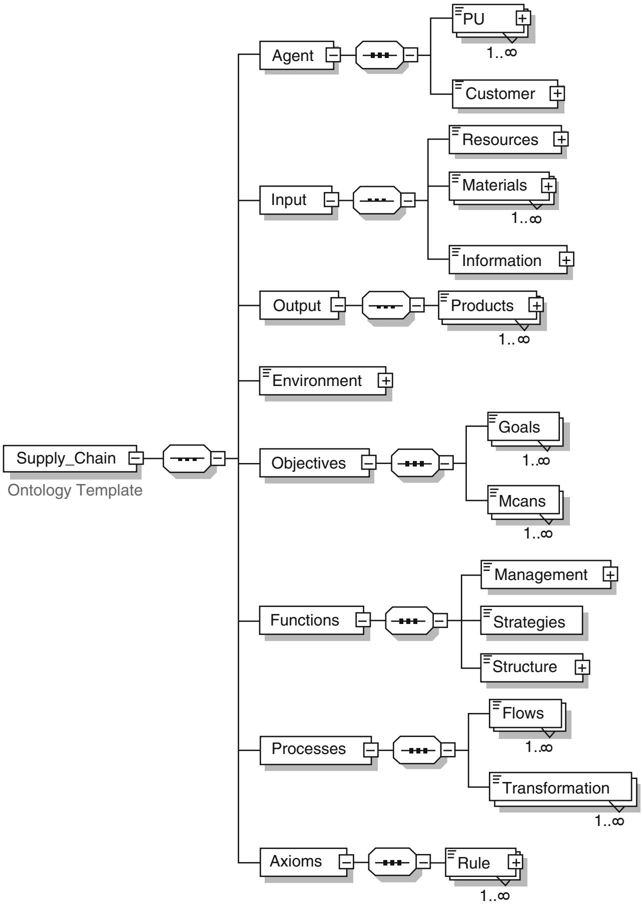
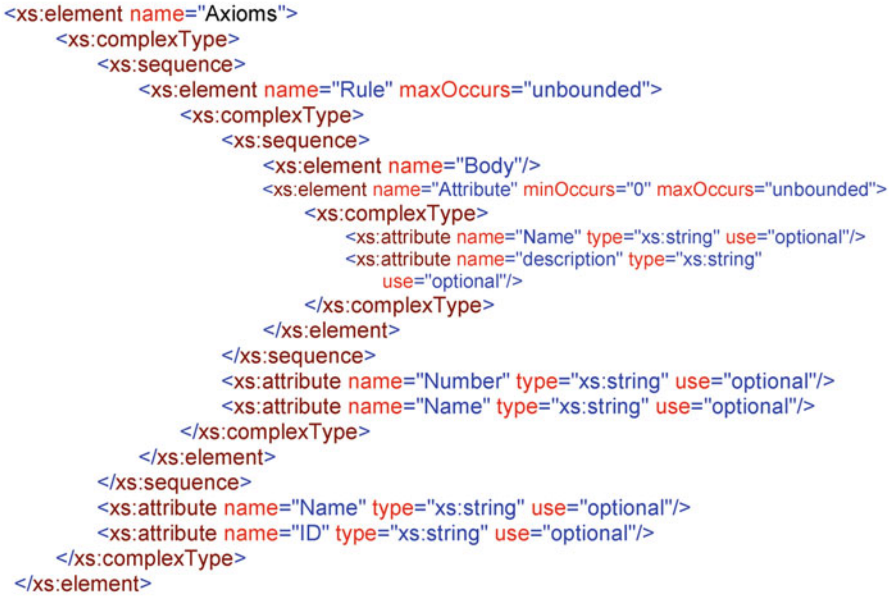


Fig. 6.2 Data schema for supply chain markup language

An axiom model XML example is represented in Fig. 6.5. The rule demonstrated here is about the relationship between inventory and demand, in case the service level is 100 %. Ontology calculus for this rule looks like:

$$Poss(ServiceLevel = 100\%) = Less(CurrInventory, Demand)$$

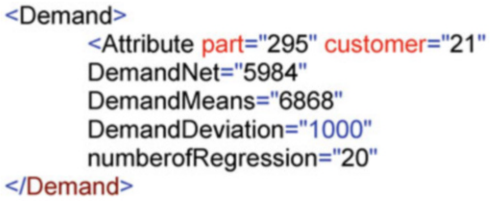
Ontology calculus formalism is transformed into XML formalism as follows. The entity type Rule defines the condition Service level is 100 %. It contains two arguments, inventory and demand. The entity Body defines the relationship between these two arguments according to the condition identified in the parent Rule entity’s Name property.



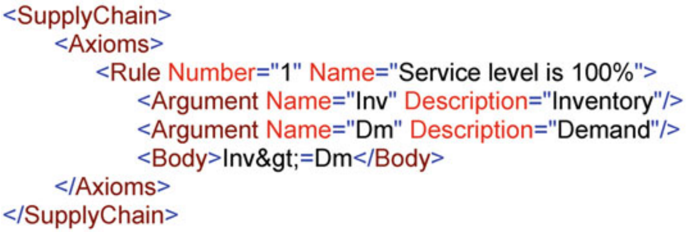
```
<xs:element name="Axioms">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Rule" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Body"/>
                        <xs:element name="Attribute" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="Name" type="xs:string" use="optional"/>
                                <xs:attribute name="description" type="xs:string"
                                    use="optional"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute name="Number" type="xs:string" use="optional"/>
                    <xs:attribute name="Name" type="xs:string" use="optional"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="Name" type="xs:string" use="optional"/>
        <xs:attribute name="ID" type="xs:string" use="optional"/>
    </xs:complexType>
</xs:element>
```

Fig. 6.3 Supply chain markup language example axioms

Fig. 6.4 Data model XML fragment



```
<Demand>
        <Attribute part="295" customer="21"
        DemandNet="5984"
        DemandMeans="6868"
        DemandDeviation="1000"
        numberofRegression="20"
</Demand>
```



```
<SupplyChain>
    <Axioms>
        <Rule Number="1" Name="Service level is 100%">
            <Argument Name="Inv" Description="Inventory"/>
            <Argument Name="Dm" Description="Demand"/>
            <Body>Inv&gt;=Dm</Body>
    </Axioms>
</SupplyChain>
```

Fig. 6.5 Axiom model XML fragment

6.6.3 Storage

The purpose of building an ontology server is to enable technology that will facilitate the large-scale reuse of ontologies through Web interfaces for decision-

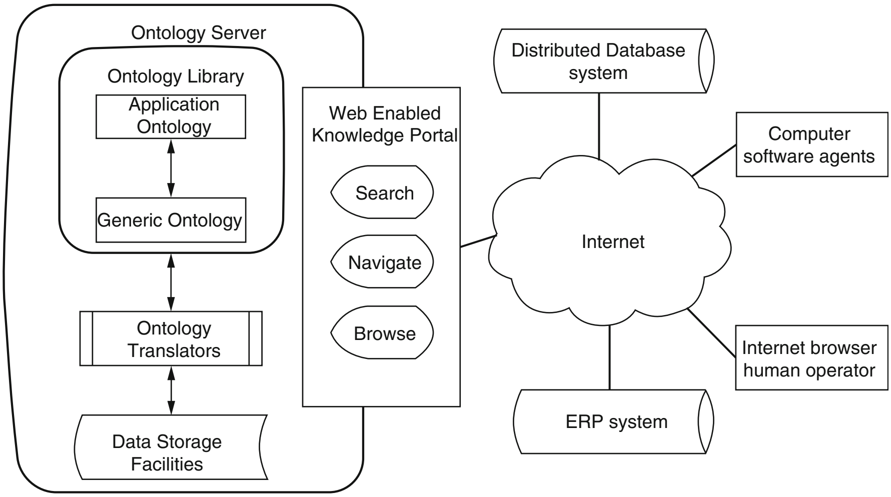


Fig. 6.6 Ontology Server architecture

making purposes throughout the complex, extended supply chain enterprise. Figure 6.6 depicts the ontology server architecture. Data are stored in data storage facilities, which are mainly relational database systems or other carriers of information, such as ERP repositories (as a complex system for maintaining data) or files (as a simpler class of data storage facilities).

The main component of the ontology server is the ontology library, which must have an index indicating how each individual item can be found. A problem classification hierarchy, Eq. (6.4), defines this index. Each node in this library corresponds to a problem and is the ontology specifying description of that problem.

6.6.4 Usage

Ontology can be utilized in a variety of ways. It can serve as an explicit medium where knowledge workers share their expertise and skills, it can be used as specifications for software engineers in developing complex software applications, and it can be used by decision makers for understanding the problem and making decisions. But the greatest advantage of having explicit ontologies is in implementing the vision of supply chain management as formulated by Fox et al. (2000). According to Fox, a supply chain is viewed as being managed by a set of intelligent agents, each responsible for one or more tasks in the supply chain, and each interacting with other agents in the planning and execution of their responsibilities.

An ontology server deployed on the Web makes the library available to supply chain members, who share the same perception of problems that they communicate to each other. In the case of agent-based SCM, ontologies provide the members with communication and interoperation. Ontologies inform the system user of the vocabulary for representing domain or problem knowledge.

Marra et al. (2012) identify that outsourcing, new product development, decision support, and risk management all are relevant to supply chain configuration. The knowledge management framework is particularly well-suited for addressing decision-making challenges in a distributed environment. Liu et al. (2014) argue that increasingly supply chain knowledge management systems should incorporate global contextual information and knowledge.

6.7 Summary

In this chapter, we explore the complexity of the supply chain configuration problem and argue that the best way to solve it is through devising a crosscutting approach that adopts concepts drawn from various disciplines in designing, developing, and implementing efficient and effective solutions. We take the representative information integration problem in the supply chain and argue that any methodology developed for supply chain configuration must explicitly take into account the systemic, reductionist, and analytic approaches available either in the published literature, or designed specifically. These approaches incorporate supply chain configuration problem details at the abstract, activity, and implementation levels, respectively. We make a case for knowledge design, development, and dissemination using taxonomy and ontology principles to incorporate system integration concepts. We propose a theoretical knowledge management system development framework, which acts as a reference model. It is based on problem solving at the above three levels. We also describe a brief implementation scenario of this framework. The utility of this framework rests on the fact that it provides a high level approach to managing the generated supply chain problem-solving knowledge, using various techniques described in Chaps. 7–9.

References

Ackoff RL (1971) Towards a system of systems concepts. Manag Sci 17:661–671

Bellamy MA, Basole RC (2013) Network analysis of supply chain systems: a systematic review and future research. Syst Eng 16:235–249

Blanchard SB, Fabrycky WJ (1990) Systems engineering and analysis. Prentice Hall, Englewood Cliffs, NJ

Cachon GP (1999) Managing supply chain demand variability with scheduled ordering policies. Manag Sci 45:843–856

Chandra C (1997) Enterprise architectural framework for supply-chain integration. Proceedings of the 6th annual industrial engineering research conference, Miami Beach, USA, pp. 873–878, Accessed 17–18 May

Chandra C, Grabis J, Tumanyan A (2007) Problem taxonomy: a step towards effective information sharing in supply chain management. Int J Prod Res 45:2507–2544

Chen F, Drezner Z, Ryan JK, Simchi-Levi D (2000a) Quantifying the bullwhip effect in a simple supply chain: the impact of forecasting lead times and information. Manag Sci 46:436–443

Chen F, Ryan JK, Simchi-Levi D (2000b) Impact of exponential smoothing forecasts on the bullwhip effect. Nav Res Logist 47:269–286

Clark AJ (1972) An informal survey of multi-echelon inventory theory. Nav Res Logist Q 19:621–650

Clark AJ, Scarf H (1960) Optimal policies for multi-echelon inventory problem. Manag Sci 6:475–490

Cohen MA, Lee HL (1989) Resource deployment analysis of global manufacturing and distribution networks. J Manuf Oper Manag 2:81–104

Deleersnyder JL, Hodgson TJ, King RE, O'Grady PJ, Savva A (1992) Integrating kanban type pull systems and MRP type push systems: insights from a Markovian model. IIE Trans 24:43–56

Diks EB, De Kok AG (1998) Optimal control of a divergent multi-echelon inventory system. Eur J Oper Res 111:75–97

Diks EB, De Kok AG, Lagodimos AG (1996) Multi-echelon systems: a service measure perspective. Eur J Oper Res 95:241–263

Drew SAA (1975) The application of hierarchical control methods to a managerial problem. Int J Syst Sci 6:371–395

Farquhar A, Fikes R, Rice J (1997) The ontolingua server: a tool for collaborative ontology construction. Int J Hum Comput Stud 46:707–727

Fensel D, Harmelen F, Horrocks I, McGuinness DL, Patel-Schnaider PF (2001) OIL: an ontology infrastructure for semantic web. IEEE Intell Syst 2:38–45

Fensel D, Hendler J, Lieberman H, Wahlster W (2002) On-to-knowledge: semantic web enabled knowledge management semantic web technology. MIT Press, Boston, MA

Fox MS, Gruninger M (1999) Ontologies for enterprise integration. Department of Industrial Engineering, University of Ontario, Oshawa, ON

Fox MS, Barbuceanu M, Teigen R (2000) Agent-oriented supply-chain management. Int J Flex Manuf Syst 12:165–188

Graves SC (1982) Using Lagrangean techniques to solve hierarchical production planning problems. Manag Sci 28:260–275

Graves SC (1999) A single-item inventory model for a nonstationary demand process. Manuf Serv Oper Manag 1:50–61

Gruber TR (1993) A translation approach to portable ontologies. Knowl Acquis 5:199–220

Gruber TR (1995) Toward principles for the design of ontologies used for knowledge sharing. Int J Hum Comput Stud 43:907–928

Grubic T, Fan I (2010) Supply chain ontology: review, analysis and synthesis. Comput Ind 61:776–786

Grubic T, Veza I, Bilic B (2011) Integrating process and ontology to support supply chain modelling. Int J Comput Integrated Manuf 24:847–863

Gruninger M (1997) Integrated ontologies for enterprise modelling. In: Kosanke K, Nel J (eds) Enterprise engineering and integration building international consensus. Springer, Berlin, pp 368–377

Gruninger M, Atefi K, Fox MS (2000) Ontologies to support process integration in enterprise engineering. Comput Math Organ Theor 6:381–394

Guarino N (1995) Formal ontology conceptual analysis and knowledge representation. Int J Hum Comput Stud 43:625–640

Gupta S, Loulou R (1998) Process innovation, product differentiation and channel structure: strategic incentives in a duopoly. Market Sci 17:301–316

Hackman ST, Leachman RC (1989) A general framework for modeling production. Manag Sci 35:478–495

Hariharan R, Zipkin P (1995) Customer-order information leadtimes and inventories. Manag Sci 41:1599–1607

Hirsch B (1995) Information system concept for the management of distributed production. Comput Ind 26:229–241

Huang C, Lin S (2010) Sharing knowledge in a supply chain using the semantic web. Expert Syst Appl 37:3145–3161

IMTR (1999) IMTR Technologies for Enterprise Integration, Rev 31; Integrated Manufacturing Technology Roadmapping Project. Oak Ridge Centers for Manufacturing Technology, Oak Ridge, Tennessee

ISO TC 184/SC 5/WG 1 (1997) Requirements for enterprise reference architectures and methodologies. http: //www.melnistgov/sc5wg1/gera-std/ger-anxshtml

Jennings R (1994) Cooperation in industrial multi-agent systems, vol 43, World scientific series in computer science. World Scientific Publishing Co., Singapore

Klir GJ (1984) Architecture of systems problems solving. Plenum Press, New York

Klir GJ (1991) Facets of systems science. Plenum Press, New York

Kosanke K (1995) CIMOSA – overview and status. Comput Ind 27:101–109

Lambert DM, Cooper MC, Pagh JD (1998) Supply chain management: implementation issues and research opportunities. Int J Logist Manag 9:1–19

Lee HL (1993) Effective inventory and service management through product and process redesign. Oper Res 44:151–159

Lee HL, Billington C (1993) Material management in decentralized supply chains. Oper Res 41:835–847

Lee HL, Padmanabhan V, Whang S (1997a) The bullwhip effect in supply chains. Sloan Manage Rev 38:93–102

Lee HL, Padmanabhan V, Whang S (1997b) Information distortion in a supply chain: the bullwhip effect. Manag Sci 43:546

Lesperance Y, Levesque HJ, Lin F, Scherl RB (1995) Ability and knowing how in the situation calculus. Stud Logica 66:165–186

Little JDC (1992) Tautologies models and theories: can we find “laws” of manufacturing? IIE Trans 24:7–13

Liu S, Moizer J, Megicks P, Kasturiratne D, Jayawickrama U (2014) A knowledge chain management framework to support integrated decisions in global supply chains. Prod Plann Contr 25:639–649

Malone TW, Crowston K (1994) The interdisciplinary study of coordination. ACM Comput Surv 26:87–119

Malone TW, Crowston K, Lee J, Pentland B, Dellarocas C, Wyner G, Quimby J, Osborn CS, Bernstain A, Hermen G, Klein M, O’Donnel E (1999) Tools for inventing organizations: towards a handbook of organizational processes. Manag Sci 45:425–443

Marra M, Ho W, Edwards JS (2012) Supply chain knowledge management: a literature review. Expert Syst Appl 39:6103–6110

Masahiko A (1984) The co-operative game theory of the firm. Oxford University Press, Oxford, UK

McCarthy J (2003) What is artificial intelligence? Computer Science Department Stanford University. http://www-formalstanfordedu/jmc/whatisai/whatisaihtml

McKelvey B (1982) Organizational systematics taxonomy evolution classification. University of California Press, Berkeley, CA

Meersman R (2001) Ontologies and databases: more than a fleeting resemblance. Rome OES/SEO Workshop

Metters R (1997) Quantifying the bullwhip effect in supply chains. J Oper Manag 15:89–100

Morris WT (1967) On the art of modeling. Manag Sci 13:B707–B717

NIST (1999) Manufacturing enterprise integration program. National Institute of Standards and Technology, Gaithersburg, MD

Pritsker AAB (1997) Modeling in performance-enhancing processes. Oper Res 45:797–804

Pyke DF, Cohen MA (1990) Push and pull in manufacturing and distribution systems. J Oper Manag 9:24–43

Shaw MJ, Solberg JJ, Woo TC (1992) System integration in intelligent manufacturing: an introduction. IIE Trans 24:2–6

Sousa P, Heikkila T, Kollingbaum M, Valckenaers P (1999) Aspects of cooperation. Distributed manufacturing systems. Proceedings of the second international workshop on intelligent manufacturing systems, pp 685–717

Sowa J (2000) Ontology metadata and semiotics. International conference on conceptual structures ICCS'2000, Darmstadt, Germany, pp 4–18

Staab S, Schnurr HP, Studer R, Sure Y (2001) Knowledge processes and ontologies. IEEE Intell Syst 16:26–34

Stader J (1996) Results of the enterprise project. Proceedings of expert systems 1996 the 16th annual conference of the British Computer Society Specialist Group on Expert Systems, Cambridge, UK

Stumptner M (1997) An overview of knowledge-based configuration. AI Comm 10:111–125

Swaminathan JM, Smith SF, Sadeh NM (1998) Modeling supply chain dynamics: a multiagent approach. Decis Sci 29:607

Thomas DJ, Griffin PM (1996) Coordinated supply chain management. Eur J Oper Res 94:1–15

Tzafestas S, Kapsiotis G (1994) Coordinated control of manufacturing/supply chains using multilevel techniques. Comput Integrated Manuf Syst 7:206–212

Uschold M, Gruninger M (1996) Ontologies: principles methods and applications. Knowl Eng Rev 11:93–155

Whang S (1995) Coordination in operations: a taxonomy. J Oper Manag 12:413–422

Wooldridge M, Jennings NR (1995) Intelligent agents: theory and practice. Knowl Eng Rev 10:115–152

Xu K, Dong Y, Evers PT (2001) Towards better coordination of the supply chain. Transport Res E Logist 37:35–54

Younis MA, Mahmoud MS (1986) Optimal inventory for unpredicted production capacity and raw material supply. Large Scale Syst 11:1–17