Individualization of Goods and Services: Towards a Logistics Knowledge Infrastructure for Agile Supply Chains

Joerg Leukel, Ansger Jacob, Paul Karaenke, Stefan Kirn, Achim Klein
University of Hohenheim, Information Systems 2, Schwerzstr. 35, 70599 Stuttgart, Germany
{joerg.leukel, ansger.jacob, paul.karaenke, stefan.kirn, achim.klein}@uni-hohenheim.de

Abstract

Our research is directed towards agile supply chains enabling enterprises to quickly respond to individual customer demand. From this perspective, agility encompasses three dimensions of adaptivity: space, time, and economy. Supply chain agility can be achieved by exploiting the most fundamental resource of any enterprise: knowledge. Studying supply chains, we regard all their tiers, participants, and potential relationships, as the search space for fulfilling individual customer demand. We study supply chains from a knowledge-based coordination perspective and regard logistics as the guiding conceptualization. The contribution of this research is a logistics knowledge infrastructure. We report about applying parts of this infrastructure to coordination problems in three selected case studies.

1. Introduction

Since 1994, the year of the appearance of Paul T. Kidd's ground-breaking book on *Agile Manufacturing*, business agility is a major concern in enterprises and in many fields of research (Kidd 1994). It dominates management literature, impacts any issue in production technology and logistics, and challenges IT research in many directions.

According to Oxford's dictionary, "agile" means to be "able to move quickly and easily" and "to think and understand quickly" (Oxford 2010). Kidd points out that agile manufacturing aims at combining "the organization, people, and technology into an integrated and coordinated whole. The agility [..] can be used for competitive advantage, by being able to respond rapidly to changes occurring in the market environment and through the ability to use and exploit a fundamental resource – knowledge" (Kidd 1994).

A major issue of business agility is customer orientation. The agile enterprise is capable of responding quickly to changes in customer demand, taking "advantage of the windows of opportunity that, from time to time, appear in the market place" (Kidd 1994). Numerous related concepts have been developed and implemented so far. Well-known examples are virtual organization, fractal factory (Warnecke 1993), and mass customization (Pine II 1993).

IT in general, and AI in particular, play a major role in developing towards the agile enterprise. This role is complemented by an important change of scope: Whereas in the 1990s, research primarily studied the interplay of people, organization, and technology *within* one enterprise, the major object of interest during the last ten years is the supply chain, which ranges from up-stream suppliers to down-stream distributors and final customers. Agility of supply chains has been contributed with competitive advantages (Christopher 2000). A major insight from supply chain management research is that the supply chain does not only constrain the ability to respond to external changes, but also provides unrecognized or unrealized potential for agility. This potential is studied in mass customization research. In contrast to mainstream literature, we consider supply chains as a solution space, which can be searched for problem solving – and in our case for meeting individual customer demand.

Example: The concept of mass customization is attributed to Joseph Pine II (1993) and was defined by Tseng and Jiao (2001) as "producing goods and services to meet individual customer's needs with near mass production efficiency". Mass customization uses flexible computer-aided manufacturing systems to produce customer-specific products. It effectively postpones the task of differentiating a product for a specific customer until the latest possible point in the supply chain. For this purpose, it combines the low unit costs of mass production with the flexibility of individual customization.

Identifying and unleashing potentials for supply chain agility is our research concern. We regard all supply chain tiers, participants, and their potential relationships as the search space for fulfilling individual customer demand. We study this problem from the perspective of knowledge-based coordination. The rationale is making supply chain structures partly transparent to decision makers and providing them with means for flexible supply chain configuration. The guiding conceptualization is logistics: the idea is that logistics provides well-defined means for describing and coordinating distributed activities. The contribution of this research is a logistics knowledge infrastructure. We report about our experiences with this approach in three case studies

The remainder of this paper is organized as follows. Section 2 introduces the basic idea and a model of supply chain. Section 3 describes the building blocks of the logistics knowledge infrastructure. In section 4, we provide a preliminary validation and discussion. Section 5 draws conclusions and outlines avenues of future research.

2. Overview and Model

This section positions the logistics knowledge infrastructure and defines a formal model of supply chains integrating product flows and contractual relationships.

2.1 Logistics Knowledge Infrastructure

Exploiting knowledge from the business sphere is a key success factor for agility of supply chains (Kidd 1994). Current approaches require dedicated efforts for making knowledge explicit and incorporating it into decision making. Our objective is to provide means for exploiting general logistics knowledge and methods in and across multiple applications (knowledge and method reuse).

Our proposal of a logistics knowledge infrastructure is positioned as an intermediate layer between domain-specific applications and current middleware for service-oriented computing (e.g., Cloud, Grid, and SOA in the most general sense). Intermediate means that it serves as a bridge between service-oriented technology and business applications. This positioning is shown in figure 1.

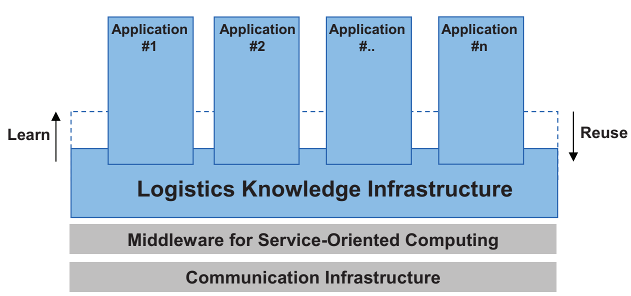


Figure 1. Logistics knowledge infrastructure

The applicability and maturity of the logistics knowledge infrastructure depend on its systematic deployment, observation and refinement in diverse application domains. In this sense, the infrastructure learns from its application in additional domains and is enriched to fulfill further requirements so that its coverage increases.

2.2 Supply Chain

A supply chain system consists of nodes participating in producing, transforming and/or moving a product, i.e., good or service, from suppliers to customers (Stevens 1989). The inter-relations between nodes are constituted by the possible flows of products, whereas nodes represent the storage and/or production of products at locations.

Definition 1 (Supply Chain System): A supply chain system is defined as a directed graph $SC=(N, F)$, where N is the set of all locations and F is the set of all possible flows of goods with $F \subseteq N \times N \times P \times TM$. Each $f \in F$ is a 4-tuple $f=(n_j, n_k, p, tm)$, with flow of product $p \in P$ from n_j to n_k using the transportation mean $tm \in TM$.

The following integrity constraints must hold:

- Let $\bullet n=\{m|(m, n) \in F\}$, the set of input nodes of n; then at least one $n \in N$ exists with $|\bullet n|=0$. Thus, at least one node has no incoming flows, i.e., it represents a real origin of products.
- Let $n\bullet=\{m|(n, m) \in F\}$, the set of output nodes of n; then at least one $n \in N$ exists with $|n\bullet|=0$. Thus, at least one node has no outgoing flows, i.e., it represents a final destination of products.
- For all $n \in N$: $|n\bullet|+|\bullet n| \geq 1$; the graph SC is (weakly) connected.

A supply chain system describes the topology of a supply chain, made up of flows of products. However, it does not necessarily match directly with respective contractual relationships between supply chain actors. For this purpose, we define a related model and map its elements to the supply chain system.

Definition 2 (Supply Chain Service Flow Model): A supply chain service flow model is a directed graph $SF=(A, S, C, M)$. A is the set of actors. S is the set of offered services. Each $s \in S$ is a tuple $s=(a_j, a_k)$, with flow s from a_j to a_k. C is a classification of S. M is a (mathematical) relation which maps each s to elements of SC. Similarly to supply chain system, integrity constraints must hold for existence of actor who does not provide a service (customer only), actor who does not consume a service, and weak connectivity.

2.3 Logistics Patterns

Logistics is concerned with the flows of products from the point of origin to the point of destination. It is closely related to supply chain, since it materializes product flows.

An interesting observation is that logistics makes extensive use of patterns, which describe flow dependencies between activities. The rationale for adopting logistics patterns is that they not only represent actual structures, but represent alternative means for realizing a flow of products from source to destination.

Two groups of patterns can be distinguished: Basic patterns and complex patterns. Basic patterns describe transportation, bundling, and unbundling as fundamental structures (table 1).

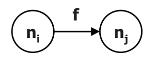
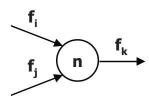
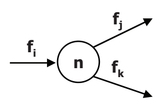
Pattern	Graph in SC
Transport (TR)	
Bundling (BU)	
Unbundling (UB)	

Table 1. Basic logistics patterns

Complex patterns are important combinations of two (or more) basic patterns (table 2): Sequence is a linkage of two transport patterns. Split-merge first unbundles a product flow via two or more intermediate nodes and then bundles them at a destination node. The loop pattern includes iterations of transportation.

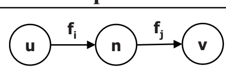
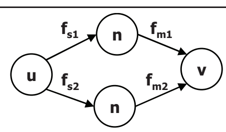
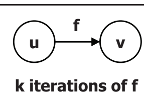
Pattern	Graph in SC
Sequence (SE)	
Split-Merge (SM)	
Loop (LO)	

Table 2. Complex Logistics Patterns

Definition 3 (Supply Chain Service Classification): *C* is a function $C \in S \rightarrow \{TR, BU, UB, SE, SM, LO\}$. The classes are as follows:

- $C(s)=\{TR\}$ is *transport*, which realizes a transformation of a product in space and time. *M(s)* contains at least two nodes n_i, $n_j \in N$ und a flow $f=(n_i, n_j, p, tm)$, thus $|M(s)|\geq 3$.
- $C(s)=\{BU\}$ is *bundling*, which realizes a quantity transformation of a product. *M(s)* contains a node $n \in N$ and at least two in-going flows f_i, f_j with $f_i=(u, n, p_i, tm_i)$ and $f_j=(v, n, p_j, tm_j)$, and one out-going flow f_k with $f_k=(n, w, p_k, tm_k)$, thus $|M(s)|\geq 4$.
- $C(s)=\{UB\}$ is *unbundling*, which realizes a quantity transformation of a product. *M(s)* contains a node $n \in N$, an in-going flow f_i with $f_i=(u, n, p_i, tm_i)$ and at least two out-going flows f_j with $f_j=(n, v, p_j, tm_j)$ and f_k with $f_k=(n, w, p_k, tm_k)$, thus $|M(s)|\geq 4$.
- $C(s)=\{SE\}$ is *sequence*, which moves a product from node *u* to *v* via *n,* with $n, u, v \in N$. *M(s)* contains these nodes and at two flows f_i, f_j with $f_i=(u, n, p_i, tm_i)$ and f_j with $f_j=(n, v, p_j, tm_j)$, thus $|M(s)|=3$.
- $C(s)=\{SM\}$ is *split-merge*, which first split a product flow at origin and then merges it at destination. *M(s)* contains origin *u*, destination *v*, at least two intermediate nodes m, $n \in N$, and for each intermediate node one in-going and one out-going flow, thus $|M(s)|\geq 8$.
- $C(s)=\{LO\}$ is *loop*, which is a k-iterative move of a product from node *u* to *v*. *M(s)* contains origin *u*, destination *v*, and $f=(u, v, p, tm)$, thus $|M(s)|=3$.

3. Logistics Knowledge Infrastructure

This section describes the building blocks of the logistics knowledge infrastructure for supply chain coordination. We introduce the rationale of coordination. Then we describe each building block in detail.

3.1 Rationale

The objective of supply chain coordination is to fulfill an individual customer demand by configuring a supply chain. Since we focus on individual demand only (e.g., product specification by the customer), a global "production plan" for this demand does not exist a-priori or only partially, i.e., a supply chain system needs to be configured. Configuration is thus the process of determining the nodes and edges of a supply chain system; nodes and edges describe the physical nature of production and delivery of the respective product. In our approach, this process does not start from scratch, but relies on explicit domain knowledge that is used to reduce the search space.

We abstract from the physical level by introducing supply chain services (as defined in section 2.2). On this contractual level, connected services are the elements of a configuration. The key idea is that setting up a configuration of services can be represented by linking and instantiating logistics patterns. By integrating the knowledge about patterns into decision making, supply chain actors can reuse these patterns for setting up local supply chain plans. Figure 2 shows the two steps of abstracting from physical supply chains by introducing services and then pattern-based configuration.

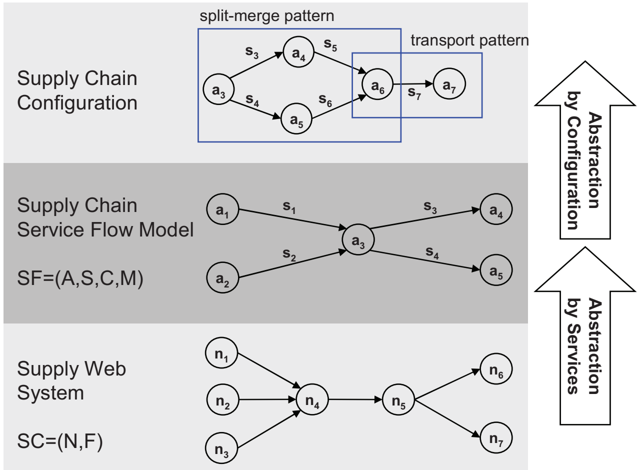


Figure 2. Two-step abstraction from physical supply chains by (1) services and (2) configuration

We propose a concrete set of methods and formalisms for realizing the basic idea. The result is a knowledge-based infrastructure consisting of four building blocks (figure 3): logistics ontology, knowledge extraction, coordination, and situated agent behavior.

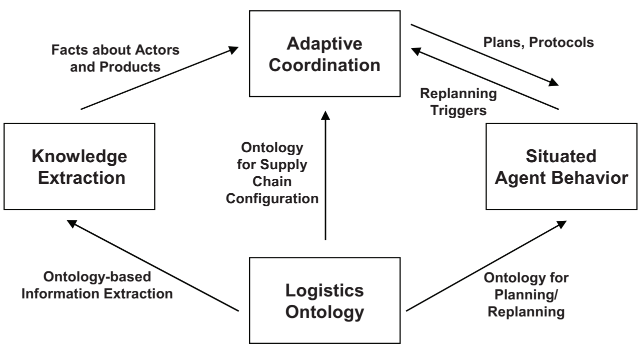


Figure 3. Building blocks and relationship

We thus ground the infrastructure on related AI concepts and technologies:

– The role of logistics ontology is to provide a formal conceptualization of actors, processes, products, and transformations in logistics. It is used to enrich the description of supply chain services to allow for reasoning about services to be connected with other services.
– Knowledge extraction is about adding facts to the knowledge base from semi- and unstructured resources such as weblogs and forums about supply chain actors, products, and supply chain services (by ontology-based information extraction).
– Coordination has to consider information asymmetries in supply chains which prevent a single participant to create, propagate, and coordinate a production plan for the entire supply chain. As a result, other coordination approaches such as interaction protocols or distributed planning are relevant.
– Situated agent behavior is about agents representing SC actors. They perceive, monitor, and reason about their situation in supply chains. Agents identify plan deviations locally and react to these by modifying the plan to some degree.

3.2 Logistics Ontology

A logistics ontology is a formal specification of a shared conceptualization for the domain of logistics (Studer, Benjamins, and Fensel 1998). Its objective is to support the configuration of supply chains. More precisely, it enriches the description of supply chain services so that reasoning over services is enabled. Reasoning supports automating tasks such as service discovery, service evaluation, service linkage, and service composition. Quality logistics ontologies, however, have not yet emerged, contrary to expectations of the Semantic Web.

Scope. Referring to definition 2 and 3, a logistics ontology must at least provide specifications of nodes *N*, products *P*, transportation means *TM*, actors *A*, and supply chain services *S*. These core elements of a "service-oriented" logistics ontology can be found in our previous work (Hoxha, Scheuermann, and Bloehdorn 2010).

Reusing Supply Chain Knowledge. The most important public knowledge source for supply chains is the Supply Chain Operations Model (SCOR) (Supply-Chain Council 2010). We integrate this knowledge into our approach and convert its intentional semantics into formal specifications. SCOR provides a comprehensive set of means for describing supply chains. SCOR is developed by an independent not-for-profit firm with more than 1,000 corporate members.

Basically, SCOR provides (1) a three-level set of modeling primitives, (2) a set of associated best practices, (3) a set of input/output data elements associated with modeling primitives on the lowest level, and (4) a set of supply chain metrics. By integrating this knowledge into the logistics ontology, we add not only a terminology from supply chain practices, but also integrity constraints over supply chain services; these can be used to determine linkages between services as part of a configuration.

SCOR differentiates supply chains by the degree of customization: (1) stocked products, (2) made-to-order products being manufactured for a specific customer order, and (3) engineered-to-order products being designed and manufactured to a specific customer requirement. This differentiation is then applied to all processes in supply chains, which are sourcing (S), manufacturing (M), and delivering (D) products. An example model is shown in figure 4; it consists of linked process categories indicated by type (character) and customization (number).

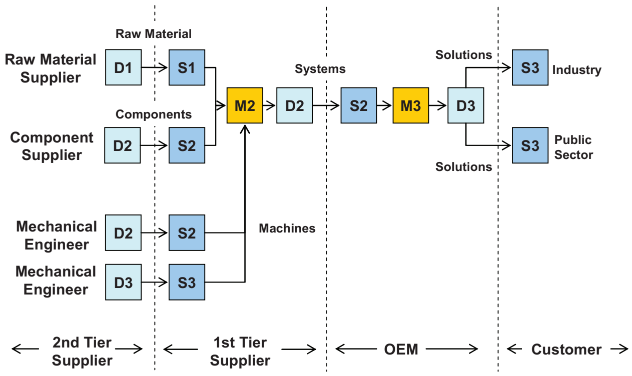


Figure 4. Example SCOR thread model

This differentiation reflects very well strategies that actors implement. For supply chain agility, it helps identifying supply chain segments of mass production and those which add individualism to products. We use this differentiation as a flat taxonomy of supply chain services.

Each supply chain service *s* implements at least a deliver process, since it delivers products from actor a_j to actor a_k (see definition 2). “Source” and “manufacture” allow for a more detailed description of processes that precede delivery. We link the supply chain service flow model to SCOR by providing a top-level view of SCOR ontology (figure 5).

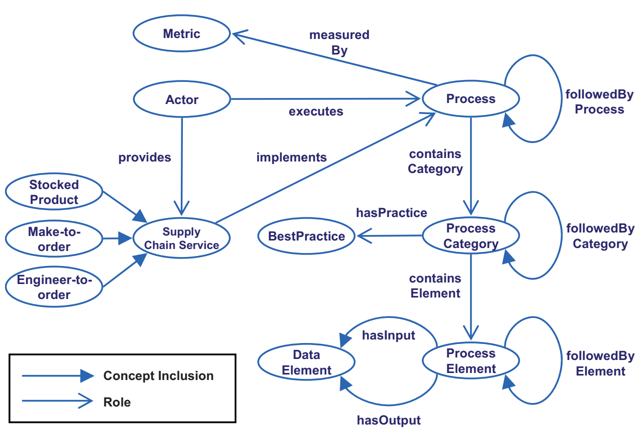


Figure 5. SCOR Ontology

First, actors provide supply chain services. For this purpose, they implement processes, which are measured by metrics. A process such as source contains more detailed process categories for stocked, make-to-order and engineer-to-order products (as shown in figure 4 by S1, S2, and S3). These are further detailed into activities, e.g., within sourcing such as schedule delivery (S1.1), receive delivery (S1.2), verify delivery (S1.3) etc.

An interesting feature of SCOR is that it not only provides modeling primitives for supply chains, but defines reference structures from which supply chains can be configured; by that it helps constraining supply chain models to allowed ones. For instance, an actor can link his own processes only as (S, M, D) or (S, D). He can link his S process with a preceding D process of another actor (supplier); similarly, he can link his D process with a proceeding S process of his customer. The rule set for linking process categories is more complex, e.g.:

– Actor’s own process categories as (S1, M1, D1), (S1, M2, D2), (S1, M3, D3), (D1, S1), (D2, S2), (D1, S1), etc.
– Actor’s source process categories must match with preceding categories of suppliers as follows: (D1, S1), (D2, S2), and (D3, S3), thus sharing the same degree of customization.

These constraints can be expressed in description logics (and OWL DL) by concept definitions, which include role restrictions over the role followedByProcess, followedByCategory and followedByElement, respectively.

Products. Knowledge about products *P* can be retrieved from product ontologies such as eCl@ss, which is also available in an OWL-based specification (Hepp 2006). The problem with reusing such domain knowledge is the lack of semantic richness; most specifications are pure taxonomies. Therefore, additional ontology engineering effort is required for adding axioms that support reasoning beyond is-a-relationships.

Semantic Supply Chain Services. Configuration of services is also a subject in Semantic Web Services research. It yields a common model entitled IOPE, which is part of OWL-S (W3C 2004), an ontology for describing Web services. IOPE structures the functional part of a service description into inputs, outputs, preconditions, and effects. We interpret the IOPE model for describing supply chain services.

IOPE describes the service functionality as an information transformation and a state change resulting from the service. *Information transformation* is subject of input and output. Valid input can be restricted by referring to a concept of the domain ontology. It is important that a supply chain service represents a physical activity taking place in the real world, thus transformation is not limited to information, but concerns the object of this physical activity; hence the product *p*. We therefore interpret IO as the *physical transformation*.

The state change is captured by precondition and effect. In Web services, preconditions are constraints over inter-dependent input information. In supply chains, these need to be related to physical state. For instance, consider a service of type transport. At least, it is required that the product is located at the origin node. We thus formulate an axiom as follows: Let *origin_o* be the origin node of *s*, then *pre_o:=isLocatedAt(p,origin_s)*. Effect is the change; here it is right the transformation of *p* in time and location. For the former we define *eff_s:=isLocatedAt(p,dest_s)*, with *dest_s* the destination node of *s*. The latter is calculated by adding the transport time (*leadtime_s*) to the start of transportation, i.e., *atTime(tu,dest_s):=atTime(tu,origin_s) + leadtime_s*. Table 3 outlines the interpretation of IOPE.

Description	Web Service	Supply Chain Service
Input	Information required for executing the service	Input factor, or product *p*
Output	Information generated by service execution	Output factor, or product *p*
Precondition	Constraint over input information	Availability of *p* at origin location, an the right time and in the right quantity
Effect	State change	Transformation of *p* in time, location, quantity, and/or quality.

Table 3. Interpretation of IOPE

IOPE supports the finding of valid linkages between services. The service functionality, however, is not sufficient to determine to which degree the request is fulfilled. This task requires a quantitative assessment, which is called Quality of Service (QoS). QoS adds non-functional properties (QoS parameters) to the service description (O'Sullivan, Edmond, and ter Hofstede 2002). Generic QoS parameters such as execution time, cost, throughput, availability, and reliability need to be adapted and extended for supply chain services. We propose to adopt such parameters from the SCOR model, which provides a comprehensive system of supply chain metrics.

QoS parameters complement the service description. We thus extend the preliminary supply chain service definition of section 2 as follows.

Definition 4 (Supply Chain Service): Supply chain service is a tuple $s=(a_j, a_k, I, O, P, E, Q)$:

- a_j is the service provider, a_k service consumer, a_j, $a_k \in A$.
- I is a set of input factors.
- O is a set of output factors.
- P is a set of preconditions, i.e., logical axioms.
- E is a set of effects, i.e., logical assertions.
- Q is a set of QoS parameters $q=(t,v)$, with parameter type t and parameter value v.

3.3 Knowledge Extraction

Knowledge-based coordination requires that all relevant knowledge is made explicit, i.e., by referring to concepts or instances of the logistics ontology or amending this ontology. Typically, information used in coordination is mostly numerical, such as cost, quantity, time, or service level. However, real-world supply chains also generate and process unstructured information. Ignoring this type of information would neglect a potentially rich source of knowledge. For instance, unedited, natural language comments and expert opinions (sentiments) published in Web sources provide such knowledge. According to Liu (2010), sentiment analysis and opinion mining are synonyms. We use the term sentiment. A sentiment on a sentiment object is a positive or negative view, attitude, emotion or appraisal from a sentiment holder. For instance, sentiments objects are reputation, reliability, quality of products, and supply chain actors.

Massive amounts of sentiment information are contained in textual documents, especially on end-user/customer-generated web sites in the Web 2.0, e.g., weblogs and forums. Automatic processing faces vast and permanently changing amounts of heterogeneous and unstructured information, which is at the same time noisy and uncertain.

With regard to supply chain coordination, text mining can be utilized for the following problems: extraction of (1) facts and (2) sentiments from unstructured textual sources providing new information, not yet considered. In both cases, the unstructured textual information is transformed into structured information, e.g., a sentiment is a tuple of date, sentiment holder, sentiment object, and sentiment polarity.

Extraction Process. The concrete objects of research are sentiments with respect to the reputation of products, actors, and supply chain services contained in Web document texts. The problem is to extract, classify and aggregate sentiment on several levels, i.e., word, sentence, document, and set of documents. The approach is ontology-based information extraction (Wimalasuriya and Dou 2010). The extraction process consists of the subsequently described steps.

Natural language pre-processing. Tasks performed are tokenization, sentence splitting, part-of-speech tagging, lemmatization, and named entity recognition (Pang and Lee 2008).

Sentiment Extraction. To extract sentiment on the sentence level, information extraction rules are used on basic features extracted by natural language pre-processing and concepts or instances routed in the logistics ontology. These rules are described as regular expressions over annotations and implemented using JAPE rules in GATE (Cunningham et al. 2002). The result is a set of tuples of sentiment object and sentiment polarity being an element of the set {positive, negative} per document.

Classification and Aggregation of Sentiments. The overall sentiment polarity of a document is the net of the number of positive and negative sentiments contained in each document. The document-normalized sentiment is then averaged over multiple documents referring to the same sentiment object from the same date. This aggregated sentiment is added to knowledge base.

3.4 Adaptive Coordination

The ability of an agile supply chain to react quickly, and precise to customer demand depends upon its capability to intelligently search the supply chain, to identify potential solutions (i.e., virtual enterprises together with appropriate collaborative production/delivery plans), to select the most appropriate of them, and to perform the required collaborative activities. The key for this challenge is adaptive coordination. Adaptivity of supply chain coordination has three dimensions: adaptivity in space, time, and economy (Kirn 2006; Kirn 2008).

The delivery of products in supply chains requires coordination across organizations; i.e., creating a create-to-order virtual enterprise by dedicated configuration of the supply chain.

Under this perspective we investigate *coordination methods* and address the *allocation of resources under distributed control by involving distributed artificial intelligence (DAI) approaches*. We represent business actors by software agents, and utilize DAI coordination methods for supply chain configuration. The results of this work are methods for solving coordination and planning problems in multi-tier supply chains.

Coordination Problems in Supply Chains. Fulfilling individual customer demand requires the existence of a production plan. The construction of the plan is itself a problem the agents need to solve (Durfee 1999). The distributed production planning problem consists of the following sub-problems (Durfee 1999; Witteveen and de Weerdt 2006):

- a decomposition problem (identify set of required input factors potentially provided by other agents and according logistics patterns),
- an allocation problem (which agent will produce which product),
- individual, local planning problems (ensure that allocated production steps can be performed), and
- a plan coordination problem (coordination of the required production steps, i.e., synthesize overall plan).

For planning the inter-organizational production process, an *abstraction on processes* is needed; i.e., processes can be abstracted from details and enriched by more specific information. This is required to hide irrelevant or confidential information across organizational boundaries. Thus, an abstraction is required if the detailed information is unknown to an agent.

Processes are described by actions and knowledge about these actions. We distinguish operational, atomic actions from non-atomic actions which need to be further refined to become operational (Unland et al. 1995). An incremental replacement of non-atomic by atomic actions is required as the (tier-wise decomposed) customer requirements are communicated along the supply chain tiers. Figure 6 shows an example of the refinement of the process nodes by decomposition.

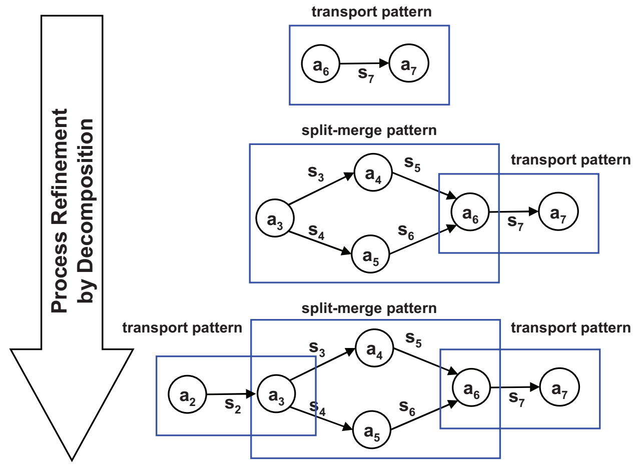


Figure 6. Process refinement by decomposition

Abstract plans can be directly reused without solving decomposition problems if every non-atomic action can be replaced by a single atomic action by allocation; i.e., instantiable action classes (abstract atomic actions) are replaced by respective instances. This results in a template-based planning approach.

An initial plan may become invalid or sub-optimal as an agent receives new information (section 3.5). Thus, the plan needs to be repaired in a replanning process. Therefore, the set of invalid or sub-optimal atomic actions needs to be replaced in the production process. This can be achieved by abstracting details for the atomic actions in question (resulting in non-atomic actions) and reprocessing the planning process without changing the remaining atomic actions. However, due to interdependencies between actions, this can also require to abstract further atomic actions to arrive at a valid and optimal solution. In contrast to the template-based planning approach, replanning can change the topology of the process. If there is neither a suitable plan template nor an initial plan available, a new plan has to be constructed from scratch.

Multiple individually optimal plans do not guarantee global plan optimality. Thus, allocation and plan coordination play a major role for solving the distributed planning problem. These steps require appropriate methods for coordinating the communicating interactions between agents (e.g., protocols). Multi-tier allocation and plan coordination problems require a consideration of the distributed nature of the supply chain as well as coordinated interactions along all tiers of the supply chain (Karaenke and Kirn 2010).

Multi-Tier Coordination. The decomposition problem in supply chains is closely related to allocation; i.e., a decomposition, which is valid on an abstract level, may become unsuitable to fulfill the customer requirements due to the aggregated properties of the input factors provided by the allocated agents. For the production of services in supply chains, we have investigated the determination of quality of service (QoS) of a composite service in (Karaenke and Leukel 2010).

Addressing the allocation problem itself, we investigate task and resource allocation in supply chain systems. We address the problem of multi-tier allocation; i.e., consideration of dependencies across supply chain tiers. If these dependencies are not considered, the fulfillment of contracts may be unaccomplishable due to missing contracts with other agents which are required for the fulfillment (overcommitment). To solve this problem, we have applied an *interaction protocol engineering perspective* (Huget and Koning 2003). We have proposed a multi-tier allocation protocol specification for composite service provision over multiple supply chain tiers in (Karaenke and Kirn 2010).

Addressing the plan coordination problem, we investigate planning and scheduling methods in supply chains and address the problem of intra- and inter-organizational interdependencies of planning problems. If these interdependencies are not considers, the construction of adapted and new plans respectively creates potentially inconsistencies regarding the global plan. To solve this problem, we apply a distributed planning perspective.

3.5 Situated Agent Behavior

This section addresses models and methods to percept and reason upon situations of agents that represent supply chain actors. The situation detection and anticipation focuses on adaptive behavior of agents in supply chains (Kirn 2008). Narrowing the DAI-perspective taken we are concerned with Belief-Desire-Intention (BDI)-agents (Bratman, Israel and Pollack 1988) situated in an environment (Ferber and Müller 1996), called situated BDI-agents. The environment is the supply chain system *SC* and all possible influences on it. To establish distributed perception of reactions and their results in this environment, we present an event-driven sensing environment allowing the definition and subscription of context events (Jacob, Mueller, and Kirn 2009).

First, we introduce technologies for establishing active perception in situated BDI-agents. We show the use of the supply chain contexts and situations in this environment. Second we introduce the event-driven sensing environment supporting data generation for active perception of the agent. The context framework provides the search space for information the agent can subscribe to. The agent uses the framework according to his situation knowledge for possible future situation detection.

Active Situation Perception. Constituting the agent property of "situated in some environment" (Wooldridge 2002), the agent needs perception capabilities for relevant changes in the supply chain to be able to act situated in his context. In cognitive science, there are models for active perception and situation assessment (Endsley 1995). The BID-agent architecture is not desgined for active perception. Situated agents (Ferber and Müller 1996, Ferber 1999) interact with their environment closely with respect to simultaneous actions, but their cognitive capabilities are not part of the situated agent model.

For active perception a model for knowledge representation is needed. Based on the *situation calculus* (McCarthy and Hayes 1969), Scherl and Levesque (1993) offer a solution for knowledge representation. It allows for establishing knowledge-based actions. Awareness about knowledge which is required for a goal-oriented sensing strategy can be derived from these knowledge representations. Knowledge-based actions aim at the generation of knowledge the agent does not have, but knows he needs to have. In the situation calculus this knowledge is represented as $Knows(\Phi,s) \stackrel{\text{def}}{=} (\forall s').K(s',s)$. $Knows(\Phi,s)$ means in situation s is all knowledge for the step to situation s' following s available by executing an atomic action. The plan execution determines relevant sensing actions resulting from the planning process (see section 3.4). The situation transition is done according to table 1 on the supply chain service model (see section 2.3). The service execution is monitored against the planed logistics patterns in each situation according to the supply chain service flow model containing a specific service execution flow f_i and participating nodes $M(S)$.

The situated agent allows active perception in the agent model to recognize situations changing based on the supply chain service model instance. Situation detection focuses on relevant perceptions linked to the service provider in order to minimize the deliberation effort of the agent. The situated agent design allows an evolution of the multiagent system through situational influences and reactions. If a deviation from an expected situation is detected, appropriate coordination methods are utilized (see section 3.4).

Event-driven Context Generation. Each agent can access its environment for its situation monitoring proposes. The environment is organized in a context framework based on an event-driven architecture (Michelson 2006) and Web services. All layers in figure 7 can be instantiated multiple times on different physical machines.

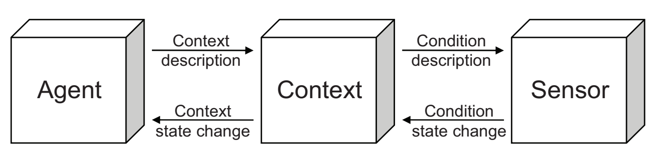


Figure 7. Context framework

A sensor in the framework can be any source of data, e.g., a data base query, a Web Service, a result from knowledge extraction or a physical sensor. The agent-layer allows the specification of a context description using the XML-based sensor specification language *ContextXML*. ContextXML can address multiple sensors in different physical locations and combine multiple sensor conditions by logical operators (AND, OR, XOR). Each ContextXML context description sent from a client to a context layer is stored and evaluated in order to separate sensor condition descriptions for each sensor.

Each extracted sensor condition description is forwarded to the sensor layer. If sensors are attached to different context layers, the according information is forwarded to the context layer. If the sensor layer reports a condition change to the context layer based on changing sensor values, all context conditions are checked to validate the complete context event. As soon as all conditions apply, the client is informed about the context change.

The agent uses the framework to generate facts about the supply chain, as long as he knows where to generate and how to interpret the data. The generation of knowledge to query is initiated by a knowledge producing action.

4. Three Case Studies

We report about implementing parts of the proposed infrastructure in three selected case studies. Each case concerns one building block from coordination, knowledge extraction, and situated agent behavior and grounds its conceptualization on the logistics ontology building block.

4.1 Coordination in Airport Ground Handling

Industrial Case. The use case setup, which has been developed in the EU project BREIN (http://www.eu-brein.com), is as follows: Stuttgart Airport has about 400 flight movements per day. Most of the flights arrive and depart in two peak time periods: in the early morning and late afternoon. The prototype system was used for conducting several experiments: Real-world operational data from an airport company was retrieved and loaded into the system; it included flight plan data, ground handling process definition, and a set of process rules (e.g., related to aircraft types, airlines, etc.).

Ground handling at airports involves (1) ground handling resources, (2) ground handling companies, and (3) airlines. Airlines have contracts with ground handling companies about handling inbound and outbound flights. Planned activities can be subject of internal (e.g., resource failure) and external disturbances (e.g., delay of arriving flight) that affect, delay or constrain the process. Thus, there can be temporary resource shortages for ground handling service providers. Hence, these providers need to outsource some tasks to another service provider at the same airport which has sufficient resources for the time frame in question.

The flight plan selected represents a peak-time with expected shortage of resources causing bottlenecks and thus the need for adapting the pre-planned delivery (82 flight movements over a period of three and a half hours).

Research Challenge. The supply chain setup with different ground handling service providers requires decentralized coordination approaches. Further, inter-organizational coordination of distributed resources in a multi-tier supply chain is required to fulfill individual operations. In contrast to central planning systems utilized in today's airport management systems, this approach facilitates inter-organizational planning and decision support. Therefore, inter-organizational interoperability is required for cross-organizational coordination.

Supply chain. We have setup an experimental ground handling supply chain, consisting of three ground handling service providers, 58 resources of three different types, providing five service types, and 15 airlines. While ground handling concerns physical items (e.g., aircrafts, baggage, passengers, etc.), it has to be noted that the handling process is being delivered by a supply chain of services, more precisely of logistics services. This fact makes it very simple to construct the supply chain system *SC* and supply chain service flow model *SF*. Actually, the latter's relation *M* is a bijection. An example *SF* is shown in figure 8 (limited to one ground handling company and airline to reduce presentation complexity).

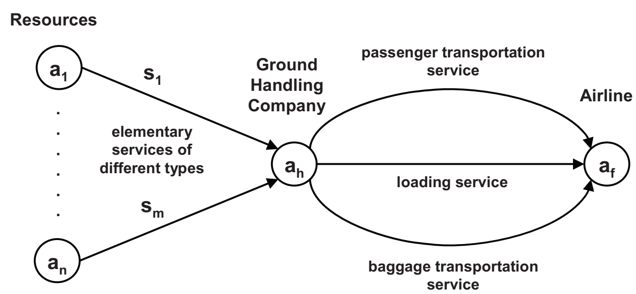


Figure 8. Supply chain service flows in ground handling

Logistics Ontology. Domain-specific knowledge from ground handling was identified and inserted into the logistics ontology by adding sub-concepts to actor *A* (e.g., ground handler, airport firm, airline, etc.), location *N* (e.g., parking positions, gates, terminals), product *P* in terms of object being moved (e.g., aircrafts, passengers, baggage, cargo). For defining supply chain services, the IATA Standard Ground Handling Agreement served as a major source and its classification was adopted. Additionally, specific physical resources (e.g., baggage carts, busses) were considered and related to supply chain services.

Coordination. The shared conceptualization provided by the logistics ontology constitutes the basis for inter-organizational coordination; it provides means for semantic interoperability. As mentioned, pre-planned schedules exist for the dispatching of aircrafts for each handling company and resource (retrieved from an airport planning system). The system checks events for potential effects on the plan. Thus, we have focused on the adaptation and replanning of the schedules for a single day due to changes in the flight plan and resource failures in an inter-organizational approach. The reallocation of conflicting services in resources' schedules is a key element of our approach.

In case of a conflict, e.g., delay of service execution not possible, because the resource is already assigned to another flight, conflicting services in the resources' plans are abstracted to abstract services on ground handling company level (cf. section 3.4). These are reallocated in reverse (procurement) auctions. A reallocation is first attempted inside the ground handling company. If this reallocation is unsuccessful, the service is outsourced to another ground handling companies' resources in an inter-organizational multi-tier allocation approach, if possible. Depending on the requirements for a service to be outsourced, it can be provided by a single resource or by multiple resources in a bundled service (e.g., number of passengers in aircraft vs. passenger capacity of busses). The auctions are combinatorial, thus may include sets of inter-related services for the same flight. Further, plan templates exist for every ground handling process which can be used for reallocation and replanning.

If the plan cannot be repaired by reallocation, further abstractions are considered. Therefore, the valid time frames in the service level agreements (SLAs) are assessed and shifting of services is considered. If the plan cannot be repaired using replanning, the penalty for SLA violations is minimized; i.e., unavoidable delays are shifted to the services on which they cause the minimal penalty costs.

Results. For the production of services in supply chains, we investigated how to determine the quality of service (QoS) of a composite service (i.e., the result of a possible allocation). The contribution is a QoS aggregation classification ontology which can be used to determine the right aggregation for annotated QoS parameters; i.e., ontology-based QoS parameter aggregation for composite services (Karaenke and Leukel 2010). This approach is based on a shared conceptualization between organizations as described in section 3.2.

Regarding the allocation, we have proposed a multi-tier allocation protocol specification for composite service provision over multiple supply chain tiers in (Karaenke and Kirn 2010). Using the model checker SPIN (Holzmann 1997), we have formally verified safety properties like the absence of deadlock, and also that the protocol enables multi-tier allocation. Further, we have demonstrated the applicability and the benefits of our solution through multiagent-based simulation of an airport logistics scenario (Karaenke and Kirn 2010).

Due to the market-based approach, the system does not give priorities to flights (customers) in advance, but delegates this function to service providers. Since each service provider aims at fulfilling its SLAs with airlines, the bidding strategy is maximizing revenue and avoiding penalty, both as defined in airline SLAs and propagated to the upstream providers.

DAI technology offers the capability for flexible and adaptive problem solving behavior, but lacks reliability, security, and robustness (Foster, Jennings, and Kesselman 2004). Service-oriented architecture (SOA) specifications foster interoperability on the technical level regarding interfaces (e.g., WSDL) and messaging protocol (e.g., SOAP); WS-* standards facilitate the building of secure, robust and reliable virtual organizations (VOs) to solve problems with distributed resources, but lack the capability to react or adapt to undesired conditions and changing requirements in dynamic environments (Foster, Jennings, and Kesselman 2004). However, specifications of the data exchanged (i.e., message contents) is beyond the scope of SOA specifications. The Semantic Web (SW) approach, in contrast, focuses on semantic interoperability. Therefore, we have applied a combination of DAI, SOA, and SW technologies to facilitate interoperability required for cooperation (Karaenke et al. 2010).

The experience made was that our system handles deviations successfully, as they occur regularly in ground handling, and that the coordination methods provide means for (1) reallocating internal resources, (2) outsourcing jobs to third party service providers, and (3) considering SLAs that exist with the final customer (airlines). A detailed description of the prototype system, the experiment, and its results can be found in (Jones 2010). Due to outsourcing of services, the system handles bottlenecks effectively by generating a valid plan in any tested case, whereas the conventional planning system stops and requires manual plan repair by qualified staff.

4.2 Knowledge Extraction in Supply Chains

Industrial Case. Documents created and used in supply chains are an important source of knowledge, in particular in supply chains that deliver highly customized and unique products, e.g., in projects involving several actors for a limited period of time.

Research Challenge. We address the extraction and classification of relationships between documents across the document repositories of actors cooperating in a supply chain. Current Document Management Systems (DMS) typically focus on single documents and metadata assigned to it, such as author, date, recipient, and topic. They fall rather short in extracting, displaying, classifying and exploiting relationships between multiple documents of different actors along supply chains. Documents contain information about actors and processes in supply chains. Thus, documents and documents relationships can be mapped to supply chains.

Supply Chain. We study a case from the construction industry (cooperative research project ProBauDok). This industry is interesting because complex construction projects involve multiple tiers and produce extremely individual products (actually, each product is unique). Customers request a construction service from the main contractor. This service is delivered based on subsection construction services and planning services, delivered by different architects. The construction company requests services from several sub-contracting construction companies which request further services. Figure 9 shows these relationships.

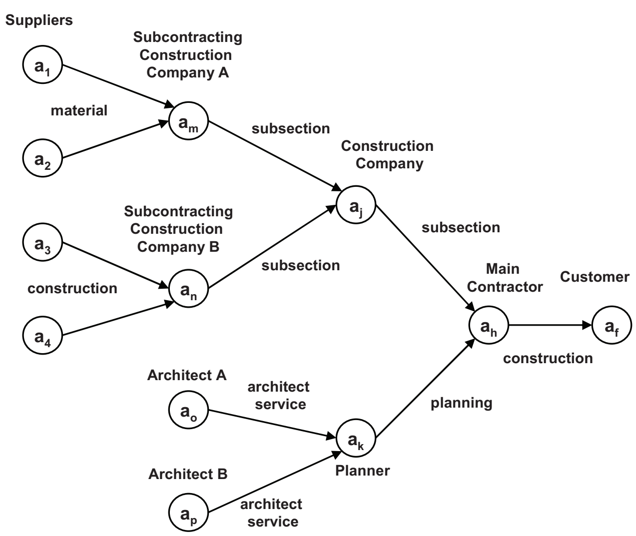


Figure 9. Supply chain service flows in construction industry

Logistics Ontology. Domain-specific knowledge from construction industry was identified and inserted into the logistics ontology by adding sub-concepts to actor *A* (according to different types of firms), product *P* in terms of objects being moved (e.g., construction material, elements, segments, machines). Supply chain services are construction works, by adopting industry classifications. The relation to documents is that both business and construction documents include references to these domain concepts. For being able to distinguish document types and constrain the information contained in a document, we added a classification of document based on *IEC 61355* ("Classification and designation of documents for plants, systems and equipment"). This classification is, however, not confined to construction. Therefore, it extends the ontology beyond the case study.

The ontology is used for (1) guiding the extraction of facts by providing language representations of concepts and instances within one document and (2) for providing a schema of the semantic repository that stores extracted facts.

Knowledge Extraction. We developed a two-tier architectural extension of DMS for semantic document management (figure 10). The rationale is to separate knowledge about business documents as follows: (1) The conceptualization is subject of the logistics ontology, which is used for semantic document storage. Inter-relations between documents are expressed using constructs of the ontology. (2) The enforcement of integrity constraints beyond the expressivity of the ontology is subject of cooperative software agents.

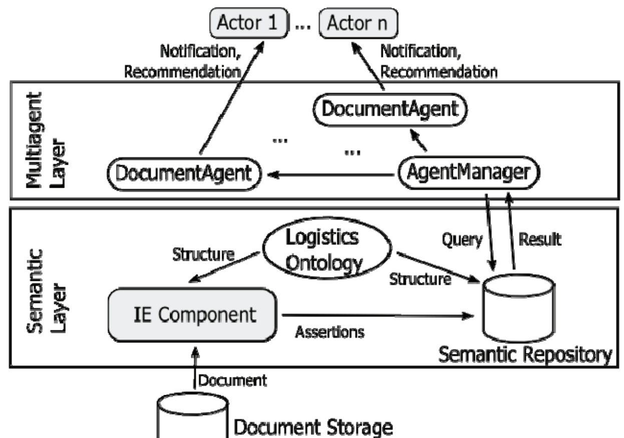


Figure 10. Two-tier extension for knowledge extraction from documents in supply chain systems

Rules based on regular expressions over annotations (created by natural language pre-processing and ontology-based information extraction) are utilized for extracting facts such as business document type, actor types, date, topic, construction subsection, invoice amount, etc. Cooperative software agents retrieve these facts, identify document relationships, and create additional facts that describe these relationships.

DocumentAgents supervise the documents that emerge over time. Each agent represents a domain-specific dynamic set of inter-related documents. It implements rules describing integrity constraints over two or more inter-related documents (e.g., sequence of order-invoice). In case of integrity violation (e.g., during invoice verification), the agent adds a respective fact to the knowledge base, which then can be used for, e.g., giving recommendations for actions to end-users via a GUI or notification messages.

4.3 Situated Agent Behaviors in Civil Engineering

Industrial Case. A civil engineering project is expected to operate at low costs, be on time, and deliver high quality documented precisely. Hindering this, several entities need to be coordinated over long distances in a high frequency according to a high rate of environmental changes such as weather, soil conditions, or machine failure. Traditional planning approaches cannot cope with the high situational change frequency of a running construction site. To counteract these challenges from a supply chain perspective, we develop and evaluate (1) coordination methods on the machine level and (2) situated BDI-agents for semi-autonomous construction vehicles to monitor situations of plan deviation. In case of detected deviations, a new resource allocation needs to be generated. The use case concerns earthworks for road construction.

Supply Chain. The supply chain flow model is made of four tiers (figure 11): The final product is a construction (e.g., road), delivered by a main contractor. This actor delegates specific activities to subcontractors, which are responsible for sections of the construction project. These subcontractors provide their services by both using own resources and external resources, which are contracted from, e.g., machine providers.

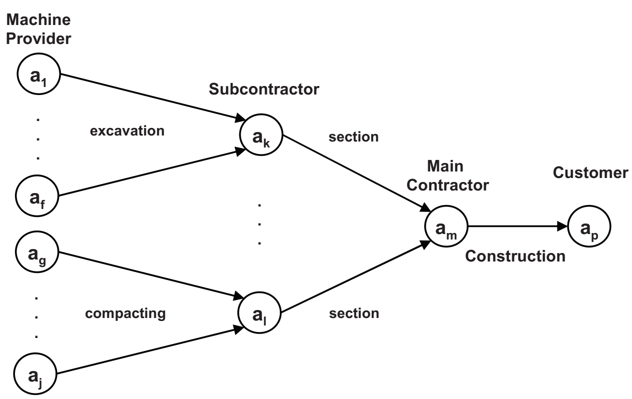


Figure 11. Supply chain service flows in civil engineering

Logistics Ontology. The ontology is extended to cater for domain-specific concepts and roles, in particular those of actors A, which provide specific construction machines and vehicles. Locations *N* relate to construction sites. Products match to the one listed in the preceding case study. Supply chain services *S* include “true” logistics services such as transporting, transshipping, and storing products on the construction site. Construction services are implemented using the bundling pattern.

Situated Agent Behavior. Based on the service flow model, various configurations of the physical supply chain model SC are applicable. To preserve the distributed manner of coordination, we adopt Generalized Partial Global Planning (Decker and Lesser 1993) for all service providers A on the construction site. This leads to a physical configuration of the supply chain transferring SF to a SC, which is object to situated adaption by the individual agents.

Agents with situated behavior need to be linked with their environment to detect situations leading to target deviations in the supply chain. The environment consists of (1) the physical world investigated using real machine sensors, and (2) common information objects, which refer to the underlying supply chain model and logistics ontology. To operate on both, we develop a situated agent architecture allowing situation detection using both sources.

The physical world integration is based on the context framework (section 3.5), which is connected to real machine sensors of excavators, dumpers, dozers and rollers. We use the head-body-agent-architecture (Steiner, Haugeneder, and Mahling 1991) differing between a head, which deliberates and interacts, and the body encapsulating domain functionality.

To integrate interaction on common, non-physical objects and information sources, we use regional synchronization based on situated multiagent systems (Weyns and Holvoet 2004). Regional synchronization allows perception of process interactions between agents in order to handle simultaneous actions on common, non-physical objects. The situated agents need to be enhanced with active perception capabilities according to construction site situations. The situations lead to estimations for plan deviations. A plan deviation triggers Generalized Partial Global Planning to generate a valid *SC*, which is monitored by situated agents during its execution.

5. Status and Future Work

This paper presents a logistics knowledge infrastructure and introduces its four major components: logistics ontology, knowledge extraction, situated agent behavior, and adaptive coordination. It is aimed at making supply chain agile to meet individual customer demand efficiently.

This ambitious goal requires significant achievements in three areas: (1) semantic interoperability by means of consensual knowledge models, (2) access to distributed knowledge resources across supply chain tiers, and (3) adaptive and knowledge-based coordination methods for supply chain operations of autonomous business entities. For this purpose, we adopt and integrate both Semantic Web Service and multiagent technology into this infrastructure. By arriving at formal representations of supply chain operations, we enable a deeper exploration of the search space for fulfilling individual demand efficiently. In particular, we propose representing supply chain operations as software-based services under the designation *Supply Chain as a Service* (Leukel, Kirn, and Schlegel 2010).

The case studies have been taken from selected cooperative research projects of the last four years. They demonstrate the advantages of our approach, and their potential impact on effectiveness, efficiency, and competitiveness of enterprise-networks. Our work in the near future will directly build on these results.

Acknowledgments

The work presented in this paper was partly funded by the European Commission under the project BREIN (Contract FP6-034556), the German Federal Ministry of Education and Research under the projects InterLogGrid (BMBF 01IG09010E) and ProBauDok (BMBF 01IS09023B), and the German Federal Ministry of Economics and Technology under the project AutoBauLog (BMWi 01MA09011). We thank Andreas Scheuermann for his helpful comments on an earlier version of this manuscript.

References

Bratman, M.E.; Israel, D.J.; and Pollack, M.E. 1988. *Plans and Resource-bounded Pratical Reasoning*. Computational Intelligence 4:349-355.

Christopher, M. 2000. The Agile Supply Chain: Competing in Volatile Markets. *Industrial Marketing Management* 29:37-44.

Cunningham, H.; Maynard, D.; Bontcheva, K.; and Tablan, V.. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, USA.

Decker, K. S., and Lesser, V. R. 1993. Generalizing the Partial Global Planning Algorithm. *International Journal of Intelligent Cooperative Information Systems* 1:319-346.

Durfee, E. 1999. Distributed Problem Solving and Planning. Weiss, G. ed. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 121-164.

Ferber, J. 1999. *Multi-Agent-Systems: An Introduction to Distributed Artificial Intelligence*. Addison Wesley.

Ferber, J., and Müller, J.-P. 1996. Influence and Reaction: a Model of Situated Multiagent Systems. In Proceedings of the 2nd International Conference on Multiagent Systems, Kyoto: 72-79.

Foster, I.; Jennings, N.R.; and Kesselman, C. 2004. Brain meets brawn: why Grid and agents need each other. In Proceedings of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004), 8-15.

Hepp, M. 2006. *eClassOWL 5.1.* Products and Services Ontology for e-Business. User’s Guide, http://www.heppnetz.de/eclassowl.

Holzmann, G. J. 1997. The model checker spin. *IEEE Transactions on Software Engineering* 23:279–295.

Hoxha, J.; Scheuermann, A.; and Bloehdorn, S. 2010. An Approach to Formal and Semantic Representation of Logistics Services. In *Proceedings of the ECAI’10 Workshop on Artificial Intelligence and Logistics*, 73-78.

Huget, M.-P., and Koning, J.-L. 2003. Interaction protocol engineering. In *Communication in Multiagent Systems, Agent Communication Languages and Conversation Policies*, M.-P. Huget, (ed.) LNCS 2650. Springer, 179–193.

Jacob, A.; Mueller, M.; and Kirn, S. 2009. An Event-Driven Approach to Dynamic Situation Detection. In Proceedings of the 17th European Conference on Information Systems, Verona, Italy.

Jones, I. ed. 2010. *BREIN Final Demonstrator*. Deliverable D8.1.1, http://www.eu-brein.com.

Karaenke, P., and Kirn, S. 2010. A Multi-tier Negotiation Protocol for Logistics Service Webs. In Proceedings of the 18th European Conference on Information Systems, Pretoria, South Africa, 1559-1560.

Karaenke, P., and Leukel, J. 2010. Towards Ontology-based QoS Aggregation for Composite Web Services. In *Proceedings of INFORMATIK 2010*, vol. 2, 120-125. Bonn: Köllen.

Karaenke, P.; Schuele, M.; Micsik, A.; and Kipp, A. 2010. Inter-organizational Interoperability through integration of Multiagent, Web Service, and Semantic Web Technologies. In Proceedings of the Agent-based Technologies and applications for enterprise interOPerability Workshop (ATOP 2010), Toronto, Canada.

Kidd, P.T. 1994. *Agile Manufacturing – Forging New Frontiers*. Reading, Mass.: Addison-Wesley.

Kirn, S. 2006. Flexibility of Multiagent Systems. In: Kirn, S.; Herzog, O.; Lockemann, P.; and Spaniol, O. eds. *Multiagent Engineering - Theory and Applications in Enterprises*, Berlin: Springer, 53-70.

Kirn, S. ed. 2008. *Individualization Engineering*. Göttingen: Cuvillier.

Leukel, J.; Kirn, S.; and Schlegel, T. 2011. Supply Chain as a Service: A Cloud Perspective on Supply Chain Systems. *IEEE Systems Journal* 5 (accepted for publication).

Liu, B. 2010. Sentiment Analysis and Subjectivity. In: Indurkhya, N. and Damerau, F. J. eds. *Handbook of Natural Language Processing*. CRC Press:Boca Raton.

McCarthy, J., and Hayes, P.J. 1969. Some philosophical Problems from the Standpoint of Artificial Intelligence. In: Meltzer, B., and Michie, D. eds. *Machine Intelligence 4*. Edinburgh University Press: 463-502.

Michelson, B. M. 2006. *Event-Driven Architecture Overview.* Boston:Patricia Seybold.

O'Sullivan, J.; Edmond, D.; and ter Hofstede, A.H.M. 2002. What's in a Service? *Distributed and Parallel Databases* 12:117-133.

Oxford University Press. 2010. *Oxford English Dictionary*. URI=http://www.oed.com.

Pang, B., and Lee, L. 2008 Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2:1-135.

Pine II, B.J. 1993. *Mass Customization: The New Frontier in Business Competition*. Boston:Harvard.

Scherl, R.B., and Levesque, H.J. 1993. The Frame Problem and Knowledge Producing Action. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Washington, D.C., 689-697.

Steiner, D.E.; Haugeneder, H.; and Mahling, D. 1991. Collaboration of knowledge bases via knowledge based collaboration. In *Proceedings of the International Working*

Conference on Cooperating Knowledge Based Systems, 113-133. Berlin: Springer:

Stevens, G. 1989. Integrating the supply chain. *International Journal of Physical Distribution & Materials Management* 19:3-8.

Studer, R., Benjamins, R. and Fensel, D. 1998. Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering* 25:161-197.

Supply-Chain Council. 2010. *Supply Chain Operations Reference Model (SCOR®)* Vers. 10.0, http://www.supply-web.org.

Tseng, M.M. and Jiao, J. 2001. *Mass Customization, in: Handbook of Industrial Engineering, Technology and Operation Management*. New York, NY: Wiley.

Unland, R.; Kirn, S.; Wanka, U.; O'Hare G. M. P.; and Abbas, S. 1995. Aegis: agent oriented organizations. *Accounting, Management and Information Technologies* 5:139-162.

W3C. 2004. *OWL-S: Semantic Markup for Web Services*. URI=http://www.w3.org/Submission/OWL-S.

Warnecke, H.-J. 1993. *The Fractal Company – A Revolution in Company Culture*. Berlin: Springer.

Weyns, D., and Holvoet, T. 2004. *A Formal Model for Situated Multi-Agent Systems.* Fundamenta Informaticae 63:1-34.

Wimalasuriya D. and Dou, D. 2010. Ontology-based information extraction: An introduction and a survey of current approaches. *Information Science* 36:306-323.

Witteveen, C., and de Weerdt, M.M. 2006. Multi-Agent Planning for Non-Cooperative Agents. In *Proceedings of the AAAI Spring Symposium on Distributed Plan and Schedule Management*, 169-170. AAAI Press.

Wooldridge, M.J. 2002. *An Introduction to Multiagent Systems*. New York: Wiley & Sons.