24

Using Ontologies for Configuring Architectures of Industrial Robotics in Logistic Processes

Matthias Burwinkel and Bernd Scholz-Reiter
BIBA Bremen, University of Bremen
Germany

1. Introduction

The provision of goods and services accomplishes a transition to greater value-added-oriented logistics processes. The philosophy of logistics is changing to a cross-disciplinary function. Therefore it becomes a critical success factor for competitive companies (Göpfert, 2009). Thus logistics assumes the task of a modern management concept. It provides for the development, design, control and implementation of more effective and efficient flows of goods. Further, on aspects of information, money and financing flows are crucial for for the development of enterprise-wide and company-comprehensive success.
According to (Scheid, 2010a) this can be ensuring, by the automation of logistic processes. Based (Granlund, 2008), the necessity for automated logistics processes raises the focus on logistics by existing dominant factors of uncertainty and rapid changes in the business area environment. Therefore, the adoption of flexible automation systems is essential. Here robotics appears very promising due to its universal character as a handling machine. This is how (Suppa & Hofschulte, 2010) characterizes the development of industrial robotics: '[...] increasingly in the direction of flexible systems, which take over new fields with sensors and innovative fiscal and regulatory approaches.' Here, logistics represents a major application field. (Westkämper & Verl, 2009) describe the broad applications for logistics and demonstrate the capability for flexibility with examples from industry and research. Besides the technological feasibility, there is also the existing demand by logistics firms concerning the need for their application.
These representations demonstrate the interaction of robotics-logistics regarding the design of technical systems for the operator strongly driven by the manufacturer (technology push) and the technological standardization of the system. Robotic-logistics concentrates on the development and integration of products. Accordingly, standardization activities of the technical systems focus on components and sub-systems that represent the manufacturer-oriented perspective.
The main goal concentrates on the planning, implementation, and monitoring of enterprise-wide process chains of technological systems under the consideration of economic criteria. In this context, the interaction of the two domains 'process' and 'technology' are essential. Thus, the configuration design of technological layouts or machines is crucial. The harmonization of the two domains requires a systematic description framework concerning their exchange of information and knowledge. A high-level abstraction of knowledge representation in the description of the relationships and connections is essential. It also

allows the description of implicit relationships such as comparative relationship notations. This applies to both qualitative and quantitative types of relationships. The outcome is a framework that is available to represent an object dependency between process and technology and to serve the described requirements for flexibility regarding logistics cargo, throughput and machine- and process-layout.
Thus, there is the need for qualitative description of relationship between process and technology by means of specific parameters and properties on a high-level abstraction.

2. Robotics-Logistics: Challenges and potentials

Since the 1970s, there has been a multifaceted development of the basic understanding of logistics. The origin of 'logistics' refers to the Greek 'logos' (reason, arithmetic) and the Romanesque-French ('providing', 'supporting'). In the past logistics were understood in delimited functions. Nowadays logistics are global networks, which are necessary to optimize. The understanding of the task itself changed from a pure functional perspective through process chains to value-adding networks:
Fig. 1 shows the historical development starting in the 1970s. Today's logistics is characterized by its value and integration in the appropriate process chains. The Federal Logistics Association designates logistic processes to the areas of procurement, production, distribution, disposal and transport logistics. (Arnold, 2006) designates differentiated performance-oriented processes as transport and storage processes. Storage processes are the processes of handling, order picking, and packing. Logistics services are evaluated based on delivery time, delivery reliability, inventory availability, delivery quality, and delivery flexibility. These are the objectives for both intra-logistics and extra-logistics. Logistics institutions, such as logistics service providers, provide value-added benefits to this process. These services are dependent of the collection and the output of their product 'commodity'. Finishing or outer packaging operations are examples here.
The logistics of the future will be essentially determined by the automation of material and information flows. In this area, automation systems in logistics already exist for several years. Application areas for these systems, such as de-palletizing and palletizing, sorting, and picking, are 'technically feasible and tested for decades' (Scheid; 2010b). The complete automation of the so-called intra-logistics is technically feasible. However, this situation is not encountered in practice due to the singular character of isolated applications. In future material flow technologies will be more modularized as (Straube & Rösch, 2008) identified. Modular automation systems maximize flexibility in the logistics systems and enable the re-utilization of technical components of handling and storage technology. To summarize the research requirements concerning these technologies (Straube & Rösch, 2008) ask for new modular constructions, which can combine different techniques based on their standardized modular features. This simplifies the integration into new systems. They describe a weakening tendency in new features for the components of conveying and storage technology. The focus is set on the configuration of system architectures composed of existing commercial components. This approach leads to process-specific integrated systems.
From an industrial point of view, multiple logistics areas display a high potential for the automation of processes (Scheid, 2010b). Thus, a high potential exists for the processes 'transport," 'storage' and 'de-palletizing." Transport processes will be automated in 2015 by nearly 30 percent. The reasons for the limiting borders for straightening the degree of automation are lying in the characteristics of the material and information flows. The existing

process dynamics and process volatility are a handicap for standardized processes. The continuous automation of specific and individual processes appears to be difficult due to these reasons. Machine application requires great flexibility for adapting changing parameters.

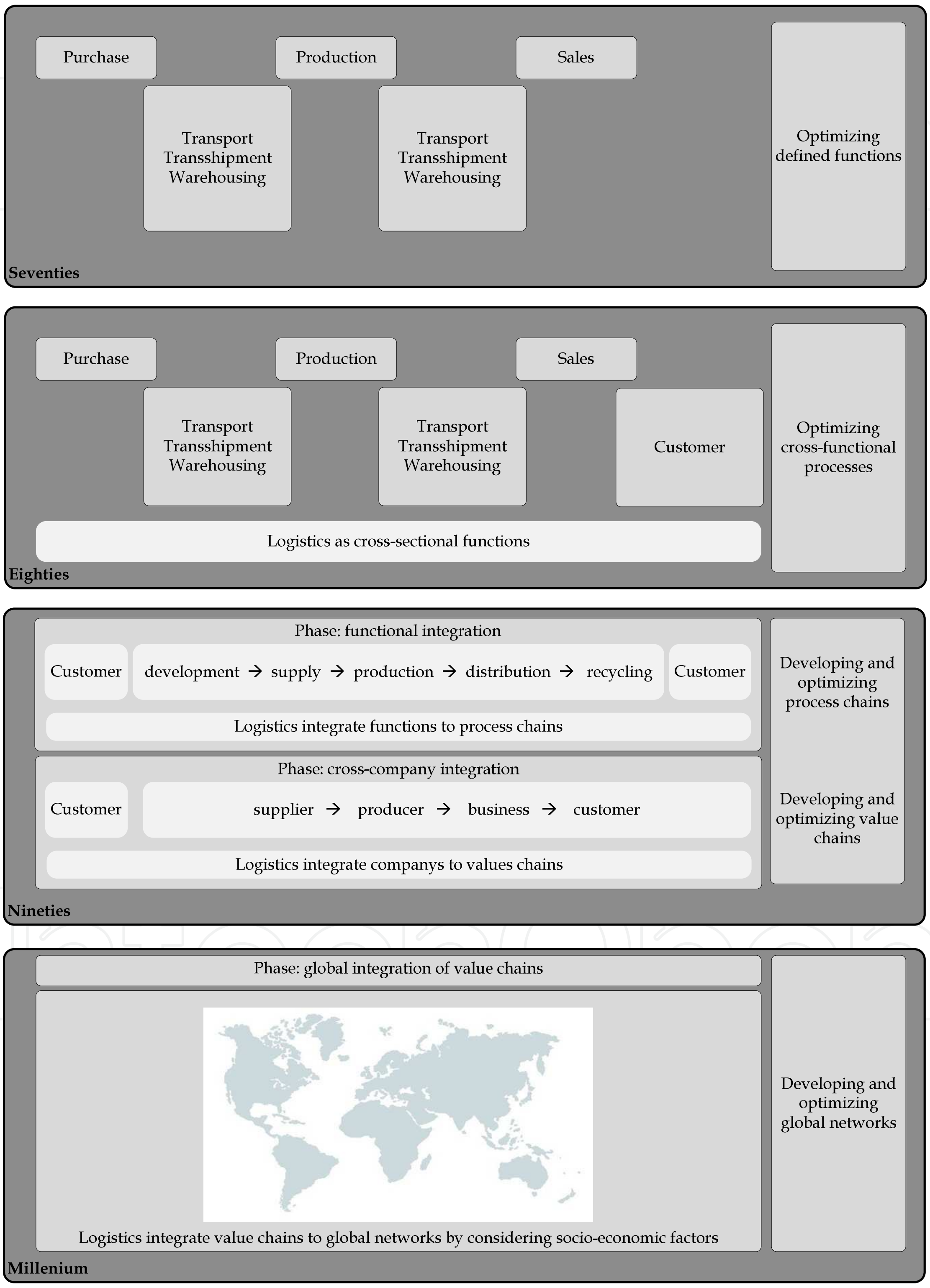

Fig. 1. Historical development of logistics philosophy [source: authors illustration following (Baumgarten, 2008)]

A fundamental role belongs to robotics. By definition, industrial robot systems are a central success factor in process automation due to their universality. The application of robotics systems in logistics factories should be designed flexibly. The automation of the processes under the customers' existing requirements can allow individually designed systems (Günthner & Hompel, 2009). In their recent study, the European Initiative (EUROP, 2009) identified the application of robotics in logistics as a central issue for the future. Thus, it highlights the broad range of application and diverse functions in this area. The operation of the systems under limited process standardization due to the complexity of the processes because of heterogeneous and manifold variables, leads to individual and special solutions in today's logistics factories. The adaption of the systems to changing process environment is hindered due to the process specific character of the systems.

(Fritsch & Wöltje, 2006) identifies the necessity for a paradigm shift from such individual system configuration and underline the relevance of standardized robot systems. This necessity establishes (Elger & Hausser, 2009) by describing the demand of more standardized solutions, which can also serve individual needs. The initiative (EUROP, 2009) characterizes the standardization of components and technical systems as an essential challenge for the so-called 'Robotics 2020." This concerns both hardware and software, and their interfaces among these components. In the authors' view, this requirement influences the system architecture essentially. In the view of the (EUROP, 2009), the system architecture accords robotics a central role. In the future architectures for robotic systems will be designed to both comprehensive configuration conditions and technical subsystems and components. They can be assigned from comparable and very different applications. Therefore, robotics systems will be more modularized in their architecture configurations in the medium term (until 2015). The interconnection between the modules is weakly configured in an overall perspective. On the one hand, this allows a rapid reconfiguration when changes of the process environment appear. On the other hand, the standardization of components and systems is besides the repeated partial usage the second driver of so-called 'adaptable configuration status." The long-term perspective for the year 2020 looks out for the development of architectures down to autonomous self-configurations.

The second crucial development is represented by the compositionality of robotic systems. A robotic system is compositional when the complexity of system architecture is based on compilation of the subsystems or components and their specific functions. The more sub systems or components will be used, the higher is the probability of complex system architecture. Thus, this configuration status is dependent on the process environment. This means that the robotic systems for self-changing or complicated processes must be explicitly designed to fit these requirements. The robotic system has to be configured process orientated. Robotics-logistics configuration conditions appear to diverge in comparison to the configuration condition of production robotics. This can be attributed to the characteristics of logistics processes. The process environment appears to have an essential influence on the technological configuration status of robotic systems. Out of the perspective of system theory, the degree of complexity can be influenced by its technological configuration status of the robotic systems and the characteristics of the process environment.

Thus, complicated processes often require robotic system architectures, which are composed of numerous components and are individually configured. This relationship can result in complicated or complex systems on the process- and on the technology-level. Additionally, procedural complexity influences technical complexity. The necessary reaction possibilities with technical components to procedurally dynamic events are the main driver here. The individual solutions counteract the intended economic standard solutions. Standardization serves to reduce complexity and have to integrate both the process environment and systems engineering. Robotic systems can be standardized by considering the two perspectives of the configuration.

The description of this relational structure is represented by an approach that works with a qualitative logical description on an abstract level. Current approaches to system modeling appear too formal. Ontological approaches with their level of abstraction are an interesting alternative. Despite the standardization of system architecture, a process-orientated configuration is to be ensured. The necessary flexibility intends to serve the dynamic and volatile processes. The construction and structure of the architecture has to be monitored and planned in its modular basic approach. To cover the historical, actual and future usage of technical systems, modular robotic systems are essential.

This book chapter describes a conceptually basic approach procedure for the representation of the relational structure between process and technology through an ontological vocabulary.

3. State of the art - modelling approaches for system representation

Examples of traditional modeling methods for representing systems where relationships between entities are described are: 'entity-relationship model," 'Petri nets,' and 'event-driven process chains' (Kastens et al., 2008, Seidlmeier, 2002, Siegert, 1996).

The Entity Relationship Model (ER-model) was developed in 1976 by Chen. It allows delimited systems to be represented in a way which is intelligible for all involved. The entities (objects) and the relationships between the objects form the basis of the modeling. Regarding the purpose of the modeling, only objects, relationships, and attributes are described (Chen, 1976). The method of Petri nets represents structural coherence between sets of events (Kiencke, 1997). In general, a Petri net is a graphic description where the transaction of the generation of sequences of event-driven networks is represented. It consists of types of nodes, which are representative of a so-called position or transition conditions and events. A directed edge connects a position with a transition. Petri nets are capable of describing a large class of possible processes (Tabeling, 2006). Event-Driven Process Chains (EPC) modeling is a process-oriented perspective on functions, events, organizational units, and information object systems. A process chain is defined by modeling rules and operators (Staud, 2006).

Systems can be also modeled by using ontologies. The concept of ontology originates from philosophy and describes the 'science of being." Many authors define ontology from different perspectives. (Gruber, 1993) describes ontologies as the explicit specification of a conceptualization. The abstract level has the advantage that many basic approaches of different research areas are defined. For example, linguistically and mathematically oriented ontologies are combined due to this definition. (Stuckenschmidt, 2009) establishes the common reference to this definition by many authors. (Studer et al., 1998) take it as a basis and defines ontologies from their formal logic: 'An ontology is a formal, explicit specification of a shared conceptualization.' They emphasize the machine-readable formality of ontology. (Neches et al., 1991) specifies this idea and describes ontologies as 'basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations, to define extensions to the vocabulary.' According to this understanding, concepts are defined through basic distinctions of objects and their rule-based relationships to each other. (Bunge, 1977) describes ontology as the only area of science besides the fields of natural and social sciences, which focuses on concrete objects and concrete reality. Ontologies are to be assigned based on philosophy since they stress the basic principles of virtual science, which cannot be proven or refused by experiments.

Ontologies represent knowledge, which is structured and provided with information technologies. They can be a crucial part of knowledge management. According to (Staab, 2002) knowledge management has the goal to optimize the requirements for employees' performance. The following factors 'persons", 'culture", 'organization" and 'basic organization processes" are the major success criteria for knowledge management. According to (Gruber, 1993) ontologies can facilitate the sharing and exchange of knowledge.

There are many kinds and types of ontologies. Depending on their internal structure, ontologies vary in their complexity, as represented in Fig. 2:

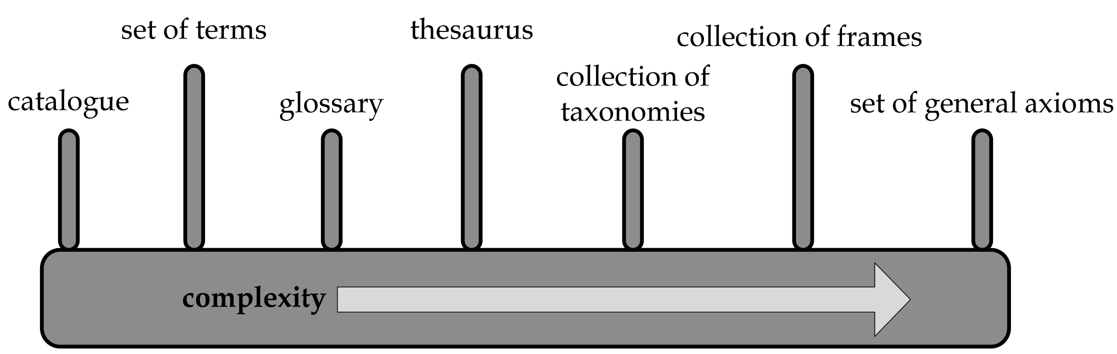

Fig. 2. Types of ontologies organized by increasing complexity [source: authors illustration following (Herb, 2006)]

Examples for trivial complex ontologies are simple catalogues or collections of concepts. Maximally complex ontologies contain an amount of general and weak-defined axioms. An interesting type is taxonomies, which can be defined as a hierarchical classification of concepts in categories.

Taxonomies are also considered as an attenuated definition of ontology. According to (Herb, 2006), they include a series of concepts that are interlinked by hereditary structures. Depending on their nature, ontologies can be applied and re-applied with different levels of intensity (Gómez-Pérez, Fernández-López, Corcho 2004). Ontologies can be classified in so-called 'lightweight ontologies' and in 'heavyweight ontologies.' 'Lightweight ontologies' describe notions (concepts), taxonomies, and relationships and properties between terms. Additionally to these properties, 'heavyweight ontologies' also consider axioms and constraints.

The ontological modeling of systems is possible through the application of existing relationships and rules. (Steinmann & Nejdl, 1999) describe the two tasks of ontologies. In the first sense, ontologies describe the nature of the constituents and the principles. He designates these as 'grammar of reality'. In the second sense, ontologies establish the objects and connections, which (Steinmann & Nejdl, 1999) designate as the 'encyclopedia of reality'. In the first sense, they function as meta-models. Abstract modeling concepts are described and provide the framework for the ontology. Specific meta-model-oriented ontologies will be designated as representation ontologies. For example, the frame-ontology in Ontolingua can be mentioned here, according to (Gruber, 1993). The ontology provides a grammar composed of concepts, relations and attributes. In the second sense, ontologies describe conceptual models are based on structures and correlation of the area of a specific application. Examples of existing conceptual models are legal texts, integration of application systems or open systems.'

Comparing classic and ontological methods, some differences can be identified. Ontologies describe the composition of reality. Traditional modeling approaches assume this information to be known. In this context (Herb, 2006) ascertains, that ontologies are applied for concept-based structuring of information. In his view, ontologies are essentially for more detailed structured information than conventional sources. (Stuckenschmidt, 2009) describes the existence of objects and items and the representation form. (Steinmann & Nejdl, 1999) detail this approach and describe it as a central factor for understanding of items. He concludes that ontologies always are based on a highly abstract level in comparison to model-based approaches.

The authors also indicate the borders of ontological modeling. The crucial difficulties are inconsistencies in classification of meta-data in ontologies, application of meta-data and the distinct classification and structuring of information. Therefore, these aspects are attributable to the highly abstract level of ontologies. Abstract notations lead to such assignment, classification, and structuring issues.

Ontologies can be differentiated in two aspects, conceptual and formal logical nature. According to (Swartout & Tate, 1999), the first aspect has the task of depicting and composing structures of reality. The second addresses the creation of the semantic framework with the definition of objects, classes and contexts. There are many basic approaches to different ontologies in the literature, oriented to the areas of application. (Bateman, 1993) describes the existence of basic types of interconnected entities and describes the so-called 'design patterns.' These are entities that can be differentiated according to the types 'endurant', 'perdurant/occurrence,' 'quality' and 'abstract.' While entities of the 'endurant' type have a continuous and predictable nature, entities of the type 'perdurant/occurrence' describe events that occur unexpectedly and unpredictably. Entities of the types 'quality' and 'abstract' unite properties, attributes, relations, and comparatives.

The ontology DOLCE is an example of the application of these basic types. DOLCE was developed by the Institute of Cognitive Science and Technology in Trento, Italy and stands for 'Descriptive Ontology for Linguistic and Cognitive Engineering.' DOLCE attempts to impart meanings to things and events. Here, entities deal with the meanings through use of agents in order to obtain consensus among all entities regarding to the meaning. (Gangemi et al., 2002) treated this principle in a plausible way. Further examples of conceptual ontologies are WordNet, the 'Unified Medical Language' ontology, 'Suggested Upper Merged Ontology," the ontology of 'ε-Connection' (Kutz et al., 2004), the ontology of 'Process Specification Language,' and the ontology 'OntoClean."

Another key component of conceptual ontologies is ontology engineering. Ontology engineering is concerned with the process design of ontology development, in order to create and to apply ontologies. There are multiple methods here. (Wiedemann, 2008) lists these as follows:

- Ontology Development
- Ontology Re-Engineering
- Ontology Learning
- Ontology Alignment/Merging
- Collaborative Ontology Construction

Ontology Development deals with the question of methodological development and the composition of ontologies. Ontology Re-Engineering focuses on existing approaches and adapts them to the current task. Ontology Learning focuses on approaches for semi- or fully-automatic knowledge acquisition. The Collaborative Ontology Construction issued guidelines for the generation of consensual knowledge. Ontology Merging combines two or more ontologies in order to depict various domains. This method allows handling knowledge that is brought together from different worlds.

(Gruninger, 2002) describes formal logical ontologies as communication, automatic conclusion and representation and re-utilization of knowledge. Formal logical ontologies aim to depict a semantic domain through syntax. The concept of semantics is to be classed in semiotics and describes the theory of signs. Semantics can be also defined as the 'theory of the relationships among the signs and the things in the world, which they denote' (Erdmann, 2001). Semantics are relevant for formal logical ontologies for modeling and generating calculations on a mathematical foundation. This basic approach with its syntax performs a key relevance by providing the mathematical grammar and the concretely denotable model. Exemplary syntaxes are algebraic terms, logical formulas or informational programs. Formal logic provides a language for formalizing the description of the real world and the tool for representing ontologies. It is differentiated according to propositional logic and predicate logic. In propositional logic, there exist exactly two possible truth-values: true or false. Predicate logic consists of terms and describes real world objects in an abstract manner by means of variables and functions. (Stuckenschmidt, 2009) presents methods and techniques of the notation.

Formal logic ontologies do not allow automatic proofs. Only computer-based evidence for sub-problems is possible. Examples of formal logical ontologies are OntoSpace, DiaSpace, OASIS-IP, CASL, OIL, and OWL.

In summary, it can be stated that both ontology types can be classified in different types according to (Guarino, 1998): 'top-level ontologies," 'domain ontologies' and 'application ontologies" which already represent known data and class models. 'Top-level ontologies' describe fundamental and generally applicable basic approaches which are independent of a specific real world. Their level of abstraction is high that allows a wide range of users.

'Domain ontologies' focus on a specific application area and describe these fundamental events and activities by specifying the syntax of 'top-level' ontologies. 'Application ontologies' make use of known data or class models which apply to a specific application area.

The following table finally summarizes the described ontologies and compares them according to the presented properties and characteristics. Furthermore, the relevance of ontologies applicable for robotic logistics is specified and the ontologies of 'Process Specification Language' and the ontology 'OntoClean' are highlighted:

ontology	author	application area	characteristics	relevance
DOLCE	Institute of cognitive Science a Technology, Italien	semantic Web	cognitive basis	partially relevant; wide knowledge, uses cognitive aspects
Onto Clean	Laboratory for Applied Ontology, Trento	hierarchical strucutre of knowledge	checking inconsistencies automatically	relevant for structuring processes and robotics technologies
PSL	National Institute of Standards and Technology, Gaithersburg	neutral representation of process knowledge	modular d	relevant for describving relations between processes and robotics
WordNet	Princeton University	representation of natural languages in IT applications	lexical database	not relevant
UMLS	National Libary of Health	database for communicating medical terminology	terminology for medical applications	not relevant; especcially for biomedicine
SUMO	Teknowledge Corporation	providing information in databases and in the internet	combined out of multiple ontologies	not relevant; designed for automated verification
E-Connection	University of Liverpool	complex correlation of different domains	connecting different domains in a formal and logical way	relevant; complex
CASL	Common Framework Initiative	first-Order-logics for subsuming specific languages	modular concept	not relevant; formal approach
F-Logic	Stony Brook University	deductive database	conceptional and object oriented	not relevant, formal approach
OIL	Vrje Universität, Amsterdam	web based language	formal infrastructure for semantic web	not relevant, formal approach
Ontology Web Language	World wide Web Consortium	representation of correlation in the semantic web	base for integration of software	not relevant, formal approach

Table 1. Comparison of selected ontologies in the context of Robotic Logistics [source: author's illustration]

4. Logical ontologies for configuration of individual system architectures

4.1 Required ontological framework

'Robotic-Logistics' formulates the central expectations to the ontology for configuration robotic system architectures. The input and output variables of the environment due to the reference process have to be defined. On this basis, the relevant domains 'technology' and 'process' can be described as to contents. Classes and variables structure them. On the process side, the reference process is addressed. In this domain, the direct upstream and downstream processes of the reference process are also relevant. The output of the upstream process provides the input of the reference process. The output of the reference process provides the input of the downstream process. The relevant technical systems and components of robotic-logistics will be structured in the technology perspective. The regulatory framework has described the following entities. Fig. 3 gives an overview of the hierarchical structure of the domains 'process' and 'technology':

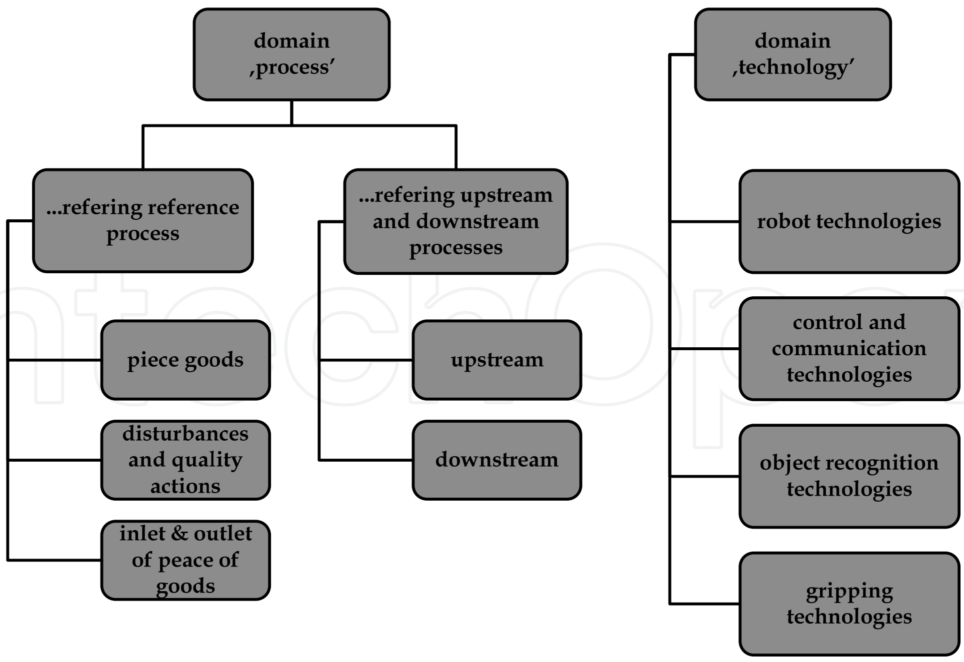

Fig. 3. Class structure of the 'process' and 'technology' domains [source: authors illustration]

The entities in these two class structures are the main processes of meta-model, which will be presented in this chapter. On this basis, process modules and process elements of the reference process are derived with application of the regulatory framework.
The reference process is situated in its systemic environment. It influences the reference process with input and output variables. Thereby the input describes the general framework and restrictions, which are valid for the robotic system. The output results directly from the target dimension of the automation task. The parameters cover technical, organizational, and economic aspects. (Kahraman et al., 2007) define a multi-criteria system for the evaluation of robotic systems, which provides a multiple key factors for the evaluation of robotic systems.
With the development of entity structures of the two domains of the reference process and the inputs and outputs of the environment, the fundamentals of ontology development are set. Based on these structures the hierarchical ontological taxonomies are created with the aid of the ontology OntoClean. This is necessary in order to be able to describe the relations between the two domains through ontology, the Process Specification Language.

4.2 Conceptional ontology for descriptive process technology relations

This section introduces a two-stage approach. In the first phase, the hierarchical structures of the respective domains are composed. The procedure model of (Stuckenschmidt, 2009) offers advantages for the creation of these taxonomies. This approach forms the taxonomies through the OntoClean ontology and analyzes potential sources of error. In the lowest level of taxonomy elements, properties and attributes of the process elements are denoted. They define the reference process. Thus, for example, the process module 'piece goods' with the process element 'bulk" displays the property 'five kilos". Due to this definition, the reference process is individualized and specified.
The second phase provides the combination of the two domains. The description of these relations is done through the ontology of 'Process Specification Language.' It is based on the

descriptive notation of functions and processes through its manifold concepts and relations in different levels of detail. Each participant in a pair of relationship is standardized and the relationship is jointly depicted. Due to the functional and procedural point of view of the ontology, the relationship can be well illustrated. The representation is done by focusing on process elements of one domain that cause an impact on the process elements of the second domain.

For preparation, the conceptual framework is defined as the delimitation of the considered environment to be covered. It is defined according to the procedure model developed by (Figgener & Hompel, 2007). They describe a regulatory framework for reference processes:

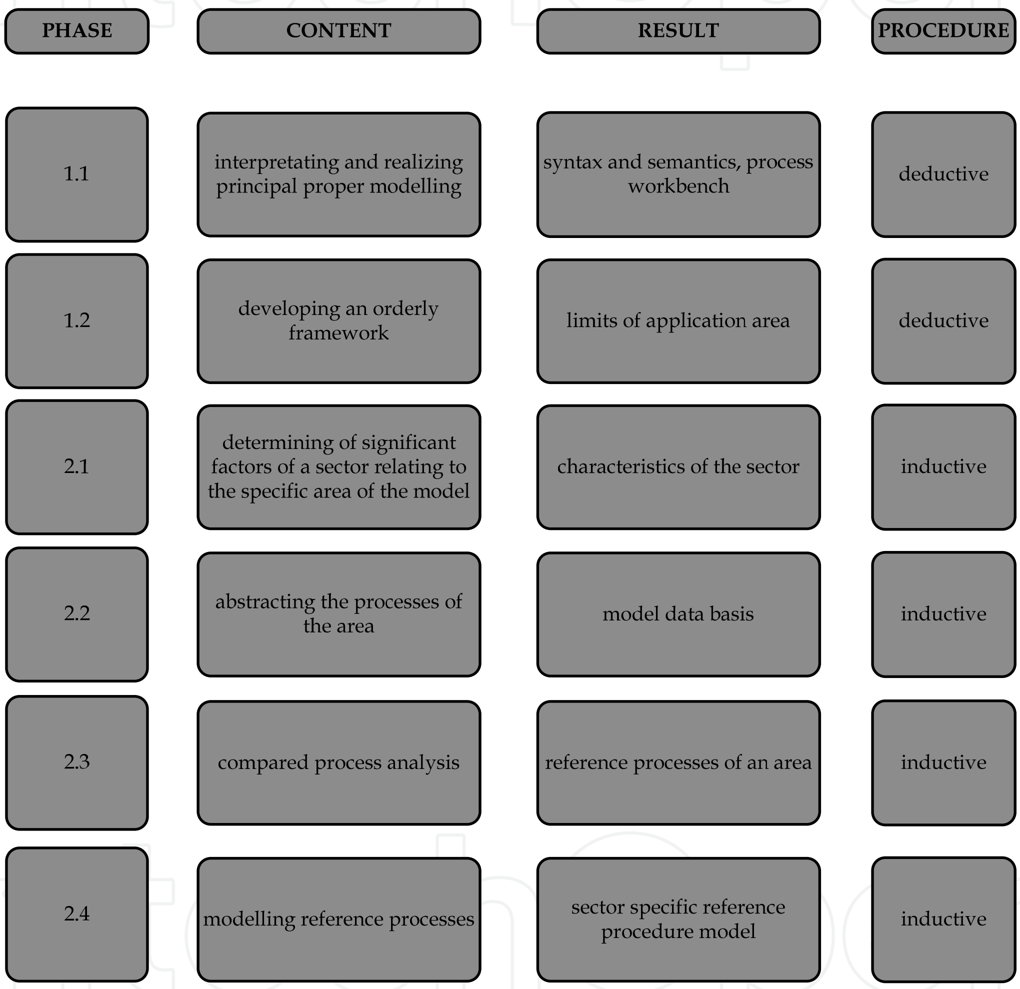

Fig. 4. Procedure model for the creation of reference process models [source: authors illustration following (Figgener & Hompel, 2007)]

The aspects of an application area are defined. Thus, displayed in fig. 4, six phases for the generation of a reference model are described. For the existing problem phase 1.2, phase 2.1, and phase 2.2 are especially relevant. Phase 1.2 describes the regulatory framework. The process modules and process elements are defined in phase 2.1 and 2.2. This distinction allows the reduction and control of the complexity and expenditure for model creation through the ontology.

With these results, both phases of the ontology model can be completed.

Fig. 5 shows the interdependence of both ontologies:

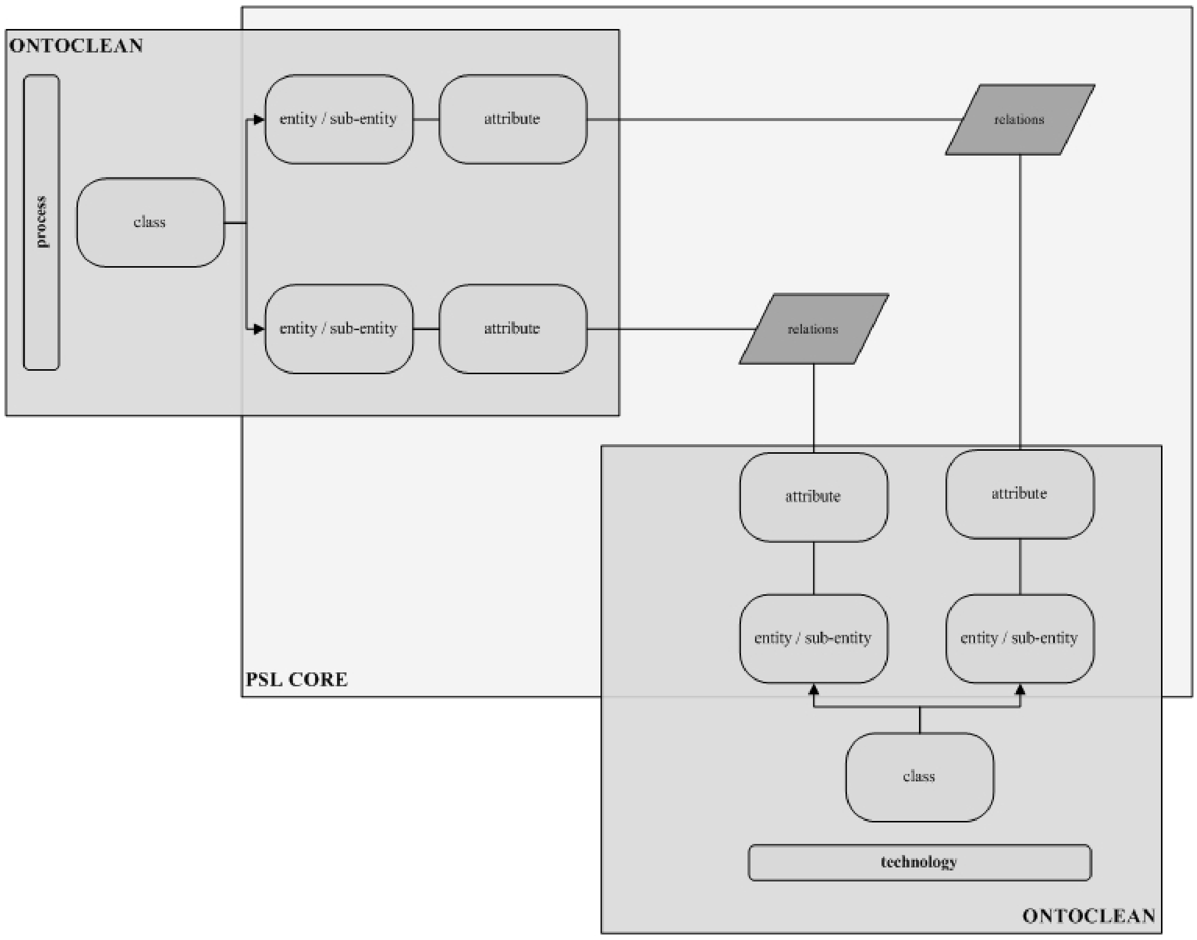

Fig. 5. Spheres of action of the ontologies 'OntoClean' and 'Process Specification Language'[source: author's illustration]

In the first sphere, the taxonomies of the domains ‘process” and ‘technology” are defined with ‘OntoClean’. Depending on the reference process, the process taxonomies will be characterized by properties. They customize the taxonomies. The fig. also displays the sphere of action of the first part namely ‘PSL Core’ of the ontology ‘Process Specification Language.” It identifies the relations, which exist among the entities and properties of the process domain and the entities and attributes of the technology domains. Summarized the figure works out 2 phases of the ontology model.
The first phase develops the taxonomies using OntoClean. The taxonomies are denoted and structured. Based on the notation of the meta-properties, the accuracy of the taxonomies is analyzed. Inconsistencies regarding the clearness of the hierarchies arise when relations are utilized incorrectly. This leads to incorrect and misleading interpretations of the ontology (Herb, 2006). The OntoClean process examines existing subsuming structures existing between classes by using meta-features.
The second phase depicts relations between the domains by using the ontology of 'Process Specification Language'. Fixed taxonomies for the domain ‘process” and the domain ‘technology” for a specific reference process, are the basis for representation of the interaction between the two domains. The main goal of the ontology is to figure out parameters or components of a domain affecting the second domain. Besides the demonstration of the existing relations, the description of the quality of the relationship is an

essential aspect. Thereby both the direction of the relationship and the qualitative description will be identified. With the ontology model, the defined requirements will be satisfied as follows:

requirement	ontology	contribution
structuring domains	OntoClean	- identification of relevant terms - structuring by using taxonomies
notation relations	PSL	- identification and notation of relations between process and technology
description relationship	PSL	- description of the relations

Table 2. Handling the requirements through the ontology model [source: author's illustration]

The ontology OntoClean structures the domains. It defines the concepts and composes the taxonomies of the domains ‘process” and ‘technology”. The ‘Process Specification Language’ note the relations between the parameters. The worked out ontology model is the basis for the individual process modularization and configuration of technical robotic systems.

4.2.1 Definition of taxonomies using the ontology ‘OntoClean’

The structuring of the domains is done by defining taxonomies. The usage of taxonomies joins and collects concepts and entities and forms a base frame for these ontologies by structuring them. Here, the relationships are developed associatively mutually. Descending rules work out the taxonomy structures. Due to the qualitative character, the taxonomies are often incorrectly distinguished. The process of ‘OntoClean’ creates taxonomies and checks the consistency and accuracy of the structures.

The procedure involves the definition of taxonomies and their meta-properties. It aims at overcoming the frequent deficit of false descent of entities in the taxonomic structure. These erroneous subsuming structures will be avoided by a philosophy-based distinction of the entities and classes with meta-properties. (Herb, 2006) describes comprehensively the meta-properties 'identity,' 'essence and rigidity,' 'dependency,' and 'unity.' Using these meta-properties, the taxonomies are distinctly defined through the concepts of class, entity, instance, and property. Entities describe the objects of taxonomy, which are collected in classes. Entities, which have a common property, instantiate a class and will be defined as instances. This is of great relevance. Especially the representation is challenging due to multiple components and parameters, which are displayed in both the domain ‘technology’ and the domain ‘process’. Table 3 provides an overview of the conceptions.

The concepts are the basis of the taxonomies to be created. They are based on the entity structures. For a specific reference process, the entities are reviewed and adapted individually. The procedural taxonomies are developed based on the meta-models of process standardization developed by (Figgener & Hompel, 2007). Depending on the type of process, the main processes, process modules and process elements are applied. Based on the structured system techniques in the domain ‘technology’, the technical taxonomies are defined due to the commercial state of the art.

The claim of universality is not maintained. The conception is defined to each reference process specifically. This increases the risk of erroneous and inconsistent definition and description of the concepts. Due to this circumstance, the analysis and validation of the developed taxonomies is an essential part of ontology development.

term	definition of procedure model due to fig. 4	commentary
class	main process	structuring of classes
entity	process module	entities which are subsumed in one class
instance	process module with same attributes	all entities with same attributes in one process module
property	process element	characterization and individualization of the reference process

Table 3. Definition of conceptions in the framework of 'OntoClean' [source: authors illustration]

The OntoClean process provides a procedure of subsuming, shown in fig. 6:

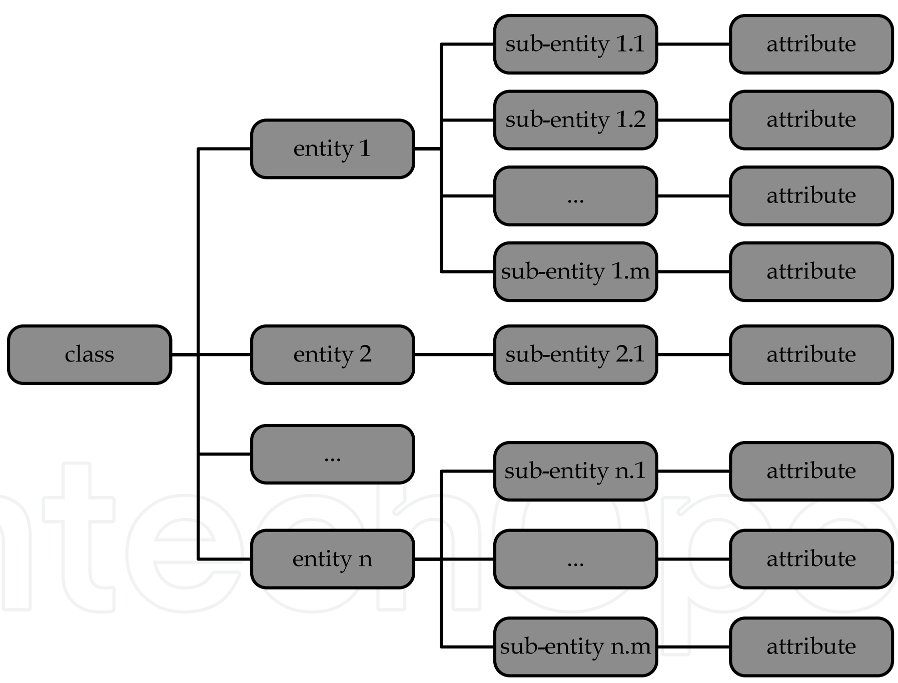

Fig. 6. General construction of taxonomy [source: author's illustration]

A class A subsumes a class B if all instances of class B are always also instances of class A. Fig. 6 presents a class with n entities. Each entity can display further lower-level hierarchical entities, known as sub-entities. Thus, the number of vertical levels is unlimited. On the lowest vertical level, the reference process is individualized by distinct properties. They provide the specific information about the reference process. These may be quantitative or qualitative. As an example here, for a procedural taxonomy, a sub-entity of type 'mass' can be specified with the quantitative property of '22 kg'.

A special case is presented due to the class structure ‘environment’. Here, both are structured the environmental framework conditions and the target dimensions. This taxonomy is independent of the reference process, and provides an example, shown in its basic structure, as follows:

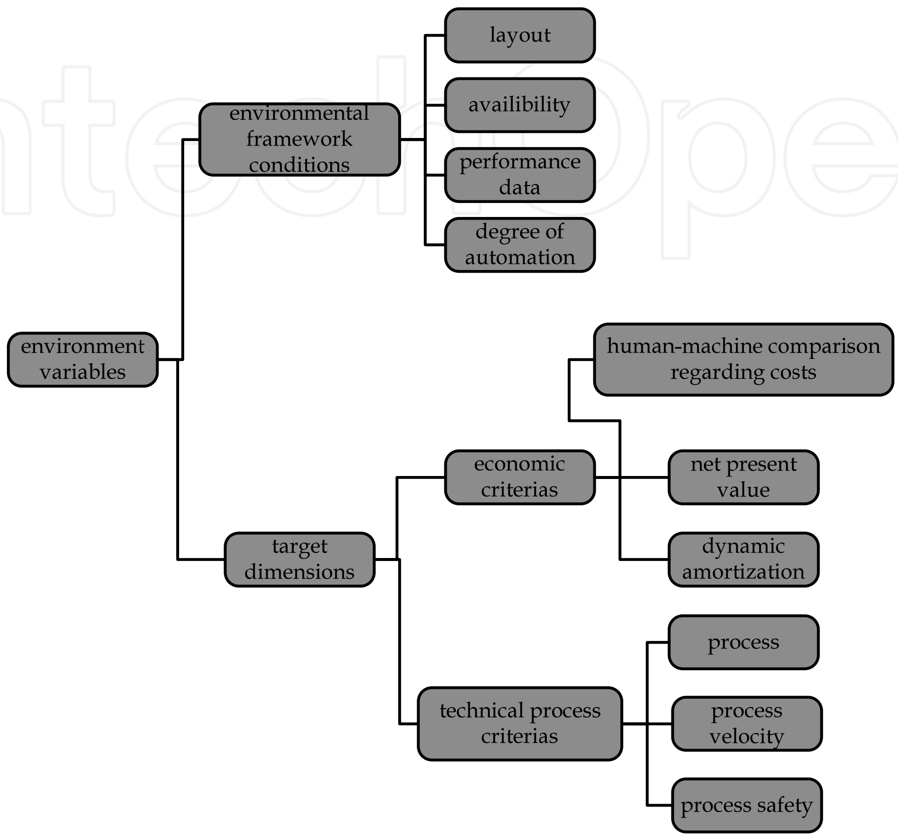

Fig. 7. Taxonomy of the environment variables [source: author's illustration]

The environmentally framework conditions define the technical requirements, such as availability and performance data. The listed properties are specifically defined for each reference process. They customize the process. By means of the target dimensions, technical and economic criteria are carried out. In this context, process safety or process velocity on the technical side are determined. On the economic side, capital value or amortization time is identified. The target dimensions represent the criteria for success of the realization of the robotic system in an ex-post manner.

For each taxonomy and its class K, a notation is defined with the property M. K is denoted as +M, when M applies to all instances of K. The notation -M is used, if not all instances of K have the property M. If M is not valid for any instance of the class K, this relationship is denoted by ~M. Each of the four meta-properties will be reviewed to that effect for each class and entity. This describes the meta-property, 'essence and rigidity' by fixing essence in general first. Second, the specific form of the essence, the rigidity, is described. A property is essential for an entity, if it occurs in every possible situation in the entity. In the next step, a property is rigid, if it is essential for all instances (+R). Non-rigid properties are referred with –R. They describe properties for those entities, which are needed not, but may be instances of the class. Anti-rigidity (~R) is available if there is no instance of associated class instance of the corresponding class.

The meta-property 'identity' describes criteria, which distinctly identify classes and differentiates instances from each other. Both classes and upper classes can provide these identity criteria. The upper classes inherit the criteria. In the first case, the classes are marked with +I. Thus, the identity criterion has been inherited by an upper class. In the second case, the criterion of identity is first defined in an upper class and is marked with +O. Classes that require a further identity criterion as restrictions for distinct definition are denoted with –I.
The third meta-property 'unity' is related to the property 'identity' and describes the affiliation of certain entities to a class. A unity criterion defines a unifying relation of all entities, which are interconnected. The corresponding classes are distinguished with +U. ~U denotes those entities of a class that cannot be distinctly described. If there is no unity criterion provided, the class is described with –U.
The fourth meta-property 'dependence' describes the dependence of a class to other. This fact is relevant if an instance of a class may not be an instance of a second class. Dependent classes will be listed with +D, while independent classes are notated with –D.
In summary, the meta-properties are defined as follows, according to (Herb, 2006):

Meta-property notification	definition
+R	a property is essential for all valid instances
-R	a property that has not inevitable an entity that is an instance of its class
~R	a property where an instance of an allowing class belongs to an instance of a regarded class
+I	classes which differentiate due to the criteria of the allowing instances
-I	class that does not have an identity criteria
+O	identity criteria that is defined for the first time and is not transmitted
+U	unity criteria that denotes connected entities
-U	none unity criteria is existing
~U	connection of entities which cannot described definitely
+D	dependent classes
-D	independent classes

Table 4. Summary definition of meta-properties, [source: authors illustration following (Herb, 2006)]

The review of meta-properties shows incorrect taxonomy structures and makes their correction possible. The next step involves reviewing the consistency of the meta-properties with each other. This will determine whether there are inadmissible combinations of meta-properties. , An example for such a combination is +O und –U. The next step focuses at the removal of all non-rigid classes from the taxonomy.

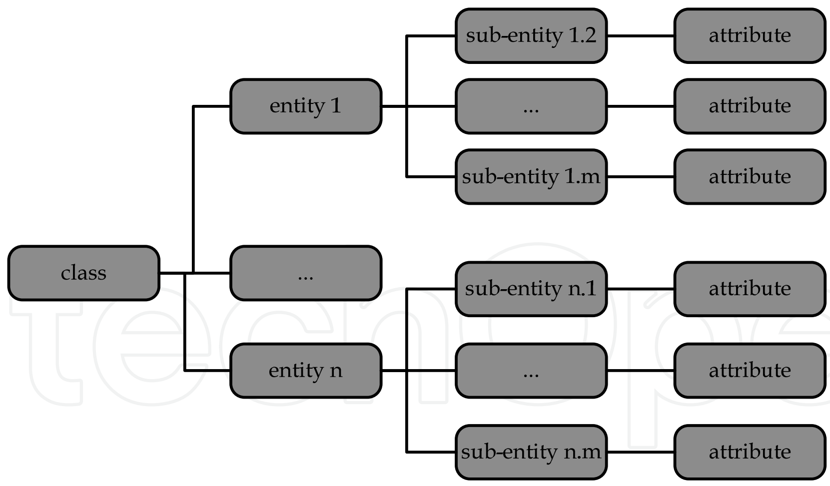

Fig. 8. Backbone taxonomy [source: author's illustration]

That figure points out for an exemplary illustration the removal of the non-rigid 'sub-entity 1.1' and the non-rigid 'entity 2'. This procedure results in the so-called backbone taxonomy. In the next step, subsuming structure has to be examined. It checks any violations of subsuming restrictions. Subsuming is described with the relation 'is-a' and is visualized by arrows. For instance, further relations are described with the notation type 'has.' To avoid false distinctions the arrows are inscribed with the relation name. The hierarchies can be described in the following ways:

- *Have*: The relationship type 'have' connects an attribute with a concept. Thereby, the Attribute is a type of the concept.
- *Att*: This type of relationship describes properties of elements. A concept can take on several properties simultaneously. They do not need to be met simultaneously at all.
- *Is a*: The relationship describes traditional subset relations (subsuming relations).

In the last step, the non-rigid classes and entities are added. Within this last step, the taxonomy is completed.

4.2.2 Description of the interaction by using the ontology of 'Process Specification Language'

In this section, the interacting entities and attributes have to be identified between the two domains by using the created taxonomies. In order to give these interactions a qualitative meaning, (Schlenoff et al., 1999) propose an approach to denote interactions of processes in independents worlds. Therefore, he develops the terminology 'Process Specification Language' (PSL). PSL is a neutral, standard language for process specification for the integration of multiple process applications within the product life cycle. The language is versatile in application and uses multiple concepts, classes, and functions to describe complex processes. Through its manifold applications and many years of further development, the language has diversified and expanded. PSL consists of several modules. The principle fundamental concepts are set in the first module which is called 'PSL-Core'. The module provides four concepts with corresponding functions. According to (Schlenoff et al., 1999), the aim of this module is to fix axioms to describe trivial process connections using a set of semantic relations.

The description of further and more complex processes is carried out with other modules, the so-called extensions. PSL offers in total three extensions: 'outer core', 'generic activities' and ‘schedules’. The module ‘PSL outer core’ deals with generic and broadly based concepts regarding to their applicability. The module ‘generic activities’ defines a terminology to describe generic activities and their relations. The module ‘schedules’ describes the application and allocation of resources to activities under the premise of satisfying the temporary restrictions:

term	definition
PSL	short notification of the ontology ‘Process Specification Language’
PSL module	group of concepts of the PSL
relation	interrelation between an entity couple of ‘process domain‘ and ‘technology domain‘. None of the entities is a ad-hoc activity.
activity	process or technical entity or sub-activity that is continuous and relates to the second domain.
ad-hoc activity	process or technical (sub-) entity including its attribute that existence is not calculable
concept	first and highest level of a PSL
class	second level of PSL
function	third and lowest level of PSL
ad-hoc relation	relation of an entity couple of the process and technology domain. Minimum one activity is an ad-hoc activity.

Table 5. Definition of relevant terms of PSL [source: author's illustration]

The definitions are based on the adaptation of the ontology to the current requirements. With these concepts, the individual concepts of this approach will be presented and adapted to this task. With the creation of taxonomies, the relations of taxonomy properties of both domains are identified and described.
This section describes the identification of existing relations and their corresponding notation using the vocabulary of the first module ‘PSL Core’ of the ontology ‘Process Specification Language’. The module contains three concepts. The first concept 'activity' describes general activities, which appear to be predictable and manageable. They do not have to be determined detailed. For instance, ‘activities’ may be standard processes of a recurring nature. The concept exhibits two different types of functions for further concepts. Function one focuses on further planned activities ('activity') ('is-occurring-at'). Function two describes the connection to unpredictable and unplanned activities ('occurrence-of').
The second concept, 'activity occurrence' describes a unique activity that proceeds unforeseen and unplanned. The concept can also exhibit two different functions for other concepts. The function 'occurrence of' is analogous to the second function of the first concept and describes the initiation of a second type of unpredictable activity of the type 'activity occurrence.’ The second function describes the relationship to a concept of the type 'object’. This function expresses the impart of the concept ‘activity occurrence' with a none further defined significance to the second concept named 'object.'

The third concept, 'object' describes all activities which do not correspond to any of the above concepts. The concept has two functions. The first relation of the type 'participate in' describes the concept 'object' which receives a non further defined relevance for a concept 'activity.' The second relation 'exists-at' describes an existing relevance to a particular point of time.
The entities are, inclusive of their properties, distinguished from the taxonomies of process and technology domains with these concepts. Here, procedural entities and properties can exist which are either calculable or definable. These activities relate to the concept 'activity'. Unpredictable, indefinable or changing conditions can be described as ad-hoc activities and assigned the concept of 'activity occurrence.' Other logistical or technical objects are called objects and assigned to the concept 'object.' An example describes the entity 'general cargo' as an activity (code 1.1) with its property 'cubic' and the entity 'stock situation' for a concept named ad-hoc activities (code 1.2) with the property 'chaotic'. An example of an object (code 1.3) is a technical process such as the process of recognizing the cargo. The following table summarizes the results of the relevant vocabulary:

PSL module	concept	definition	relation	definition	modification for robotics-logistics	code
PSL Core						1
	acitvity	a general non-defined acititvity			defined and calculable activity that notes an entity or an attribute of taxonomy.	1.1
			is-occurring-at	a primary activity generates a secondary activity at a defined time	the concept 1.1 generates a concept 1.1	1.1.1
			occurrence-of	the primary concept generates a secondary non-expected activity	the concept 1.1 generates a concept 1.2	1.1.2
	activity occurrence	a temporary activity and specific activity that occurs nonrecurring			a non-calculable and changing ad-hoc-activity that notes an entity or attribute of a taxonomy	1.2
			occurrence-of	the primary activity generates a secondary non-expected activity	the concept 1.2 initiates a new concept 1.2	1.2.1
			participates-in	a primary activity generates a non-definable relevance for an object	the concept 1.2 generates a non-defined relevance for an object at a specific time	1.2.2
	object	all entities that are not an activity or activity occurrence			entity or attribute of a taxonomy that are not concepts 1.1 or 1.2	1.3
			participates-in	a primary activity assigns a non-defined relevance to an object in a specific time	the concept 1.1 assigns a relevance to a concept 1.3	1.3.1
			exists-at	an object exists to a specific time	the concept is relevant in a specific time	1.3.2

Table 6. Vocabulary PSL module 'Core' [source: author's illustration]

In a first step, the implementation of ontologies for a specific reference process is associated with concepts and properties of the valid entities. The second step identifies and denotes the relations between the concepts. A matrix representation is provided which is shown in the general structure in tab. 7. The columns show the entities and properties of the technology domains. The lines depict the process domains. The individual hierarchy steps of taxonomies are presented. As described, they were indicated by the hierarchic structure. The coding of the lines and columns indicates the respective levels of the hierarchic structure. Additionally, the identified concepts of the respective sub-entities and properties are noted on the lowest structural level. In the cells, the interaction from tab. 6 are noted and distinguished by means of the coding. For example, the procedural sub-entity 1.1.1 affects the technical components 1.1.1 through the relationship 'object-participates-in' (code 1.3.1).

Process Specification Language				**Code**	**T.1**	**T.1.1**	**T.1.1.1**	**...**	**...**	**T.n**	**T.n.m**	**T.n.m.o**
				technical- taxonomy	**system technique 1**			**...**	**...**	**system technique n**		
						system technique 1.1		**...**	**...**		**system technique n.m**	
PSL Core							**component 1.1.1**	**...**	**...**			**component n.m.o**
Code	**process taxonomy**			**PSL-concept**			**concept 1.z**	**...**	**...**			**concept 1.z**
P.1	**Class 1**							…	…			
P.1.1		**entity 1.1**						…	…			
P.1.1.1			**sub-entity 1.1.1**	**concept 1.x**			1.3.1	…	…			1.x.y
...	**...**	**...**	**...**	**...**	…	…	…	…	…	…	…	…
...	**...**	**...**	**...**	**...**	…	…	…	…	…	…	…	…
...	**...**	**...**	**...**	**...**	…	…	…	…	…	…	…	…
P.n	**class n**							…	…			
Pn.m		**entity n.m**						…	…			
P.n.m.o			**sub-entity n.m.o**	**concept 1.x**			1.x.y	…	…			1.x.y

Table 7. General matrix representation of the process-technology relations in accordance with PSL module 'core' [source: author's illustration]

The vocabulary allows the description of the relational structure for a dedicated reference process, which describes the relations among the procedural entities and the technical components.

4.3 Industrial application: Depalletizing plastic boxes with a robotic system

This section presents the robot based automation of a simple industrial application by using the presented ontological framework. The presented example focuses on the interaction between 'piece good' of the 'process domain' and 'gripper' of the 'technology domain'. Here the automation of a logistics process by using the ontological framework will be presented. Online books shops package their goods in plastic boxes. Logistics Providers handle these boxes for delivering to the customer. Hence, the boxes are send on pallets in swap bodies by using trucks. The logistics provider unloads the trucks and imports them in their distribution center which operates with a high degree of automation. Therefore the boxes have to be depalletized and brought onto the conveyor technology system. In general this separation is done manually. A robotic systems was configured and integrated by using the ontological framework to automate this reference process. The following figure displays the process with the implemented robotic system:

Fig. 9. Industrial application of a robotic system for depalletizing plastic boxes: result of the ontological configuration [source: author's illustration]

The illustration points out that the configuration of the technical system depends on the parameters of the process. The upstream conveyor supply box pallets including a buffer function. The downstream conveyor conveyors the single boxes into the distribution cycle. The task of an robot-based automation system focuses on the handling of single or multiple boxes and the lay down onto the roller conveyor. Using the presented framework for configuring the robotics architecture, the first step of generating the procedural and technical taxonomies has to be executed. The class structure with its entities and attributes of the 'process domain' can be assigned with attributed as followed:

Entity	meta- property	sub-entity	attribute	meta-property
geometry	+R, +I, -U, +D	form	cubic	+R, -I, -U, +D
		dimension (min, max)	length = 55 [cm] width = 40[cm] height = 30 [cm]	+R, +I, -U, +D
		volume	V = 27 [l]	-R, -I, -U, -D
		surface	closed	+R, -I, -U, -D
material	+R, -I, -U, -D	art	plastic	+R, +I, +U, +D
		stability	high	-R, -I, -U, -D
packaging	+R, +I, -U, +D	strapping	1	-R, +I, -U, -D
		type of packaging	single	-R, +I, +U, +D
mass	+R, +I, -U, +D	weight	28 [kg]	+R, +I, -U, +D

Table 8. Entities and attributes of the class structure 'piece goods' of the 'process domain' [source: author's illustration]

The table displays the various entities with its attribute regarding the class structure 'piece goods'. For instance, the meta-properties define the taxonomy of this class. Also the hierarchical structure is defined. For instance, incorrect assignments of sub-entities will be avoided. Additionally the table assigns relevant attributes of the reference process to entities. The meta-properties figure out that the sub-entities 'geometry' and 'mass' are quite important due to their essential (+R). Furthermore they give identity to their class (+I). Finally, the corresponding taxonomy is presented in figure 10:

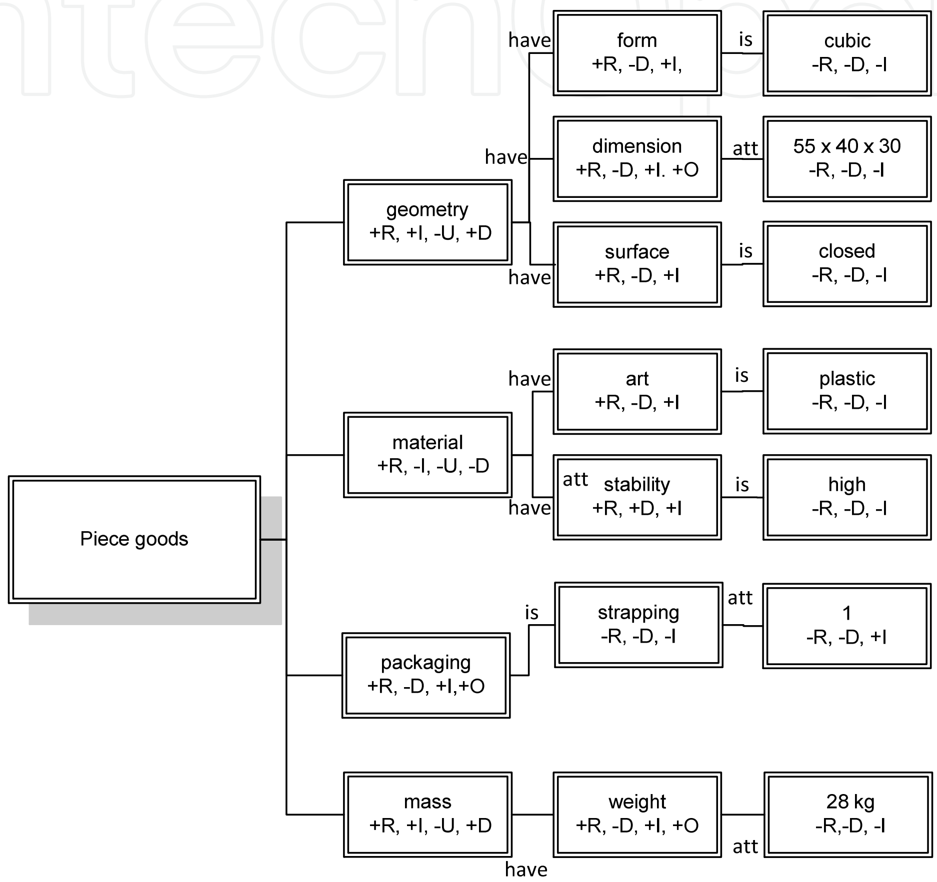

Fig. 10. Backbone taxonomy of the class structure "piece goods" of the "Process domain" [source: author's illustration]

It displays the corrected taxonomy of the exemplary class structure of the industrial application. The class "piece goods" consists of four entities (2nd level) and seven sub-entities. These are related with to attributes. The relations 'subsumption' (is) or 'attribute' (att) describes the connection to the entities. Subsumption are given if the attribute is part of the sub-entity.

The following step focuses on interactions between the "process domain" and "technology domain". Therefore, the relations are noted. Table 9 exhibits the relationship to the system technology "robotics" which unite the entities "kinematics", "geometry", "load", "accuracy" and "installation". Table 9 displays the relation codes between the two domains. For instance, there are some relations of the type "ad-hoc-activity" and "activity". For instance, the strapping has an influence to the accuracy of the robot. Also the mass defines the type of the robot. The table resumes these relations:

Process Specification Language	**PSL Core**	T.1 robotics	T.1.1	T.1.2	T.1.3	T.1.4	T.1.5
			kinematics	geometry	load	accuracy	installation
P.1	piece goods						
P.1.1	geometry		1.1.2	1.1.1			
P.1.2	material		1.1.1		1.2.1	1.1.1	
P.1.3	packaging					1.1.1	
P.1.4	mass				1.1.1		

Table 9. Entities and attributes of the class structure 'piece goods' of the 'process domain' [source: author's illustration]

This example clarify the potential of the ontological framework. The framework offers a general and systemic knowledge to configure the best technical components and modules for the specific application due to the system technologies "robotics", "gripping technology", "pattern recognition" and "robot control and communication".

5. Conclusion

The paper presents an ontological approach to standardize robotic systems in logistic processes. Ontologies allow the systematic depiction of the technical systems in the procedural environment. Through their high level of abstraction, this chapter describes the conceptualization and elaboration of an ontological vocabulary for configuration process customized robotic architectures. The vocabulary allows the description of the relational structure for a dedicated reference process. It describes the relations among the procedural entities and the technical components. This ontology framework is the basis for the formation of modules and the configuration of the modules in robotics architectures.

The main goal provides a descriptive approach to the relationship between process and technology. Here, representations of conceptual ontologies were consulted. Due to the conceptual approach, the notation is on an abstract level, so that an automatic conclusion through formal ontologies is realistic. The representation of a dedicated solution space of possible technical configuration states of robotics system architectures is feasible, too.

In further research requirements, the development of formal ontologies in the context of this scope reduces the level of abstraction and enables the mechanical and automatic generation of ontologies.

In this way, interpretation and manipulation opportunities will be reduced and the interconnections of relationships between process and technology detailed. In this connection, formal ontologies allow the development of so-called architecture Configurator. They are based on the provided procedural and technical information and the possible ontological interrelationships. With this information, automatically development, including economic criteria, prioritizes configurations of robotic system architectures for dedicated reference processes. This approach can also serve as an appreciation of the nature of a 'Rapid Configuration Robotics' approach, which can digitally review prototyping activities such as technical feasibility and economic usefulness. The requirement for this type of IT-based configuration planning is shown by the RoboScan10 survey:

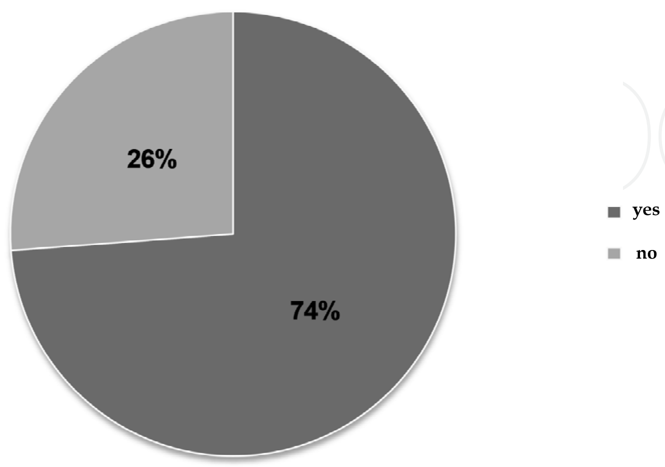

Fig. 11. Study RoboScan10: Answers about necessity for an IT-based system that plans the configuration of robotic systems [source: (Burwinkel, 2011)]

Fig. 11 shows the field of opinions in the context of RoboScan10 about the necessity for IT-based configuration planning of robotic systems. The question asked was: 'from a planning perspective: Could you envisage using an IT-based planning tool, which enables the configuration of both single robot systems and multi robot systems? 75% of all respondents could envisage the application of such tools.

6. References

Arnold, D. (2006). *Intralogistik : Potentiale, Perspektiven, Prognosen,* Springer, ISBN 978-3-540-29657-7, Berlin, Germany

Bateman, J. (1993): On the relationship between ontology construction and natural language: a socio-semiotic view. *International Journal of Human-Computer Studie,* Vol.43, No.5/6, pp.929-944, ISSN 1071-5819

Baumgarten, H. (2008). *Das Beste der Logistik : Innovationen, Strategien, Umsetzungen,* Springer, ISBN 978-3-540-78405-0, Berlin, Germany

Bunge, M. (1977). *Treatise on basic philosophy,* Reidel, ISBN 9027728399, Dordrecht, Netherlands

Burwinkel, M. & Pfeffermann, N. (2010). Die Zukunft der Robotik-Logistik liegt in der Modularisierung : Studienergebnisse "RoboScan'10". *Logistik für Unternehmen,* Vol.24, No.10, pp.21-23, ISSN 0930-7834

Chen, P. (1976). The entity-relationship model - toward a unified view of data. *ACM Trans. Database Syst*, Vol. 1, No. 4, pp. 9–36

Elger, J., & Haußener, C. (2009). Entwicklungen in der Automatisierungstechnik. In: *Internet der Dinge in der Intralogistik,* W. Günthner (Ed.), 23–27, Springer, ISBN 978-3-642-04895-1, Berlin, Germany

Erdmann, M. (2001). *Ontologien zur konzeptuellen Modellierung der Semantik von XML,* University Karlsruhe, ISBN 3-8311-2635-6, Karlsruhe, Germany

EUROP. (2009). Robotics Visions to 2020 And Beyond : The Strategic Research Agenda For Robotics in Europe, In: *European Robotics Technology Platform,* 20.06.2011, available from: www.robotics-platform.eu

Figgener, O. & Hompel, M. *Beitrag zur Prozessstandardisierung in der Intralogistik.* In: *Logistics Journal* (2007). ISSN 1860-5923 S. 1–12

Fritsch, D. & Wöltje, K. (2006). Roboter in der Intralogistik : Von der Speziallösung zum wirtschaftlichen Standardprodukt. *wt Werkstattsstechnik Online,* Vol.96, No.9, pp. 623–630, ISSN 1436-4980

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A. & Schneider, L. (2002). Sweetening Ontologies with DOLCE, In: *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web,* Gómez-Pérez, A. & Benjamins, V. (Ed.),. pp.223–233, Springer, ISBN 978-3-540-44268-4, Berlin, Germany

Gómez-Pérez, A., Fernández-López, M., Corcho, O. (2004). *Ontological engineering: With examples from the areas of knowledge management, e-commerce and the semantic web,* Springer, ISBN 978-1-852-33551-9, London, England Göpfert, I. (2009). *Logistik der Zukunft - Logistics for the future*, Gabler, ISBN 978-3-834-91082-0, Wiesbaden, Germany

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition,* Vol. 5, No.2, pp.199-220, ISSN 1042-8143

Gruninger, M. (2002). Ontology - Applications and Design. *Communication of the ACM,* Vol.45, No.2, pp.39–41, ISSN 00010782

Guarino, N. (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS '98), June 6 - 8, Trento, Italy,* FOIS, ISBN 905-1-993-994, Amsterdam, Netherlands

Günthner, W. & HompelM. (2009.). *Internet der Dinge in der Intralogistik,* Springer, ISBN 978-3-642-04895-1, Berlin, Germany

Herb, M. (2006). Ontology Engineering mit OntoClean. In: *IPD University Karlsruhe,* 10.06.2011, available from:
http://www.ipd.uni-karlsruhe.de/~oosem/S2D2/material/1-Herb.pdf

Kahraman, C., Cevik, S., Ates, N.Y. & Gulbay, M. (2007). Fuzzy multi-criteria evaluation of industrial robotic systems. *Computers & Industrial Engineering,* Vol.52, No.4, pp. 414-433, ISSN 0360-8352

Kastens, U. & Kleine Büning, H. (2008). *Modellierung : Grundlagen und formale Methoden,* Hanser, ISBN 978-3-446-41537-9, München, Germany

Kiencke, U. (1997). *Ereignisdiskrete Systeme: Modellierung und Steuerung verteilter Systeme,* Oldenbourg, ISBN 348-6-241-508, München, Germany

Kutz, O., Lutz, C., Wolter, F., Zakharyaschev, M. (2004). ε-connections of abstract description systems. *Artif. Intelligence,* Vol.156, No.1, pp.1-73, ISSN 0004-3702Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. & Swartout, W. (1991). Enabling technology for knowledge sharing. *AI Mag,* Vol.12, No.3, pp.36–56

Scheid, W. (2010). Perspektiven zur Automatisierung in der Logistik : Teil 1 - Ansätze und Umfeld. *Hebezeuge Fördermittel - Fachzeitschrift für technische Logistik,* Vol.50, No.9, pp. 406–409, ISSN 0017-9442

Scheid, W. (2010): Perspektiven zur Automatisierung in der Logistik : Teil 2 - Praktische Umsetzung. *Hebezeuge Fördermittel - Fachzeitschrift für technische Logistik* Vol.50, No.10, pp. 482–483, ISSN 0017-9442

Schlenoff, C., Gruninger, M., Tissot, F., Valois, J. Lubell, J. & Lee, J. (1999). The Process Specification Language (PSL) Overview and Version 1.0 Specification, In: *www.mel.nist.gov/psl/,* 12.06.2011, available from: http://www.mel.nist.gov/msidlibrary/doc/nistir6459

Seidlmeier, H. (2002). *Prozessmodellierung mit ARIS® : Eine beispielorientierte Einführung für Studium und Praxis,* Vieweg, ISBN 352-8-058-048, Braunschweig, Germany

Siegert, H. (1996). *Robotik: Programmierung intelligenter Roboter,* Springer, ISBN 3540606653, Berlin, Germany

Staab, S. (2002). Wissensmanagement mit Ontologien und Metadaten. *Informatik-Spektrum,* Vol.25, No.3, pp.194–209, ISSN 0170-6012

Staud, J. (2006). *Geschäftsprozessanalyse : Ereignisgesteuerte Prozessketten und objektorientierte Geschäftsprozessmodellierung für betriebswirtschaftliche Standardsoftware,* Germany, ISBN 978-3-540-24510-0, Berlin, Germany

Steinmann, F. & Nejdl, W. (1999). Modellierung und Ontologie, In: Institut für Rechnergestützte Wissensverarbeitung, 25.05.2011, available from: www.kbs.uni-hannover.de/Arbeiten/Publikationen/1999/M%26O.pdf

Straube, F. & Rösch, F. (2008): *Logistik im produzierenden Gewerbe,* TU Berlin, ISBN 978-3-000-24165-9, Berlin, Germany

Stuckenschmidt, H. (2009). *Ontologien : Konzepte, Technologien und Anwendungen,* Springer, ISBN 978-3-540-79330-4, Berlin, Germany

Studer, R., Benjamins, V., Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering,* Vol.25, , No.1-2, pp.161–197, ISSN 0169-023X

Suppa, M. & Hofschulte, J. (2010). Industrial Robotics. *at – Automatisierungstechnik,* Vol.58, No. 12., pp.663–664,

Swartout, W. , Tate, A. (1999). Ontologies. *IEEE Intelligent Systems and their Applications,* Vol.14, No.1, pp.18–19, ISSN 1094-7167

Tabeling, P. (2006): *Softwaresysteme und ihre Modellierung : Grundlagen, Methoden und Techniken,* Springer, ISBN 978-3-540-25828-5, Berlin, Germany

Westkämper, E., Verl,A. (2009). *Roboter in der Intralogistik : Aktuelle Trends - Neue Technologien - Moderne Anwendungen,* Verein zur Förderung produktionstechnischer Forschung, Stuttgart, Germany

Wiedemann, G. (2008). Ontologien und Ontology Engineering, In: Seminar 'Semantic Web Technologien', 12.06.2011, available from: http://www.informatik.uni-leipzig.de/~loebe/teaching/2008ss-seweb/08v-ontengineering-gwiedemann.pdf