Logist. Res. (2011) 3:29–36
DOI 10.1007/s12159-010-0043-4

ORIGINAL PAPER

Ontology-based communication for the decentralized material flow control of a conveyor facility

Sergey Libert · Michael ten Hompel

Received: 30 August 2010 / Accepted: 10 December 2010 / Published online: 30 December 2010

Abstract Decentralized material flow control is a promising idea to deal with the growing complexity of modern material handling systems. The following paper introduces an ontology-based model for the description of information needed for communication among software agents in distributed material flow control systems. The presented basic ontology was developed by a group of experts from research and industry and holds the most important concepts, which can be easily customized for specific application purposes. The application of the proposed communication ontology is shown in a real-world example.

Keywords Distributed material flow control · Internet of things · Conveyor system · Multiagent systems · Communication ontology

1 Introduction and problem definition

One of the current trends in modern facility logistics is the modularization of mechanical components with the simultaneous distribution of the control function [1, 2]. This idea leads to the breakup of the hierarchical control structure used in the classic system design. The desired material flow control system features a distributed flat structure of standalone control entities. Various research projects accompanied by pilot implementations show the first steps toward the depicted goal [3–5].

S. Libert (✉) · M. ten Hompel
Chair for Materials Handling and Warehousing,
TU Dortmund University, Dortmund, Germany
e-mail: sergey.libert@flw.mb.tu-dortmund.de

M. ten Hompel
e-mail: michael.tenhompel@flw.mb.tu-dortmund.de

A promising step on this way is to use the Internet as an example for decentrally organized and highly flexible systems. Similar to data packages and routers in a computer network, transport objects and conveyors can create user-defined transport networks and organize the material handling process themselves. This vision is a subject of the research program "Internet of Things for facility logistics" funded by the German Federal Ministry of Education and Research (http:\\www.internet-of-things.net).

The term Internet of Things (IOT) originates from the Auto-ID Center of the Massachusetts Institute of Technology (MIT) [6] and found widespread use in the last several years. The Internet of Things for facility logistics is an application of the IOT idea for the domain of the material flow control. This application combines the up-to-date identification methods like radio frequency identification (RFID) with the modern software technologies like multiagent systems [7, 8].

In earlier works, we showed advantages of this concept using analytical methods [9] or simulation models [10]. In cooperation with German logistic enterprises, several test-bed systems based on real-world scenarios are currently under development [8]. However, some common models and standards are still missing for a large-scale implementation of the Internet of Things in industrial practice.

One of the most important points in the organization of a decentralized material flow control is communication in the heterogeneous multiagent environment. Indeed, the control agents, which represent system components and other entities of a logistics system, can originate from different manufacturers and need a common communication basis, accepted by all participants. In the praxis, the most of the existing specifications for the communication with and within material flow controls are proprietaries of the corresponding enterprises. The efforts for the standardization, such as VDMA 15276 and VDI/VDMA 5100, can be considered as meta-standards that does not provide the level of detail necessary for implementation. There are only some de facto standards, such as Modbus and OPC UA, specifying communication in the field of industrial automation. These standards define protocols and semantics for the communication within the process automation. However, semantics of higher application levels is beyond the scope of these standards. On the other hand, some standards for business-to-business communications, such as EDIFACT (Electronic Data Interchange For Administration, Commerce and Transport) and EPCIS (Electronic Product Code Information Service), can be used conditionally for the communication in the agent-based material flow control. In this case, the specifics of the agent interactions as well as the application specifics have to be mapped carefully to the existing standards [11].

In this paper, we introduce an extendable model for describing the domain knowledge in material handling controls in the form of communication ontology.

This model is a compromise of the project participants and includes the minimum semantics needed for the realization of the distributed material flow control scenario. This scenario is briefly described in the beginning of the second chapter of this paper. Following that description, an introduction to the agent communication as well as the definition of the communication ontology is given. The third chapter starts with an overview of our approach and proceeds with the description of the basic ontology for the given scenario. The application of the proposed communication ontology is shown in a real-world example in the fourth chapter.

2 Background

2.1 Application scenario "Internet of Things for facility logistics"

In the application scenario "Internet of Things for facility logistics", the transported goods themselves take over the control and use the transport resources as well as other material handling functions of the material handling plant. The information relevant for transportation is stored on the RFID-tag of the corresponding transportation good. Hence, the dependency on external information systems and thus the integration efforts are reduced.

The material handling plant is supposed to consist of elementary functional units called modules. These modules should have standardized energetic, mechanical, and data interfaces and should be easily interchangeable.

The distributed control system is a multiagent system, wherein the transport unit agents represent the unit loads and the module agents represent single conveyors of different conveyor types. The transport unit agents follow a given workflow and interact with module agents to access the plant resources or to interchange the transport-related information. The service agent is the other type of agents in the system. They are responsible for gathering information from the distributed environment, processing it internally, and providing new information to the other agents. A service agent for example can realize a human–machine interface, a monitoring client or a directory service.

2.2 Communication in multiagent systems

In a multiagent system, agents obtain information about their environment via sensors and execute their tasks by means of actuators. If agents do not possess all the necessary abilities, resources, and information to fulfill their tasks independently, they have to interact with each other. Depending on goals, competences, and existing resources, various ways of agent interaction can be differentiated, from cooperation and coordination up to mutual prevention and competition [2, 12].

From a technical point of view, the interaction between agents is an exchange of information via direct communication. In 1992, the Defense Advanced Research Projects Agency (DARPA) developed a model for direct communication via message exchange within the context of the Knowledge Sharing Effort (KSE). This model consists of three levels: communication, messages, and contents [13]. Building upon this model, the Foundation for Intelligent Physical Agents (FIPA) specifies three levels of agent communication [14]:

- Protocol,
- Communication Language, and
- Ontology.

A protocol describes the communication flow, which consists of individual speech acts. The information representation and semantics in each speech act are irrelevant to the protocols. The structure of a message, however, is an important element of a well-specified protocol [15, 16]. For example, FIPA specifies a wide range of communication protocols, which are sufficient for most tasks in an agent-based material flow control. These protocols utilize the Agent Communication Language (ACL) as a message envelope. Four relevant examples of FIPA interaction protocols are listed in Table 1.

Protocols define the message structure but do not describe the message content. The semantics is determined by the communication ontology and is represented in a content language. Examples for content languages are special agent languages like KIF, SL, or LEAP, but also XML dialects like DAML + OIL [15, 17–19].

Table 1 Relevant FIPA interaction protocols

Shortcut	Protocol name	Purpose/meaning
QUERY	Query interaction	Query for information
REQUEST	Request interaction	Request for agent action
CFP	Contract net interaction	Call for proposal
SUBSCRIBE	Subscribe interaction	Subscription for events

The selection of a suitable content language is up to the software developer. Some relevant decision criteria are transparency (plain text or encoded), the amount of data (overhead) that needs to be transmitted, but also the acceptance among other system developers. If two agents, developed independently from each other, have to communicate with each other, they have to master at least one common language.

The ontology represents the third level of the agent communication. It describes the communication context in order to create unambiguity in the exchange of information. For this reason, a special ontology needs to be determined for each application domain.

2.3 Communication ontology

The term ontology originates in philosophy and describes in the field of theoretical informatics a formalized concept creation. Studer et al. [20] specifies a common definition given by Gruber [21] and defines ontology as "*an explicit formal specification of a shared conceptualization*". The following terminology explains this definition [20]:

- *Conceptualization* is an abstract, simplified model of the world, usually limited to a particular set of concepts, relevant for a particular domain of interest.
- *Shared* reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.
- *Formal* means that the ontology specification must be machine readable.
- *Explicit* indicates that the type of domain concepts and the constraints imposed on their use are defined explicitly.

In the application domain of the distributed material flow control, we use an ontology to restrain the communication context for the participating control agents. In this case, the communication ontology can be seen as an agreement among the agents within a community sharing interest in a common application domain. The ontology represents available knowledge by terms semantically associated to each other. These terms describe the objects and other entities of the application domain and are called *concepts*.

According to the FIPA, Ontology Service Specification [22] agents communicate by making logical assertions, requesting information or action, and posing queries. Therefore, a communication ontology has to define the vocabulary allowing queries and assertions to be exchanged among agents. For this purpose, we use two special concepts that normally envelop actual concepts [22, 23]. These special concepts are *predicates* and *actions*:

- *Predicates* are concepts that declare something about environment conditions and which can be either true or false.
- *Actions* express the requests of agents to perform some activity.

3 Basic ontology for the agent-based material flow control

3.1 The approach and the applied methods

In this section, the basic ontology for the agent-based material flow control [24] is presented and briefly explained. This ontology is the summary of results worked out by a team of experts from research and industry. In order to produce these results, several real-world scenarios primarily from two application domains, baggage-handling systems and large-scale distribution systems, were considered. For the gathering and the arrangement of the concepts, use-case analysis and clustering methods were used.

There are several notation and representation formats for authoring ontologies, such as the Web Ontology Language (OWL) [25]. Since the primary purpose of the current work is the development of the communication ontology for multiagent systems, we resort to a special diagram of the PASSI-Methodology [23]. However, notation languages like OWL can be useful for the later implementation of the developed ontology.

The PASSI-Methodology is an approach in the scope of the Agent-Oriented Software Engineering (AOSE) which enables a systematic development of agent-based applications [26]. One reason why we have chosen PASSI from a number of existing AOSE-methods [2, p. 53 ff.] is the continuing support of the system development process as well as the consideration of the most significant aspects of the agent orientation (such as ontology, roles). Another reason is the utilization of the Unified Modeling Language (UML), which is an accepted standard in the software development, by PASSI.

A presentation of the entire PASSI-Methodology exceeds the scope of this contribution. The only relevant part of the PASSI-Methodology needed here is the ontology development step. In this part, the Domain Ontology Description Diagrams (DODD) utilizes the UML class diagrams to visualize the ontology terms. The semantic relations between the terms are represented by association and inheritance connections. At this point, the inheritance mechanism allows extension and further specialization of the basic terms.

The terms of the basic ontology are divided into four groups: entity ontology, functions ontology, workflow ontology, and transport ontology and are explained in the following.

3.2 Entity ontology

The entity ontology conceptualizes the participants of a distributed material flow control system. The provided concepts *transport unit* (*tu*), *module*, and *service* are visualized in Fig. 1.

3.3 Function ontology

Functions are realized by modules and services. The function ontology defines the vocabulary needed to ask for offered services or module functions and their executions. By means of this, ontology requests like “Which functions are available in the material flow system?”, or “Which service or which module can offer a certain function for a transport unit at what costs?” can be expressed (Fig. 2).

Special functions like *store*, *picking*, or *transport* can inherit from the function concept for certain module types. Other examples for special functions are *weight checking*, *security checking*, *outline checking*, or *packaging*.

Each function is characterized by its costs. Costs describe the effort of a module or service for executing a certain function. They can be factors that characterize the system performance, e.g., the time needed for execution of a certain material handling function.

3.4 Workflow ontology

While the function ontology refers to the execution of individual operational steps, the workflow ontology helps

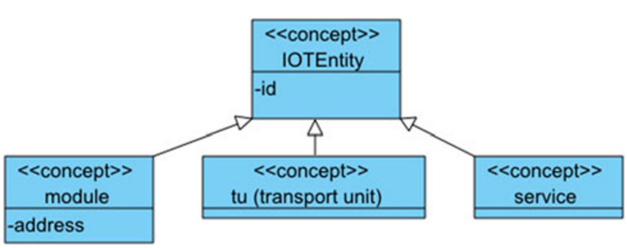

Fig. 1 DODD for the entity ontology

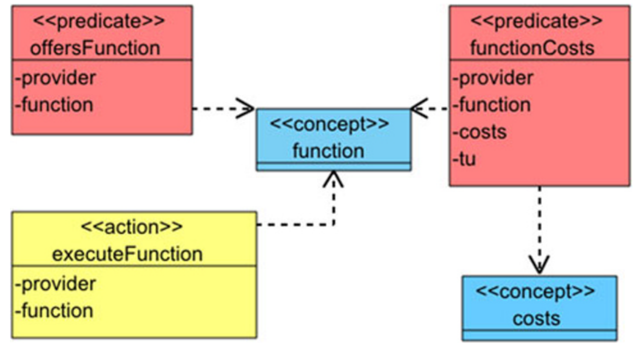

Fig. 2 DODD for the function ontology

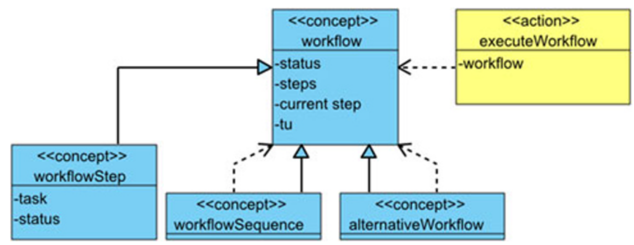

Fig. 3 DODD for the workflow ontology

to specify and assign the operation process as a whole (Fig. 3). A *workflow* can consist of several *workflow steps*. These steps can be processed in a determined or non-determined order. Special workflows can be refined by inheritance from the *workflow* concept.

3.5 Transport ontology

Transportation is the function which is provided by the most modules in conveyor systems and is a prime task of material handling. For this reason, the terms necessary for the transport realization are emphasized in the special transport ontology, shown in Fig. 4.

The concepts *location*, *transferPoint*, and *route* of the transport ontology specify elements of the system topology. They are necessary for solving the routing problem in a distributed environment. For this purpose, the predicates *isSuccessor*, *locationBelongsToModule*, and *routeRealizesOrder* can be used. The predicate *tuLocated* allows exchange information about the current location of a transport unit, and the predicate *tuTransferPermited* enables coordination of agent activities during a load transfer between two modules.

3.6 Extensibility

The proposed basic ontology defines the minimum set of terms needed to model the interagent communication in

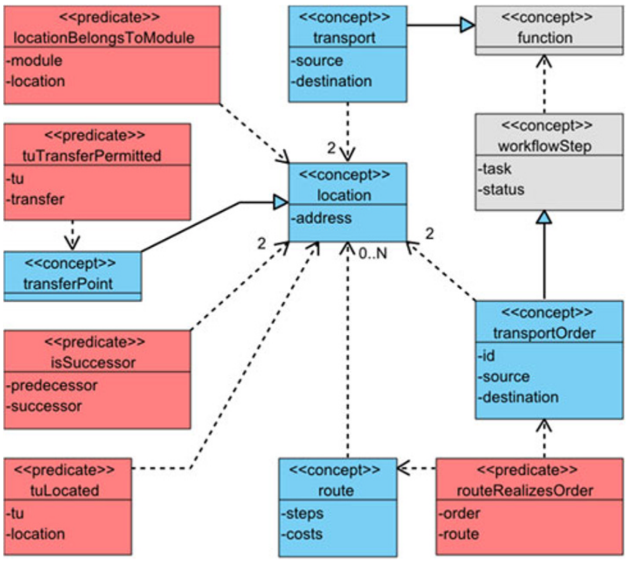

Fig. 4 DODD for the transport ontology

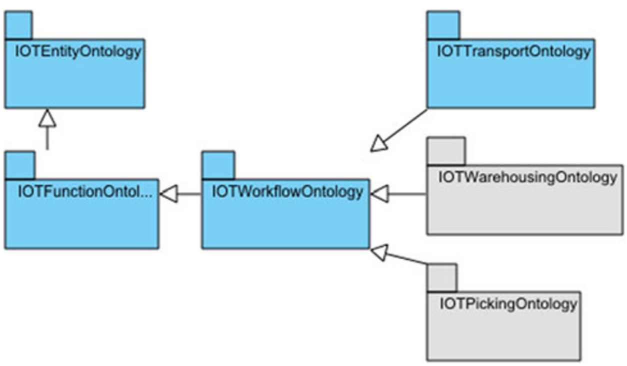

Fig. 5 Hierarchy and extensibility of the basic ontology

distributed material flow controls. The four ontology parts are ordered hierarchically (Fig. 5). The extensibility of this ontology is guaranteed through the inheritance mechanism. Moreover, the already existing terms can be supplemented with additional concepts, predicates, and actions.

4 Real-world example

The proposed ontology is used in a prototypic agent-based control system developed in connection with the research project "Internet of Things". This system is currently applied to control a conveyor system which was built up at the Fraunhofer Institute for Material Flow and Logistics (IML). In the following, this test-bed is briefly described, and implementation details of the agent-based control are given. Finally, we explain the main idea of the ontology-based communication for this real-world example.

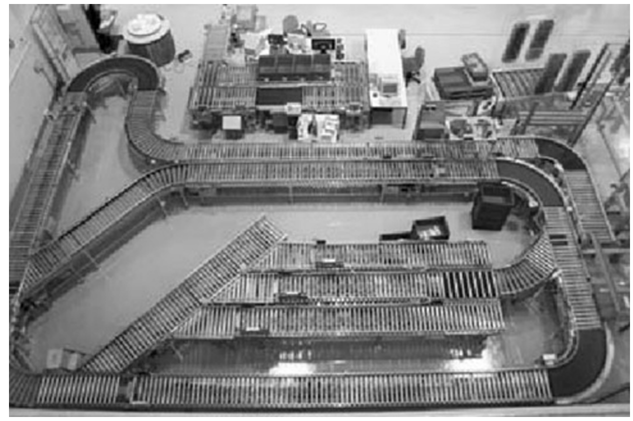

Fig. 6 Bird's eye view of the test-bed plant

4.1 Test-bed

The facility used to test the agent-based material flow control is part of a picking cell installation. The whole system adjoins an automated small-parts warehouse at the system entry and an automated guided vehicle (AGV) system at the system exit (Fig. 6).

The part of the system controlled by agents is a picking loop (in the top area of the picture), which includes a buffered store for empty picking boxes, an automatic scale, a picking station and a transfer to the guided vehicle system.

The field control is a simple program running on several embedded PCs. This program passes the sensor events to the corresponding conveyor agents and translates the transport commands to electrical signals for actuators and drives. For this purpose, a newly developed hardware abstraction layer (HAL) is used.

4.2 Multiagent control system

The picking loop is divided into seven sections, each controlled by its own conveyor agent. These seven conveyor agents represent the single conveyor sections. The automated guided vehicle (AGV) on the system output is represented by an AGV agent. The picking box agents represent the transport boxes. Additionally, the picker agent and the order management agent represent the picker interface and the interface to the ordering system correspondingly.

The operation scenario includes the workflow shown in Fig. 7. Every task of each workflow step refers to a certain function provided by the conveyor modules. For this purpose, the modules provide functions like weight control, buffering, picking, and shipping in addition to the module transport function.

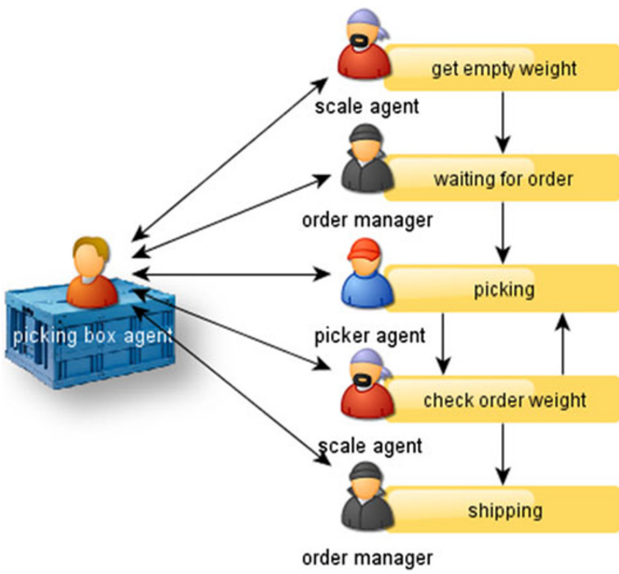

Fig. 7 Workflow in the picking loop

Following the workflow, a picking box agent processes it step-by-step. For each workflow step, the agent communicates to the directory facilitator in order to find out which conveyor module provides the currently needed function. After that, the picking box agent looks for the route to a target module and follows it. After arriving at the module, a conversation between the picking box agent and the function provider (module agent) takes place. For the picking order assignment as well as for the interactions by picking additional information, interchange with the picker agent and with the order management agent is needed (see Fig. 7).

4.3 Ontology-based communication

Developing the multiagent control system, we apply the PASSI-Methodology named above [23]. In this methodology, UML-like diagrams are utilized to support the development steps. Thus, for the visualization of the communication model, the Communication Ontology Description Diagram (CODD) is used. In this diagram, the communication participants (agents) are depicted as classes and the communication as directed associations between these classes. The association classes represent the pieces of communication and are specified by ontology, protocol, and content language.

The simplified Communication Ontology Description Diagram for the picking loop scenario is shown in Fig. 8. We suggest the communication using FIPA interaction protocols and omit the specification of the content language as irrelevant. Based on this communication model, some communication examples are explained below.

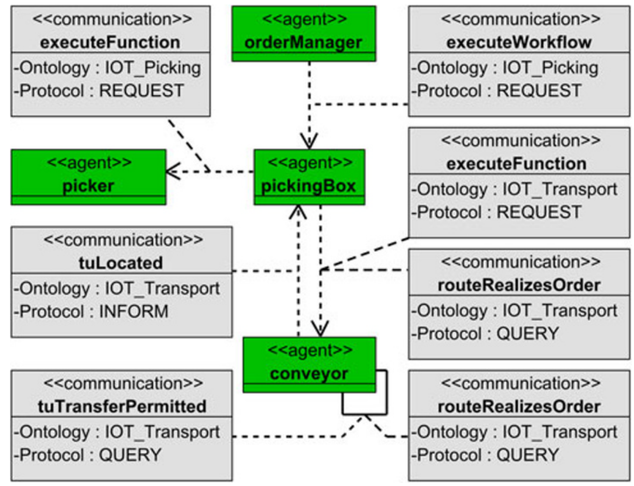

Fig. 8 CODD for the picking loop scenario

- Workflow allocation: Every picking box agent is requested from the order manager to fulfill the workflow (*executeWorkflow*). One step of the workflow specifies the picking order. The picking order structure is defined in a special picking ontology which is not part of the basic ontology.
- Picking box localization: The conveyors detect and identify transported boxes (e.g., using RFID), and the conveyor agent informs the box agent about its position in the plant (*tuLocated*).
- Routing: A simple variant of dynamic source routing is realized using the predecessor–successor relations between conveyor modules. In order to do this, every conveyor agent is able to process the *routeRealizeOrder* query and send it to all its successors until the target module is found.
- Function utilization: The functions transport, weight, and pick provided by conveyor and picking agents can be requested by using the *executeFunction* action. However, requesting the picking function needs the function definition in the picking ontology.
- Transfer: Conveyor agents synchronize the transfer of picking boxes between modules by means of the *tuTransferPermitted* predicate. Based on this communication, different place reservation strategies can be realized.

To show how the communication works, we present an agent communication diagram as an example of a simple route discovery protocol.

In Fig. 9, conveyor agents exchange messages according to the FIPA Query Interaction Protocol (simplified depicted). The two roles defined in this communication are the initiator and the participant. The initiator is the conveyor agent that starts the route discovery. It sends to all its successors the same messages that contain the

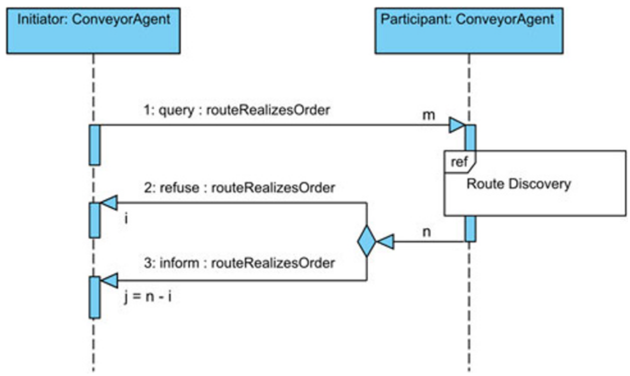

Fig. 9 Example of a recursive route discovery protocol

routeRealizeOrder predicate (filled transport order and empty route). It is the goal of the participants to fill the route message with information and forward it or send it back if the route is accomplished. In order to forward the message, the participant becomes the initiator and the search continues recursively. If a participant conveyor cannot fill the route, a refuse message is sent back to the initiator.

The given example is not to show the advantages or disadvantages of the explained distributed routing algorithms. It aims for the demonstration of how a well-specified communication context makes the communication unambiguous and easy to understand even for complex control tasks like routing.

5 Summary

The formal conceptualization of the communication context is an important condition for sharing information in distributed environments like multiagent systems. The communication ontology proposed in this paper defines a common conceptual basis for communication in agent-based material flow control systems. The participants of the communication are control agents that represent the transport units, conveyor modules, and IT services following the application scenario of the Internet of Things for facility logistics.

The ontology was developed by a group of experts from research and industry and includes the minimum set of concepts needed for requesting the material handling activities as well as for the exchange of control-relevant information. The usage of the ontology is shown in a real-world example in the last part of this paper.

The introduced basic ontology for the Internet of Things defines the essential terms and their relations to each other. It allows for the modeling and implementation of various control tasks in distributed control environments. The terms are kept highly abstract and generic allowing a wide applicability to be reached. However, a specialization of the basic ontology can be necessary for particular applications. For this purpose, the existing terms can be extended by means of the built-in inheritance mechanism. Another extension possibility is to supplement the basic ontology with new terms and relations. In this way, new ontologies for particular applications requirements and special processes can be created.

Acknowledgments The authors are grateful to partners in the project "Internet of Things" for interesting and useful ideas regarding the topic of this paper and to the German Federal Ministry of Education and Research for the financial support of this project.

References

1. Wilke M (2008) Control concept for autonomous changeable material flow systems. Logist J (online). http://www.logistics-journal.de
2. Bussmann S, Jennings NR, Wooldridge M (2004) Multiagent systems for manufacturing control. Springer, New York
3. Hallenborg K, Demazeau Y (2006) Dynamical control in large-scale material handling systems through agent technology. In: Proceedings of IEEE/WIC/ACM international conference on intelligent agent technology. IEEE Computer Society, Washington (DC, USA), pp 637–645
4. ten Hompel M, Sondhof U, Libert S (2005) Distributed control nodes for material flow system controls on the example of unit load conveyor and sorter facilities. In: Grote C, Ester R (eds) Proceedings of embedded world 2005 conference, Nürnberg (Germany), pp 998–1007
5. Bussmann S, Schild K (2001) An agent-based approach to the control of flexible production systems. In: Proceedings of 8th IEEE international conference on emergent technologies and factory automation (ETFA 2001), Nice (France), pp 169–174
6. Sarma S, Brock DL, Ashton K (2000) The networked physical world. Technical Report MIT-AUTOID WH-001, Massachusetts Institute of Technology, Auto-ID Center
7. ten Hompel M, Bullinger H-J (2007) Internet der Dinge: http:\\www.internet-der-dinge.de. In: Bullinger H-J, ten Hompel M (eds) Springer, Berlin (Germany) (in German)
8. Günthner W, ten Hompel M (2009) Internet der Dinge in der Intralogistik. In: Günthner W, ten Hompel M (eds). Springer, Berlin (in German)
9. Nopper JR, ten Hompel M (2009) Analysis of the relationship between control strategy, layout complexity and performance. In: Prpc. Facility Logistics, Orlando (Florida, USA)
10. Follert G, Roidl M (2008) Evaluation of routing strategies for decentralized self-organisation in large-scale conveyor systems. In: Kimberly E (ed) Progress in material handling research. The Material Handling Institute of America, Charlotte
11. Hribernik K, Kramer C, Hans C et al (2010) In: Poppelwell K, Harding J, Poler R et al (eds) A semantic mediator for data integration in autonomous logistics processes. Springer, London
12. Consoli A, Tweedale J, Jain L (2006) An overview of agent coordination and cooperation. In: Knowledge-based intelligent information and engineering systems. Springer, Berlin, pp 497–503
13. Patil RS, Fikes RE, Patel-Schneider PF et al (1992) The DARPA knowledge sharing effort: progress report. In: Rich C, Nebel B, Swartout W (eds) Principles of knowledge representation and reasoning: proceedings of the third international conference, pp 243–254

14. Foundation for Intelligent Physical Agents, FIPA Homepage. www.fipa.org
15. Defense Advanced Research Projects Agency (1993) The DARPA Knowledge Sharing Initiative External Interfaces Working Group, Specification of the KQML Agent-Communication Language. Draft. http://www.cs.umbc.edu/kqml/
16. Foundation for Intelligent Physical Agents (2002) FIPA ACL Message Structure Specification (SC00061). http://www.fipa.org/specs/fipa00061/
17. Bellifemine F, Caire G, Greenwood D (2007) Developing multi-agent systems with JADE. Wiley, Chichester
18. Wooldridge M (2002) An introduction to multiagent systems. Wiley, Chichester
19. Botelho L, Zhang T, Willmott S et al (2009) Review of content languages suitable for agent-agent communication, EPFL I&C Technical Report #200233
20. Studer R, Benjamins VR, Fensel D (1998) Knowledge engineering: principles and methods. Data Knowl Eng 25(1–2): 161–197
21. Gruber TR (1995) Toward principles for the design of ontologies used for knowledge sharing. Int J Human-Comput Stud 43:907–928
22. Foundation for Intelligent Physical Agents (2000) FIPA Ontology Service Specification (XC00086). http://www.fipa.org/specs/fipa00086/
23. Cossentino M, Potts C (2002) A CASE tool supported methodology for the design of multi-agent systems. In: Proceedings of international conference on software engineering research and practice (SERP 02)
24. Libert S, Chisu R, Keutner K. Eine ontologie für das internet der Dinge. In [8] (in German)
25. W3C (2009) OWL 2 Web Ontology Language Document Overview Homepage. http://www.w3.org/TR/owl2-overview/
26. Institute of High Performance Computing and Networking (ICAR), PASSI Homepage. http://www.pa.icar.cnr.it/passi/