Impacts of Data Integration Approaches on the Limitations of Autonomous Cooperating Logistics Processes

Karl A. Hribernik, Christoph Kramer, Carl Hans, Klaus-Dieter Thoben

Karl A. Hribernik

Dipl.-Inform. Karl A. Hribernik studied Computer Science at the University of Bremen. He joined the Bremen Institute of Industrial Technology and Applied Work Science (BIBA) as a research scientist in 2002. His research focus is on the integration of heterogeneous data sources into item-level Product Lifecycle Management systems. He is currently involved in the subproject "Integration of Logistics Data to support Autonomous Cooperating Logistics Processes" in the CRC637 "Autonomous Cooperating Logistics Processes - A Paradigm Shift and Its Limitations".

Christoph Kramer

Dipl.-Inf. Christoph Kramer studied Computer Science at the University of Bremen. He joined the Bremen Institute of Industrial Technology and Applied Work Science (BIBA) as a research scientist in 2009. Focus of the work of Mr. Kramer is the development of integration approaches for heterogeneous data sources. Mr. Kramer's research area covers the definition of semantic and syntactic descriptions of heterogeneous data sources as well as implementations of transformation mechanisms in an ontology-based mediator system.

Carl Hans

Dr.-Ing. Carl Hans is the head of the department "Intelligent ICT for Co-operative Production" at BIBA. As a research scientist, he focuses on the development and integration of ICT solutions in industrial and research projects in areas including Computer Simulation, Artificial Intelligence and Knowledge Management.

Prof. Dr.-Ing. Klaus-Dieter Thoben

After finishing his studies in mechanical engineering, Klaus-Dieter Thoben worked as a research assistant at the Department of Production Engineering at the University of Bremen, where he received his Doctor of Engineering degree in 1989. He received the state doctorate (Habilitation) and the related "venia legendi" for the domain of Production Systems in 2002. In the same year he was appointed to the professorship "IT Applications in Production Engineering" at the University of Bremen. Since 2003 he is director of BIBA (Bremen Institute for Production and Logistics GmbH) Department: ICT Applications in Production. In 2008 he

was appointed to the professorship “Integrated Product Development” at the University of Bremen.

Abstract

Autonomous cooperating logistics processes are characterized by the ability of intelligent logistics objects to process information, and to render and to execute decisions on their own. In order to do so, these objects need to be able to access data relevant to the decisions to be made. Due to the highly dynamic and heterogeneous nature of the relevant data sources in systems supporting autonomous cooperating logistics processes, the selection and implementation of an adequate data integration approach proves to be a challenging task. Furthermore, the selection, implementation and configuration of different data integration approaches impacts the characteristics of autonomous cooperation in different ways. Aspects such as the timeliness, reactivity, scalability, robustness and adaptability of the data integration mechanisms employed factor into limiting the degree of possible autonomous cooperation. Consequently, this contribution endeavours to systematically analyse the limitations thus imposed by different data integration approaches, and to establish a scheme of categorization for identifying adequate data integration approaches for different degrees of autonomous cooperation.

Introduction

Autonomous cooperating logistics processes are characterized by the ability of intelligent logistics objects to process information, to render and to execute decisions on their own. In order to do so, these objects need to be able to access data relevant to the decisions to be made. For example, an intelligent container, which “wants” to be transported from A to B, needs to interact with many different entities, such as different sensor networks, freight forwarders or cold storages. This means that, in order for an IT infrastructure to truly support autonomous cooperating logistics processes on an operational level, its intelligent logistics objects not only need to be able to communicate with each other, but also be suitably integrated into the overall logistics IT landscape. The “traditional” IT landscape in logistics is already a highly complex, distributed and heterogeneous one even without taking autonomous cooperating processes into account. Significant effort was and still is spent in order to achieve at least integration between systems of certain business partners by bridging the technological islands through specific ICT solutions (Hannus 1996). However, most of these solutions sooner or later become obsolete due to both the continuous development of individual standards and systems and the highly dynamic partnerships found in today’s enterprise networks. Instead of developing solutions for 1:1 relationships, it would be preferable to develop a general solution which allows a unique access to all relevant logistics data while accepting the diversity of existing systems and standards (Hans et al. 2008).

This situation is exacerbated by developments in modern logistics such as autonomous cooperating logistics processes and the Internet of Things. Both developments lead to the creation of "new islands" of technology development in the IT logistics landscape. Depending on the application, relevant data may be stored in heterogeneous enterprise systems, such as Warehouse Management Systems (WMS), Enterprise Resource Planning Systems (ERP) or disposition systems. At the same time, data from item-level tracking and tracing systems needs to be taken into account, in particular that pertaining to RFID. Data may also be generated and stored in systems embedded into logistics objects such as trucks or containers, or be generated dynamically, for example by sensor networks monitoring the temperature of a refrigerated container.

Whilst the specific requirements towards data integration differs according to the characteristics of each individual application of autonomous control, it can be said that, in general, digital counterparts representing individual logistics entities need to be able to access data relevant to their decision making processes, regardless which "island" that data may be located on. The heterogeneity of the data sources, their highly distributed nature, along with their availability makes the selection of an adequate data integration mechanism a highly challenging task.

Furthermore, research in autonomous cooperating logistics processes shows that different control problems arise from different applications of autonomous control, resulting in a wide spectrum of degrees of autonomy. (Windt et al. 2008) The resulting requirements towards the characteristics of intelligent logistics objects involved in these processes as well as the underlying data processing, decision making and consequently data integration strategies vary in accordance with the degree of autonomy. When considering an adequate approach to data integration for the support of autonomous cooperating logistics processes, this needs to be taken into account along with the characteristics of the underlying IT landscape outlined above.

Numerous approaches to data integration exist which may be taken into consideration. Each approach exhibits a number of strengths and weaknesses which characterise its applicability to the problem of data integration for the support of autonomous cooperating logistics processes. It follows that these characteristics impact on the autonomous cooperating logistics processes themselves. For example, a data integration approach which is weak in providing real-time access to heterogeneous data sources might negatively impact the reactivity of the overall autonomous cooperating logistics system. That means the selection, configuration and implementation of a data integration approach has a direct effect upon the degree and limits of the autonomous cooperating logistics system it is chosen to support. Aspects such as the timeliness, reactivity, scalability, robustness and adaptability of the data integration mechanisms employed factor into limiting the degree of possible autonomous cooperation.

This contribution endeavours to systematically analyse the limitations thus imposed by different data integration approaches, and to establish a scheme of categorization for identifying adequate data integration approaches for different degrees of autonomous cooperation. It is structured as follows: first, the theoretical background of the problem area is discussed. This encompasses autonomous cooperating logistics processes, the types of data source involved in such processes, and an overview of data integration approaches. The next section deals with the analysis and categorisation of different data integration approaches. An approach to judging the effect the different categories of data integration approach will have on different dimensions of autonomous cooperating logistics processes is derived. The next section applies the approach to the different categories of data integration in order to present a comparison of the different approaches to data integration and their effect on autonomous cooperating logistics processes. A summary and outlook concludes this contribution.

Theoretical Background

The following sections present an overview of the theoretical background relevant to this contribution. First, autonomous cooperating logistics processes are introduced. Then, an overview of the IT landscape in those processes illustrates the integration targets required to be handled by potential data integration mechanisms.

Autonomous Cooperating Logistics Processes

In the context of this contribution, the term "Autonomous Control" is used following Böse and Windt (2007) to describe

> "...processes of decentralised decision-making in heterarchical structures. It presumes interacting elements in non-deterministic systems, which possess the capability and possibility to render decisions independently."

The research area of autonomous cooperating logistics processes (Freitag et al. 2004) aims to meet today's logistics challenges such as the goods structure, logistics and structural effects identified by Aberle (2003), by introducing autonomy and self-organisation into control, information processing and decision-making in logistics (Ehnert et al. 2006). The argumentation is that central control and planning of logistics processes has reached its limits in addressing these issues (Scholz-Reiter et al. 2004). Here, the term "autonomy" describes

> "...the capability of a system, process or an item to design its input-, throughput- and output-profiles as an anticipative or reactive answer to changing constraints of environmental parameters."

The application of autonomous control to logistics processes is expected to increase their robustness, flexibility, adaptability and reactivity to respond to changing business environments, requirements and to changing or partially conflicting objectives (Freitag et al. 2004). A prominent characteristic of this understanding is the decentralisation of decision-making responsibilities in contrast to traditional, hierarchical process control. A dynamic heterarchy in which otherwise passive logistics entities are equipped with the ability to process information, to render and execute decisions on their own replaces the strict centralised top-down management of traditional logistics processes. Artificial agents are entrusted to act in their own “best interest" within the bounds of their operational, tactical or strategic (Timm 2006) autonomies. The motivation for this approach is, amongst others, an expected improved robustness and increased scalability of process control.
The concept of an intelligent logistics object is inherent in the understanding of autonomous control in logistics systems proposed by Böse and Windt (2007). Here,

> “...autonomous control in logistics systems is characterized by the ability of logistics objects to process information, to render and to execute decisions on their own.”

Logistics objects are defined in this context as both,

> “...material items (e.g. parts, machines or conveyors) and immaterial items (e.g. production orders) of a networked logistics system, which have the ability to interact with other logistics objects of the considered system.”

In Scholz-Reiter et al. (2007), the former are further differentiated as commodities and all types of resources whilst constraining the immaterial logistics objects to orders.

According to this understanding, an intelligent logistics object is consequently either a material or immaterial logistics object which is capable of communicating and interacting with other logistics objects. It is a broader understanding than that of the Internet of Things which additionally encompasses autonomous objects without physical representations.

The IT Landscape in Autonomous Cooperating Logistics Processes

(Hribernik et al 2010) categorises the major data sources which comprise the potential IT landscapes supporting autonomous cooperating logistics processes (cf. Table 1). Here, four types of data source are differentiated:

1. Logistics IT systems, describing IT systems in logistics such as ERP, WMS, disposition and other “traditional” enterprise systems used in logistics

2. Intelligent material logistics objects – which relate to material intelligent logistics objects, which exhibit characteristics of the PEID (Product Embedded Information Device) classification scheme (The PROMISE Consortium 2008)
3. Digital counterparts – these relate to the decision making components of intelligent logistics objects, whether located in the object or in the network
4. Sensors and actuators – relating to sensors, sensor networks and actuators, which fall outside of the previous categories

Table 1: Major Data Sources Supporting Autonomous Cooperating Logistics Processes

Data Sources	Type(s)	Interface/standard
Logistics IT systems	General	EDIFACT EANCOM
		EANCOM XML
		ebXML
	SAP compliant	SAP RFC (Remote Function Call)
	Other	Bespoke proprietary
Intelligent logistics objects	EPC compliant	EPCIS
	ID@URI compliant	Dialog
	PEIDs	PMI
	OSGi-based	OSGi
	Other	Bespoke proprietary
Digital counterparts	Multi-agent based (e.g. JADE, PlaSMa, Dialog)	ACL (Agent Communication Language)
		Agent proxies
		Dialog agent
		EDIFACT EANCOM
Sensors & actuators	Java-based	OSGi
	OGC compliant	SensorML
	PEIDs	PMI
	Other sensors	Bespoke proprietary formats
	OPC	OPC DA
		OPC XML DA
		OPC AU
	General	GDI
		ORiN API
	Smart Embedded Devices in Manufacturing	SOCRADES
	Other actuators	Bespoke proprietary formats

With regards to logistics IT systems, EDIFACT EANCOM and SAP RFC are the most prominent targets. However, the more than 30% systems with proprietary interfaces cannot be neglected. (Hribernik et al 2010) Consequently, a data integration approach must be able to cope with both semi-structured, standard data exchange formats as well as function interfaces and be flexible enough to cope with arbitrary proprietary interfaces.

To integrate intelligent material logistics objects, the support of RFID middleware standards such as the EPCglobal Framework Architecture, foremost EPCIS, is mandatory. In addition, a means to interfacing emerging standards for the integration of PEIDs and other embedded devices is necessary. The PROMISE Messaging Interface PMI currently offers the most comprehensive and structured approach to this.

The field of digital counterparts is dominated by software agent technology. The PlaSMa platform is dedicated to the support of autonomous cooperating logistics processes and is consequently of highest priority. Other approaches favour service interfaces. The possibility of agent communication via EANCOM strengthens the need for EANCOM support, but is at the present time not widespread.

Sensor and sensor network integration is at the present time largely a case-by-case decision, with most interface using proprietary approaches. However, emerging standards such as PMI or SensorML are increasing in importance and should not be neglected. A data integration approach therefore needs to be highly flexible towards sensor data sources. With regards to actuators, a promising contribution can be found in the Unified Architecture standards put forwards by OPC. A proposed data integration approach should also take into account the standards emanating from ISO 20242 and factory automation initiatives such as SOCDRADES.

Categories for Data Integration Approaches

Various approaches for the integration of heterogeneous data sources exist. They all offer not only consistent access to data but also the ability to resolve existing integration conflicts. Literature suggests different categorisation schemata for the classification of these approaches. One scheme differentiates between the different information system architecture layers: *manual integration, common user interface, integration by application, integration by middleware, uniform data access* and *common data storage* (Ziegler and Dittrich 2004). In the context of autonomous logistics processes, both manual integration and common user interface can be disregarded – the data to be integrated is consumed not only by human users but primarily by other system components, including enterprise systems and distributed decision making components, e.g. agents in intelligent logistics objects. The remaining four categories describe different levels of ways of coupling information systems along a continuum moving from placing full responsibility for data integration with the querying application (integration by application) through to accomplishing logical integration at the data access level (uniform data access).

The final category describes transferring all data to be integrated into a new data storage system (common data storage). Another widely accepted categorization scheme also focuses on the type of coupling between integrated data sources (Wache 2002). Here, *tightly coupled*, *loosely-coupled* and *object-oriented* approaches are distinguished. Tightly-coupled approaches correspond roughly to above category uniform data and include common data storage access, whilst loosely-coupled approaches find their place at the opposite end of the continuum. The category "object-oriented" introduces a further level of detail into the classification of data integration approaches. It is important to understand that these are not absolutes - the properties of a data integration mechanism might signify it belongs to more than one category. In the following, the three categories are described in more detail.

Tightly-coupled approaches

An integration mechanism is considered tightly-coupled if it is based on one or more federated schemata which solve the integration conflicts. The federated schemata constitute a consolidation of the local schemata of the integrated database systems. An integration approach which uses federated schemata is called a "federated database management system".
Applications that want to access data from the individual subsystems of a federated database management system interact only with the federated schema. The various schemata of the subsystems remain hidden from the application. Direct access is not possible. If a request is made to the federated database management system, it is responsible for breaking down the request into component queries that correspond to the respective local schemata of the component systems. Subsequently, the data supplied by the subsystems is assembled. In developing a federated schema, care needs to be taken to ensure it results in a superset of the local schemata. If this is not the case, data is lost because it can't be retrieved by the applications.
Creating a federated schema is a challenging task in the development of closely coupled systems (Tatarinov & Halevy 2004). Semantically equivalent data has to be identified and the resulting integration conflicts have to be resolved. The effort for resolving the various conflicts of integration varies considerably. This allows resolving syntactical problems, such as data type conflicts, relatively quickly and easy. Other conflicts however, are difficult and expensive solvable.

Loosely coupled approaches

Loosely coupled integration mechanisms use query formalisms to enable applications to define the mappings to data sources themselves. Such a query formalism needs to support a multi-database query language. The query language must, in addition to its ability to query different data sources simultaneously, allow the definition of integration rules which are able to solve the different integration problems. This means that a loosely coupled integration mechanism does not constitute a ready-made solution for all integration problems, but only provides the means to solve such problems in the form of a complex query language. This means the integration mechanism isn't responsible for solving the integration conflicts – this responsibility is passed on to the querying application.

Object-oriented approaches

Object-oriented integration mechanisms are very similar to tightly ones. Both exhibit global schemata for the elimination of the integration conflicts. In object-oriented approaches, data is encapsulated in objects. Semantically equivalent objects from different data sources are combined into the federated schemata from a super type. Unlike traditional object orientation, functions and methods of each object are inherited backwards. Thus, the super type inherits all the functionality of collected objects. Functions are created within the super type which access the different fields and functions of the collected objects and return them. The integration conflicts are resolved in these functions. An overview about different object-oriented approaches is given by (Pitoura, Bukhres & Elmagarmid 1995).

Classification of Data Integration Mechanisms

The following sections present a number of major data integration mechanisms, which may be classified according to the scheme outlined in the previous section. Some mechanisms cannot clearly be classified. In order to reflect this, Figure 1 presents a sketch showing how roughly the major data integration mechanisms relate to the three categories at the corners. Subsequently, a classification of the mechanisms is shown according to the categories outlined in the previous section. The inclusion of Message Oriented Middleware, Service-Oriented-Architecture and Enterprise Service Bus is motivated by the classification according to (Ziegler and Dittrich 2004) and represent widely adopted approaches for hte facilitation of integration by application.

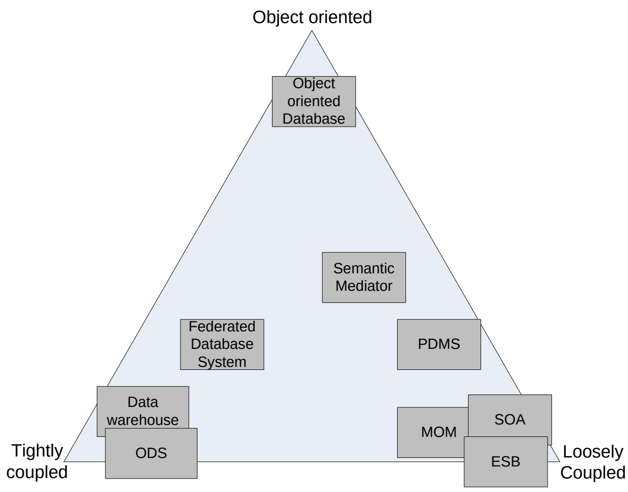

Figure 1: Classification of Data Integration Mechanisms

Data warehouses

A data warehouse is a centralized data pool which mirrors data provided by heterogeneous and often distributed data sources in a local database with a previously defined schema (Widom 1995). Data warehouses are often used for the Online Analytical Processing (OLAP) or as basis for subsequent data mining (Inmon 1996).
A data warehouse usually has three main functions: data extraction and updating, data integration and data storage. The data warehouse extracts the required information from the various heterogeneous data sources. This extraction is executed periodically in a preset interval in order to keep the data up to date. Accordingly, the extracted information is transformed and integrated into the local schema provided by the data warehouse. This local schema and the respective database tables have been initially developed at the creation of the data warehouse. For integration and transformation, a mapping between the local schema and the schema of the respective data source is needed. By means of this mapping the different integration problems are solved. Following to the integration, the data is stored in a local database. Afterwards, the integrated information is accessible by users through this local database. With the use of a local schema, data warehouses falls into the category of tightly-coupled integration approaches.

Operational Data Stores (ODS)

This kind of integration approach is very similar to the data warehouse approach (Inmon 2000). Just as a data warehouse, the operational data store extracts data from the various data sources to be integrated. The extracted data is transformed and integrated towards the global schema and stored in an own local database. The main difference between Operational data stores and data warehouses is the fact, that usually in operational data stores only current data sets are stored, while in data warehouses historical data is stored also. Whereas a data warehouse would store data from last 50 days for example, an ODS would only store the most current data (Baragoin, Marini, Morgan, Mueller, Perkins & Yim 2001). So, the ODS stores only data useful for the current operational world.

Federated Database Systems (FDBS)

Federated Database-Systems are systems that offer access to autonomous and often heterogeneous data sources (Sheth & Larson 1990). Unlike data warehouses or operational data stores, the FDBS don't mirror all the data from the various data sources in an own database. Instead, in case of a query, the required data sets are queried from the data sources, integrated and preprocessed at runtime.

A FDBS offers a federated global schema which is a combination of the local schemata from the data sources (Heimbinger & McLeod 1985) and which was build at construction time manually.

There are two main approaches for creating a federated schema. In the global-as-view-approach, the relations of the global schema are defined as view over the relations of the local schemata. In contrast, the local-as-view-approach defines relations of the local schemata as view over the relations of the global schema. There are also approaches which are combinations of the local-as-view and global-as-view approaches like the one introduced by (Xu & Embley 2004). These approaches try to combine the various characteristics of the two approaches in order to meet their needs.

Message Oriented Middleware (MOM)

In Message-oriented Middleware environments, communication participants interchange data via messages in an asynchronous way. Thereby the messages aren't exchanged directly between the participants; instead the communication takes place via a middleware. The middleware receives messages from the sender and passes them to the receiver. Thus, an asynchronous communication between sender and receiver is possible. The used middleware is a kind of abstraction layer between the various communication partners. Since the single sender and receiver just interact with the middleware, they don't need to know many details about each other; the message could be transformed by the MOM to fit the requirements of the receiver(Menge, 2007). Thus, the participants in a system are loosely coupled (Curry, 2004). Since the messages send over a MOM are normally used to transfer information and data, the MOM systems represent a form of data integration.

Service-Oriented-Architecture (SOA)

In service oriented architecture, applications offer their functionality as reusable services. These services implement a standardized and implementation independent interface. A service in a SOA represents a complete self-contained business function(Natis, 2003). Existing services may be reused by applications or by other services (Endrei, et al., 2004), therefore interoperable protocols exist.
In order to share knowledge about services and their existence, a SOA normally makes use of a naming service. All available services are registered by their providing application to a central naming service with a service description. Consuming applications can discover the available services with this naming service and query the service description to obtain the knowledge about the functionality and how to access the service. Combining different services to implement a complex business process is called service orchestration and offers one of the main goals of business integration.

Enterprise Service Bus (ESB)

Out of the message-oriented middleware and service-oriented architecture the concept of the enterprise service bus (ESB) was evolved. The concept of ESB tries to solve the disadvantages of MOM and SOA such as increasing point-to-point interfaces through services in a SOA (Degenring, 2005) or the use of proprietary protocols and platform specific interfaces in MOM solutions (Menge, 2007).
An ESB is an open-standards based integration infrastructure for applications or services. It provides a messaged-based communication between distributed applications or services in a secure and trustworthy manner. Therefore, the a ESB System provides transformation of messages to fit the requirements of the services, routing, message acceptance, message processing and message routing as well as sending messages to multiple receivers (Ortiz, 2007).

ESB are usually realised out of three elements: A message broker (similar to MOM) or a Message protocol, adapters or service container and mediation/integration services. Services are connected to the ESB by using adapters and communicate over the message broker. The elements of the ESB which deals with the previous mentioned features of the ESB are implemented and attached to the ESB as services also.

Object-Oriented Databases

In object-oriented databases the data is stored in an object-oriented manner, similar to object-orientation in programming languages. Thus it supports an object-oriented data model instead of a relational data model like traditional relational databases. The object-oriented data model provides concepts like encapsulation, classes, inheritance, overriding and object identification (Bancilhon, 1988).

Multidatabase Systems are an approach for data integration where a single module is located on top of the data sources to be integrated. This component offers a global schema to the user. A restriction of this approach is that it is only applicable to data sources with database management system capabilities. With the introduction of object orientation to multidatabase systems, this restriction has been removed (Dogac, Dengi, & Özsu, 1996). With the concepts of object-orientation it's possible to encapsulate every possible data source with the use of implanted wrapper-components.

Peer-Data-Management Systems

In Peer-Data-Management Systems there is no central intermediate layer. Instead the data sources are encapsulated by peers. Peers interact as autonomous unit which are able to answer queries. Each peer may be connected through mappings to other peers. To answer a query, the peer use the data of the encapsulated data source as well as data from peers with which he is connected through mappings.
According to (Roth & Naumann 2006) and (Tatarinov & Halevy 2004), a peer consist of a peer schema, a set of local schemata, local mappings and peer mappings. The peer schema issues what data the peer provides, it can be denoted as export schema. The local mappings connect the peer schema with the local schemata of the local data source. The peer mappings defines the relationships between the different peers in the peer-data-management system.

Semantic Mediators

The concept of mediator according to (Wiederhold 1992) provides the mediator as an independent intermediate layer between the applications which want to query data and the associated data sources. The applications of this concept put their questions to the mediator instead of being sent directly to the data sources. The mediator will forward those requests to the data sources, evaluates their returns and then generates the result for the question of the application.

To fulfill this role, the mediator is composed of, according to (Calvanese & De Giacomo 2005) the mediator itself and the wrappers. The wrappers provide a uniform access to heterogeneous data sources. They have the knowledge that is required for access the particular data source and thus represent an abstraction of the

individual characteristics of the data sources. Generally, one wrapper is necessary per heterogeneous data source.
If the mediator receives a request for the schema provided by him, it splits the request and forwards the request to the wrappers. Only these parts of the request will be forwarded to the wrappers that can be answered by the appropriate data sources. The different wrappers of a mediator receive the requests of the mediator in a uniform query language, transform this request to the appropriate query language of the data source and send that request to the data source. The result of this request is then converted into a common format and returned to the mediator by the wrapper (Gupta, Ludäscher & Martone 2003).
Then the mediator is responsible for integrating the different results of each wrapper and passing them to the requesting application.

In semantic mediators, both syntactic and semantic descriptions of the data to be integrated are applied. The semantic mediator is capable of extracting knowledge regarding the data structures of the underlying data sources and subsequently transforming, decomposing and recomposing data requests according to that knowledge. Given a user query, the mediator first decides which data sources are responsible for the query, based on the semantic descriptions. Then, the queries for the responsible data sources are built with use of the syntactic descriptions. The results from the data sources are afterwards transformed and integrated using both, syntactic and semantic, descriptions. For the semantic description ontologies are a commonly used mean as mentioned by (Wiederhold 1994).

Identification of Limitations and Potentials

A catalogue of criteria for gauging the degree of autonomous control in logistics systems is presented in (Windt, Böse and Philipp 2008). Here, criteria affecting the degree of autonomous control are categorised by system layer (decision, information and execution) and each described using a number of properties. To give an example, "Location of data processing" is a criterion in the system layer "Information System." Its properties describe a continuum between "central" and "decentralised" data processing. The further towards decentralised data processing this criterion tend in a given system, the higher the degree of autonomous control. At the present time, this catalogue of criteria encompasses three criteria for measuring the effect of the information system layer on the level of autonomous control. These are "location of data storage", "location of data processing", and "interaction ability". Their level of detail is not able to capture well the impact of data integration of autonomously controlled systems. Therefore, this contribution intends to suggest additional criteria by which that impact may be captured. In order to do so, the following sections first discuss criteria of data integration mechanisms in relation to their impact on autonomous cooperating logistics systems. These criteria and their properties may then be used to extend the existing catalogue of criteria. Taken separately, they can be used to investigate the value of different data integration mechanisms for use in autonomous cooperating logistics processes.

Criteria of Data Integration Mechanisms for Autonomous Cooperating Logistics Processes

Table 2 shows properties of mechanisms which provide data integration in the context of autonomous cooperating logistics processes. The properties are divided into two categories. The first deals with properties related to the *reliability* of a data integration mechanism, the second with its *flexibility*. The following sections describe in more detail both the categories and their respective properties.

Table 2: Properties of Data Integration Mechanisms for Autonomous Cooperating Logistics Processes

Category	Criterion	Description
Reliability	Data Timeliness	Ability to guarantee up-to-date data
	Reactivity	Ability to guarantee a timely response to a query
	Robustness	Ability to function reliably under any circumstance
	Quality of Data	Ability to guarantee determinable quality of data
Flexibility	Data Volume Scalability	Ability to manage increases in data volume
	Data Source Scalability	Ability to manage increases in the amount of data sources
	Data Source Agnosticism	Ability to integrate different types of data source
	Adaptability	Ability to react to changes in data sources

Criteria of Reliability

The first category is that of reliability. This refers to the capability of a data integration mechanism to perform its function correctly in a specified period of time under stated operation conditions.

Timeliness

This criterion describes how up-to-date the data retrieved using the data integration mechanism is. A service-oriented architecture, for example, the data retrieved from a system via its service interface can be guaranteed to be valid at the time the query was accepted by the service. However, the timeliness of data retrieved from a data warehouse is dependent on the scheduling of data extraction – depending on the configuration of the system; it might be minutes or days old. In both cases, in the period of time between the query and the receipt of the data, it may have become outdated with new data. The former can be said to be timelier due to the more direct access to the data. This criterion directly impacts the behaviour of autonomous cooperating logistics processes in that a decision system may operate more dynamically and reliably the more the timeliness of the data it requests can be guaranteed.

Reactivity

This criterion refers to the ability of a data integration mechanism to respond to a query within a determinable amount of time. This is dependent on the one hand on the connectivity interface used by data integration mechanism and on the other on the architecture underlying it. For example, querying a data warehouse using ODBC can be expected to generate a timely response within a determinable and short amount of time, whilst querying a Peer Data Management System might timeout without returning a complete result set from the respective peers at all.

Robustness

Robustness refers to the ability to perform *reliably* under any circumstance. Like reliability, it is also related to scalability and timeliness as it implies the guarantee of a determinable quality of service.

Data Quality

Data quality (DQ) is understood as the level of fitness for the use of the data by the data consumer in an information system. (Strong, Lee, & Wang, 1997) sub classify data quality in four Categories: intrinsic, accessibility, contextual and representational data quality. Each of these categories has several dimensions as shown in Table 3. In autonomous logistic processes it’s necessary to perform on high quality data, to avoid defective decision making based on measurement errors in sensors for example. Hence, the quality of the provided data is an important measuring point for the various integration approaches.

Table 3: Data quality categories and dimensions (Strong, Lee, & Wang, 1997)

DQ Category	**DQ Dimensions**
Intrinsic DQ	Accuracy, Objectivity, Believability, Reputation
Accessibility DQ	Accessibility, Access security

Contextual DQ	Relevancy, Value-Added, Timeliness, Completeness, Amount of data
Representational DQ	Interpretability, Ease of understanding, Concise representation, Consistent representation

Criteria of Flexibility

Heterogeneity and dynamism are cornerstone characteristics of autonomous cooperating logistics processes. Consequently, the second category of criteria refers to the ability of a data integration mechanism to, on the one hand react flexibly to changes in the surrounding system environment, and on the other be applicable to disparate types of data source.

Data Source Scalability

Data Source Scalability refers to the ability of a data integration mechanism to facilitate the growth of the surrounding system. Scalability in this sense refers to the data administration difficulties in creating and maintaining large systems (Rosenthal and Seligman 2001). The central issue addressed revolves around the administrative effort required to add new data sources to an integrated system. This criterion directly impacts on the scalability of autonomous cooperating logistics systems. For example, if a restriction is put on the scalability of the underlying data integration mechanisms, this limit may also apply to the introduction of new intelligent logistics objects.

Data Volume Scalability

In contrast to data source scalability, this dimension of scalability describes the ability of a mechanism to handle increases in data volume with regards to its runtime performance.

Data Source Agnosticism

Data source agnosticism describes the capability of a given data integration mechanism to be applied to different types of data source and interface (cf. Table 1), if necessary simultaneously. Due to the expected heterogeneity, hierarchy and degree of distribution of IT systems and data sources in logistics systems exhibiting a high degree of autonomy, this capability impact directly on a mechanism's ability support such systems. For example, a specialised mechanism which is capable of adequately integrating enterprise systems may be inadequate for the integration of sensor data. Such a mechanism alone would not do the requirements toward heterogeneity justice.

Adaptability

Adaptability describes several dimensions of a given data integration mechanism in its relation to autonomous cooperating logistics processes. Foremost, it refers to the ability of the mechanism to adapt changes in the data sources in the IT logistics landscape. This can mean adding or removing a data source to or from the pool of integration targets. For example, an intelligent logistics object such as a parcel is introduced into the IT landscape of a logistics provider it has hitherto not been involved with. It may need, for example, to be able to access that provider's IT systems to identify a suitable means of transportation. Using a closely-coupled approach to data integration, adding such a new data source would mean the modification of the federated data scheme. This would result in considerable effort. Using a loosely-coupled approach, the integration could take place immediately, but the responsibility for interpreting the data and solving heterogeneity conflicts would be placed with the intelligent logistics object.

Results – Limitations on Autonomous Cooperating Logistics Processes

This section presents an evaluation of the data integration mechanisms discussed previously against the criteria defined in the previous section. An overview of the evaluation is presented in Table 4, which is elaborated in the following.

Although approaches to realising near real-time data warehouses exist, the decision-making process in traditional data warehouse environments is often delayed because data cannot be propagated from the source system to the data warehouse in time. (Bruckner, List and Schiefer 2002) This is especially disadvantageous to autonomous cooperating logistics processes – the degree of autonomous control increases with the dynamism of its decision system. (Windt, Böse and Philipp 2005)

Since operational data stores are very similar to data warehouses, the evaluation of ODS is very similar to the results of the data warehouses. Their data timeliness is slightly superior, because the interval between requesting and storing new data sets from the data sources is typically much smaller. But this also leads to a lack of data volume scalability, because only the most current data sets are available. Older data records, as needed for object tracking for example, aren't available.

Traditional, tightly-coupled FDBS are robust, response and data timely. Due to the strict definition of a federated data scheme over all local schemata, a high level of data quality may be guaranteed. However, data source scalability is not an advantage of these FDBS – each time a new data source of a new type is added, the federated data schema needs to be modified. It also needs to be changed each time local data schemata are altered.

SOA-based integration approaches are highly flexible towards data source scalability, agnosticism towards data sources and changes made to local data sources. Services are implemented at the local systems and their specifications published in repositories. Services interfacing new or modified data sources can thus be quickly implemented and published, ready for immediate integration. A system integrated using SOA is highly robust against the failure of individual services. The drawback of this approach is that the integration effort is placed upon the querying application. Furthermore, the large amount of data overhead generated for each service call and response using, for example SOAP RPC (cf. Gudgin et al. 2003), make SOA less scalable with respect to data volume

Table 4: Evaluation of Data Integration Mechanisms

Category	**Criterion**	**Data Warehouse**	**Operational Data Store**	**Federated Database Management System**	**Service-oriented Architecture**	**Message-oriented Middleware**	**Enterprise Service Bus**	**Peer Database Systems**	**Object-oriented Databases**	**Semantic Mediator**
Reliability	Timeliness	○	◔	●	●	●	●	●	●	●
	Reactivity	●	●	●	◑	◑	◑	◔	◕	◑
	Robustness	●	●	◕	●	◑	●	●	●	◕
	Quality of Data	●	●	●	◔	○	◕	◑	●	●
Flexibility	Data Volume Scalability	◕	◔	◑	◔	◑	◑	◕	◑	◑
	Data Source Scalability	◕	◕	◔	◕	◕	◕	●	◔	●
	Data Source Agnosticism	◑	◑	◑	●	◕	◕	◕	◑	◕
	Adaptability	◔	◔	◔	●	◔	●	◕	◔	●

○ weakly, ◔ slightly, ◑ averagely, ◕ largely and ● strongly fulfilled

The Enterprise Service Bus provides good data timeliness because the data can be queried directly from the data sources instead of querying local images like in data warehouses. With the use of SOA the ESB can be referred to be robust. There are disadvantages in ESB with response timeliness (messages can decay in messages queues) and data volume scalability (high data overhead and transformations in communication between components).

The main advantages of object oriented (multi-)databases are the data timeliness (ad-hoc querying of data) and quality of data (data transformation and cleaning mechanism can be applied). Similar to FDBS, the object oriented databases constitute a global schema which narrows down their adaptability and data source scalability.

Semantic Mediators have advantages in timeliness, quality of data, data source scalability and adaptability. Like many of the other approaches, the semantic mediator queries the data sources at query time, thus the timeliness is very good. The amount of data sources integrated using a semantic mediator can be theoretically infinite. This is because the mediator queries only data sources which may supply data according to the initial user-query, thus there is only little overhead with unaffected data sources. Due to the loose coupling of data sources in a semantic mediator, the adaptability is also very good.

The weak spots of the mediator are the reactivity and data volume scalability. The mediator queries the data sources at query time, whereat the responses may take a while. In addition this implies that most of the integration tasks, like data transformation or duplicate recognition for example, have to be done at query time also. Hence, huge amounts of data sets increase response time.

Summary and Conclusions

This contribution has endeavoured to systematically analyse the limitations imposed by different data integration approaches on autonomy in autonomous cooperating logistics processes, and to establish a scheme of categorization for identifying adequate data integration approaches for different degrees of autonomous cooperation. To achieve this, first possible classification of data integration approaches was presented. Then, the major approaches which show potential for the application to the data integration in autonomous cooperating logistics processes were discussed. Subsequently, a catalogue of criteria for gauging the impact of specific data integration approaches on the degree of autonomy cooperating logistics processes exhibit was proposed. The criteria were grouped according to whether they relate to the reliability or flexibility of the data integration mechanism. The former encompass timeliness, reactivity, robustness and quality of data. The latter consist of data volume scalability, data source scalability, data source agnosticism and adaptability. Finally, the data integration mechanisms discussed previously were evaluated in the light of these criteria.

The results of the evaluation do not disqualify any of the mechanisms from being used in information systems supporting autonomous cooperating logistics processes. They do, however, provide a guide to gauging what effects a data integration mechanism may have on the degree of autonomy in such processes and for what reasons. They also allow for an informed decision on a case-by-case basis as to which data integration mechanism is most suitable for which scenario of autonomous control.

For example, in a scenario with few data sources and little fluctuation in the amount and type of intelligent logistics object involved, but high demands towards reactivity, robustness and quality of data, either a data warehouse or operational data store might be a sensible choice. However, it is also clear that this type of data integration mechanism will further limit the degree of autonomy that scenario will exhibit.

On the other end of the spectrum, an information system for the support of a highly dynamic, heterogeneous and fluctuating IT landscape involving many different intelligent logistics objects will necessitate a different data integration approach. Here, an approach which strongly supports data source scalability, agnosticism and adaptability is preferable. A number of approaches such as SOA-based integration mechanisms and semantic mediation fulfil these requirements. Both also strongly support timeliness and fulfil other criteria well. Consequently, the use of these data integration approaches is potentially less limiting on the degree of autonomy the autonomous cooperating logistics system may exhibit.

Acknowledgements

This research was supported by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 “Autonomous Cooperating Logistics Processes”.

References

Aberle G (2003) Transportwirtschaft: einzelwirtschaftliche und gesamtwirtschaftliche Grundlagen, 4th edn. Oldenbourg, Munich

Bancilhon, F. (1988). Object-Oriented Database Systems. Proceedings of the Seventh Acm Sigact-Sigmod-Sigart Symposium on Principles of Database Systems. Austin: Association for Computing Machinery.

Baragoin, C., Marini, M., Morgan, C., Mueller, O., Perkins, A. & Yim, K. (2001). Building the Operational Data Store on DB2 UDB. San Jose, California: IBM.

Böse, F. & Windt, K. (2007). Catalogue of Criteria for Autonomous Control in Logistics. In M. Hülsmann, & K. Windt, Understanding Autonomous Cooperation & Control in Logistics - The Impact on Management, Information and Communication and Material Flow (pp. 57-72). Berlin: Springer.

Bruckner, R., List, B. and Schiefer, J. (2002) Striving towards Near Real-Time Data Integration for Data Warehouses. In: Goos, G., Hartmanis, J. and van Leuwen, J. (Eds): Lecture Notes in Computer Science, 2002, Volume 2454/2002, Springer. pp. 173-182, DOI: 10.1007/3-540-46145-0_31

Calvanese, D. & De Giacomo, G. (2005). Data integration: A logic-based perspective.

Curry, E. (2004). Message-Oriented Middleware. In Q. H. Mahmoud, Middleware for communications (pp. 1-28). Wiley.

Degenring, A. (2005, 4). Enterprise Service Bus. Java Spektrum , pp. 16-18.

Dogac, A., Dengi, C., & Özsu, M. T. (1996). Building Interoperable Databases on Distributed Object Management Platforms. Communications of the ACM .

Ehnert I, Arndt L, Mueller-Christ G (2006) A sustainable management framework for dilemma and boundaries in autonomous cooperating transport logistics processes. Int J of Environ and Sustainable Dev 5: 355-371. doi: 10.1504/IJESD.2006.011555

Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., et al. (2004). Patterns: Service-Oriented Architecture and Web Services. IBM.

EPCglobal Inc. (2007, September 21). EPC Information Services (EPCIS) Version 1.0.1 Specification. Retrieved 12 23, 2010, from http://www.gs1.org/sites/default/files/docs/epcis/epcis_1_0_1-standard-20070921.pdf

Gudgin M, Hadley M, Mendelsohn N, Moreau J-J, Nielsen H F (2003) SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation, World Wide Web Consortium. http://www.w3.org/TR/soap12-part1/. Accessed 06 September 2010.

Gupta, A., Ludäscher, B. & Martone, M. E. (2003). Registering Scientific Information Sources for Semantic Mediaton. Lecture Notes in Computer Sciences , pp. 182-198.

Hannus M (1996) Islands of automation in construction. In: Žiga Turk (ed) Construction on the information highway. CIB publication, University of Ljubljana

Hans C., Hribernik K. A., Thoben K-D (2008) An Approach for the Integration of Data within Complex Logistics Systems. In: Haasis H D, Kreowski H-J, Scholz-Reiter B (eds) Dynamics in Logistics. First International Conference LDIC 2007 Proceedings. Springer, Heidelberg.

Heimbinger, D. & McLeod, D. (1985, July). A federated architecture for information management. ACM Transactions on Information Systems , pp. 253-278.

Hribernik, K.; Hans, C.; Kramer, C.; Thoben, K.-D.: Data Integration for an Internet of Things Supporting Autonomous Cooperating Logistics Processes. In: Uckelmann, D.; Harris, M.; Michahelles, F. (eds.): Architecting the Internet of Things. Springer, Berlin, 2010

Inmon, W. H. (2000). Information Architecture for the 90's: Legacy Systems, ODS and DW.

Inmon, W. (1996, November). The Data Warehouse and Data Mining. Communications of the ACM , pp. 49-50.

Menge, F. (2007). Enterprise Service Bus. Free And Open Source Software Conference

Natis, Y. V. (2003, April 16). Service-Oriented Architecture Scenario. Stamford: Gartner.

Ortiz, S. J. (2007, April). Getting on Board the Enterprise Service Bus. Computer , pp. 15-17.

Pitoura, E., Bukhres, O. & Elmagarmid, A. (1995, June). Object orientation in multidatabase systems. ACM Computing Surveys (CSUR) .

The PROMISE Consortium (2008) PROMISE Architecture Series Volume 1: Architecture Overview. Finland: Promise Innovation Oy

Timm I J (2006) Strategic Management of Autonomous Software Systems: Overview Article. Technical Report 35. University of Bremen, Centre for Computing and Communication Technologies (TZI)

Rosenthal, A. and Seligman, L. (2001) Scalability Issues in Data Integration. AFCEA Federal Database Colloquium, San Diego, 2001

Roth, A. & Naumann, F. (2006). Benefit and Cost of Query Anwering in PDMS. In G. Moro, S. Bergamaschi, S. Joseph, J.-H. Mori, & A. M. Ouksel, Databases, Information Systems, and Peer-to-Peer Computing, International Workshops (pp. 50-61). Springer.

Scholz-Reiter B, Windt K, Freitag M (2004) Autonomous logistic processes: New demands and first approaches. In: Monostori L (ed) Proc. 37th CIRP International Seminar on Manufacturing Systems. Hungarian Academy of Science, Budapest.

Scholz-Reiter, B., Kolditz, J. & Hildebrandt, T. (2007). Specifying adaptive business processes within the production logistics domain - a new modelling concept and its challenges. In M. Hülsmann, & K. Windt, Understanding Autonomous Cooperation & Control in Logistics - The Impact on Management, Information and Communication and Material Flow (pp. 275-201). Berlin: Springer.

Sheth, A. P. & Larson, J. A. (1990, September). Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys (CSUR) - Special issue on heterogeneous databases .

Soon T J, Shin-ichii I (2007) EPCIS and Its Applications. Synthesis Journal, 109-124. Information Technology Standards Committee. IDA Singaport & SPRING Singapore. http://www.itsc.org.sg/pdf/synthesis07/Five_EPCIS.pdf. Accessed 06 September 2010.

Strong, D. M., Lee, Y. W., & Wang, R. Y. (1997, May). Data Quality In Context. Communications of the ACM , pp. 103-110.

Tatarinov, I. & Halevy, A. (2004). Efficient query reformulation in peer data management systems. SIGMOD '04 Proceedings of the 2004 ACM SIGMOD international conference on Management of data .

United Nations Economic Commission for Europe (UNECE). (n.d.). United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport. Retrieved 12 23, 2010, from http://www.unece.org/trade/untdid/welcome.htm

Wache, H. (2002). Semantische Mediation für heterogene Informationsquellen. Berlin: Akademische Verlagsgesellschaft Aka GmbH.

Widom, J. (1995). Research problems in data warehousing. In Proceedings of the fourth international conference on Information and knowledge management (pp. 25-30). Baltimore, Maryland, United States: ACM.

Wiederhold, G. (1994). Interoperation, Mediation, and Ontologies.

Wiederhold, G. (1992). Mediators in the Architecture of Future Information Systems. Computer , pp. 38-49.

Windt, K., Böse, F. & Phillipp, T. (2005). Criteria and Application of Autonomous Cooperating Logistics Processes. In Proceeding of the 3rd International Conference on Manufacturing Research - Advances in Manufactoring Technology and Management. Cranfield.

Windt K, Philipp T, Böse F (2008) Complexity Cube for the Characterization of Complex Production Systems. Int J of Comp Integr Manuf 21: 195-200. doi: 10.1080/09511920701607725

Windt, K.; Böse, F.; Philipp, T.: Autonomy in Production Logistics - Identification, Characterisation and Application. In: International Journal of Robotics and CIM, 24(2008)4, pp. 572-578

Xu, L. & Embley, D. W. (2004) Combining the Best of Global-as-View and Local-as-View for Data Integration. In Information Systems Technologies and Its Applications - ISTA , pp. 123-136.

Ziegler, P. and Dittrich, K. (2004) Three Decades of Data Intecration — all Problems Solved? In: IFIP International Federation for Information Processing, 156/2004. pp. 3-12, DOI: 10.1007/978-1-4020-8157-6_1