

PROCEEDINGS

# ARTIFICIAL INTELLIGENCE AND LOGISTICS (AILOG-2010)

WORKSHOP AT ECAI 2010

Kerstin Schill, Bernd Scholz-Reiter, Lutz Frommberger (eds.)

August 17, 2010

European Conference on Artificial Intelligence (ECAI)

Lisbon, Portugal





# ARTIFICIAL INTELLIGENCE AND LOGISTICS (AILOG-2010)

## ORGANIZERS

### **Kerstin Schill**

SFB/TR 8 “Spatial Cognition”, University of Bremen, Germany

[kschill@informatik.uni-bremen.de](mailto:kschill@informatik.uni-bremen.de)

### **Bernd Scholz-Reiter**

SFB 637 “Autonomous Cooperating Logistic Processes”, University of Bremen, Germany

[bsr@biba.uni-bremen.de](mailto:bsr@biba.uni-bremen.de)

### **Lutz Frommberger**

SFB/TR 8 “Spatial Cognition”, University of Bremen, Germany

[lutz@informatik.uni-bremen.de](mailto:lutz@informatik.uni-bremen.de)

## INVITED SPEAKER

Anthony G. Cohn (University of Leeds, UK)

## PROGRAM COMMITTEE

Ana Bazzan (Universidade Federal do Rio Grande do Sul, Brazil)

John Bateman (University of Bremen, Germany)

Susanne Biundo (University of Ulm, Germany)

Jürgen Branke (University of Warwick, UK)

Stefan Kirn (University of Hohenheim, Germany)

Martin Lauer (Karlsruhe Institute of Technology, Germany)

Ramon López de Mántaras (IIIA-CSIC, Spain)

Carles Sierra (IIIA-CSIC, Spain)

Gerhard Weiss (Maastricht University, Netherlands)

Stefan Wölfl (University of Freiburg, Germany)

## ORGANIZING INSTITUTIONS

SFB/TR 8 “Spatial Cognition”

University of Bremen and University of Freiburg, Germany

<http://www.sfbtr8.spatial-cognition.de/>

SFB 637 “Autonomous Cooperating Logistic Processes”

University of Bremen, Germany

<http://www.sfb637.uni-bremen.de/>



## TABLE OF CONTENTS

|                                                                                                                        |    |
|------------------------------------------------------------------------------------------------------------------------|----|
| Preface: Artificial Intelligence and Logistics . . . . .                                                               | 1  |
| <i>Kerstin Schill, Bernd Scholz-Reiter, Lutz Frommberger</i>                                                           |    |
| Invited Talk: Acquiring spatio-temporal behaviour models (Abstract) . . . . .                                          | 3  |
| <i>Anthony G. Cohn</i>                                                                                                 |    |
| TECHNICAL PAPERS                                                                                                       |    |
| Qualitative spatial reasoning with topological information in BDI agents . . . . .                                     | 7  |
| <i>Michael Schuele, Paul Karaenke</i>                                                                                  |    |
| Managing Contingencies in Timed Transportation Networks by Agent<br>Technology . . . . .                               | 13 |
| <i>Natalja Pulter, Jens Nimis, Peter C. Lockemann</i>                                                                  |    |
| Oil Industry Supply Chain Management as a Holonic Agent Based Distributed<br>Constraint Optimization Problem . . . . . | 19 |
| <i>Fernando J. M. Marcellino, Jaime S. Sichman</i>                                                                     |    |
| Ensuring Privacy through Distributed Computation in Multiple-Depot Vehicle<br>Routing Problems . . . . .               | 25 |
| <i>Thomas Léauté, Brammert Ottens, Boi Faltings</i>                                                                    |    |
| Residual resources exploitation in a cooperative MAS . . . . .                                                         | 31 |
| <i>Gaël Hette, Sylvia Estivie, Emmanuel Adam, René Mandiau</i>                                                         |    |
| Bringing Agents into Application: Intelligent Products in Autonomous Logistics . . . . .                               | 37 |
| <i>Farideh Ganji, Ernesto Morales Kluge, Bernd Scholz-Reiter</i>                                                       |    |
| Context-based Learning of Adaptive Vehicle Dispatching Strategies . . . . .                                            | 43 |
| <i>Gerd Heiserich, Ludger Overmeyer</i>                                                                                |    |
| Modeling a Neural Network Based Control for Autonomous Production Systems . . . . .                                    | 49 |
| <i>Daniel Rippel, Florian Harjes, Bernd Scholz-Reiter</i>                                                              |    |
| Towards intelligent material handling planning systems – Status quo and steps<br>to be taken . . . . .                 | 55 |
| <i>Tobias Sommer, Daniel Neuhäuser, Dirk Marrenbach, Karl-Heinz Wehking</i>                                            |    |
| Reasoning About Geometry: An Approach Using Spatial-Descriptive Ontologies . . . . .                                   | 61 |
| <i>Markus Eich, Frank Kirchner</i>                                                                                     |    |
| Working Simulations with a Foundational Ontology . . . . .                                                             | 67 |
| <i>Robert Porzel, Tobias Warden</i>                                                                                    |    |
| An Approach to Formal and Semantic Representation of Logistics Services . . . . .                                      | 73 |
| <i>Julia Hoxha, Andreas Scheuermann, Stephan Bloehdorn</i>                                                             |    |
| Author Index . . . . .                                                                                                 | 79 |



## PREFACE

Logistics is concerned with the organization and control of processes in space and time in order to transport or manufacture goods and with the coordination of the resulting flows of material and information, as well as monetary flows. Complexity and dynamics of logistic networks have reached a level that more and more prevents the processes from being entirely observable and fully controllable, caused by, e.g., a globalized economy, a shift from buyer's to seller's markets, and shorter product life cycles. Thus, recent trends in logistics point towards solutions with distributed and self-organizing processes.

Such logistic networks are characterized by a wide variety of heterogeneous properties, constraints, and requirements: temporal (very quickly, just in time), economic (low cost, given budget), physical (smooth, secure travel), informational (public, secure data), environmental (low energy, ecology-preserving). Different qualities or modalities need different representations; transitions between different subsystems require interfaces between different representations, transformations of processes, and methods for their selection; knowledge about situations and goals is crucial for the selection of the most suitable strategy; etc. This manifold set of demands and the emerging uncertainty requires the inclusion of methods from artificial intelligence (AI) and cognitive science. AILog-2010 tries to bring the research communities of logistics and AI closer together as we believe both could benefit from a closer collaboration. Therefore the workshop addresses researchers from logistics that utilize and develop AI approaches as well as researchers from AI that apply their work in logistics or see possibilities to connect to logistic problems or methods.

The idea to organize this workshop emerged from increased cooperation between the two DFG funded collaborative research centers "SFB 637 Autonomous Cooperating Logistic Processes" and "SFB/TR 8 Spatial Cognition" at the University of Bremen. The SFB/TR 8 intensively investigates how to describe and manage spatio-temporal processes, be it on a human or machine level. The basic research done there provides a solid foundation for connection with the field of logistics that is also concerned with processes in space and time and their interaction with human and artificial agents. The research focus of the SFB 637 on autonomous logistics has naturally led to the intensive use of AI methods to address emerging uncertainty and increased complexity. Activities in both research centers converged on and manifested in cooperative approaches bringing together both fields for a mutual extension of the research done so far. The AILog-2010 workshop is supposed to evaluate the potential of AI approaches in logistics in an international context.

From the submitted manuscripts we selected 12 papers for presentation at the workshop after a peer-review process. We would like to thank all the authors for their contribution and the members of the program committee and the external reviewers for the substantial feedback they provided on the submitted manuscripts. We also would like to thank Christian Freksa, Bernd Krieg-

Brückner, and Dieter Hutter for advice and support and Torsten Hildebrandt for his significant help in the workshop organization. Furthermore, we thank the organizers of ECAI 2010 and the workshop chair Ulle Endriss for the smooth organization. Financial support by the German Research Foundation (DFG) and the University of Bremen is gratefully acknowledged.

Finally, we hope for interesting input and fruitful discussion at the workshop.

*Kerstin Schill*

*Bernd Scholz-Reiter*

*Lutz Frommberger*

*(AILog-2010 organizers)*

## INVITED TALK

### ACQUIRING SPATIO-TEMPORAL BEHAVIOUR MODELS

ANTHONY G. COHN  
UNIVERSITY OF LEEDS, UK

In this talk I will present ongoing work at Leeds on building models of video activity. I will present techniques, both supervised and unsupervised, for learning the spatio-temporal structure of events from video or other sensor data, in terms of which activity can be understood. In both cases, the representation exploits qualitative spatio-temporal relations. For supervised learning I will show how the supervisory burden can be reduced using what we term "deictic supervision", whilst in the unsupervised case I will present a method for learning the most likely interpretation of the training data. I will also show how objects can be "functionally categorised" according to their spatio-temporal behaviour and how the use of type information can help in the learning process, especially in the presence of noise. I will present results from several domains including a kitchen scenario and an aircraft apron.



# TECHNICAL PAPERS



# Qualitative spatial reasoning with topological information in BDI agents

Michael Schuele<sup>1</sup> and Paul Karaenke<sup>1</sup>

**Abstract.** The paper investigates belief-desire-intention (BDI) agents models in multiagent systems. In the spatial context, the agents' knowledge about their environment is often uncertain. We focus on the problem of autonomous, collision-free motion of agents on base of uncertain knowledge in transport logistics. This problem is addressed from a qualitative spatial reasoning perspective. The contribution of this paper is an agent model that integrates BDI concepts and spatial analysis capabilities. The approach is evaluated by means of multiagent simulation in a scenario from the transport logistic domain.

## 1 INTRODUCTION

The trends and changes in logistics lead to decentralized approaches of control systems for logistic processes that reduce the complexity of central systems [11]. The former standard approach limits decisions of local entities to a minor scope of action. The decentralized approach addresses aspects like heterogeneity, adaptivity, and reactivity to dynamically changing external influences. Thus, local entities which follow global goals of central interest with the capabilities of dynamic decision-making on an operational level are required. A well-known example of this approach is distributed control at automated container terminals. In this paper, we consider autonomous logistic entities in individual, closed, dynamic environments like construction sides, airports, and ports. Figure 1 shows a service chain model example of a construction side. The individual requirements of the customer determine the requirements on the actors which contribute to the provision of services. In order to achieve a high grade of individualization of the rendered services of the logistics entities, the whole service chain and thus the single logistic entities need to be equipped with the feature of adaptivity. Especially in the spatial dimension, the logistic entities have to react adaptively on dynamically changing requirements, like changing transportation target positions and transportation routes, and on the dynamic environment.

Logistics is mainly characterized by interacting logistics processes. Therefore, effective logistics requires a holistic approach to take inter- and intra-processual dependencies into account. Failures of local resources can affect the execution of logistics processes that use these resources. Due to inter-processual dependencies this can also affect further processes. Thus, low-level problems, such as collision-free motion, are relevant for logistic problems from a high-level perspective along the whole service chain.

In order to address the listed aspects, software agents are used as representatives of logistic entities that act on their behalf. The paper

investigates goal-oriented belief-desire-intention (BDI) agents models in multiagent systems. In the spatial context, the agents' knowledge about their environment is often uncertain. Agents cannot base their decisions on static information because of the dynamics of their environment. In this paper, we consider the problem of autonomous, collision-free motion of agents on base of uncertain knowledge in the context of transport logistics. BDI agents are enriched with spatial analysis features. For this purpose, qualitative spatial reasoning based on the RCC-8 calculus provides suitable concepts [19]. We concentrate on the agent model itself. Sensors and robotic capabilities, that would be required for a real-world scenario, are not considered. We integrate standard features of the OpenGIS standard [6] in an agent environment, and map the functions to the RCC-8 base relations. The environment is represented as thematic layers (e.g., obstacles). The presented approach is evaluated by means of multiagent simulation in a scenario from the transport logistic domain.

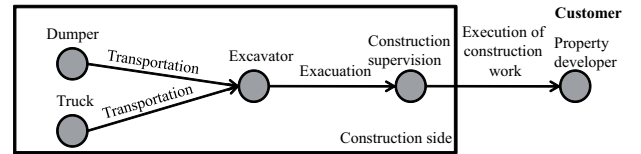


Figure 1. Service chain model example of a construction side

The remainder of this paper is as follows. Section 2 describes the theoretical grounding of this research: deliberative agents and qualitative spatial reasoning. In section 3, we describe the integration that enables BDI agents to use the qualitative spatial reasoning, selected spatial capabilities, and the architecture of our approach. Section 4 describes the evaluation experiments and analyses the evaluation results. Related work is discussed in section 5. Section 6 gives a conclusion and an outlook on future research.

## 2 RESEARCH APPROACH

### 2.1 Deliberative agents

The paradigm of BDI architecture for software agents bases on the concepts of beliefs, representing information about the agent's current environment; desires, representing the agent's goals; and intentions, representing the agent's current focus, that leads to concrete actions. The decision making process, what actions to perform in order to achieve the goals, is based on the philosophical approach of practical reasoning [2]. Practical reasoning consists of two activities,

<sup>1</sup> University of Hohenheim, Information Systems 2, Germany, email: {michael.schuele, paul.karaenke}@uni-hohenheim.de

deliberation and means-end reasoning. Deliberation means the decisions on what goals an agent wants to achieve. Means-end reasoning denotes the process of inferring how these goals can be achieved [26].

## 2.2 Qualitative spatial reasoning

Representing and analyzing spatial information is an essential problem in the context of the agents' environment in multiagent systems [23]. Often, spatial information is only available as qualitative information or as a large amount of quantitative data. These circumstances require an efficient analysis in qualitative form. One approach, the region connection calculus (RCC) for qualitative spatial reasoning was developed by Randell, Cui and Cohn [19]. This calculus is based on binary topological relations. The variant of this calculus used in this paper, the RCC-8, is based on eight mutually exhaustive and pairwise disjoint relations, which describe the topological relations between two spatial regions. The base relations of the RCC-8 are DC (disconnected), EC (externally connected), PO (partial overlap), EQ (equal), TPP (part tangential proper), NTPP (non-tangential proper part),  $TPP^{-1}$  (tangential proper part inverse) and  $NTPP^{-1}$  (non-tangential proper part inverse). Reasoning is deduced from the composition ( $\circ$ ) of two base relations  $R$  and  $S$  of two regions  $x$  and  $y$ , that is formalized as

$$\forall x, y : x(R \circ S)y \Leftrightarrow \exists z : (xRz \wedge zSy) \quad (1)$$

The compositions of the eight base relations are shown as a composition table in [21]. The content of each cell determines the possible relations that results from the reasoning regarding the relation of the respective column and line header. The composition table shows that the reasoning results are not unique in the majority of cases.

## 3 QUALITATIVE SPATIAL REASONING IN BDI AGENTS

### 3.1 Agent model

The objective of this approach is to enable BDI agents to move autonomously and collision-free in a spatial environment. The spatial environment is based on real geographic data with regions that are declared as movement areas and regions that are declared as obstacles. Agents have two reciprocally affecting goals: (1) to move to the agent's target position, and (2) to avoid collisions with obstacles and other agents. If an agent has reached its target position, a new target position can be determined and the agent starts again to reach this position. For the formal notation of this BDI agent pattern we use the AgentSpeak(L) language [20]. The AgentSpeak(L) snippets below are confined to the relevant parts.

The agent model has been developed on the condition that it is accepted by Jason [1], a Java-based interpreter for an extended version of AgentSpeak(L). Agents have the following set of beliefs. The belief *agent\_id* is used as an unique identification attribute of an agent. *speed* determines the speed of an agent. An agent's target position is represented by the coordinates *target\_position\_x* and *target\_position\_y*, the agent's current position by *current\_position\_x* and *current\_position\_y*. The position that is calculated for the next step is represented by *calculated\_position\_x* and *calculated\_position\_y*. The belief *expansion* defines the expansion radius of an agent. Thus, agents are not represented as single points but as polygons. The belief *perception\_factor* determines the factor in relation to the agent's expansion, that is used to define the perception area of an agent. This region is represented by the belief *perception\_area*.

The belief *movement\_area* represents the regions in which an agent is able to move. *obstacle\_area* represents the regions that are declared as obstacles. The belief *rcc8\_movement\_area\_relation* defines the RCC-8 base relation of *perception\_area* and *movement\_area*. *rcc8\_obstacle\_area\_relation* defines the RCC-8 base relation of *perception\_area* and *obstacle\_area*. The belief *angle* determines the angle that defines the direction of motion for an agent relatively to the coordinate system.

```
/* Initial beliefs */
agent_id(ID).
speed(S).
target_position_x(TX).
target_position_y(TY).
current_position_x(CUX).
current_position_y(CUY).
calculated_position_x(CAX).
calculated_position_y(CAY).
expansion(E).
perception_factor(PF).
perception_area(PA).
movement_area(MA).
obstacle_area(OA).
rcc8_movement_area_relation(RMR).
rcc8_obstacle_area_relation(ROR).
angle(A).
```

The initial goal is *start*. Thus when it starts running the event a plan is triggered. This plan executes the function *register\_at\_environment* that registers the agent as a part of the environment. The representation of an agent can be interpreted as a dynamic obstacle. *start* also generates the goal *move*.

```
/* Initial goals */
!start.

/* Plans */

+!start: true <-
    register_at_environment(ID, CUX, CUY,
        CAX, CAY, E, PF);
    !move(PA, A).
```

The activities that are triggered by *move* represent the beginning of the process. The goal *calculate\_new\_position* is generated.

```
+!move(PA, A): <-
    !calculate_new_position(PA, A).
```

*calculate\_new\_position* executes the function with the same name. This function calculates the new position for the next step by using the angle and speed of an agent. The initial angle is determined by the relation between the current position and the target position, so that in the first instance the direct path is favored. The new calculated position and the polygon that represents the agent are located in the current perception area of the agent. After executing the calculation function the *avoid\_collision* goal is generated.

```
+!calculate_new_position(PA, A): true <-
    calculate_new_position(CUX, CUY, TX, TY,
        S, E, A, CAX, CAY);
    !avoid_collision(PA, A).
```

*avoid\_collision* executes three functions that enable the agent to reason in the spatial context. The function *detect\_spatial\_relations\_for\_movement* calculates the RCC-8 base relation between an agent's perception area and the movement area. *detect\_spatial\_relations\_for\_obstacles* calculates the RCC-8 base relation between an agent's perception area and the obstacle area, e.g., obstacles and other agents. The function *reason\_spatial\_composition* provides the underlying spatial reasoning; i.e. the function determines if the calculated position is in a region that is declared as an agent's movement area and if the position is not within an obstacle or intersects with an obstacle. The polygon representing an agent can be assumed, as described above, to be within the perception area of this agent. This can be formally described by the RCC-8 base relation NTPP [21]. By means of the composition table of the RCC-8 it can be reasoned about the possible RCC-8 base relation between the calculated position of the agent, the movement area, and the obstacle area. A collision can be assumed, if the reasoned set of base relations with the movement area contains DC, EC,  $TPP^{-1}$ , PO or  $NTPP^{-1}$ , and if the reasoned set of base relations with the obstacle area contains TPP,  $TPP^{-1}$ , PO, EQ, NTPP or  $NTPP^{-1}$ . In case of an existing collision the perception area is decreased, the motion angle is modified, or both modifications are performed. This is done in order to reach a collision-free event in the next iteration. As final activity *avoid\_collision* generates the goal *move\_to\_position*.

```
+!avoid_collision(PA, A): true <-
  detect_spatial_relations_for_movement(
    ID, PA, RMR, MA);
  detect_spatial_relations_for_obstacle(
    ID, PA, ROR, OA);
  reason_spatial_composition(ID, RMR, ROR,
    PA, MA, OA, CUX, CUY, CAX, CAY, TX, TY,
    S, E, A, CA);
  -+collision(CA);
  !move_to_position(PA, MA, OA, A).
```

The following AgentSpeak(L) sections are triggered regarding a belief change. If a result of the previous plan is *collision(0)*, the agent beliefs *-avoided(collision)*. If it is *collision(1)*, the agent beliefs *+avoided(collision)*.

```
+collision(0) : true <-
  -avoided(collision).
```

```
+collision(1) : true <-
  +avoided(collision).
```

The following plan is triggered if the motion to the calculated position is collision-free. The function *move\_to\_calculated\_position* executes the actual motion; i.e. the current position of the agent is set to the calculated position. The agent's perception area is recalculated according to the new position. The belief *at\_target\_position*, containing the information if the agent has reached its target position, is set. The goals *change\_target\_position* and *move* are generated in order to enable a new iteration of motion.

```
+!move_to_position(PA, MA, OA, A):
  avoided(collision) <-
  move_to_calculated_position(ID, PA, MA,
    OA, CUX, CUY, CAX, CAY, TX, TY, PF, E, A,
    S, PR);
  -+at_target_position(PR);
  !change_target_position;
  !move(PA, A).
```

If there is a potential collision with the calculated position, *calculate\_new\_position* has to be executed again with the new perception area and the new angle.

```
+!move_to_position(PA, MA, OA, A):
  not avoided(collision) <-
  !calculate_new_position(PA, A).
```

If the result of the previous plan is *at\_target\_position(0)*, the agent beliefs *-reached(target\_position)*. If it is *at\_target\_position(1)*, the agent beliefs *+reached(target\_position)*.

```
+at_target_position(0): true <-
  -reached(target_position).
```

```
+at_target_position(1): true <-
  +reached(target_position).
```

The following plan is triggered if the target position is reached. Then, a new target position can be determined.

```
+!change_target_position:
  reached(target_position) <-
  change_target_position(TX, TY).
```

### 3.2 Spatial Capabilities

The environment the agents perceive is formed from the environment and other agents. The environment is equal for all agents in the system. In this paper, the environment is limited to a two-dimensional space. The data structure is based on thematic layers, that are implemented as database tables with geometric objects. Geometric objects are defined analogous to the types of the Simple Feature Specification for SQL of the Open GIS Consortium [6]. In this model, spatial objects are limited to polygons and multi-polygons. All data is based on real-world coordinates. With this vector representation of the entities we obtain a continuous environment model. The thematically separated polygon layers are defined explicitly as obstacle areas or movement areas. For the individual layers, it is not relevant whether the objects are static or dynamic. The decision process of the agents is based on a snapshot of the current state of the perceived environment. The perception of the agents is limited to a defined region of perception in the area of their site.

For the use of the RCC-8, regions are defined as polygons or multi-polygons. Base relations are determined between polygons or multi-polygons. To carry out the analysis, the topological features and spatial analysis capabilities provided by the Simple Feature Specification for SQL are used. A unique definition of the base relation between two polygons or multi-polygons, as shown in Table 1, can be determined by using these functions.

In detail, the relations between two geometric objects *A* and *B* are analyzed. The used topological features are *Contains*, *Touches*, *Intersects*, *Within*, *Equals*, *Disjoint*, and *Overlaps*. *Contains* checks if object *A* spatially contains object *B*. *Touches* is true if the only points in common between object *A* and object *B* lie in the union of the boundaries of *A* and *B*. *Intersects* checks if two geometric objects spatially intersect. *Within* is true if the object *A* is completely inside object *B*. *Equals* verifies if two geometric objects are spatially equal. *Disjoint* checks if two objects have no common point. *Overlaps* is true if two geometric objects share space, but are not completely contained by each other. For an unique determination of the base relation the application of other spatial analysis function is necessary. *GeometryType(Intersection(A, B))* returns the data type of the geometric

**Table 1.** Topological and spatial features for the determination of the base relation of the RCC-8

|                                 | DC                  | EC         | TPP     | TPP <sup>-1</sup> | PO      | EQ      | NTPP    | NTPP <sup>-1</sup> |
|---------------------------------|---------------------|------------|---------|-------------------|---------|---------|---------|--------------------|
| Contains(A,B)                   | false               | false      | false   | true              | false   | true    | false   | true               |
| Touches(A,B)                    | false               | true       | false   | false             | false   | false   | false   | false              |
| Intersects(A,B)                 | false               | true       | true    | true              | true    | true    | true    | true               |
| Within(A,B)                     | false               | false      | true    | false             | false   | true    | true    | false              |
| Equals(A,B)                     | false               | false      | false   | false             | false   | true    | false   | false              |
| Disjoint(A,B)                   | true                | false      | false   | false             | false   | false   | false   | false              |
| Overlaps(A,B)                   | false               | false      | false   | true              | true    | false   | false   | false              |
| GeometryType(Intersection(A,B)) | Geometry-Collection | Point/Line | Polygon | Polygon           | Polygon | Polygon | Polygon | Polygon            |
| Touches(Boundary(A),B)          | false               | true       | false   | true              | false   | true    | false   | false              |
| Touches(Boundary(B),A)          | false               | true       | true    | false             | false   | true    | false   | false              |

object, that represents the shared geometric object of two geometric objects *A* and *B*. *Touches(Boundary(A), B)* is true if the combinatorial boundary of the object *A* touches object *B*.

To be interpreted as one of the base relations of the RCC-8, all values of a column in Table 1 have to be fulfilled. In order to enable the qualitative spatial reasoning, the composition table as shown in [21] has to be implemented.

### 3.3 Architecture

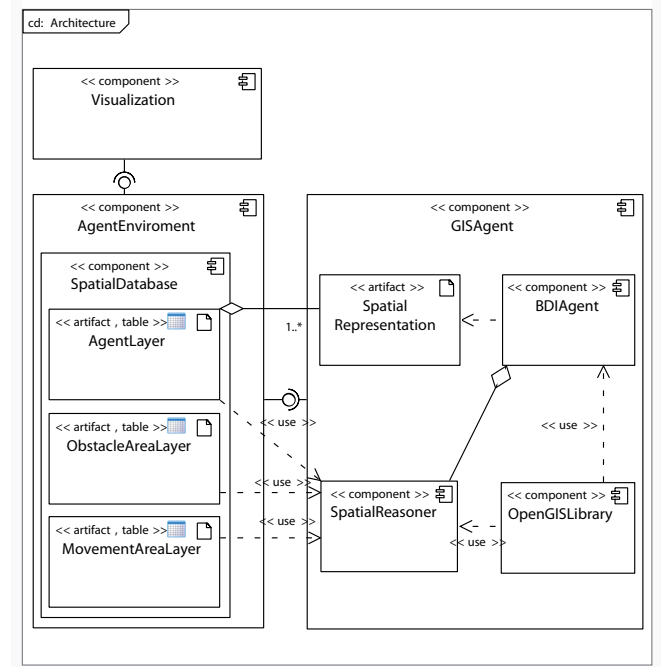
This approach links spatial data, qualitative representation and reasoning, and reasoning in a multi-agent context. The architecture is aligned with the fundamental principles of geographical information systems (GIS). A GIS is an information system that stores, analyses and displays spatial data from the real world [3]. Figure 2 shows an UML component diagram of the system architecture of our approach. The AgentEnvironment component contains a spatial database and provides interfaces for interactions with the other components. The spatial database is compliant to the OpenGIS standard [6] that supports storage and analysis of geometric objects. For the spatial representation of the layers (AgentLayer, ObstacleLayer, MovementLayer) a database table exists following the thematically storage approach of geographic data.

The GISAgent component consists of the BDIAgent with its SpatialRepresentation and SpatialReasoner described in the previous section. The SpatialReasoner implements the RCC-8 composition table and analyzes the spatial data from the AgentEnvironment using the spatial features of the OpenGISLibrary. The OpenGISLibrary provides spatial analysis functions compliant to the OpenGIS standard [6]. The SpatialRepresentation artifact constitutes the geometric object representing an agent. The table AgentLayer of the AgentEnvironment is a composition of all SpatialRepresentation artifacts of the participating agents.

The Visualization component displays the spatial data of the AgentEnvironment divided into the underlying layer structure (Figure 3). An iterative update ensures the currentness of the visualization of the dynamic environment.

## 4 EVALUATION

For the evaluation a multiagent system based on the following components was implemented. For the implementation of the BDI agent

**Figure 2.** UML component diagram of system architecture

system Jadex [16] is used. The spatial analysis functions are implemented using the JTS Topological Suite [24] and PostGIS [17], a GIS extension for the PostgreSQL database system [18]. The database system also acts as a data representation layer of the environment of the agent system as described in the section above. The environment of the agents and the associated visualization is developed with a further development of the Hopix platform [10]. Figure 3 shows a screenshot of the system.

To demonstrate the applicability and utility of the presented approach, we conduct three experiments of a simulated construction site as a part of the service chain shown in Figure 1. The logistics task of the autonomous resources (dumper) is to transport spoil automatically from one point of the construction side to another point as an iterative action. In the experiments both positions are equal

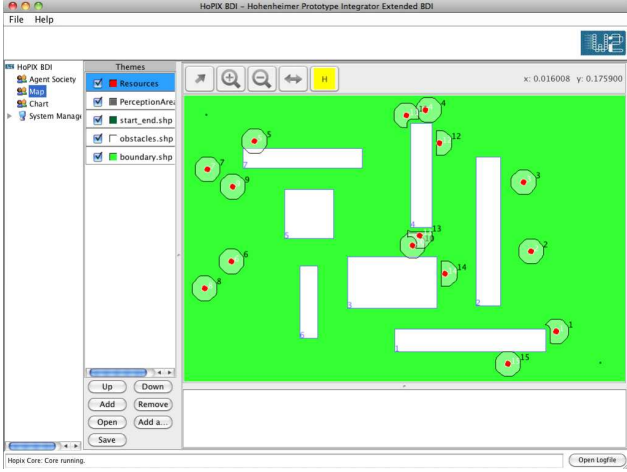


Figure 3. Screenshot of spatial visualization

for all resources. The resources are represented by agents that simulate the autonomous, collision-free motion. The general setup for all experiments is based on the same environment and parameters. The boundary of the simulation area is a rectangle with a width of approximately 600 meters and a length of approximately 400 meters based on real geographic data. The boundary also defines the thematic layer, that determines the agents' movement area. Another polygon layer defines rectangles that determine the obstacles. Collisions with these obstacles and with other agents have to be avoided. The speed parameter is set to 4 meters per simulation interval, the radius of a resource to 4 meters and the diameter of the agents' perception area to 24 meters. All agents have a start/current position and a target position. The initial angle for movement is defined by the angle between the start position and the target position.

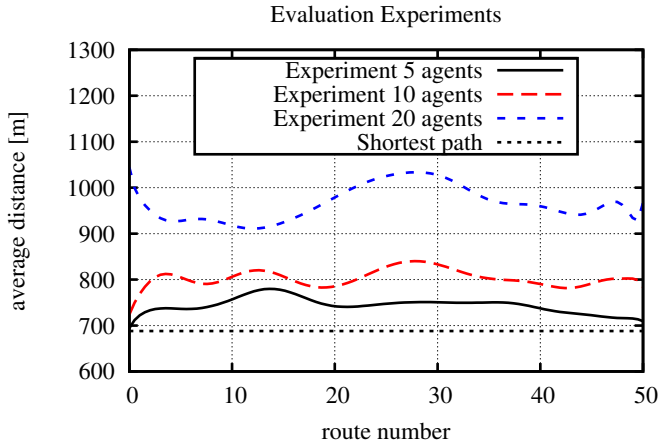


Figure 4. Analyze of mean passed distances of agents in evaluation experiments

The scenario has been executed with 5, 10, and 20 agents in three experiments. We measure the covered distance between the two predefined target positions. The optimal path has a length of 688 meters. Each experiments is conducted with 50 routes. Figure 4 shows the mean of the covered distance for all agents in the experiments per route number. The measurements confirm the expected result, that

the more agents participate in the simulation the greater the covered distance gets. Table 2 gives an impression about the deviation between the optimal distance of the shortest path and the mean of all covered distances in one experiment.

From the service chain perspective the requirements regarding the adaptivity in the spatial dimension are fulfilled. The agent model enables the logistic entities to react autonomously on a dynamic environment, like dynamic obstacles. The transportation routes are calculated adaptively regarding the dynamic parameters. Changing target positions lead to an immediate internal re-planning of the autonomous logistic entities.

Table 2. Mean and deviation of experiments

|                   | 5 agents | 10 agents | 20 agents |
|-------------------|----------|-----------|-----------|
| Mean distance [m] | 743.7    | 803.1     | 965.0     |
| Optimum [m]       | 688.0    | 688.0     | 688.0     |
| Deviation[%]      | 7.5%     | 14.3%     | 28.7%     |

## 5 RELATED WORK

The integration of qualitative spatial reasoning mechanisms with software agents in a similar approach is investigated in [22]. This approach utilize a different spatial calculus based on 23 base relations and make questions about perception in a real-world environment explicit. However, the authors remain on a high level of abstraction and do not give details about the concrete model of their software agents.

In [27], Zimmermann and Freska enhance spatial reasoning by the concept of motion. They show how the static representation can be interpreted as motion and how the conceptual neighborhood structure can be used. Though this approach is not suitable for dynamic environments.

In [25], Wolter et. al. provide an approach that simulates continuous, collision-free movements of vessels in an open sea scenario. They formalize common traffic rules by the  $OPRA_m$  calculus, which describes the relations between oriented points. Whenever two or more vessels meet below a predefined distance, the system calculates rule-compliant maneuvers for the involved vessels. However, details about the agent model are not given. The authors report that the representation and reasoning processes occur at the level of a control system with a bird's eye view – not at the level of the individual agents.

The problem class of motion planning problem addresses a similar research question [12]. In particular the path planning problem of multiple mobile robots in an environment provides relevant approaches. The existing techniques can be partitioned in three subcategories: centralized, decoupled, and decentralized approaches. The latter category have the closest relation to this contribution. In [13] the problem of multi-robot decentralized motion planning is formulated as a maze-searching problem. The main emphasis of [13] and [4] is on the algorithmic issues of decentralized decision-making. The authors of both contributions ignore real life uncertainties of input information – they assume precise knowledge and perfect sensing within a reactive approach. The authors of [8] and [5] also follow a decentralized approach. Whenever a collision between robots in a predefined perception area is anticipated, groups of robots are formed in order to solve the problem.

In [15], Müller et. al. present a navigation approach that generates human-like motion behavior for mobile robots in highly pop-

ulated environments. The approach detects and tracks people in the surroundings of the robot and integrates this knowledge into the planning process based on the A\* algorithm. The presented solution does not base on software agents and is not suitable for the application in a logistics scenario, because of missing communication features.

Automated guided vehicles (AGVs) are common for transportation tasks in intra-logistic scenarios such as container terminals. Several contributions (e.g., [9], [7], and [14]) address this topic in centralized approaches that require a control instance based on an operation research approach. Furthermore the whole system has to be equipped with localization systems or solutions based on wire-guided tracks or optical surface markings.

## 6 CONCLUSION

We have shown the applicability of qualitative spatial reasoning with topological information in BDI agents for autonomous, collision-free motion in dynamic environments. In particular collision detection profits from the spatial reasoning. The multiagent paradigm matches with the decentralized approach for logistic control systems. The BDI architecture allows to make the agents' decision process explicit – based on beliefs, desires and intentions. The results of the evaluation experiments show that the approach is applicable in simulation environments based on real geographic data. The relevant spatial capabilities show how standard features of the OpenGIS standard can be integrated into an agent environment, and how functions can be mapped to the RCC-8 base relations.

From the motion planning perspective, future research has to investigate the suitability of different common strategies for calculating the agent's new position for the proposed agent model. The support of further multiagent technologies like agent communication, agent interaction protocols and coalition formation can decrease the mean deviation measured in the evaluation experiments. From the logistics perspective, we have to investigate further how agent technologies, especially the BDI paradigm, can support the decentralized approach of logistic control systems along the complete supply chain, particularly in the spatial dimension. We also have to investigate how further spatial cognition and spatial analysis functions can be employed for software agents in the logistic domain. In order to refine the evaluation, the effectiveness and efficiency of the model has to be proven. Further, the model has to be evaluated against existing (non-)qualitative approaches.

## ACKNOWLEDGEMENTS

The work presented in this paper was partly funded by the German Federal Ministry of Education and Research under the project Geolog (BMBF 01IS09019A).

## REFERENCES

- [1] R. H. Bordini, J. F. Hübner, and M. Wooldridge, *Programming Multi-Agent Systems in AgentSpeak Using Jason*, John Wiley & Sons, 2007.
- [2] M. Bratman, *Intentions, Plans, and Practical Reason*, Harvard University Press, 1987.
- [3] P. A. Burrough and R. A. McDonnell, *Principles of geographical information systems*, Oxford University Press, 1998.
- [4] L. Chun, Z. Zheng, and W. Chang, 'A decentralized approach to the conflict-free motion planning for multiple mobile robots', in *International Conference on Robotics and Automation (ICRA)*, pp. 1544–1549, (1999).
- [5] C. Clark, S. M. Rock, and J. C. Latombe, 'Motion planning for multiple mobile robot systems using dynamic networks', in *IEEE Int. Conference on Robotics and Automation*, pp. 4222–4227, (2003).
- [6] Open GIS Consortium. OpenGIS Simple Features Specification For SQL, 1999.
- [7] M.P. Fanti, 'Event-based controller to avoid deadlock and collisions in zone-control AGVs', *International Journal of Production Research*, **40**(6), 1453–1478, (2002).
- [8] M. Jäger and B. Nebel, 'Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots', in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1213–1219, (2001).
- [9] C. W. Kim and J. M. A. Tanchoco, 'Conflict-free shortest-time bidirectional AGV routing', *International Journal of Production Research*, **29**(12), 2377–2391, (1991).
- [10] S. Kim and T. Bieser, *HoPIX: Simulationswerkzeug zur Analyse von Wertschöpfungsnetzwerken*, 161–171, Individualization Engineering, Cuvillier-Verlag, Göttingen, 2008.
- [11] H. Langer, J. D. Gehrke, J. Hammer, M. Lorenz, I. J. Timm, and O. Herzog, 'A framework for distributed knowledge management in autonomous logistic processes', *International Journal of Knowledge-Based Intelligent Engineering Systems*, **10**(4), 277–290, (2006).
- [12] S. M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.
- [13] V. J. Lumelsky and K. R. Harinarayan, 'Decentralized motion planning for multiple mobile robots: The cocktail party model', *Autonomous Robots*, **4**(1), 121–135, (1997).
- [14] R. H. Möhring, E. Köhler, E. Gawrilow, and B. Stenzel, 'Conflict-free Real-time AGV Routing', in *OR*, eds., H. A. Fleuren, D. den Hertog, and P. M. Kort, pp. 18–24, (2004).
- [15] J. Müller, C. Stachniss, K. O. Arras, and W. Burgard, 'Socially inspired motion planning for mobile robots in populated environments', in *International Conference on Cognitive Systems (CogSys'08)*, Karlsruhe, Germany, (2008).
- [16] A. Pokahr, L. Braubach, and Lamersdorf, W., 'Jadex: Implementing a BDI-Infrastructure for JADE Agents', in *EXP - In Search of Innovation (Special Issue on JADE)*, pp. 76–85, Turin, (2003).
- [17] PostGIS. Website. Available online at <http://postgis.refractor.net> visited on April 8th 2010.
- [18] PostgreSQL. Website. Available online at <http://postgresql.org> visited on April 8th 2010.
- [19] D. A. Randell, Z. Cui, and A. G. Cohn, 'A spatial logic based on regions and connection', in *Proc. 3rd International Conference on Knowledge Representation and Reasoning*, pp. 165–176, Los Altos, (1992).
- [20] A.S. Rao, 'Agentspeak(L): BDI agents speak out in a logical computable language', in *Proceedings of Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pp. 42–55, Eindhoven, (1996).
- [21] J. Renz and B. Nebel, 'Spatial reasoning with topological information', in *Spatial Cognition*, eds., C. Freksa, C. Habel, and K. F. Wender, volume 1404 of *Lecture Notes in Computer Science*, pp. 351–372. Springer, (1998).
- [22] A. Schultdt and B. Gottfried, 'Spatiotemporal cooperation at the geographic scale', in *Interdisciplinary College (IK 2008)*, eds., B. Loos and V. Micelli, p. 32, (2008).
- [23] M. Schüle, R. Herrler, and F. Klügl, 'Coupling GIS and Multi-agent Simulation - Towards Infrastructure for Realistic Simulation', in *MATES*, eds., G. Lindemann, J. Denzinger, I. J. Timm, and R. Unland, volume 3187 of *Lecture Notes in Computer Science*, pp. 228–242. Springer, (2004).
- [24] JTS Topology Suite. Website. Available online at <http://www.vividsolutions.com/jts> visited on April 8th 2010.
- [25] D. Wolter, F. Dylla, S. Wöfl, J. O. Wallgrün, L. Frommberger, B. Nebel, and C. Freksa, 'Sailaway: Spatial cognition in sea navigation', *KI*, **22**(1), 28–30, (2008).
- [26] M. J. Wooldridge, 'Intelligent agents', in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, ed., Gerhard Weiß, 27–77, MIT Press, Cambridge, MA, USA, (1999).
- [27] K. Zimmermann and C. Freksa, 'Enhancing spatial reasoning by the concept of motion', in *Prospects for Artificial Intelligence*, eds., A. Sloman, D. Hogg, G. Humphreys, and A. Ramsay, 140–147, Amsterdam: IOS-Press, (1993).

# Managing Contingencies in Timed Transportation Networks by Agent Technology

Natalja Pulter and Jens Nimis and Peter C. Lockemann<sup>1</sup>

**Abstract.** Lean production is characterized by low levels of inventories at the production sites. Because classical strategies such as just-in-time or just-in-sequence are susceptible to deviations from the original plan (“contingencies”) supply chains based on regular and guaranteed delivery times (“timed” transportation networks) have been proposed which potentially seem more robust if one can compensate for the contingencies within a single clock cycle. Since contingencies may be of different severity, compensation strategies may vary from the local to the global and, consequently, may sometimes pose serious challenges if one wishes to stay with one cycle. Consequently, intelligent contingency management becomes an essential factor for the success of a timed supply chain. The paper introduces a multi-level model that escalates contingencies to the appropriate compensation level. It examines which kinds of contingencies and, hence, which compensation level is particularly well suited for agent technology, and discusses an agent solution within the escalation model.

## 1 Introduction

Lean production is characterized by low levels of inventories at the production sites. Typically, these are achieved by strategies such as just-in-time or just-in-sequence, but have as a drawback a high susceptibility to disturbances in the supply chain with a concomitant high cost in downtime and/or high effort of replanning. To overcome these drawbacks, a timed approach has lately been proposed where parts are delivered in a timed fashion of regular intervals. While such a strategy requires a modest amount of inventory, production planning is eased by working from fixed and guaranteed delivery times, and the hope is that delays or other disturbances can be compensated for more robustly within a single clock cycle. This paper addresses the issue of such compensations. More specifically, it concentrates on the part of the supply chain where parts are transported from a supplier to the manufacturer. Since deviations from the original plan (“contingencies”) may be of different severity, compensation strategies may vary from the local to the global and, consequently, may sometimes pose serious challenges if one wishes to stay within a single clock cycle. Knowing the kind of exceptions and the compensation techniques and their cost in terms of resource demands may also allow the stakeholders to trade off compensation for longer cycles and concomitant higher inventories.

There is also an economical and an ecological side to timed transportation networks. Due to its central geographical location German highways carry an extremely heavy load of truck traffic: 77% of goods are transported by trucks. Within the present highway system

the load limits have practically been reached. There is little chance of much further extension of the road network, both because of the high cost and because of popular resistance. On the other hand carriers like trains or ships still offer considerable spare capacities. The problem so far has been that rail and ship transports run on regular schedules and thus were considered too inflexible. Once manufacturing is based on timed processes, rail and ship become natural partners in a timed transportation network. Consequently, intelligent contingency management becomes an essential factor for the success of a timed supply chain.

Today the intelligence is provided by human dispatchers. With the continuous growth of the logistic networks, the increasing number of criteria to be observed and the ever shorter times for decision making the dispatchers become overextended. Hence, contingency management should be computer-based to the extent that simple measures can be triggered automatically so that the dispatchers can attend to the more complex contingencies, and for these the system should provide valuable assistance.

The paper is organized as follows. The second section introduces the logistical and organizational difficulties, which make dealing with incidents so complicated in the logistical area. Section 3 proposes a multi-level model that escalates contingencies to the appropriate compensation level. As part of this model, the fourth section discusses how contingencies can be detected and then classified in order to determine how to attempt compensation. Section 5 describes the escalation procedure with the specific purpose of identifying the kind of what contingencies for which agent technology seems particularly well suited. We argue that this is the case whenever negotiations become an essential part of the solution. Section 6 discusses the agent approach in more detail and illustrates it by an example. Section 7 concludes the paper.

## 2 Problem Reach

Contingencies are fairly local events, such as a traffic congestion, engine breakdown, incomplete load, driver sickness, etc. Their effects, however, may become more widespread if not treated properly. Treatment may involve securing new resources, and these may have to be withdrawn from other supply processes. Clearly then, the more local the compensation is the more limited the effects will be on the same or other logistical processes. We give a few examples below.

**Business level** Take a dispatcher who has to compensate for late provision of a parts load. She can choose among a variety of solutions, e.g., change the order of load points in the tour, send an idle vehicle at a later time, cooperate with other company in the network passing the load point. Ideally she will evaluate the different solutions in light of various attributes deemed important for the

<sup>1</sup> All authors are with FZI Forschungszentrum Informatik an der Universität Karlsruhe, Germany, email: {pulter, nimis, lockemann}@fzi.de

carrier: additional cost, tour distance, tour period, current time or load buffers, etc. More often than not the evaluation must consider several such attributes in combination. Further, making use of time or load buffers or of idle vehicles uses up precious resources that are not available for subsequent contingencies. Several such contingencies may occur concurrently, making the decision even more complicated.

**Network level** Mostly dispatchers try to solve contingencies by mobilization their own resources. Sometimes they will resort to help by other network partners. In this case a dispatcher has to make phone contact to the dispatchers of other partners and try to negotiate an offer, which can be very difficult because of competing interests. The fact that a dispatcher can only negotiate with one participant at any one time complicates the situation, at least if she is under high time pressure. There are indications that in a timed network cooperation among otherwise competing partners must increase in order to keep the cost of collection runs (“milk runs”) under control so that negotiations become more of a norm.

**Supply chain level** A supply chain involves a good number of partners, e.g., the local distributors at the supplier’s and manufacturer’s ends, and one or more large carriers (rail, ocean-going vessel) in the middle. Contingencies at any one of them that remain unresolved impact all the ones that follow, and even a solution may impact the preceding ones if it imposes new constraints on them. In general then, contingencies often have a habit of spreading across a large part of the network. Hence, any solution should be checked on its effects on the remainder of the supply chain in order to ensure that no follow-up contingencies will occur. In summary, a dispatcher should find a solution as fast as possible, while making as few changes as possible, and always keep an eye on follow-up effects.

The simultaneous management of all this requirements under the high time pressure is difficult enough, but it becomes even more difficult when multiple incidents happens at the same time or during a short time span. Clearly, coping with contingencies would benefit from automated help.

### 3 Contingency Management Model

The basic concept behind our contingency management is the notion of *buffer*. A buffer is a spare resource kept in reserve, like an extra vehicle or a tolerance in delivery time, and is provided as part of a planning process. Buffers are the means for resolving contingencies. Contingency management, then, offers mechanisms for determining which among the available buffers offers the most effective and efficient solution, i.e., for selecting among the buffers those that add to the robustness at least cost and least spread of effects. The search starts with the most local buffers, and only if no solution can be found successively escalates to buffers with more global reach. Of course, before any remedial action can be taken a contingency must be detected as such, and must be classified in order to determine the buffers that have the potential to overcome the contingency. Consequently, contingency management consists of two main parts: contingency detection and classification, and the actual contingency management (Figure 1).

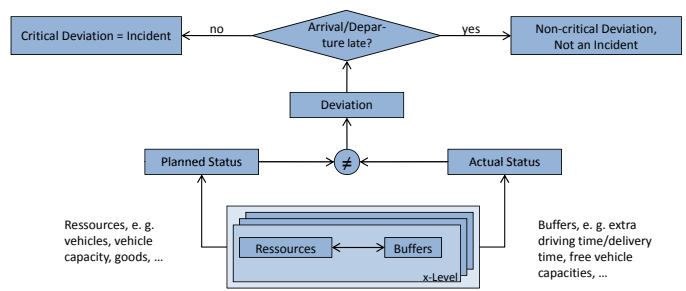


Figure 1. Contingency model

## 4 Contingency Detection and Classification

### 4.1 Related Work

Contingencies are events that fall outside the (planned) norm. Tools for detecting such events are known under the name “supply chain event management (SCEM)” and monitor the logistics processes and compare the observed events with predefined events. If an event does not occur when expected or occurs when not expected it is reported. Events that characterize critical plan deviations are filtered out and reported to the user in realtime [3].

One of the better known supply chain management systems is SAP SCM. With SAP EM as a part of SAP SCM supply chain processes can be monitored for exceptions [7]. Each relevant process is represented by an object that includes all relevant milestones. The observed events are categorized, and a rule engine determines whether follow-up activities are necessary. If a logistics partner knows early enough whether to take action the management of the logistics processes has a decidedly pro-active flavor.

Pro-active process management is also the objective of arvarto services [4], an approach of a pro-active process management by proactive messages to important process participants. The appearance or absence of events triggers messages like SMS or Email to the best-suited participants. The services include a clear trend towards automatic actions like generation of an electronic order if the original delivery time is by far exceeded. For the generation one may draw on suggestion from a knowledge base.

### 4.2 Contingency Detection

Supply chains are a sequence of transportation processes executed by successive carriers. Today for all practical purposes, events can only be observed when the load is handed over from one carrier (or service provider like a warehouse facility) to the next, i.e., at the so-called transshipment points. In the future, events will become more numerous, e.g., by reading RFID tags also during shipment [5], or by tracking the position of trucks by GPS. At any rate, most events (hopefully all) are of no further relevance. What is needed, then, is a filter that compares the characteristics of each event to a set of prescribed criteria drawn from the plan, and only pass along those events that are determined to be critical deviations, such as time delays, missing, damaged or incorrect loads. An interesting issue is what to do with those events that can easily be satisfied by one of the most local buffers, like an arrival time within the tolerance level. Should it be filtered out or passed on to contingency management? In the former case the number of events can further be reduced and processing of the contingency can be sped up, but at the price of functional redundancy because one must register that the buffer used

has been reduced in size, a functionality that is also an essential part of contingency management. Notice that because not all contingencies are known in advance events must follow an abstract (generic) model.

Our present solution utilizes a commercial platform by ptv AG and opts for the first approach of strong filtering. The input consists of the actual and planned states of the transportation resources, e.g., vehicles, orders etc. In the first step corresponding states are compared regarding properties like quantities, arrival and departure times. When a deviation is detected, it is tested against the existing local buffers in the affected transportation chain like driving time, handling time and capacity range. If the deviation can be compensated by any of these buffers it is classified as a non-critical deviation. A buffer must then be chosen for this deviation type, and adjusted.

If the deviation cannot be compensated for by such a buffer, it is classified as critical, the details of the event are stored in a database and the contingency management is notified.

## 5 Contingency Management

### 5.1 Escalation and Optimization

Take a contingency where a load is not provided on time. Contingency management may try to reorganize buffers of the affected tour. For example the tour points of the affected tour could be rearranged according to the earliest possible and latest acceptable pickup or delivery time. But this may have an effect on other tours visiting some of the same tour points. Instead one may try to use buffers of other tours or orders. Clearly the effect becomes much more global and must be dealt with on another level. Clearly too, the higher up the level is where one seeks a solution, the wider is the range of potential buffers, but also the effect on buffers. Hence, escalation quickly becomes an issue of optimization. It is not sufficient to optimize only with respect to one criterion, e.g., costs, because every resolution of a contingency consumes buffers which may also be needed in case of another contingency.

### 5.2 Escalation Levels

Since buffers are resources, and one can associate a certain reach with each, it seems only natural to define the escalation levels by classifying buffers into solution levels. To follow the typical planning strategy, the demand for transportation arises either from the need for production material or from the provision of a certain amount of goods. So a transportation chain starts with a transportation *order* between participants in the *logistic network*, which describes the amount of which goods that have to be transported from a certain source to a certain destination. Then the transportation order is mapped to a *transport*, which has to pass several transshipment points on its way to the recipient. This is especially true for multimodal transportation networks where the goods may change between (several) trucks, rail or ship, and often also when transports start and end in milk-runs rather than hub-and-spoke transports to avoid half-empty trucks. Next the planning process assigns the transports to *vehicles* and these to *tours*. Accordingly the basic solution levels are order, transport, and tour level. Thus leaves out two other kinds of buffers: Substitute trucks and substitute carriers. Both are costly remedies and may have a wide reach. Consequently, these two – fleet and network – are placed on top of the order level as the levels “of last resort”. Besides every process step consists of several sub-steps. Each of these levels can further be divided into sub-levels. The sub-levels of the network level reflects the participants of the logistic

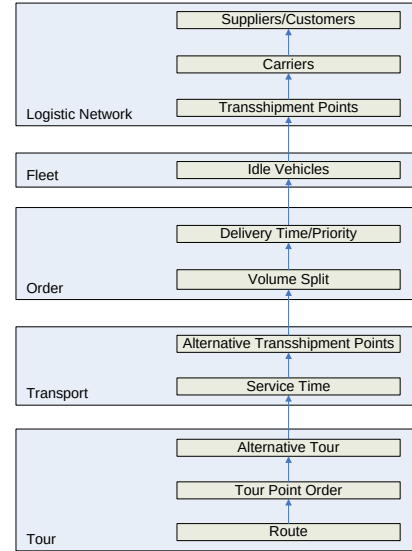


Figure 2. Escalation levels with sub-levels

network: customers/suppliers, carriers and transshipment point owners. The fleet level concentrates only on the available vehicles, while the order level is divided into one for latest delivery time sub-level and another for a volume split. Alternative transshipment points and the tolerance within pickup/delivery times are the sub-levels of the transport level. Buffers on the tour level are grouped according to alternative tours, tour points and routes. Figure 2 shows the complete set of levels.

### 5.3 Escalation Procedure

The idea underlying search by escalation is to find a solution on the lowest possible level. This way the number of affected buffers and, hence, of additional resources can be kept to a minimum (and in particular, the use of additional vehicles or carriers may be avoided). Therefore the search for a solution always starts at the lowest suitable level, and only if no solution can be found the search is extended to the next higher level, and so on. Take a vehicle breakdown. There is no use in looking for an alternative route or an alternative tour point order. In a first attempt one would examine the service time level in order to determine whether there is enough latitude in, e.g., the delivery time. If an escalation is necessary, one could review the transshipment points and decide, e.g., on replacing rail by a light truck to make up for the delay. Notice that solutions on higher levels may necessitate an adjustment of buffers on lower levels.

In general, the search must be based on accompanying information gathered over time. Some of the information can be obtained by automatic means, e.g., whether and how fast a vehicle had recently been moving. We expect that most of the information will be entered manually via PDAs by persons such as drivers, transshipment point owners, suppliers, using predefined masks (e.g., contingency cause – traffic jam, estimated delay – 60 min). Often more than one solution may be found on a given level, for example two different alternative transshipment points on the transport level. We supply ranking functions in order to make an intelligent choice.

To give a bit more detail, the various levels may offer the following solution possibilities:

**Tour – Route** A new route for a planned tour is generated based on additional information such as road closures.

**Tour – Tour Point Order** A tour is rescheduled based on revised earliest arrival times or load capacities resulting in a new tour point order.

**Tour – Alternative Tour** Other tours in the vicinity of a transport point are searched and an attempt is made to place the load in such a tour.

**Transport – Service Time** A transport is picked up or delivered the next regularly scheduled visit.

**Transport – Alternative Transshipment Points** A load is picked up at or delivered to an alternative transport point without changing the carrier. Conditions to be observed are whether there is still enough storage space in the chosen transshipment point, or if there is some other transshipment point that has goods that could be used as a replacement because they have sufficient latitude in their delivery time. The tour is then rescheduled according to the new pickup/delivery location.

**Order – Volume Split** Sometimes the supplier or the customer can tolerate that a volume of parts is picked up or delivered in several batches. Alternatively, if there is enough slack at later transshipment points, an order may still be split during the transportation as long as the batches can be consolidated in time before the delivery.

**Order – Delivery time/Priority** The planned tours are checked for transportation orders with later delivery times than the critical transports. If such transports are found, they are removed from a tour and replaced by the critical transports.

**Fleet – Idle Vehicles** The critical transports are rescheduled with a vehicle that is idle at the time of the pickups/deliveries of the transports.

**Network – Transshipment Points** Other participants of the logistic network are contacted, e.g., to negotiate the use of their transshipment point, to replace the missing load by goods stored with them.

**Network – Carriers** Other carriers of the logistic network are contacted whether they could take over the transportation of one or several orders.

**Network – Suppliers/Customers** Other suppliers/customers are contacted to negotiate with them whether they could provide certain critical goods or take excess goods.

The goal is to maintain robust schedules (reliable regular schedules) for suppliers and manufacturers: No matter what the contingencies are they remain invisible to the two groups. That is true unless the escalation reaches the top with no solution. In this case the participants will have to renegotiate the entire order.

## 6 Agent-based Contingency Management

### 6.1 Choice of Network Level

Logistics is a highly developed technology with a wealth of optimizing algorithms for almost all the escalation levels [2]. Unfortunately most of them aim at the optimization of a collection of tours rather than a single tour. Consequently, they are very inefficient if one tries to perform replanning. Mes et al. have shown that agent-based transportation planning may be more efficient and almost equally effective [11]. Neagu et al. use agents to optimize a collection of tours when a single order is being added ([12]).

None of these planning algorithms optimized more than one criterion, whereas in dealing with contingencies we believe that several criteria (several buffers) should be considered. Outside logistics sev-

eral promising approaches for agent-based multi-criteria negotiation exist in the e-commerce area ([6, 10]).

In summary then we feel that the use of agents is a promising technique in contingency management. This is supported by Weiss, who claims in [13] that software agents can be both efficient and effective in combinatorially complex settings. The network level provides such a setting. Below we develop an agent-based approach on the network level as a first example before we try to extend agent-based solution to the lower levels.

### 6.2 Related Work

Agents have already been used in several fields of logistics. One is supply chain event management. [15] introduces an approach of an agent-based SCEM. It defines monitoring criteria, gathers information on the criteria and interprets it and in case of an unexpected event generates alerts and directs them to the actors. These tasks are realized by different agents. E.g., the communication with supply chain partners is managed by a discourse agent, while the gathering of information is realized by a surveillance agent.

A second field is transportation management. [8] introduces an agent-based approach for optimization of tour plans. Thereby a shipping company allocates orders to its trucks. The trucks represented by truck agents bid on whole or on parts of transportation orders while the shipping company agent combines the best bids to a tour plan. Instead the shipping company agent may elect not to execute an order on its own, but to cooperate with another company. In this case a negotiation between the two company agents takes place.

Another approach to optimize tour planning is presented in [12]. Here the transportation area is divided into regions and the tours are optimized by the AgentClusterManager of the single region. If a transportation order concerns different clusters, the agents of the clusters agree which of them can handle the transportation in a cheaper way. In order to optimize tours within a region, trucks negotiate with one another about transfers of the transportation orders to other tours until cost-saving changes are no longer possible.

### 6.3 Agent Types and Tasks

The network scenario is typical for situations that tend to be approached by software agents. All participants pursue their own egoistic goals and, therefore, try to keep the information about their orders, customers, resources, and problem details secret from the other participants. On the other hand the participants need to collaborate from time to time in pursuance of a common goal. Negotiations have the purpose of maximizing one's own benefits and still satisfy the common goal [13]. In our scenario we distinguish between the following participants and, consequently, agents, which become active only in case of a contingency, and they only deal with contingencies (Figure 3):

**Supplier agents** The supplier agents have two main tasks and two goals. On the one hand the supplier agents are activated when the goods committed to a customer cannot be provided in time for the pickup. In this case a supplier agent can either contact the carrier agent and try to schedule a new or additional pickup date or it can contact other supplier agents for help and negotiate with them the conditions for their support. To meet such a contingency each supplier agent maintains a list of suppliers producing the same kind of products together with their location information. The negotiation goal of a supplier agent is to find a partner who can provide the

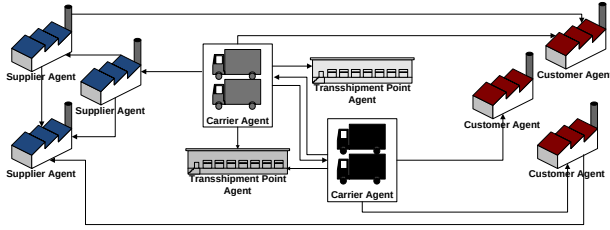


Figure 3. Network agents and their possible negotiation partners

necessary goods at the earliest possible time for the lowest possible price. On the other hand a supplier agent may be on the receiving end when contacted for help. In this case the agent checks whether the requested goods can be provided at which time and in which quantity. Its negotiation goal is to get the help order at an acceptable price but to achieve the highest possible price.

An additional function may be to contact customer agents in the case of an overproduction or of an order cancellation by a customer. While not a logistics problem by itself, the supplier agent would be the source of one by trying to find a customer (it may maintain a customer list for this purpose) who would be willing to buy the goods, and to negotiate the price. If all these negotiations come to a successful end one or more agents must contact their carrier agents.

**Customer agents** Similarly, customer agents pursue two main tasks and two goals. One has to do with the need for additional goods if the demand increases over plans or if goods cannot be delivered on time. In this case a customer agent contacts other supplier agents to find one who is willing to sell the goods. The goal is to do so and achieve the lowest possible price. The complementary task is to provide offers for suppliers that try to sell spare goods. When a customer agent receives an offer for certain goods it checks for available inventory space and negotiates the price of the goods with the goal of obtaining the lowest possible price. Again, if all these negotiations come to a successful end one or more agents must contact their carrier agents.

**Transshipment Point Agents** A transshipment point agent is purely reactive. Requests for helping out with goods involve checking the stocks at hand, their recipients and delivery times and potential resupplies. If suppliers help out customers, or vice versa, the transshipment point agent checks for spare capacity for intermediate storage of goods.

**Carrier Agents** At its core, transport logistics is about scheduling carriers and their resources. Hence, carriers are the ones who suffer the consequences of all short-term changes, and to cope with them carrier agents must be able to contact all other agents, be they supplier, customer, transshipment point or other carrier agents. Every carrier agent maintains a list of the other partner agents together with additional information about them such as location information or types of produced goods. To give an example, if a carrier needs to offload an order to some other carrier its agent will determine the most suitable negotiation partners according to additional information like the transshipment points within a certain radius of the contingency. The negotiation goal of the agent requesting help is to keep the additional cost as low as possible, to find a solution as close to the original plan as possible, and presumably to avoid new long-term competition. On the other hand,

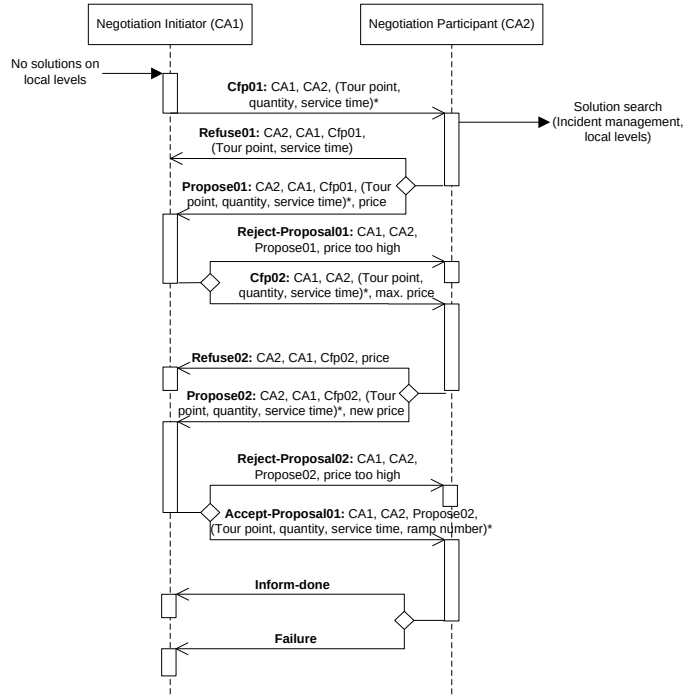


Figure 4. Example of a negotiation procedure between two carrier agents

a carrier agent receiving a request for help will check whether it can provide the requested resources without endangering its own plans. If this is the case it tries to get the additional transport order at an advantageous price.

## 6.4 Agent-based Negotiations

We developed first solutions for all these negotiations based on the FIPA Iterated Contract Net Interaction Protocol [9], a modification of the original Contract Net protocol [14]. In the Contract Net protocol an agent generates a task announcement and sends it to one or more other agents. The agents listen to task announcements and bid on a task they would like to execute. The requesting agent selects one bidder and informs them by an award message. Finally the requesting agent is informed when the task is done [14].

The FIPA Iterated Contract Net Interaction Protocol extends this basic protocol by adding confirmation and rejection communicative acts. It also allows the requesting agent to alter the original announcement (call-for-proposal/cfp) rather than accept offers in the first negotiation round, and start a second negotiation round in the hope of getting more suitable proposals [9]. We decided to allow only two negotiation rounds to stay reasonably reactive. Notice that for reasons of competition the protocol must meet its purpose under the constraint that no information other than absolutely necessary for the task is provided with the call-for-proposal and the proposals. In particular, no information is exchanged on orders, resources etc.. The standard solution procedure on the carrier sub-level of the network solution level including negotiations looks as shown in Figure 4.

If for example the contingency detection discovered a critical delay of a vehicle because of a break down and no solution was found on lower levels, the carrier agent is activated. The carrier agent looks up other carrier agents having storage facilities in the vicinity of the affected tour points and sends a call-for-proposal to these agents. The

call-for-proposal looks as follows:

```
(cfp01
:sender CA1
:receiver CA2
:content
(Tour point 1, 10 pallets, 30.09.10, 09:00;
 Tour point 2, 15 pallets, 30.09.10, 14:00))
```

The message contains the performative (cfp), the sender (CA1), the receiver (CA2) and the actual message content describing the task. In this case it would be the location information of the affected tour points, the quantity in the single tour points and the time constraints in the single tour points (earliest pickup time/delivery time, latest pickup time/delivery time).

This is a wonderful example for demonstrating the recursive nature of contingency management. The carrier agents receiving the call-for-proposal try to reschedule their plans based on the received information. From their point of view this is nothing but a contingency, albeit one of ad-hoc orders. Consequently, they will turn to contingency management with the ad-hoc orders. Recursion is easily limited because only solutions on local levels make sense now. If a carrier agent does not find a suitable solution, it sends a refuse-message to the requesting agent. This message can contain the reason for refusal (e.g., no free capacities on scheduled vehicles, no idle vehicles) or a set of conditions which have to be met in order to fill the proposal (e.g., maximum quantity: x pallets, earliest pickup time: 01.10.10, 9 am). If contingency management is successful the carrier agent estimates the additional cost of the help offer, adds the desired profit and makes a proposal. The in-reply-to (cfp01) proposal contains the same information as the cfp-message augmented by the location information of the tour points, the possible volume, the earliest and latest possible service times, and the desired price.

The requesting carrier agent evaluates all proposals. If there is a suitable proposal, the requesting carrier agent informs the winner (or several winners if only a combined proposal covers all affected transports) and all the other carrier agents about the reject of their proposals. In case of an acceptance the message contains additionally the details of the transports (e.g. the supplier name, the ramp number, ...). The reject-proposal message contains besides the usual information the reason for the reject (e.g., price too high).

If none of the proposals contains a suitable offer the requesting carrier agent starts the second negotiation round mentioned above by sending a modified call-for-proposal. This message will only go to the most promising negotiation partners based on their proposals, e.g., those that deviate from the original request only within prescribed limits. All other negotiation partners receive a reject-proposal message. The reduced group of recipients can again decide whether they bid on the new proposal or refuse to do so. At the end of this round the carrier agent has either to accept one of the bids or to escalate the solution search to the next level.

The negotiations between supplier and customer agents or between carrier and supplier/customer agents follow the same protocol. They differ only in the content of the message text, e.g., product type instead of earliest and latest service times. Where they differ drastically is, of course, how the agents reach their decisions, but this part of the solution process remains hidden from the protocol.

## 7 Conclusion and Further Research

This work has partly been funded by the Bundesministerium für Wirtschaft und Technologie (BMWf, German Federal Ministry of

Economics and Technology, contract number 19 G 7036 E) in the LogoTakt project [1] where carriers, suppliers and manufacturers collaborate with a vendor of logistics platforms and research institutions. For the SCEM we use the vendor's commercial platform. Contingency management is designed as an add-on component to the platform. At present the lower escalation levels have been implemented to the extent that they resolve contingencies locally. As mentioned before, currently we develop the agent-based network-wide contingency resolution discussed above. All these solutions will be part of large simulation studies involving two supply chains, one where parts are delivered from several geographically distributed central warehouses by truck, and a second where the main transport segment is by rail and the recipient is an automobile manufacturer.

Contingency management based on escalation is a fairly new concept. Since the concept involves measures that may spread across several levels, response times become a critical issue. This may often preclude replanning via optimization algorithms. The question of central interest to us is whether agent approaches offer faster response times while still preserving an acceptable degree of result quality. We consider the work discussed in this paper a first step towards that objective.

## REFERENCES

- [1] Logotakt. <http://www.logotakt.de>, 2009.
- [2] D. Arnold, H. Isermann, A. Kuhn, K. Furmans, and H. Tempelmeier, *Handbuch Logistik*, Springer-Verlag, Berlin Heidelberg, 2008.
- [3] A. Baader and S. Montanus, *Transparency in Global Supply Chain Networks Methods and Tools for Integrated Supply Chain Event Management*, 3–11, Strategies and Tactics in Supply Chain Event Management, Springer-Verlag, Berlin Heidelberg, 2008.
- [4] Th. Becker, *Supply Chain Event Management: Innovation in Logistics Services*, 3–11, Strategies and Tactics in Supply Chain Event Management, Springer-Verlag, Berlin Heidelberg, 2008.
- [5] Th. Blecker, W. Kersten, and C. Herstatt, *Innovative IT Systems in Logistics and Supply Chain Management - RFID*, 169–175, Key Factors for Successful Logistics - Services, Transportation Concepts, IT and Management Tools, Erich Schmidt Verlag, 2007.
- [6] E. de Oliveira, J. M. Fonseca, and A. Steiger-Garo, 'Multi-criteria negotiation in multi-agent systems', in *CEEMAS'99, St. Petersburg, Russia*, (1999).
- [7] P. Diener and M. Rosemann, *Supply Chain Event Management: Managing Risk by Creating Visibility*, 3–11, Strategies and Tactics in Supply Chain Event Management, Springer-Verlag, Berlin Heidelberg, 2008.
- [8] K. Fischer, J. P. Mueller, and M. Pischel, 'Cooperative transportation scheduling: An application domain for dai', *Applied Artificial Intelligence*, **10**, 1–33, (1996).
- [9] Foundation for Intelligent Physical Agents. Fipa iterated contract net interaction protocol specification, March 2002.
- [10] C. M. Jonker, V. Robu, and J. Treur, 'An agent architecture for multi-attribute negotiation using incomplete preference information', *Autonomous Agents and Multi-Agent Systems*, **15**(2), 221–252, (2007).
- [11] M. Mes, M. van der Heijden, and A. van Harten, 'Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems', *European Journal of Operational Research*, **181**, 59–75, (2007).
- [12] K. Neagu, N. and Dorer, D. Greenwood, and M. Calisti, 'Lsatn: Reporting on a successful agent-based solution for transport logistics optimization', in *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, (2006).
- [13] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT PRESS LTD, 2001.
- [14] M. J. Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2009.
- [15] R. Zimmermann, S. Winkler, and F. Bodendorf, 'Agent-based supply chain event management concept and assessment', in *Proceedings of the 39th Hawaii International Conference on System Sciences*, (2006).

# Oil Industry Supply Chain Management as a Holonic Agent Based Distributed Constraint Optimization Problem

Fernando J. M. Marcellino<sup>1,2</sup> and Jaime S. Sichman<sup>2</sup>

**Abstract.** Very few industries can benefit more from maximizing supply chain efficiencies than the oil and gas companies. However, the behavior of such supply chains is too complex to be modeled analytically. Multi-agent systems show great similarity with respect to supply chains and provide the flexibility to model the complexities and dynamics of real world supply chains without excessive simplifying assumptions. Since supply chain management problem has a recursive structure, it becomes appropriate to employ holonic agents for its modeling. In addition, the type of relationships between its entities and the required global optimization make it natural to model its interactions as constraints, and the entire system as a Distributed Constraint Optimization Problem (DCOP).

## 1 Introduction

Supply chain management can be defined as the configuration, coordination and continuous improvement of an organized set of operations. Its goal is to provide maximum customer service at the lowest cost possible, where a customer is anyone who uses the output of a process. Since the goal of a company is to maximize profits, it must weigh the benefits versus the costs of its decisions along the supply chain. Very few industries can benefit more from maximizing supply chain efficiencies than the oil and gas companies [2]. In this industry, there is a need to ensure that each entity along the supply chain can respond quickly to the exact needs of its customers. On the other hand, one of its weaknesses is that each entity is likely to act in its best interests to optimize its own profit. In general, that doesn't meet the goal of the entire supply chain. Integrating the management decisions of different parts of the supply chain may provide huge gains and benefits, which are usually concealed by the pitfalls of short-term and local angles. Historically, the decisions of business supply chains have been highly concentrated, and researchers have used analytical approaches focusing on how to improve the efficiency of the individual entities along the chain instead of improving its combined performance. Although, in general, great corporations own a global planning application that tries to take into account the main activities and goals of its components, the complexity of the resulting system makes it unfeasible an integration of these latter. In addition, even if the centralized planning system is simple enough to work, the frequent and unforeseeable changes in the business environment render its results obsolete and useless very fast. Since multi-agent systems show great similarity with supply chains, multi-agent systems become a suitable means to manage the supply chain with a

better performance than that attained by the current human-software approach. Furthermore, the recursive nature of supply chains suggests the use of holonic agents, which are fractal structures that deal better with the communication between different abstraction levels of the supply chain. Finally, the distributed constraint optimization formalism fits very well to the supply chain management problem, which is naturally distributed but tightly coupled as a network of facilities. This approach makes the local and selfish nature of the agents meet the requirement of a global optimization. They do so using only local communication with neighboring agents, since for such problems communication with a single central agent is unacceptable. In addition, it allows agents to operate in an asynchronous and more efficient way.

This paper proposes a preliminary model that includes both holonic agents and distributed constraint optimization for the oil industry supply chain management. The next section synthesizes the basic concepts involved in the work. Section 3 describes the problem in focus, and the section 4 presents the proposed model, which is divided into an organization model and a constraint optimization model, which are then refined to a distributed constraint optimization model. Finally, the paper concludes with a summary of the current results, and an outlook on future research activities.

## 2 Basic Concepts

### 2.1 Oil Supply Chain

A typical oil industry supply chain includes exploration of new petroleum (crude oil) reservoirs, drilling of crude wells, crude extraction at onshore and offshore platforms, its transport to the refineries, the refining of the crude oil (raw material) in the refineries in order to produce the final products (petroleum derivatives), such as gasoline and diesel, the transport of those products to distribution terminals where they are dispatched to distribution companies, and finally the delivery of the derivatives to the final customers (e.g. gas stations). There are different types of crude, with distinct qualities. Each of them has a specific production profile, yielding definite proportions of each derivative product. As to the transport of crude and its derivatives, it is carried out by ships, trains, trucks, and mainly by pipelines.

Oil supply chain management is intrinsically associated with integrated planning. First, it is concerned with functional integration of acquisition of raw material (crude oil), manufacturing (refining), transportation, and warehousing activities [9]. In the oil industry supply chain, that integrated planning links fundamental processes, like

<sup>1</sup> PETROBRAS S.A., Brasil, email: fmarcellino@petrobras.com.br

<sup>2</sup> Universidade de São Paulo, Brasil, email: jaime.sichman@poli.usp.br

logistics (responsible for transportation planning), production management (oil extraction and refinery production planning), inventory management (hedge against the uncertainties of supply and demand, or reserve for seasonal demand), and demand forecasting (statistical prediction of future demand for final products). The goal of the inventory management is to determine the trade-off for the inventory levels, so that they are not too low that lead to lose opportunities, or too high that impose useless inventory hold costs. The usual way to attain this goal is to use either probabilistic analytical models or simulation.

An important component in supply chain analysis is the choice of performance measures, which are used to determine the efficiency of a system, or to compare alternative ones. The literature categorizes these measures as either qualitative or quantitative. In general, quantitative measures are related to monetary values, as cost and profit, whereas qualitative ones are based on customer satisfaction [1]. Traditionally, mathematical optimization models are used to solve supply chain management problems; however, supply chains are too complex to allow realistic models to be evaluated analytically.

## 2.2 Multi-Agent Systems (MAS)

MAS offer such useful features as parallelism, robustness and scalability. They are highly applicable in domains where centralized approaches meet their limits, and are composed of autonomous, reactive, proactive, and interacting entities called *agents*, engaged in the realization of a joint goal. Negotiation is a fundamental capability of agents to reach their goals. The agent-based framework may lead to more effective integration of production, logistics, and trading processes. Agent descriptions provide an ability to specify both static and dynamic characteristics of various supply chain entities [11]. Since a supply chain is a network of facilities, each agent can be assigned to model a facility, and relationships can be defined as links to connect these agents. An approach that considers the supply chain by means of agents and simulation is very well-suited to analyze these systems. Compared to analytical techniques, agent simulation provides the flexibility to model the complexities and dynamics of real world supply chains without having to perform excessive simplifying assumptions [7]. However, the simulation approach can't guarantee optimality.

## 2.3 Holonic Agents

MAS has become a natural tool for modeling and simulating complex systems. However, in those systems there usually exist a great number of entities interacting among themselves, and acting at different levels of abstraction. In this context, it seems unlikely that MAS will be able to faithfully represent complex systems without multiple granularities. That's why holonic systems have attracted the attention of researchers [4]. The term *holon* was coined by Arthur Koestler [5], based on the Greek word *holos* for whole and *on* for part. Thus, a *holon* is a self-similar or fractal structure that consists of several *holons* as components, and is itself a part of a greater whole. It should meet three conditions: to be stable, to be autonomous and to be able of cooperating. Gerber et al. [3] propose three types of structures for *holons*, which vary with respect to the autonomy of the members. The moderated group is the intermediary structure, which was chosen for this work due to its greater flexibility. It specifies a holonic organization with two main roles: *head* and *part*. The *head* is a kind of moderator of the *holon*, represents its shared intentions and negotiates them with agents outside the *holon*, as well as with those inside

the *holon*. Only the *head* can communicate with the outer world. On the other hand, the *part* is any other member of the *holon*. Figure 1 outlines the holonic organization of this work.

## 2.4 Distributed Constraint Optimization Problem (DCOP)

DCOP is a formalism that can model problems distributed due to their nature. These are problems where agents try to find assignments to a set of variables that are subject to constraints. The reason for the distribution of the solving process comes from the assumption that only a subset of the agents has knowledge of each given constraint [10]. Unlike agent-based simulation, this approach allows global optimization. It is assumed that agents maximize their cumulated satisfaction by the chosen solution. This is different from other related formalisms involving self-interested agents, which try to maximize their own utility individually. Thus, the agents can optimize a global function in a distributed fashion communicating only with neighboring agents, and even in an asynchronous way. That framework has proved to be naturally suitable to model different types of complex combinatorial problems, including the oil industry domain [6]. A DCOP can be formalized as a tuple  $(A, V, D, C, F)$  [8]:

- $A = a_1, a_2, \dots, a_n$  is a set of  $n$  agents,
- $V = x_1, x_2, \dots, x_n$  is a set of  $n$  variables, each one associated with an agent,
- $D = D_1, D_2, \dots, D_n$  is a set of finite and discrete domains each one associated with the corresponding variable,
- $C = f_{ij} : D_i \times D_j \rightarrow N$ , with  $i, j = 1..n$ ,  $i \neq j$  is a set of constraints, represented by a cost function  $f_{ij}$  for each pair of variables  $x_i$  and  $x_j$ ,
- $F = \sum F_{ij}(d_i, d_j)$ , where  $x_i \leftarrow d_i, x_j \leftarrow d_j$ ,  $x_i, x_j \in V$ , is the objective function.

Only the agent has knowledge and control over values assigned to variables associated with it. The agents' goal is to choose a valuation for all variables in order to minimize or maximize the objective function  $F$ , which is modeled as a set of valued constraints.

## 3 Problem Description

This work focuses on an oil industry supply chain, which starts at the crude oil extraction, and finishes when its derivative products are delivered to distribution companies, which are considered here as final customers. Thus, all activities that occur before oil extraction, such as exploration and drilling, as well as those that occur after delivery to distribution companies, such as the delivery of products to the gas stations, are out of the scope of this work. The supply of crude oil and its derivatives must be made preferentially by the oil company itself, which may be a single verticalized petroleum enterprise or a set of cooperating companies of the oil business. Henceforth it will be called extended enterprise (EE) the general situation that comprises both cases. If necessary or eligible, the EE can purchase from the spot market (SM), which satisfies any extra demands of crude oil and its derivatives at higher prices. In the same way, SM can buy any exceeding inventories of those items at lower prices. The EE operation area is spread geographically, and this physical space is visualized as a partition of regions, which are in their turn grouped into continents, which finally are gathered into a global area. A region is divided into trading areas or oil extraction areas. A facility, i.e., a refinery or distribution terminal, is responsible for each of the first

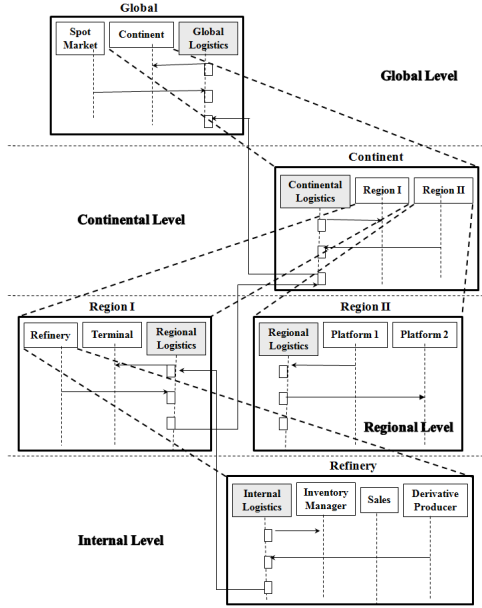


Figure 1. Holonic organization with the head agents highlighted

ones, which serves specific final customers (distributing companies). On the other hand, each of the latter ones is composed of oil extraction platforms, that yield a certain type of crude oil at a rate that may be considered constant for the time scale of this work.

All the areas are connected by transportation modals, like ships and pipelines. Furthermore, each area owns a specific logistic entity, which is responsible for the transportation planning of crude oil and its derivatives, which, henceforward, will be named just products. Each transportation modal has a set of routes, which connect two entities of the chain (platforms, refineries, terminals, and SM) and has a maximum transportation capacity, which is ignored in this work for the sake of simplicity. If the connected entities belong to a same region, that transport is managed by its regional logistics. Otherwise, the transport is managed by the continental logistics or even the global one, depending on the range of the particular route. Since the facilities are connected through a transportation network, they can cooperate with each other to supply the different trading areas, instead of being restricted to their own ones.

A refinery can produce multiple derivatives, and it does so according to different production plans, which are characterized by processing a definite quantity of a particular type of crude oil and producing a certain quantity of each resulting derivative. In addition, each facility is responsible for the management of its inventories of each product, but platforms can't store the extracted crude oil, which must be drained continuously by ships or pipelines. Thus, each facility should choose and follow a specific inventory policy for each product. Figure 2 illustrates an example of the described oil supply chain.

## 4 Proposed Model

### 4.1 Introduction

The model proposes a multi-period time approach. Thus the supply chain system evolves over a given time horizon  $H$ , which is divided into periods  $P_\tau, \tau \in \{1, \dots, T\}$ , which, from now on, will be represented by their index  $\tau$ . At the beginning of each period  $\tau$ , decisions

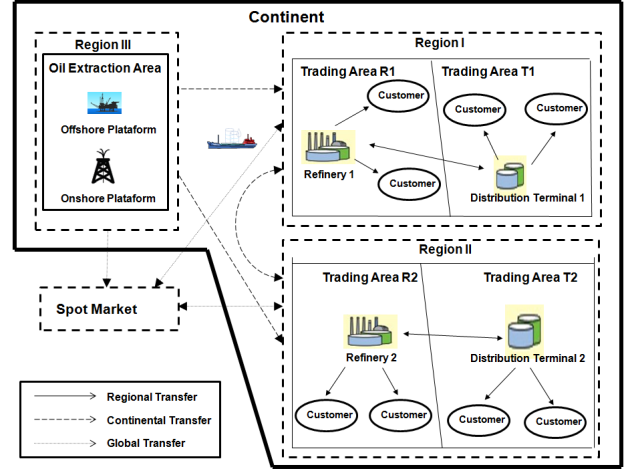


Figure 2. Example of an oil supply chain scenario

are made based on past and future decisions. The sequence of such decisions along  $H$  will be called a decision strategy ( $S$ ), and a decision means the assignment of values to all the decision variables, which are described next.

#### 4.1.1 Decision Variables

A refinery can produce multiple derivatives, and it does so according to different production plans  $plan_{r\tau}$ , which are chosen from a discrete set generated by the production planner of each refinery  $r$  at each period  $\tau$ . On the other hand, the inventory management of each product  $p$  is treated by choosing the inventory  $Inv_{fp\tau}$  between a minimum and a maximum level for each facility  $f$  at each period  $\tau$ . The first represents the safety inventory, and the latter the physical capacity limit. The inventory policy to be adopted must guarantee that the corresponding inventory level will be equal to  $Inv_{fp\tau}$  at the end of the period  $\tau$ , by replenishing or draining the required product quantity. Thus, the decision variables of the model are :

- $plan_{r\tau}$ : production plan adopted by refinery  $r$  during period  $\tau$ ;
- $Inv_{fp\tau}$ : inventory level of product  $p$  (derivative or type of crude oil) in facility  $f$  at the end of period  $\tau$ ;

where  $\tau \in \{1, \dots, T\}$ ,  $d \in D$  (set of crude derivatives),  $o \in O$  (set of types of oil crude),  $p \in P = D \cup O$ ,  $f$  is a facility (refinery  $r$  or terminal  $t$ ).

#### 4.1.2 Input Parameters

- $In(r, o, plan_{r\tau})$ : quantity of crude oil  $o$  processed in refinery  $r$ , using production plan  $plan_{r\tau}$  during period  $\tau$ ;
- $Out(r, d, plan_{r\tau})$ : quantity of derivative  $d$  produced in refinery  $r$ , using production plan  $plan_{r\tau}$  during period  $\tau$ ;
- $Dm_{fd\tau}$ : total expected demand of derivative  $d$  in the trading area of facility  $f$  during period  $\tau$ ;
- $extRate_{\phi o}$ : quantity of crude oil  $o$  extracted in platform  $\phi$  during a time period;
- $extCost_{\phi o}$ : extraction cost of a unit of crude oil  $o$  in platform  $\phi$ ;
- $prdCost_{rd}$ : production cost of a unit of derivative  $d$  in refinery  $r$ ;
- $prc_p$ : price of a unit of product  $p$  in EE's market;

- $SMbuyPr_{cp}$ : price at which SM buys a unit of product  $p$ ;
- $SMsellPr_{cp}$ : price at which SM sells a unit of product  $p$ ;
- $InvCost_{fp}(Inv_{fp\tau})$ : inventory cost function of inventory level  $Inv_{fp\tau}$  for product  $p$  in facility  $f$  during period  $\tau$ ;
- $RfRate_{Rp}$ : freight rate for product  $p$  in region  $R$ ;
- $CfRate_{Cp}$ : freight rate for product  $p$  between regions in continent  $C$ ;
- $GfRate_p$ : freight rate for product  $p$  among continents and SM;

In this work it is proposed a framework based on holonic agents and distributed constraint optimization to model the oil supply chain. The model is divided into two submodels: organization model, and constraint optimization model. The latter gives rise to the distributed constraint optimization model, which is the one actually considered in this work. These models are described in the sections 4.2, 4.3 and 4.4.

## 4.2 Organization Model

As a result of the features of a holonic system, its organization model can be represented in two dimensions: a *holonic organization*, which is common to all the holonic systems, and an *internal organization*, which is specific to a problem domain [4]. Thus, a same agent can have a role in the holonic organization, and another role in the internal organization. The holonic organization is composed basically of the roles *head* and *part*, whereas, in general, the internal organization comprises more roles, which are associated with physical entities, services and functions of the supply chain. In addition to these specific roles of the internal organization, two generic roles are shared by several of them. The first is the *supplier* role, which refers to any entity that provides products to other entity of the chain. The other is the *client* role, which is played by any entity which needs these products. Figure 3 presents the UML class diagram which represents the roles of the internal organization and the relationships between them. Thus, the *refinery* is modeled as a *terminal* with a *derivative producer*, since both of them own a *sales* and an *inventory manager* area, and an *internal logistics*. The *logistics* is the *head* of the respective holon in all levels of the chain, i.e., the *internal logistics* for the holon *refinery* or *terminal*, the *regional logistics* for the holon *region*, and so on, up to the *global logistics* for the holon *global*. Each *logistics* is responsible for the balance of products between the *suppliers* and the *clients*. The *external logistics*, which represents all the logistics roles, but the internal one (operates inside a facility), is responsible for the transportation planning as well. As to the *production planner* role, it is played inside the refinery by a software or a team of human experts, which support the *derivative producer* role. The *sales* role, in turn, is responsible for the forecast of the total demand for each derivative at the corresponding trading area. Figure 4 shows an UML collaboration diagram, which represents the relationships between the roles in the organization model.

The holonic agent architecture allows a complicated process to be decomposed into smaller processes, which are further decomposed, until each of them can be handled by a single agent. The *header* agent acts as a mediator, which supports communication and control for each level of process decomposition. Although it would be possible a smaller granularity, with agents representing operational units of the refinery, in this work that “descent” comes to an end at the *derivative producer*, which puts up a discrete set of production plans, that are obtained from the *production planner*.

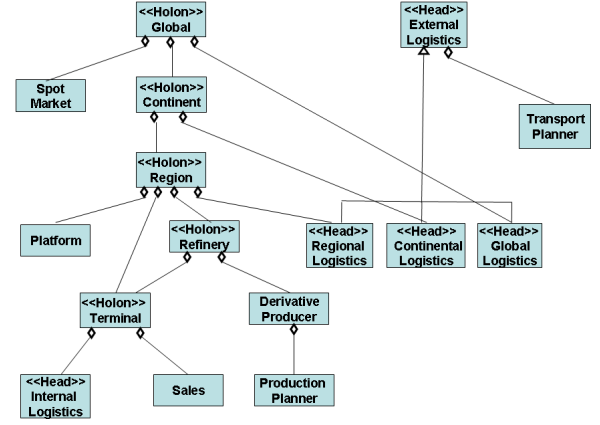


Figure 3. Class diagram of the organization model

## 4.3 Constraint Optimization Model

A constraint network is a declarative structure, which expresses relations and tolerances among entities. It consists of a number of nodes that are connected by constraints. Multi-agent is one of the approaches that can be used to model and implement constraint networks. Therefore, agent technology and constraint network can be integrated together to support coordination and sharing of information among facilities in supply chains. When the relationships between agents are modeled by constraints, they can be integrated as a network, and global optimization becomes possible. Each agent can be assigned to model a facility and relationships can be defined as constraints between these agents. These relationships pertain to how the products flow from suppliers to clients, and can be represented in quantities of products, in cost or scheduling of deliveries, etc. The environment imposes limitations to the supply chains, which have their significant variables assigned to values within finite domains. As a result, the constraint model considers a finite set of variables, which are associated with a finite domain, and a set of constraints that restrict the values which the variables can simultaneously take.

### 4.3.1 Constraints

The constraints of the model are expressed in terms of the decision variables and the input parameters. However, to make it easier, the model employs dependent variables, which are defined as follows:

- $\Delta_{\alpha p \tau}$ : the difference between output and input quantities of product  $p$  with respect to the entity  $\alpha$  during period  $\tau$ ;
- $\Delta_{\alpha p \tau}^+ = \begin{cases} \Delta_{\alpha p \tau}, & \text{if } \Delta_{\alpha p \tau} \geq 0 \\ 0, & \text{if } \Delta_{\alpha p \tau} < 0 \end{cases}$
- $\Delta_{\alpha p \tau}^- = \begin{cases} -\Delta_{\alpha p \tau}, & \text{if } \Delta_{\alpha p \tau} \leq 0 \\ 0, & \text{if } \Delta_{\alpha p \tau} > 0 \end{cases}$

Using all the variables and parameters, the constraints are presented for each level of the organizational model.

#### i. Internal Level

$$\begin{aligned} \Delta_{rd\tau} &= Out(r, d, plan_{r\tau}) - Dm_{rd\tau} + Inv_{rd\tau-1} - Inv_{rd\tau} \\ \Delta_{td\tau} &= Inv_{td\tau-1} - Inv_{td\tau} - Dm_{td\tau} \\ \Delta_{ro\tau} &= Inv_{ro\tau-1} - Inv_{ro\tau} - In(r, o, plan_{r\tau}) \\ \Delta_{\varphi o \tau} &= extRate_{\varphi o} \end{aligned}$$

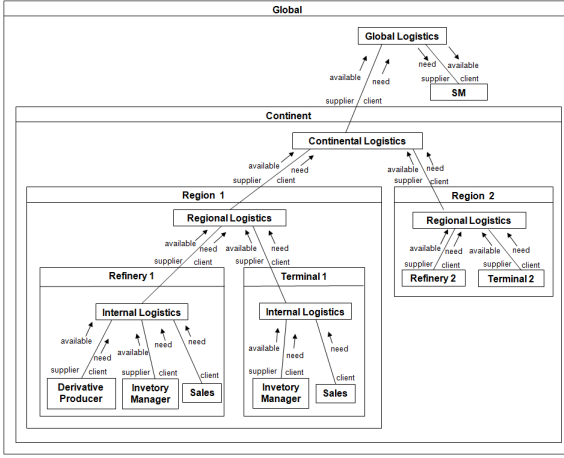


Figure 4. Collaboration diagram of the organization model

## ii. Regional Level

$$\Delta_{Rp\tau} = \sum_{f \in R} \Delta_{fp\tau}$$

## iii. Continental Level

$$\Delta_{Cp\tau} = \sum_{R \in C} (\Delta_{Rp\tau}^+ - \Delta_{Rp\tau+1}^-)$$

## iv. Global Level

$$\begin{aligned} \Delta_{Gp\tau} &= \sum_C (\Delta_{Cp\tau}^+ - \Delta_{Cp\tau+1}^-) \\ \Delta_{Gp\tau}^+ - \Delta_{SMp\tau+2}^- &= 0 \\ \Delta_{SMp\tau}^+ - \Delta_{Gp\tau+2}^- &= 0 \end{aligned}$$

It is important to point out the relationship between past and future, which appears in the continental and global levels. It comes from the physical time constraints associated with the transport of products between the entities of the chain. In fact, it is assumed in this work that the products take about one time period to be transported between different regions inside a continent (continental level), whereas they take about two time periods to travel between two continents, or from a continent to SM (global level). As to the regional level, it is assumed that two different facilities of a same region are close enough to move their products between each other within a single time period.

Since the time period is limited, i.e.,  $\tau \in \{1, \dots, T\}$ , it is necessary to take into account the boundary conditions  $\Delta_{RpT+1}^- = 0$ ,  $\Delta_{CpT+1}^- = \Delta_{CpT+2}^- = 0$ ,  $\Delta_{GpT+1}^- = \Delta_{GpT+2}^- = 0$ ,  $\Delta_{SMpT+1}^- = \Delta_{SMpT+2}^- = 0$ , and  $Inv_{fpT-1} = cte$ .

### 4.3.2 Objective Function

The supply chain manager's motivation is to choose a decision strategy  $S(plan_{\tau\tau}, Inv_{fp\tau})$  such that the chosen performance measure

is the best possible. In this work, the adopted performance measure is the profit of the entire supply chain during H. However, one of the basic assumptions of the model is that all the customer demands must be supplied. That is always possible due to the flexibility offered by SM, which can buy or sell unbounded quantities of any product. Therefore, the objective is to maximize the total profit with the customer satisfaction automatically guaranteed.

The total profit can be expressed as :

$$\begin{aligned} Profit(S) &= Income(S) - Cost(S) \\ Income(S) &= COInc(S) + DPInc(S) \\ Cost(S) &= COCost(S) + DPCost(S) + \\ &\quad + Freight(S) + InvCost(S) \end{aligned} \quad (1)$$

$COInc$  is the income obtained by selling the surplus crude oil to SM;  $DPInc$  comes from the sale of all the derivative products, internally and to SM as well;  $COCost$  is the cost resulting from crude extraction or its purchase from SM in order to feed the refineries;  $DPCost$  is the total cost to produce the derivatives in the refineries, and the prices paid to SM when a purchase is necessary to supply the whole demand of customers;  $Freight$  is the total freight of the transportation of crude and derivatives, and is obtained by the sum of the freights of transfer of products inside each region ( $Rf_{rt_{Rp\tau}}$ ), between different regions in a same continent ( $Cf_{rt_{Cp\tau}}$ ), and among continents and SM ( $Gf_{rt_{p\tau}}$ ); finally,  $InvCost$  is the total inventory cost to store derivatives and crudes during H.

$$\begin{aligned} COInc(S) &= \sum_{\sigma\tau} SMbuyPrc_o \cdot \Delta_{Go\tau}^+ \\ DPInc(S) &= \sum_{fd\tau} prc_d \cdot Dm_{fd\tau} + \sum_{d\tau} SMbuyPrc_d \cdot \Delta_{Gd\tau}^+ \end{aligned} \quad (2)$$

$$\begin{aligned} COCost(S) &= \sum_{\sigma\tau} SMsellPrc_o \cdot \Delta_{Go\tau}^- + \sum_{\phi o\tau} extCost_{\phi o} \cdot \Delta_{To\tau} \\ DPCost(S) &= \sum_{rd\tau} prdCost_{rd} \cdot Out(r, d, plan_{r\tau}) + \\ &\quad + \sum_{d\tau} SMsellPrc_d \cdot \Delta_{Gd\tau}^- \\ InvCost(S) &= \sum_{fp\tau} InvCost_{fp}(Inv_{fp\tau}) \end{aligned} \quad (3)$$

$$\begin{aligned} Freight(S) &= \sum_{Rp\tau} Rf_{rt_{Rp\tau}} + \sum_{Cp\tau} Cf_{rt_{Cp\tau}} + \sum_{p\tau} Gf_{rt_{p\tau}} \\ Rf_{rt_{Rp\tau}} &= \begin{cases} \sum_{f \in R} \Delta_{fp\tau}^+ \cdot RfRate_{Rp}, & \text{if } \Delta_{Rp\tau} < 0 \\ \sum_{f \in R} \Delta_{fp\tau}^- \cdot RfRate_{Rp}, & \text{if } \Delta_{Rp\tau} \geq 0 \end{cases} \\ Cf_{rt_{Cp\tau}} &= \begin{cases} \sum_{R \in C} \Delta_{Rp\tau}^+ \cdot CfRate_{Cp}, & \text{if } \Delta_{Cp\tau} < 0 \\ \sum_{R \in C} \Delta_{Rp\tau}^- \cdot CfRate_{Cp}, & \text{if } \Delta_{Cp\tau} \geq 0 \end{cases} \\ Gf_{rt_{p\tau}} &= \begin{cases} \left( |\Delta_{Gp\tau}| + \sum_C \Delta_{Cp\tau}^+ \right) \cdot GfRate_p, & \text{if } \Delta_{Gp\tau} < 0 \\ \left( |\Delta_{Gp\tau}| + \sum_C \Delta_{Cp\tau}^- \right) \cdot GfRate_p, & \text{if } \Delta_{Gp\tau} \geq 0 \end{cases} \end{aligned} \quad (4)$$

### 4.4 Distributed Constraint Optimization Model

From the previous constraint model, it is possible to derive a distributed constraint model, which employs the same parameters, decision and dependent variables of the first one. Basically, whatever the role in the organization model, but the *logistic* ones, one agent is created for each tuple  $(p, \tau)$ . On the other hand, a single agent exists per

logistic entity in the case of logistic roles. Since the logistic agents of all the levels have also the head role, they are responsible for the computation of the dependent variables, which are updated according to the decision variables assignments and sent upward from a logistic agent to the other located in the upper level. Furthermore, the appropriate match between the correct time periods while balancing availabilities of *suppliers* and needs of *clients* in the continental and global levels is also a responsibility of the corresponding logistic agent. The agents of the *derivative producer* role manage the decision variables  $plan_{r\tau}$ , whereas the agents of the *inventory manager* role take care of the decision variables  $Inv_{fp\tau}$ .

The objective function represents the total profit, and therefore is to be maximized. It is written as  $F = \sum F_{L:\alpha}$ , where the valued constraint functions  $F_{L:\alpha}$  are defined as relationships between two agents (neighbors). In the focused holonic model, one agent of the couple has always one of the logistic roles ( $L$ ), whereas the other can play any of the other roles ( $\alpha$ ). Each of these functions is presented next, along with the pair of neighbor agents, and the variables exchanged by them, as well as the parameters, since these latter can change dynamically according to the environment. In case of a holon, it is followed by its head agent (symbol  $\equiv$ ), which represents the entire holon in the communication. All the variables act directly on the objective function, whereas some parameters do so and others not ( $F_{L:\alpha} = 0$ ).

#### 4.4.1 Constraints

##### i. Internal Level

$$Sales_{fd\tau} \xrightarrow{Dm_{fd\tau}} Internal\ Logistics_f$$

$$F_{L_f:S_{fd\tau}} = prc_d \cdot Dm_{fd\tau}$$

$$Inventory\ Manager_{fp\tau} \xrightarrow{Inv_{fp\tau}} Internal\ Logistics_f$$

$$F_{L_f:M_{fp\tau}} = -InvCost_{fp}(Inv_{fp\tau})$$

$$Derivative\ Producer_{ro\tau} \xrightarrow{In(r,o,plan_{r\tau})} Internal\ Logistics_r$$

$$F_{L_r:D_{ro\tau}} = 0$$

$$Derivative\ Producer_{rd\tau} \xrightarrow{Out(r,d,plan_{r\tau})} Internal\ Logistics_r$$

$$F_{L_r:D_{rd\tau}} = -prdCost_{rd} \cdot Out(r,d,plan_{r\tau})$$

##### ii. Regional Level

$$Facility_{fp\tau} \equiv Internal\ Logistics_f \xrightarrow{\Delta_{fp\tau}} Regional\ Logistics_R$$

$$F_{L_R:F_{fp\tau}} = -RfRate_{Rp} \cdot \Delta_{fp\tau}^+$$

$$Platform_{\varphi o\tau} \xrightarrow{\Delta_{\varphi o\tau}} Regional\ Logistics_R$$

$$F_{L_R:P_{\varphi o\tau}} = -(RfRate_{Ro} + extCost_{\varphi o}) \cdot \Delta_{\varphi o\tau}^+$$

##### iii. Continental Level

$$Region_{Rp\tau} \equiv Regional\ Logistics_R \xrightarrow{\Delta_{Rp\tau}} Continental\ Logistics_C$$

$$F_{L_C:R_{Rp\tau}} = RfRate_{Rp} \cdot \Delta_{Rp\tau} - CfRate_{Cp} \cdot \Delta_{Rp\tau}^+$$

##### iv. Global Level

$$Continent_{Cp\tau} \equiv Continental\ Logistics_C \xrightarrow{\Delta_{Cp\tau}} Global\ Logistics$$

$$F_{L_G:C_{Cp\tau}} = CfRate_{Cp} \cdot \Delta_{Cp\tau} - GfRate_p \cdot \Delta_{Cp\tau}^+$$

$$SM_{p\tau} \xrightarrow{SMsellPrc_p, SMbuyPrc_p} Global\ Logistics$$

$$F_{L_G:SM_{p\tau}} = -GfRate_p \cdot \Delta_{SM_{p\tau}}^+ - SMsellPrc_p \cdot \Delta_{SM_{p\tau}}^+ + SMbuyPrc_p \cdot \Delta_{SM_{p\tau}}^-$$

## 5 Conclusion

In this paper, we have developed a theoretical model that combines a constraint network approach and a holonic multi-agent approach to support coordination and management of a typical oil industry supply chain. Agents capture the distributed behavior of the supply chain, and the holonic approach makes it possible to represent its intrinsic recursive nature. On the other hand, the constraint network model allows a tight integration of the chain entities, which are spread in space and time, and allows optimization. In a usual simulation approach, each decision is based only on past decisions, but this model integrates both the past and the future, and allows an actual optimization.

The proposed model is divided into an organization and a constraint model. The first consists of a holonic organization of the kind moderated group, and an internal organization suited to the oil supply chain domain. The constraint model is refined to a distributed constraint model, which is the one to be considered.

In a future work the model will be extended to include transport constraints and the transport planning by the distribution of product transfers among the available routes. The responsibility for these tasks will be assigned to the corresponding logistic agent. The inclusion of such constraints will restrict the search space to be explored, aiming at the optimization. The process of negotiation with the *production planner* will also be detailed within the *derivative producer* of each refinery, aiming at the choice of the eligible best production plans for the decision variable domain. Finally, it will be developed a prototype based on a case study, as a proof-of-concept to validate the proposed model, which will be compared with actual historical data.

## References

- [1] B. M. Beamon, 'Supply Chain Design and Analysis : Models and Methods', *International Journal of Production Economics*, (1), 1–22, (1998).
- [2] C. M. Chima and D. Hills, 'Supply-chain management issues in the oil and gas industry', *Journal of Business*, 5(6), 27–36, (2007).
- [3] C. Gerber, J. Siekmann, and G. Vierke, 'Holonc multi-agent systems', *Research Report*, 99(3), (1999).
- [4] V. Hilaire, A. Koukam, and S. Rodriguez, 'An Adaptive Agent Architecture for Holonic Multi-Agent Systems', *ACM Transactions on Autonomous and Adaptive Systems*, 3(1), (2008).
- [5] A. Koestler, *The Ghost in the Machine*, Hutchinson 'I&' Co, London, 1st edn., 1967.
- [6] F. J. M. Marcellino, N. Omar, and A. V. Moura, 'The Planning of the Oil Derivatives Transportation by Pipelines as a Distributed Constraint Optimization Problem', in *IJCAI-DCR 2007*, Hyderabad, India, (2007).
- [7] F. D. Mele, G. Guillén, A. Espuña, and L. Puigjaner, 'A Simulation-Based Optimization Framework for Parameter Optimization of Supply-Chain Networks', *Industrial&Engineering Chemistry Research*, 45(9), (2006).
- [8] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo, 'Adopt: Asynchronous distributed constraint optimization with quality guarantees', *Artificial Intelligence*, 161, 149–180, (2006).
- [9] J. F. Shapiro, *Modeling the Supply Chain*, Duxbury Press, Pacific Grove CA, 2006.
- [10] M. C. Silaghi and M. Yokoo, 'ADOPT-ing : unifying asynchronous distributed optimization with asynchronous backtracking', *Autonomous Agent Multi-Agent Systems*, 19, 89–123, (2009).
- [11] J. M. Swaminathan, S. F. Smith, and N. M. Sadeh, 'Modeling Supply Chain Dynamics : A Multiagent Approach', *Decision Sciences*, 29(3), 607–632, (1998).

# Ensuring Privacy through Distributed Computation in Multiple-Depot Vehicle Routing Problems

Thomas Léauté and Brammert Ottens and Boi Faltings<sup>1</sup>

**Abstract.** The *Vehicle Routing Problem (VRP)* has been extensively studied over the last twenty years, because it is an abstraction of many real-life logistics problems. In its *multiple-depot* variant (MDVRP), the routes of vehicles located at various depots must be optimized to serve a number of customers. In this paper, we investigate how to protect the privacy of delivery companies, when each depot is owned by a different company with a limited view of the overall problem. Companies then need to exchange messages with each other to coordinate the assignment of customers to depots. We show how *Distributed Constraint Optimization (DCOP)* can be used to solve the assignment problem using distributed computation, and we study the guarantees that can be provided with respect to the protection of each company’s knowledge about the problem.

## 1 INTRODUCTION

Consider a delivery company that has subcontracted the delivery of packets to several independent contractors. These contractors are all interested in maximizing their own profit, while the global delivery company wants to serve its customers as efficiently as possible. Using a centralized approach to assign customers to the different contractors would involve the disclosure of sensitive information on the side of these contractors, which in general is not something that they are eager to accept. One could resort to the use of auctions to allocate customers to different contractors, but auctions are, in general, not able to give any guarantees on the quality of the solution found.

This paper proposes a distributed optimization approach that is able to find an efficient allocation of customers, while retaining a certain level of privacy for the different contractors. This approach makes use of the fact that contractors are generally geographically separated, meaning that not every contractor will be able to serve every customer, i.e. a contractor will not have to coordinate with every other contractor, greatly reducing the complexity of the problem.

The formalism used in the approach presented in this paper is the *Distributed Constraint Optimization (DCOP)* framework [21]. The DCOP framework is used to model loosely coupled multi-agent optimization problems, where the knowledge of the problem is distributed over different agents. Several algorithms exist that are able to solve DCOP problems. These algorithms can be sub-divided into two main families. One family of algorithms uses a form of distributed search to find an optimal solution (such as SynchBB [21], ADOPT [31], or AFB [16]), while the other family of algorithms is based on dynamic programming and uses inference to solve the problem (like DPOP [34], O-DPOP [36], or ASO-DPOP [32]).

We use the DCOP framework to address the problem of assigning customers to depots in a distributed variant of the *Multiple-Depot Vehicle Routing Problem (MDVRP)*. The *Vehicle Routing Problem (VRP)* and its numerous variants have been the subject of much research in the past few decades [47, 15]. The main purpose of this paper is to show the merits of distributed optimization in solving MDVRPs where each depot is controlled by a separate company. Special emphasis is put on protecting the privacy of the different contractors.

The rest of the paper is structured as follows. Section 2 first presents related work on the VRP and DCOPs. Section 3 then defines the problem, shows how it can be formulated as a DCOP, and solved using the DPOP algorithm. Section 4 discusses the privacy issues in DCOPs, and describes the P-DPOP algorithm that was designed to address these issues. Section 5 presents some other, *asynchronous* DCOP algorithms, and Section 6 compares algorithm performances based on benchmark problems. Section 7 finally concludes.

## 2 RELATED WORK

Section 2.1 first summarizes some of the previous work on solving the Multiple-Depot Vehicle Routing Problem. Section 2.2 then reviews the literature on Distributed Constraint Optimization.

### 2.1 The Multiple-Depot Vehicle Routing Problem

The *Vehicle Routing Problem* [47] is a well-known NP-hard problem in Operations Research: optimal routes must be designed for a fleet of vehicles so as to serve customers while satisfying certain constraints, e.g. on capacity or on route length. Variants of the problem have been proposed [15]; this paper focuses on the *multiple-depot* version.

Laporte *et al.* [23] proposed the first complete MDVRP algorithm, based on integer linear programming. Most algorithms proposed later are approximate heuristics that use a two-phase approach: 1) assign customers to depots, and 2) solve the resulting VRP. One of the first such heuristics was introduced by Chao *et al.* [6]. Other work proposed to use tabu search [40, 8], genetic clustering [46], large neighborhood search [39], or hybrid genetic algorithms [22].

Previous work also applied *Multi-Agent Systems* to solve various flavors of the VRP [9]. Some of this work is not truly applicable to the Distributed MDVRP we consider in this paper, because it uses *virtual* agents, simulated on a central computer that knows the whole problem; this includes memetic algorithms [4], or ant colony optimization [41]. A more relevant thread of research applied the *Contract Net Protocol* [42, 13] to solve VRPs in a truly distributed manner.

This paper proposes a comparable approach, but uses Distributed Constraint Optimization to solve the distributed problem of assigning customers to depots, and focuses on *privacy*. For the VRP part of the problem, we use an off-the-shelf, centralized algorithm.

<sup>1</sup> Artificial Intelligence Laboratory (LIA), École Polytechnique Fédérale de Lausanne (EPFL), Switzerland – email: [firstname.lastname@epfl.ch](mailto:firstname.lastname@epfl.ch)

## 2.2 Distributed Constraint Optimization

Distributed Constraint Optimization is used to model multi-agent optimization problems, in which problem knowledge is *distributed*: initially no agent has a global overview of the overall problem.

**Definition 1 (DCOP)** A discrete Distributed Constraint Optimization Problem is defined as a tuple  $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$ , where:

- $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$  is a set of agents;
- $\mathcal{X} = \{x_1, \dots, x_n\}$  are variables, each owned by an agent;
- $\mathcal{D} = \{D_1, \dots, D_n\}$  is a set of finite domains for the variables such that  $x_i$  takes values in  $D_i$ ;
- $\mathcal{C} = \{c_1, \dots, c_m\}$  is a set of soft constraints, where each constraint is a function  $c_i : D_{i_1} \times \dots \times D_{i_l} \rightarrow \mathbb{R}$  that defines a cost for each combination of assignments to a subset of variables.

A solution to the DCOP is an assignment to all variables that minimizes the sum of all costs  $\sum_i c_i$ .

Hard constraints that allow or disallow specific variable assignments can be modeled as soft constraints, using large positive costs. Any agent knows all the constraints expressed over at least one of its variables, but does not know any constraint that it is not involved in.

The DCOP formalism has been applied to a wide variety of problem domains, such as distributed meeting scheduling, sensor networks, or combinatorial auctions. Ottens *et al.* [32] also applied it to a distributed vehicle pickup and delivery problem.

Most DCOP algorithms are complete and based on distributed search. Some use a linear ordering of the variables (SynchBB [21], OptAPO [30], AFB [16]). Others use a *pseudo-tree* ordering to better exploit problem structure (ADOPT [31], DPOP [34], NCB [7]). DPOP (Section 3.2.2) is the only one based on dynamic programming instead of search. Because DCOPs are NP-hard, approximate algorithms have also been proposed (DSA [49], MGM-2 [29]).

Several metrics can be used to assess the performances of complete DCOP algorithms. Since they are message-passing algorithms, two important metrics are the number of messages exchanged, and the total amount of information exchanged (as message sizes may vary significantly from one algorithm to another). Runtime can be estimated in terms of wall clock time (if each agent has a dedicated processor), number of *Non-Concurrent Constraint Checks* [17], or using *simulated time* [45].

Finally, the framework of *DCOP under stochastic uncertainty* [25] was recently proposed to model multi-agent optimization problems involving uncertain, stochastic problem data. This extended formalism can be applied for instance to distributed VRPs in which customer demands are stochastic.

## 3 PROBLEM DEFINITION

Section 3.1 first formally defines the problem; Section 3.2 then describes how it can be reformulated into a DCOP.

### 3.1 Distributed Multiple-Depot VRP

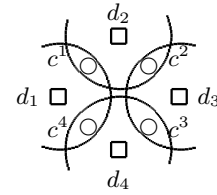
The Distributed Multiple-Depot VRP is simply a distributed version of the Multiple-Depot VRP, in which each depot is controlled by a specific company. The problem is defined more formally as follows.

**Definition 2 (DisMDVRP)** The Distributed, Multiple-Depot, Vehicle Routing Problem consists of a set  $D = \{d_1, \dots, d_{n_D}\}$  of depots in the Euclidian plane, each controlled by a different delivery

company. Each company owns a common number  $n_V$  of vehicles, with a common maximum load  $Q_{\max}$  and a common maximum route length  $L_{\max}$ .<sup>2</sup> The companies must serve a set  $C = \{c^1, \dots, c^{n_C}\}$  of customers at known locations, so that each customer  $c_i$  has a demand  $q_i \in \mathbb{N}$  and must be served by exactly one vehicle. Each vehicle must come back to its initial depot at the end of its route.

In addition to the usual MDVRP constraints, each company has a common visibility radius  $R \leq L_{\max}/2$  that defines the boundaries of its knowledge of the overall DisMDVRP. The company owning depot  $d_i$  is only aware of the customers that are within distance  $R$  of  $d_i$ , and only knows another company if their areas of visibility overlap.

The goal is for the companies to come to an agreement on who should serve which customers, using which vehicle routes, so as to serve all visible customers at minimal total route length.



**Figure 1.** A simple DisMDVRP with four depots and four customers.

Figure 1 shows a DisMDVRP instance with 4 depots and 4 customers. Each depot is at distance 1 of its two closest customers, and owns  $n_V = 2$  vehicles, with  $Q_{\max} = 1$  and  $L_{\max} = 2.5$ , so that a vehicle can only serve one customer, of common demand  $q = 1$ .

The visibility radius is  $R = 1.25$ , such that, for instance, depot  $d_1$  only knows customers  $c^1$  and  $c^4$ , and ignores the existence of  $c^2$  and  $c^3$ . It also knows the two companies  $d_2$  and  $d_4$ , and will have to coordinate with each of them to decide who serves  $c^1$  and  $c^4$ , but  $d_2$  and  $d_4$  do not know about each other, and neither do  $d_1$  and  $d_3$ .

The local nature of the agents' problem knowledge adds a new challenge to the traditional MDVRP. Any algorithm proposed to solve such a distributed, multi-agent problem will necessarily involve exchanging information between the agents, in order to efficiently assign customers to depots. While the performances of traditional MDVRP algorithms are usually compared based on runtime and solution quality, DisMDVRP algorithms should also be evaluated based on the amount of information they require the agents to exchange.

A simplistic approach could consist in the companies electing a leader, sharing with it all their respective information about the problem, and letting it solve the resulting MDVRP in a centralized fashion. This approach would have a number of important drawbacks:

- The companies must be willing to disclose all of their problem knowledge to an elected leader. This is unlikely to be the case in real life: even if they work under the same franchise, delivery companies would want to protect their trade secrets, such as the exact nature of their fleets, or their internal costs.
- This would be computationally very inefficient, since the entire computational burden would lie on the leader. An efficient DisMDVRP algorithm should distribute the effort between the participants in order to leverage their computational power.

<sup>2</sup> These restrictions are only made for simplicity and could be removed without making our approach invalid.

- This would also introduce a bottleneck in network bandwidth usage, since all agents must report their information to the leader.
- The algorithm would have a poor response time to changes in the problem, since the information about the changes would first have to propagate to the leader, which would have to find a new solution to the overall problem, and notify the other companies. A distributed approach should allow agents to directly react to changes, particularly if they only have a local impact.
- The leader becomes a central point of failure of the algorithm.

To solve the distributed part of the DisMDVRP, namely the assignment of customers to depots, we propose to use the DCOP approach, which has been successful at solving such multi-agent optimization problems. While we use a specific flavor of the VRP, this approach also applies to other variants of the problem, as long as they involve the necessity for multiple agents to coordinate their decisions. In particular, the issue of protecting the companies' private knowledge about the problem can become more important in variants of the VRP involving heterogeneous vehicle fleets.

### 3.2 Problem reformulation as a DCOP

Section 3.2.1 first presents how the DisMDVRP can be formulated as a DCOP. Section 3.2.2 then presents the DPOP algorithm.

#### 3.2.1 Formal DCOP model

A DisMDVRP can be reformulated as a DCOP, with one agent per company/depot. Depot  $d_i$  owns one Boolean variable  $x_i^j$  for each visible customer  $c^j$ , modeling its decision to serve the customer ( $x_i^j = 1$ ) or not ( $x_i^j = 0$ ). The DCOP constraints are of two types:

1. For each customer  $c^j$ , if the set  $D^j$  of depots within visible distance  $R$  of  $c^j$  is non-empty, then the following  $|D^j|$ -ary hard constraint enforces that  $c^j$  must be assigned to exactly one depot:

$$\sum_{d_i \in D^j} x_i^j = 1. \quad (1)$$

2. For each depot  $d_i$ , if the set  $C_i = \{c^{j_1}, \dots, c^{j_n}\}$  of visible customers is non-empty, then the following  $|C_i|$ -ary soft constraint represents the cost of the optimal solution to  $d_i$ 's own VRP, as a function of whether it serves each customer in  $C_i$ :

$$vrp_i(x_i^{j_1}, \dots, x_i^{j_n}) = \text{optimal cost of } d_i\text{'s VRP} \in [0, \infty]. \quad (2)$$

Notice that evaluating a constraint  $vrp_i$  on a given assignment of values to its variables requires solving a (centralized) single-depot VRP. Such an evaluation can be seen as a call by agent  $d_i$  to a local subroutine, which does not require exchanging additional information with other agents, and can be implemented using any existing VRP algorithm. This modeling approach decouples the distributed, master problem of assigning customers to agents, solved using a DCOP algorithm, from the local, slave problem of routing a given agent's vehicles, solved using a traditional VRP algorithm.

The DCOP model for the DisMDVRP instance in Figure 1 is illustrated in Figure 2, in the form of a *constraint graph*. The nodes in the graph correspond to the variables in the DCOP, and constraints are represented by edges between the variables in the constraint scope. In this example, each agent (represented by a dotted box) owns two variables, whose respective values model whether or not the company is assigned each of the two customers in its area of visibility. Internally, each agent has to solve a VRP problem, represented by a  $vrp_i$  constraint over its variables. Inter-agent *sum* constraints enforce that each customer be served by exactly one company.

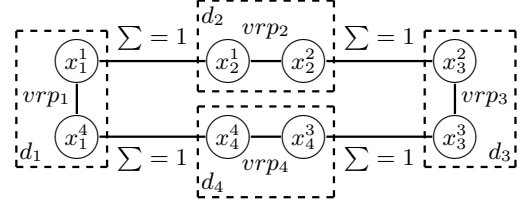


Figure 2. The DCOP constraint graph corresponding to Figure 1.

#### 3.2.2 The DPOP algorithm

DPOP [34] is an instance of the general *bucket elimination scheme* from [10], adapted for the distributed case. It proceeds in four phases: the first two construct a *pseudo-tree* ordering of the variables (i.e. a tree allowing *back edges* between a variable and an ancestor), the third propagates information bottom-up along the pseudo-tree, and the last top-down phase infers optimal values for the variables.

**Leader election:** The agents first run a leader election algorithm to choose one variable as the root of the pseudo-tree ( $x_1^4$  in Figure 3). Each variable is initially assigned a score; the root is the one with the maximum score, computed by viral propagation. Various scoring heuristics have been proposed to generate pseudo-trees with specific good properties; the one that elects the most connected variable tends to produce low-width pseudo-trees.

**DFS traversal:** The root variable then initiates a depth-first traversal of the constraint graph to compute a spanning tree, using a token-passing algorithm described in [11]. The *tree edges* of the resulting pseudo-tree correspond to the edges in the spanning tree, and the back edges to the remaining edges in the constraint graph. The pseudo-tree in Figure 3 consists of a single branch; however, in the more general case, the pseudo-tree may have multiple branches, each executing the remaining two phases in parallel.

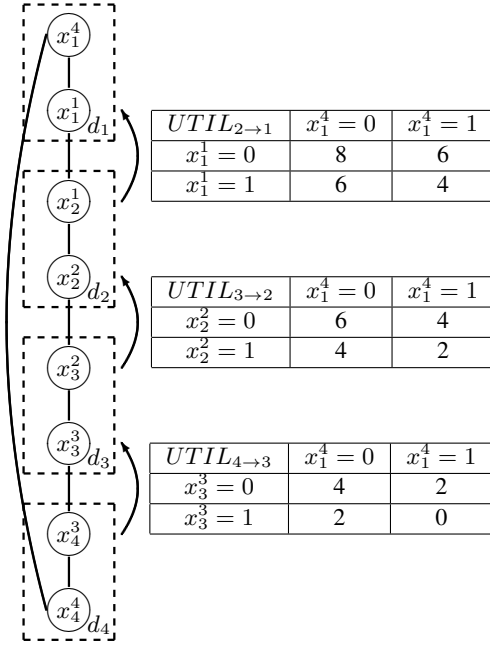
**UTIL propagation:** During this bottom-up phase, each variable sends a *UTIL* message to its parent with the aggregated, optimal cost achievable by its whole subtree, as a function of some ancestor variables. In Figure 3, the message  $UTIL_{3 \rightarrow 2}$  sent by  $d_3$  contains the optimal cost of  $d_4$  and  $d_3$ 's combined VRPs, as a function of whether  $d_2$  serves  $c^2$  and whether  $d_1$  serves  $c^4$ .

**VALUE propagation:** After adding the costs of its local VRP to  $UTIL_{2 \rightarrow 1}$ , agent  $d_1$  can infer the optimal values for its variables that minimize the overall objective function. In the example, due to the symmetries in Figure 1, all combinations of decisions have cost 8. Agent  $d_1$  can choose to serve both customers  $c^1$  and  $c^4$  ( $x_1^1 = x_4^1 = 1$ ); these decisions are then sent downwards in a top-down phase, after which all variables have been assigned optimal values. For instance, agent  $d_2$  can infer from  $d_1$ 's decisions and from  $UTIL_{3 \rightarrow 2}$  that its optimal decisions are  $x_2^1 = 0$  and  $x_4^2 = 1$ .

DPOP's computational bottleneck is in the sizes of the UTIL messages: they are exponential in the width of the pseudo-tree. [34]

## 4 PROVIDING PRIVACY GUARANTEES

In many real-life situations, companies want to keep their problem knowledge private. While the DCOP approach does not require full disclosure to one agent, agents can still make inferences based on received messages. Section 4.1 presents the types of information that can be leaked, and previous work on this issue. Section 4.2 then describes a recent variant of DPOP that improves the state of the art.



**Figure 3.** A possible pseudo-tree for the constraint graph in Figure 2, with the bottom-up UTIL messages exchanged by the agents in DPOP.

#### 4.1 Privacy leaks in DCOP algorithms

Faltings *et al.* [11] defined *semi-private information* as the part of an agent’s knowledge that might be considered private, but that can *inevitably* be leaked by the solution chosen to the DCOP. In Figure 2, assume that, after running a DCOP algorithm,  $d_1$  chooses  $x_1^1 = 0$ . Although the algorithm might not explicitly reveal to  $d_1$  the value that  $d_2$  chose for its variable  $x_2^1$ , agent  $d_1$  can infer it from the constraint  $x_1^1 + x_2^1 = 1$ . In other words, since  $c^1$  must be served by either  $d_1$  or  $d_2$ , if  $d_1$ ’s decision is to not serve  $c^1$ , then  $d_1$  can infer the identity of the agent serving  $c^1$ . However, if three companies can serve a customer, the identity of the assigned agent is no longer semi-private information, and may or may not be protected by the algorithm.

Putting aside semi-private information, Faltings *et al.* [11] also defined four types of privacy that may be desirable, based on four categories of problem knowledge that agents may want to protect.

**Agent privacy:** No agent should discover the existence of agents it does not share constraints with. In Figure 2,  $d_1$  and  $d_3$  should not discover each other’s existence. Linear variable orderings (SynchBB, AFB) violate agent privacy (Table 1): they can require message exchanges between non-neighboring variables. Pseudo-trees (DPOP, ADOPT, NCBB) do not, but some agent privacy can still be leaked. For instance, the presence of variable  $x_1^4$  in the message  $UTIL_{4→3}$  reveals to agent  $d_3$  the existence of agent  $d_1$ .

**Topology privacy:** No agent should discover topological constructs (variables, constraints, cycles...) it is not involved in. Agent  $d_1$  must not learn how many customers  $d_2$  can see. This can be leaked by the variable ordering heuristic: the *most connected* heuristic reveals to all agents the degrees of their neighboring variables.

**Constraint privacy:** No agent should discover the costs in any constraint it is not involved in. DPOP violates constraint privacy: the message  $UTIL_{4→3}$  reveals to agent  $d_3$  the optimal costs of

agent  $d_4$  for serving customers  $c^3$  and  $c^4$ .

**Decision privacy:** If no agent can learn the values taken by another agent’s variables, then the algorithm protects *full decision privacy*. *Partial decision privacy* allows agents to learn the values only of neighboring variables. DPOP violates both, since, in Figure 3, the VALUE propagation phase reveals to all agents the value of  $x_1^4$ .

Most of the literature on privacy in DCOP focuses on constraint privacy. Frameworks have been proposed to measure leaks of constraint privacy [14, 28, 19]. Methods have been developed to partially protect constraint privacy in search-based algorithms [5] and in DPOP [18]. Cryptography has also been applied to fully protect constraint privacy [48, 43], but the algorithms do not scale to realistic problems. The following section describes a more recent algorithm that scales better, while addressing all four types of privacy.

**Table 1.** The privacy guarantees of various complete DCOP algorithms.

|                      | agent   | topology | constraint | decision |
|----------------------|---------|----------|------------|----------|
| SynchBB, OptAPO, AFB |         |          |            |          |
| DPOP, ADOPT, NCBB    | partial | partial  |            |          |
| P-DPOP               | full    | full     | partial    | partial  |
| P <sup>2</sup> -DPOP | full    | full     | full       | full     |

#### 4.2 The P-DPOP algorithm

P-DPOP [11] is a variant of DPOP that guaranties all four types of privacy defined in Section 4.1, through the following techniques:

**Codenames:** To protect agent privacy, P-DPOP obfuscates names and values of back-edge variables using *codenames* (Table 3).

**Anonymous leader election:** To provide topology privacy, agents are allowed to lie a random number of times during the viral propagation of scores, to hide the position of the root. A randomized scoring heuristic is used to avoid leaking topological information.

**Cost obfuscation:** Constraint privacy is leaked by the message  $UTIL_{4→3}$ , which reveals to  $d_3$  the dependency of  $d_4$ ’s optimal cost function on  $x_4^1$ . To obfuscate this dependency,  $d_4$  adds a secret, large number to each column; for instance 1234 for  $x_4^1 = 0$ , and 2345 for  $x_4^1 = 1$  (Table 3). Agents  $d_3$  and  $d_2$  can then perform their operations on the obfuscated costs without decrypting them; only  $d_1$  does so by subtracting the secret numbers.

**Table 3.** Obfuscated message  $UTIL_{4→3}$  in P-DPOP.

|             | $\alpha = \beta$ | $\alpha = \gamma$ |
|-------------|------------------|-------------------|
| $x_3^3 = 0$ | 1238             | 2347              |
| $x_3^3 = 1$ | 1236             | 2345              |

Silaghi *et al.* [44] showed that most DCOP algorithms also violate constraint privacy through their runtime, because it depends on the *tightness* of the constraints, i.e. how soft or hard they are. Since they are based on dynamic programming instead of search, DPOP and P-DPOP do not suffer from this flaw.

Finally, codenames also guarantee *partial* decision privacy. *Full* decision privacy is however violated, since each variable learns the value of its upper neighbors in the pseudo-tree. The P<sup>2</sup>-DPOP algorithm [26] provides a patch to this privacy leak (Table 1), but this comes at a high price in terms of memory and runtime complexity.

**Table 2.** Experimental results for various DCOP algorithms on some of the Cordeau instances from [2]. ADOPT timed out on all instances.

| prob. | $ D $ | $R$ | $ C $ | wall clock time (ms) |                |        |              |         | information exchanged (bytes) |               |              |          |            |
|-------|-------|-----|-------|----------------------|----------------|--------|--------------|---------|-------------------------------|---------------|--------------|----------|------------|
|       |       |     |       | SynchBB              | DPOP           | O-DPOP | ASO-DPOP     | P-DPOP  | SynchBB                       | DPOP          | O-DPOP       | ASO-DPOP | P-DPOP     |
| p01   | 4     | 14  | 30    | <b>535</b>           | 802            | 712    | 850          | 1 437   | 23 900                        | <b>6 525</b>  | <b>6 427</b> | 8 405    | 28 616     |
|       |       | 16  | 32    | <b>1 177</b>         | 2 182          | 1 934  | 2 237        | 4 035   | 53 507                        | <b>7 772</b>  | <b>7 984</b> | 11 405   | 45 612     |
|       |       | 18  | 35    | 9 061                | <b>6 939</b>   | 7 232  | <b>6 881</b> | 11 141  | 992 643                       | <b>10 099</b> | 11 505       | 19 001   | 186 823    |
|       |       | 20  | 38    | -                    | <b>103 153</b> | -      | -            | -       | -                             | <b>22 249</b> | -            | -        | -          |
| p03   | 5     | 10  | 35    | <b>224</b>           | 373            | 385    | 397          | 701     | 7 803                         | <b>1 981</b>  | <b>1 911</b> | 2 338    | 7 731      |
|       |       | 12  | 40    | <b>273</b>           | 636            | 713    | 813          | 1 266   | 8 278                         | <b>2 450</b>  | <b>2 497</b> | 3 375    | 13 225     |
|       |       | 14  | 46    | <b>1 443</b>         | 4 028          | 3 813  | 4 037        | 5 804   | 47 671                        | <b>7 543</b>  | 7 840        | 10 979   | 45 059     |
|       |       | 16  | 58    | <b>55 320</b>        | 99 209         | -      | -            | 268 804 | 10 192 397                    | <b>12 824</b> | -            | -        | 14 603 058 |
| p11   | 5     | 22  | 45    | <b>323</b>           | 846            | 985    | 975          | 1 437   | 8 171                         | <b>1 981</b>  | <b>1 911</b> | 2 338    | 7 867      |
|       |       | 24  | 53    | <b>338</b>           | 5 763          | 6 753  | 6 996        | 8 932   | 9 133                         | <b>2 450</b>  | <b>2 497</b> | 3 232    | 95 051     |
|       |       | 26  | 60    | <b>449</b>           | 19 186         | 45 370 | 43 872       | 33 513  | 9 365                         | <b>2 450</b>  | <b>2 497</b> | 3 435    | 13 055     |
|       |       | 28  | 68    | <b>751</b>           | 91 185         | -      | -            | 165 211 | 12 905                        | <b>3 374</b>  | -            | -        | 59 348     |
|       |       | 30  | 72    | <b>875</b>           | 197 478        | -      | -            | 476 268 | 14 981                        | <b>3 970</b>  | -            | -        | 352 032    |

P-DPOP has the same complexity as DPOP, i.e. it is exponential in the width of the pseudo-tree used. However, because P-DPOP cannot use elaborate heuristics to choose the pseudo-tree that would leak private information, the pseudo-trees produced by P-DPOP have on average a higher width than for DPOP.

## 5 ANY-TIME BEHAVIOR VIA ASYNCHRONY

Although DPOP is a very efficient algorithm in terms of the number of messages that are exchanged, the sizes of the messages are exponential in the width of the pseudo-tree. Faltings *et al.* [12] observed that, in general, one does not need full knowledge of a constraint problem to find the optimal solution. Although this observation was made for centralized problems, it is also valid in a distributed setting. Since DPOP sends information on *all* possible assignments, there is obvious room for improvement. In Open-DPOP (O-DPOP) [36], the UTIL propagation phase is replaced by a new phase in which an agent reports the entries in its UTIL table one by one, and in a best first order (i.e. lowest costs first), until its parent has enough information to determine its own next best assignment. An optimal solution to the DCOP is found when the root finds its first best assignment.

One disadvantage of both DPOP and O-DPOP is that they are not any-time algorithms, i.e. there are no intermediate solutions and a decision can only be made at termination. To mitigate this problem, ASO-DPOP [32] was proposed. The difference between O-DPOP and ASO-DPOP is twofold. First of all, the UTIL propagation phase and the VALUE propagation phase run concurrently, allowing agents to make decisions before the termination of the algorithm. Furthermore, agents are also allowed to report partial information, speeding up the propagation of information through the network, enabling the agents to make better informed intermediate decisions. The resulting privacy properties are the same as for DPOP (Table 1).

## 6 EXPERIMENTAL RESULTS

Simulations were run with the FRODO platform [27], on the Cordeau MDVRP benchmarks from [2]. To solve the local VRPs, we used the OR-Objects library [1], with the best out of randomized versions of the Clarke & Wright and the Gillett & Miller algorithms (5 iterations, strength 10) as construction algorithm, and the 3-Opt TSP improvement algorithm [24, 3]. The simulations were carried out on a 2.9 GHz, 16-core PC (i.e. one agent per core) with 2 GB of Java heap, and a timeout of 15 min. The results are presented in Table 2.

One can first observe that SynchBB is almost always faster than all other algorithms, except for two instances with large  $R$ , including one that DPOP was the only algorithm to solve. This can be explained by the fact that algorithms in FRODO are implemented at the variable level, not at the agent level. This means that variables within the

same agent exchange virtual messages (not counted in the information exchange metric), as if they were independent agents. Therefore, DPOP-based algorithms perform *Dynamic Programming (DP)* also inside each agent, while SynchBB performs *Branch & Bound (B&B)*, which is more efficient in the centralized setting.

At the inter-agent level, however, DP proves to be dramatically more efficient than B&B to reduce network bandwidth usage: all DPOP-based algorithms (except P-DPOP) send much less information than SynchBB. On p03 and for  $R = 16$ , DPOP even sends three orders of magnitude less information than SynchBB, and is only twice slower. This is due to the fact that, by sending few large messages instead of many small messages, DPOP saves on the overhead of sending a single message. Future work should investigate the use of a DP/search hybrid such as *PC-DPOP* [37].

Also, note that in these simulations, agents exchanged messages simply by accessing shared memory space. Therefore, the runtime metric does not take into account the cost of sending messages. It is very probable that, in a more realistic setting in which agents would communicate over a network, SynchBB's runtime performance would be significantly degraded by the large amounts of information (and the large numbers of messages) it exchanges.

When it comes to the performance of O-DPOP in terms of information exchange, O-DPOP actually sends just as much information as DPOP. This is probably due to the fact that, while O-DPOP needs to explore fewer combinations of assignments to the variables, and does not report all the information contained in DPOP's UTIL messages, it has to pay the overhead of sending a *context* in each of its UTIL messages. As far as ASO-DPOP is concerned, it sends only slightly more information; future work should study its convergence properties compared to other any-time algorithms like SynchBB.

Finally, the price that P-DPOP has to pay compared to DPOP in order to provide privacy guarantees is only moderate as far as runtime is concerned. P-DPOP however sends significantly more information.

## 7 CONCLUSION AND FUTURE WORK

In this paper, we have introduced the Distributed, Multiple-Depot Vehicle Routing Problem, in which each depot is controlled by a separate company with only partial information. We have proposed to use Distributed Constraint Optimization to formalize the distributed assignment of customers to depots. We have evaluated the performances of various DCOP algorithms on distributed versions of existing MDVRP benchmarks, in terms of runtime, information exchange, and any-time behavior. We have also put a special emphasis on the guarantees that can be provided in terms of protecting the companies' private information.

Future work should evaluate the DCOP approach against the ap-

proach based on the CNP. One could also investigate *dynamic* MDVRPs in which the companies must react to changes such as arrivals of new customers; this could be done using *S-DPOP* [35]. Another thread of research is designing *incentives* for the companies to honestly report their true costs. Petcu *et al.* [38] already proposed the *M-DPOP* algorithm as an incentive-compatible DCOP algorithm.

## REFERENCES

- [1] Operations Research – Java Objects. <http://opsresearch.com>.
- [2] VRP Web. <http://neo.lcc.uma.es/radi-aeb/WebVRP/>.
- [3] *Local Search in Combinatorial Optimization*, eds., Emile Aarts and Jan Karel Lenstra, Wiley, 1st edn., 1997.
- [4] Dariusz Barbucha and Piotr Jędrzejowicz, 'An agent-based approach to vehicle routing problem', *WASET*, **26**, 36–41, (2007).
- [5] Ismel Brito and Pedro Meseguer, 'Distributed forward checking may lie for privacy', In Pearce [33].
- [6] I-Ming Chao, Bruce L. Golden, and Edward Wasil, 'A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions', *American Journal of Mathematical and Management Sciences*, **13**(3–4), 371–406, (1993).
- [7] Anton Checheta and Katia Sycara, 'No-commitment branch and bound search for distributed constraint optimization', in *Proceedings of AAMAS'06*, pp. 1427–1429, Hakodate, Japan, (May 8–12 2006).
- [8] Jean-François Cordeau, Michel Gendreau, and Gilbert Laporte, 'A tabu search heuristic for periodic and multi-depot vehicle routing problems', *Networks*, **30**(2), 105–119, (December 7 1998).
- [9] P. Davidson, L. Henesey, L. Ramstedt, J. Tornquist, and F. Wernstedt, 'An analysis of agent-based approaches to transport logistics', *Transportation Research Part C*, **13**, 255–271, (2005).
- [10] Rina Dechter, *Constraint Processing*, Morgan Kaufmann, May 5 2003.
- [11] Boi Faltings, Thomas Léauté, and Adrian Petcu, 'Privacy guarantees through distributed constraint satisfaction', in *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'08)*, pp. 350–358, (December 9–12 2008).
- [12] Boi Faltings and Santiago Macho-Gonzalez, 'Open constraint programming', *Artificial Intelligence*, **161**(1–2), 181–208, (January 2005).
- [13] Klaus Fischer, Jörg P. Müller, and Markus Pischel, 'Cooperative transportation scheduling: An application domain for DAI', *Applied Artificial Intelligence*, **10**(1), 1–34, (February 1996).
- [14] M. S. Franzin, E. C. Freuder, Francesca Rossi, and R. Wallace, 'Multi-agent constraint systems with preferences: Efficiency, solution quality, and privacy loss', *Computational Intel.*, **20**(2), 264–286, (May 2004).
- [15] Michel Gendreau, Jean-Yves Potvin, Olli Bräysy, Geir Hasle, and Arne Løkketangen, 'Metaheuristics for the vehicle routing problem and its extensions : A categorized bibliography', Technical Report 2007-27, CIRRELT, (August 2007).
- [16] Amir Gershman, Amnon Meisels, and Roie Zivan, 'Asynchronous forward-bounding for distributed constraints optimization', in *Proceedings of ECAI'06*, pp. 103–107, (August 29–September 1 2006).
- [17] Amir Gershman, Roie Zivan, Tal Grinshpoun, Alon Grubshtein, and Amnon Meisels, 'Measuring distributed constraint optimization algorithms', in *Proceedings of DCR'08*, pp. 17–24, (May 13 2008).
- [18] Rachel Greenstadt, Barbara Grosz, and Michael D. Smith, 'SSDPOP: Using secret sharing to improve the privacy of DCOP', In Pearce [33].
- [19] Rachel Greenstadt, Jonathan P. Pearce, and Milind Tambe, 'Analysis of privacy loss in distributed constraint optimization', in *Proceedings of AAAI'06*, pp. 647–653, (2006).
- [20] Katsutoshi Hirayama, William Yeoh, and Roie Zivan, eds. *Proc. Distributed Constraint Reasoning Workshop (DCR'09)*, July 13 2009.
- [21] Katsutoshi Hirayama and Makoto Yokoo, 'Distributed partial constraint satisfaction problem', in *CP'97*, volume 1330, pp. 222–236, (1997).
- [22] William P. C. Ho, George T. S. Ho, Ping Ji, and Henry C. W. Lau, 'A hybrid genetic algorithm for the multi-depot vehicle routing problem', *Engineering Applications of AI*, **21**(4), 548–557, (June 2008).
- [23] G. Laporte, Y. Nobert, and D. Arpin, 'Optimal solutions to capacitated multidepot vehicle routing problems', *Congressus Numerantium*, **44**, 283–292, (1984).
- [24] *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, eds., E. L. Lawler, Jan Karel Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, Wiley, August 1985.
- [25] Thomas Léauté and Boi Faltings, '[DPOP]: Distributed constraint optimization under stochastic uncertainty using collaborative sampling', In Hirayama *et al.* [20], pp. 87–101.
- [26] Thomas Léauté and Boi Faltings, 'Privacy-preserving multi-agent constraint satisfaction', in *Proc. of the 2009 IEEE Intl Conf. on Privacy, Security, Risk and Trust (PASSAT'09)*, pp. 17–25, (August 29–31 2009).
- [27] Thomas Léauté, Brammert Ottens, and Radoslaw Szymanek, 'FRODO 2.0: An open-source framework for distributed constraint optimization', In Hirayama *et al.* [20], pp. 160–164. <http://liawww.epfl.ch/frodo/>.
- [28] Rajiv T. Maheswaran, Jonathan P. Pearce, Emma Bowring, Pradeep Varakantham, and Milind Tambe, 'Privacy loss in distributed constraint reasoning: A quantitative framework for analysis and its applications', *Autonom. Agents and Multi-Agent Systems*, **13**(1), 27–60, (July 2006).
- [29] Rajiv T. Maheswaran, Jonathan P. Pearce, and Milind Tambe, 'Distributed algorithms for DCOP: A graphical-game-based approach', in *Proceedings of ISCA PDCS'04*, pp. 432–439, (September 15–17 2004).
- [30] Roger Mailler and Victor R. Lesser, 'Solving distributed constraint optimization problems using cooperative mediation', in *Proceedings of AAMAS'04*, volume 1, pp. 438–445, (July 19–23 2004).
- [31] Pragnesh J. Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo, 'ADOPT: Asynchronous distributed constraint optimization with quality guarantees', *Artificial Intelligence*, **161**, 149–180, (2005).
- [32] Brammert Ottens and Boi Faltings, 'Coordinating agent plans through distributed constraint optimization', in *Proceedings of the ICAPS'08 Multiagent Planning Workshop (MASPLAN'08)*, (September 14 2008).
- [33] Jonathan P. Pearce, ed. *Proceedings of the 9th Intl Workshop on Distributed Constraint Reasoning (CP-DCR'07)*, September 23 2007.
- [34] Adrian Petcu and Boi Faltings, 'DPOP: A Scalable Method for Multi-agent Constraint Optimization', in *Proc. 19th Intl Joint Conf. on Artificial Intelligence (IJCAI'05)*, pp. 266–271, (July 31 – August 5 2005).
- [35] Adrian Petcu and Boi Faltings, 'S-DPOP: Superstabilizing, fault-containing multiagent combinatorial optimization', in *Proc. of the 20th National Conference on A.I. (AAAI'05)*, pp. 449–454, (July 9–13 2005).
- [36] Adrian Petcu and Boi Faltings, 'O-DPOP: An algorithm for open/distributed constraint optimization', in *Proc. of the 21st National Conf. on Artificial Intelligence (AAAI'06)*, pp. 703–708, (2006).
- [37] Adrian Petcu, Boi Faltings, and Roger Mailler, 'PC-DPOP: A new partial centralization algorithm for distributed optimization', in *Proceedings of IJCAI'07*, pp. 167–172, (January 6–12 2007).
- [38] Adrian Petcu, Boi Faltings, and David C. Parkes, 'M-DPOP: Faithful distributed implementation of efficient social choice problems', *Journal of Artificial Intelligence Research (JAIR)*, **32**, 705–755, (July 2008).
- [39] David Pisinger and Stefan Ropke, 'A general heuristic for vehicle routing problems', *Computers and OR*, **34**(8), 2403–2435, (August 2007).
- [40] Jacques Renaud, Gilbert Laporte, and Faye F. Boctor, 'A tabu search heuristic for the multi-depot vehicle routing problem', *Computers and Operations Research*, **23**(3), 229–235, (March 1996).
- [41] A.E. Rizzoli, R. Montemanni, E. Lucibello, and L.M. Gambardella, 'Ant colony optimization for real-world vehicle routing problems: From theory to applications', *Swarm Intelligence*, **1**, 135–151, (Sept. 2007).
- [42] Tuomas Sandholm, 'An implementation of the contract net protocol based on marginal cost calculations', in *AAAI'93*, pp. 256–262, (1993).
- [43] Marius-Călin Silaghi, Boi Faltings, and Adrian Petcu, 'Secure combinatorial optimization simulating DFS tree-based variable elimination', in *Proc. of the 9th Intl Symp. on A.I. and Math.*, (January 4–6 2006).
- [44] Marius-Călin Silaghi and Debasis Mitra, 'Distributed constraint satisfaction and optimization with privacy enforcement', in *Proceedings of IAT'04*, pp. 531–535, (September 20–24 2004).
- [45] Evan A. Sultanik, Robert N. Lass, and William C. Regli, 'DCOPolis: A framework for simulating and deploying distributed constraint optimization algorithms', In Pearce [33].
- [46] Sam R. Thangiah and Said Salhi, 'Genetic clustering: An adaptive heuristic for the multidepot vehicle routing problem', *Applied Artificial Intelligence*, **15**(4), 361–383, (2001).
- [47] *The Vehicle Routing Problem*, eds., Paolo Toth and Daniele Vigo, SIAM, 2001.
- [48] Makoto Yokoo, Koutarou Suzuki, and Katsutoshi Hirayama, 'Secure distributed constraint satisfaction: Reaching agreement without revealing private information', *Artif. Intel.*, **161**(1–2), 229–245, (Jan. 2005).
- [49] Weixiong Zhang, Guandong Wang, Zhao Xing, and Lars Wittenburg, 'Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks', *JAIR*, **161**(1–2), 55–87, (January 2005).

# Residual resources exploitation in a coopetitive MAS

Gaël Hette and Sylvia Estivie and Emmanuel Adam and René Mandiau<sup>1</sup>

**Abstract.** In a dynamic system, some disturbances could occur during the execution of agents task schedules. In this paper, we aim to propose a framework to solve such kind of problems by pooling residual resources to reallocate tasks during the execution of the system. In our context, the initial task schedules of agents are defined such that during some given time windows agents could perform additional tasks with their residual resources. In this paper, we study this problem of reallocation of residual resources in order to propose a solution to these disturbances.

## 1 Introduction

Over the last decades multi-agent paradigm has been used as a framework to solve various kinds of resource allocation and task scheduling problems in distributed systems. In order to cope with the characteristics of each problem types, different approaches have been proposed in the field of multi-agent systems. One approach consists in considering and formalizing such problems as Distributed Constraint Satisfaction Problem (*DCSP*). Another approach which forms a noticeable growing field of research, known as *Multi-agent Resource Allocation (MARA)*, is dedicated to issues regarding allocation of resources within a system of autonomous agents [2]. A multi-agent resource allocation problem is defined as a process of distributing a number of resources (or tasks) among several agents. This distribution is conducted by a centralized or distributed allocation procedure wherein agents participate in the elaboration of the allocation by reporting their preferences over the set of resources (or tasks) they will receive. In the case of a task allocation procedure, an usual objective is to find a feasible allocation. Whereas a typical objective of resource allocation procedure is to find an optimal allocation in the sense of a measure of the social welfare arisen by the considered society of agents.

In this paper we choose to focus on the case of task allocation over a multi-agent system, where each agent disposes of non shareable resources to achieve tasks. In this context, some specific allocation problems lead to solutions where the totality of the resources are not necessary allocated. These classes of problems implique the fact that non-allocated resources (i.e. *residual resources*) are still available during the execution of such allocated system instances. In this paper we propose to use *MARA* techniques to re-allocate tasks during the execution of a defined schedule. Thus considering a failure during the execution of such schedule, some tasks become impossible to service. These tasks could be reallocated to agents disposing of *residual resources* corresponding to their achievement constraints. We aim to propose a framework to resolve such problems by pooling

*residual resources* to reallocate tasks during the execution of the system. Considering an agent in need of resources, our approach is based on a model and a set of methods to find corresponding available resources within the system and then use negotiation to determine a new tasks allocation.

The paper is structured as follows. Section 2 is a detailed presentation of the problem addressed by our study and goes on by presenting a brief state of the art. Section 3 specifies the basic elements of our computational model. Section 4 details how we can apply our approach to the problem of adaptive resource management in a transportation network.

## 2 Preliminaries

### 2.1 Our problem

We consider a multi-agent system with an initial resource allocation amongst agents. This initial resource allocation is computed in order to accomplish an associated task schedule. Each tasks of this schedule have to be performed by a specific agent within a given time window during which the required resources for this task are allocated to this agent.

In addition, we assume that the initial resource allocation is such that some allocated resources are not fully used by agents, i.e. an agent could perform additional tasks during given time windows of its initial task schedule. So, the main idea of this work is to consider that agents pool their *residual resources* in order to execute tasks for other agents and to enable task reallocation. The main goal of each agent is to increase the value of their *residual resources* by means of monetary valuation and also to be able to use cheap resources to delegate the achievement of one or more punctual tasks.

Our approach aims to cover a wide range of possible applications such as computer network, grid computing, shipping logistics, transportation network, supply chain, manufacturing system, search and rescue mission, etc...

For example, if we consider a computer network conducted by a multi-agent system and with a defined resource allocation, during the execution of the task schedule some failure may occur. As a consequence some tasks may become unfeasible. Moreover some resources may be allocated to agents and may not be fully used by them, i.e. *residual resources*. Our idea is that by pooling those *residual resources* it could be possible to achieve some of the unfeasible tasks with a new ad-hoc task allocation.

As we focus on the case of tasks reallocation during the execution of a defined schedule, the main constraint of our problem is that we cannot modify the initial resource allocation. We must rather use this allocation in order to assign the execution of additional tasks to agents based upon their allocated resources and initial task schedule (i.e. without modifying the agents initial task schedule).

<sup>1</sup> Univ Lille Nord de France, F-59000 Lille, France; UVHC, LAMIH, F-59313 Valenciennes, France; CNRS, FRE 3304, F-59313 Valenciennes, France; email: {gael.hette}{sylvia.estivie}{emmanuel.adam}{rene.mandiau}@univ-valenciennes.fr

## 2.2 State of the art

A possible approach to solve resource allocation and task scheduling problems in distributed systems is to consider and formalize such problems as Distributed Constraint Satisfaction Problem (DCSP) in which variables and constraints are distributed among multiple agents. As described by M. Yokoo [16], resolution of DCSP are mainly achieved by use of distributed algorithms. Like algorithms for solving CSP, DCSP algorithms can be divided into two groups : search algorithms (i.e. backtracking and iterative improvement) and consistency algorithms (arc and path consistency). DCSP algorithms are efficient to solve large-scale resource allocation problems within a reasonable amount of time [16]. But as noted by Harvey et al. [8] DCSP algorithms use ordering over the variable which induces an order among agent authority and implies priority over resource allocation between agents. Thus we ignore this approach, since for the purpose of our work, we do not want any authority order between agents.

The remainder of this section starts with a presentation of one major problem where the need of task reallocation could arise in case of disturbance or failure during the execution of a defined schedule. Thus, under some assumptions, it could be possible and profitable to apply the pooling of *residual resources* in order to resolve such kind of problem. Furthermore, to the best of our knowledge there is no study in the multi-agent literature which addresses the problem of task reallocation during the execution of a schedule and without any modification of the initial resource allocation. Therefore we give an overview of different approaches for resource allocation and tasks scheduling which deal with problems that are closely related to our (Section 2.1). We present these approaches as possible leads and insights to cope with task reallocation without modifying the initial resource allocation and task schedule of agents.

The *Vehicle Routing Problem with Time Windows (VRPTW)* [10] is a major problem of interest as it can be used to model many applied problems. This problem has been studied widely in both operational research [14, 3] and multi-agent systems literature [5, 7]. The VRPTW concerns the design of least cost routes over a fleet of vehicles, in order to deliver goods from a central depot to a set of geographically scattered points during given time windows<sup>2</sup>. The VRPTW is considered as *static* in the case that all relevant informations are known beforehand. Otherwise the problem is considered as *dynamic* and is referred as *Real-Time VRPTW* [11, 6].

In the multi-agent literature, the market based approach is the main approach adopted to solve VRPTW. A common model is based on a manager agent to represent the shipping company and a set of agents to represent trucks of the company. The task allocation process comprises a sequence of successive or parallel allocation procedures where delivery orders are proposed to the truck agents by the manager agent. These allocation procedures are often based on the Contract Net Protocol [4, 5] or on the Vickrey auction model [15, 7, 9].

Fischer et al. [5] presented a multi-agent system simulation framework where agents interactions are driven by negotiation protocol leading to different task scheduling mechanisms. They produced an implementation and experimental results of their approach to solve the *Vehicle Routing Problem with Time Windows (VRPTW)*. This study focuses on a problem that is similar to ours in the sense of resource pooling, as it proposes a peer-to-peer negotiation between companies to buy and sell free loading capacities (i.e. *residual re-*

*sources*). But these considerations differ with ours because this negotiation does not occur in the case of task reallocation. As a matter of fact, a task reallocation procedure is used to overcome disturbance during the execution of a plan. This procedure allocates the unfeasible tasks of an agent to other agents of the same company. Considering the constraints of our problem, the outcomes of this procedure are interesting. Indeed, in the best case the procedure can lead to local rescheduling of agents plans, whereas in the worst case it leads to a more global rescheduling.

Another multi-agent system simulation framework for resource allocation and task scheduling which is based on auction has been proposed by V. Godoretski et al. [7]. They produced an implementation and experimental results to solve the VRPTW. Their study differs with our problem because it focuses on the case of one shipping company and there is no consideration about dealing with disturbance during the execution of a plan.

Hoen et al. [9] studied the concept of decommitment in a large scale logistics setting of competing transportation companies. The action of decommitment consists in replacing a won contract by another (more profitable) contract. As in our problem, their approach deals with a set of competitive companies and their decommitment strategy can be seen as a process of tasks reallocation with few modifications of agents task schedule. The decommitted tasks are reallocated among the agents of the company which holds the contract, whereas by pooling *residual resource* we aim to reallocate tasks among agents of all companies. In this study the objective of decommitment is to increase the profit, but we consider that a similar decommitment strategy could be used to overcome disturbances during the execution of multiple VRPTW solution instances.

Sadri et al. [13] proposed an approach to solve the *Temporal Resource Reallocation Problem (TRRP)*, based on multiple stages of negotiation using dialogue moves in a society of self-interested agents sharing resources. The context of TRRP is such that agents aim to accomplish tasks that require a specific resource, have a specific duration and must be done within a given time window. At start, the initial distribution of resources is such that agents do not necessary own the resources required by their activities. Thus agents negotiate to exchange the possession of resources over specific time windows. The negotiation process leads to compute a resource allocation and schedules such that agents can perform their tasks within the given time constraints. P. Alexopoulos et al. [1] extended the work of Sadri et al. [13], by modifying the protocol and policy of the third negotiation stage in order to allow multiple resources exchange dialogues. In regard with our problem these approaches are interesting because they aim to compute a resource allocation within time windows. The difference between these two works and our problem is that under the TRRP context agents exchange resources and modify their schedule. As a matter of fact, it could be interesting for us to adapt these negotiation protocols such that agents exchange and/or delegate tasks execution within specified time windows and according to their *residual resources*.

## 3 Computational model

We consider a multi-agent system populated by  $n$  agents  $A_1, \dots, A_n$ . In order to enable agents to perform an argumentation based negotiation protocol (Section 3.1), we use a Belief-Desire-Intention model [12] inspired by Sadri et al. [13] to define the knowledges of agents. Thus, the knowledges of an agent  $A_i$  are represented by a tuple  $\langle B_i, I_i, R_i, O_i, D_i, DS_i, G_i \rangle$  with :

<sup>2</sup> Delivery locations must be visited only once and by exactly one vehicle during a given time window. Each route starts and ends at depot/warehouse. The total load of a vehicle for a given route does not exceed its capacity.

- $B_i$  : *beliefs* (negotiation policy, informations about self and other agents),
- $\mathcal{I}_i$  : *intentions* (plan, task schedule of agent  $i$ ),
- $\mathcal{R}_i$  : *allocated resources*,
- $\mathcal{O}_i$  :  $\{O_{i1}, \dots, O_{iu}\}$  a finite set of *task offers* of agent  $i$ ,
- $\mathcal{D}_i$  :  $\{D_{i1}, \dots, D_{iv}\}$  a finite set of *task demands* of agent  $i$ ,
- $\mathcal{DS}_i$  : *store* of past dialogues,
- $\mathcal{G}_i$  : *agent goals*.

In our model, the desires that an agent wishes to satisfy corresponds to its task demands. The intentions  $\mathcal{I}_i$  of an agent  $A_i$  represent its task schedule. This schedule is an ordered set of tasks. Each task is specified by a time window and its required resources. As Sadri et al. [13] we assume that each task needs only one resource and that all tasks requiring the same resource have disjoint time windows. In regard with our problem, the initial resource allocation is such that each agent of the system owns all the required resources of its initial task schedule. In addition, the task schedules of agents are defined such that, during some time periods, all allocated resources are not needed by the agents (i.e. *residual resources*). We assume that during such time periods agents can perform additional tasks requiring these *residual resources*. Thus, agents dispose of a set of *task offers* as a mean to propose to other agents to execute tasks for them. According to the dynamic of the system, some disturbances occur during the execution of agents task schedules. In our model we define a disturbance as a loss of one or more initially allocated resources of an agent. Therefore, agents incurring disturbances cannot perform the tasks requiring such unavailable resources. To overcome such situation agents dispose of a set of *task demands* which define the set of tasks that an agent cannot perform and could delegate to other agents. Agents create a *task demand* for each scheduled tasks it cannot perform due to the loss of a resource. At start this set of *task demands* is empty. The content of this set evolves along the dynamic of the system.

Given an agent  $A_i$ , we define each elements of  $\{O_{i1}, \dots, O_{iu}\}$  and  $\{D_{i1}, \dots, D_{iv}\}$  as finite sets of data respectively describing the *residual resource* associated to a *task offer* and the resource associated to a *task demand* of  $A_i$ . Each elements of  $\{O_{i1}, \dots, O_{iu}\}$  and  $\{D_{i1}, \dots, D_{iv}\}$  are composed of the following data set  $(A_{rc}, R_{rc}, T_{rc}, V_{rc}, P_{rc}, Loc_{rc}, Pref_{rc})$ , where :

- $A_{rc}$  : the agent offering or needing resource,
- $R_{rc}$  : the type of the offered or needed resource,
- $T_{rc}$  : a time interval defining a time constraint over the period when the resource is available or needed to perform a task delegation,
- $V_{rc}$  : the volume or amount of the offered or needed resource,
- $P_{rc}$  : the price (a monetary valuation) of the offered or needed resource,
- $Loc_{rc}$  : the location of the offered or needed resource,
- $Pref_{rc}$  : a finite set of preferences regarding the use of the resource (Section 3.2).

Depending on the type of problem addressed, some of the above constraints are not necessarily affected, e.g. for a computer network problem the  $Loc_{rc}$  constraint may be useless. Our brief state of the art along task and resource reallocation problems lead us to think that temporal constraints are a common and crucial point of interest in such problems. So in our approach we consider that temporal constraint is the only necessary constraint for the core class of task allocation problems addressed by our framework. Thus we assume that the following constraints are optional and used in particular cases :

- $V_{rc}$  : used in VRPTW<sup>3</sup> as a constraint on freight load, or in TRRP<sup>4</sup> as the constraint on required resource type;
- $P_{rc}$  : to address problems with a market based approach or if the addressed problem involves selfish agents;
- $Loc_{rc}$  : used if the addressed problem concerns situated agents (ie. VRPTW).

In the reallocation procedure (Section 3.1), agents try to find the best *task offer* corresponding to their needs by asking other agents about their *task offer*. The reallocation procedure also involves a negotiation step. In order to perform the reallocation procedure, agents must communicate according to the application context of the addressed problem. Hence, we distinguish two possible alternatives over agent's communication ability :

- the communication between agents are constrained, such that at a given time an agent is only able to communicate with a restricted number of other agents (e.g. communication can be constrained by topological consideration);
- there is no communication constraint, agents can communicate any other agent at any time.

In the context of our study, we assume that we are in the second case and there is no communication constraint between agents.

### 3.1 Task reallocation process

In the context of our problem, we consider a multi-agent system which evolves over time. Agents have to perform tasks in a dynamic environment, and we assume the following *system cycle* :

1. *Environment dynamics* : the environment evolves according to the defined rules of the system dynamics. *Disturbances* occur at random, i.e. some resources become unavailable to one or several agents.
2. *Perception step* : agents get perceptions from the environment. These perceptions concern the current availability of the initially allocated resources. Agents facing disturbances update their knowledges in regard with their unavailable resources. As a matter of fact, these agents generate and add new *task demands* to their own set of *task demands*.
3. *Reasoning step* : agents facing disturbances (i.e. agents with a non empty set of *task demands*) create a list of agents which may propose each one or more appropriate *task offers*.
4. *Communication step* : agents facing disturbances ask the previously selected agents about their *task offers*. If one or more *task offers* fulfill the needs of a *task demand*, the agent choice between the different offers is based upon its preferences model (Section 3.2). If there is no appropriate *task offer*, the agent performs a negotiation process with agents offering less appropriate *task offers*.
5. *Action step* : agents performs their scheduled tasks in regard with the availability of the required resources.

We propose to decompose the task reallocation process of an agent  $A_i$  into three main steps. The first step of the reallocation process occurs during the reasoning step of agent  $A_i$ . During this step  $A_i$  computes a list of agents to query about their *task offers*. This list of agents is determined upon the beliefs of  $A_i$  about other agents. These beliefs about other agents have been acquired during former interactions with agents.

<sup>3</sup> Vehicle Routing Problem with Time Windows

<sup>4</sup> Temporal Resource Reallocation Problem

The second step of the reallocation process occurs during the communication step. During this step,  $A_i$  queries all the previously selected agents about their *task offers*. In response, each of the selected agents compute their current set of *task offers* without any consideration of the *task demand* of agent  $A_i$ . Then, all the selected agents reply to  $A_i$  by sending their current set of *task offers*. Thus  $A_i$  retrieves from these agents a set of acceptable *task offers* in regard with its needs (i.e. the preferences of  $A_i$  *task demand*). If the set of acceptable offers is empty, then agent  $A_i$  can perform the following actions:

- restart the reallocation process and determine a new list of agents according to its new beliefs about others in order to ask new agents about their *task offers*,
- negotiate upon time constraints, with agents offering less acceptable *task offers*.

The third step of the reallocation process occurs if the set of acceptable *task offers* is not empty. During this step the agent  $A_i$  chooses, with respect to the preferences of its *task demand*, the best offer from the set of acceptable *task offers*. Then  $A_i$  concludes the delegation of its task with the agent proposing the chosen *task offer*.

### 3.2 Agents preferences representation

In order to define how agents should make a decision over a set of acceptable solutions, we introduce a set of ad hoc preferences. Preferences are a mean to express the satisfaction of an agent towards a specific offer. During the second step of the reallocation process it is up to the agent  $A_i$ , who is missing a resource, to perform a research heuristic as a way to find acceptable offers. To reduce the computational cost of this heuristic, the selection of acceptable offers is only based upon agent  $A_i$  satisfaction, and as a matter of fact only upon  $A_i$  preferences. As a result we define a class of preferences specific to the *task demand* of an agent as follows :

- $T_{pref}$  : preferences among time constraints of a set of offers. This preference enables agents to express how much a task delegated to another agent can be postponed.
- $P_{pref}$  : preferences among prices of different *task offers*. For instance, it determines if the agent chooses the cheapest priced or the average priced offer.
- $PT_{pref}$  : preference between price and time period proposals, over a set of acceptable solutions. This enable to define if an agent chooses to focus on price constraint rather than time constraint or vice versa.
- $Loc_{pref}$  : preferences over the location of different *task offers* to choose with.
- $Co_{pref}$  : preferences over a specific group of agents to work with. Thus, between different acceptable *task offers* an agent chooses the offer made by an agent specified by  $Co_{pref}$ .

In regard with all the above different preferences associated to a *task demand*, a major question of interest is how to deal with this set of preferences to determine the choice of an agent between two or more acceptable *task offers*. In a future work, we will define a procedure to aggregate this heterogeneous preferences.

During the second stage of the reallocation process, agents exchange dialogues of negotiation by pairs. Both agents participating in the dialogue use a negotiation policy to formulate dialogue moves. The negotiation policy of an agent is based upon its preferences regarding its *task demand* or its *task offer*. As needed by an agent in-

involved in a negotiation dialogue, we define the following specific class of preferences :

- $T_{pref}$  : preferences over the time period to execute the offered task. Agent negotiate over the time period during which the task should be executed,
- $Loc_{pref}$  : agent negotiate over the location of the offered *residual resources*,

## 4 An application for logistic networks

We consider a framework which aim to manage pooling capacity for transportation and storage of goods/freight between different logistic companies sharing their resources. Unexpected events may occur during the execution of the contracts (transportation or storage), e.g. truck breakdown or warehouse failure. Such unexpected events must be solved by finding new transportation or storage means within the pooled resources of the framework, and by creating new transportation plans.

We consider a multi-agent system which represent a pooling capacity framework. Each agent represent transportation or storage mean, e.g. trucks or warehouses. The environment is described as a spacial grid divided into small non overlapping areas. Agents are localized by their position in a unique area of the spatial grid. Time is considered continuous.

There is no communication constraint between agents. Each agent can handle a specific bounded capacity (volume and weight) of freight. During the execution of the system agents' freight load evolve. Actually agents capacity are not necessary fully loaded, so unused transportation and storage capacity are still available.

To cope with unexpected events agents must be able to find new means of transportation and/or storage within the agent society. Thus they must communicate with each others to find agents with corresponding residual capacity and elaborate new plans.

## 5 Conclusion

In this paper, after a presentation of the problem addressed by our study and a short state of the art, we proposed a computational model of the multi-agent system that allows agents to negotiate the reallocation of tasks. We introduced the possible exploitation of residual resources in a multi-agent system where agents dispose of non shareable resources to execute a defined task schedule. To overcome disturbances during the execution of such schedule, we propose to reallocate tasks to agents over time periods where they can perform additional tasks without modifying their initial schedule. During the reallocation process agents try to delegate to other agents the execution of tasks they can no longer perform due to a loss of resources.

There are many lines of future research related to this work. A major question concerns the definition of an aggregation procedure for the heterogeneous set of agents' preferences. Another issue that is not covered in this work and that we will study in the future is the agents negotiation protocol and policy based on the specific class of preferences proposed in Section 3.2. We will also intend to refine the second step of the reallocation procedure. At this step of the procedure, if an agent is unable to find any acceptable *task offers*, it can restart the search of acceptable *task offers* or negotiate with agents offering less acceptable *task offers*. The issue of this refinement is to enable agents to determine if they either initiate a negotiation process or restart the search of *task offers*. It is obvious that in certain cases it is not efficient to perform a negotiation process. Thus we will

have to exhibit such cases and find a mean to avoid agent to negotiate in these cases. Another issue of future work is to implement our computational model and propose an experimental evaluation of the reallocation procedure and its' computational complexity based on the number and the size of the messages exchanged between agents.

## ACKNOWLEDGEMENTS

We would like to thank the referees for their comments, which helped improve this paper. The present research work has been supported by International Campus on Safety and Intermodality in Transportation, the Nord-Pas-de-Calais Region, the European Community, the Regional Delegation for Research and Technology, the French Ministry of Higher Education and Research, the French National Center for Scientific Research, the Ile-de-France Region, the "Ministre de l'économie, de l'Industrie et de l'Emploi", and ADVANCITY. The authors gratefully acknowledge the support of these institutions.

## REFERENCES

- [1] Panos Alexopoulos, 'Resource sharing in multi-agent systems through multi-stage negotiation', in *AIAI*, eds., Christos Boukis, Aristodemos Pnevmatikakis, and Lazaros Polymenakos, volume 247 of *IFIP*, pp. 57–64. Springer, (2007).
- [2] Yann Chevaleyre, Paul E. Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaître, Nicolas Maudet, Julian A. Padget, Steve Phelps, Juan A. Rodríguez-Aguilar, and Paulo Sousa, 'Issues in multiagent resource allocation', *Informatica*, **30**(1), 3–31, (2006).
- [3] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis, 'VRP with time windows', 157–193, (2001).
- [4] R. Davis and R. G. Smith, 'Negotiation as a metaphor for distributed problem solving', *Artificial Intelligence*, **109**, 20–63, (1983).
- [5] Klaus Fischer, Brahim Chaib-draa, Jörg P. Müller, Marcus Pischel, and Christian Gerber, 'A simulation approach based on negotiation and co-operation between agents: A case study', *IEEE Trans. on Systems, Man, and Cybernetics*, **29**, 531–545, (1999).
- [6] Zheng gang Dan, Lin ning Cai, and Li Zheng, 'Real-time distribution system based on multi-agent', in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Volume I, IMECS '08, March 19 - 21, 2008, Hong Kong*, eds., S. I. Ao, Oscar Castillo, Craig Douglas, David Dagan Feng, and Jeong-A Lee, Lecture Notes in Engineering and Computer Science, pp. 218–221. International Association of Engineers, Newswood Limited, (2008).
- [7] Vladimir I. Gorodetski, Oleg Karsaev, and Victor Konushy, 'Multi-agent system for resource allocation and scheduling', in *CEEMAS*, eds., Vladimír Marík, Jörg P. Müller, and Michal Pechoucek, volume 2691 of *Lecture Notes in Computer Science*, pp. 236–246. Springer, (2003).
- [8] Peter Harvey, Chee Fon Chang, and Aditya Ghose, 'Support-based distributed search: a new approach for multiagent constraint processing', in *AAMAS*, eds., Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, pp. 377–383. ACM, (2006).
- [9] Pieter Jan't Hoen and J. A. La Poutré, 'A decommitment strategy in a competitive multi-agent transportation setting', in *AMEC*, eds., Peyman Faratin, David C. Parkes, Juan A. Rodríguez-Aguilar, and William E. Walsh, volume 3048 of *Lecture Notes in Computer Science*, pp. 56–72. Springer, (2003).
- [10] Gilbert Laporte and Ibrahim H. Osman, 'Routing problems: A bibliography', *Annals of Operations Research*, **61**(1), 227–262, (1995).
- [11] H. N. Parfaits, 'Dynamic vehicle routing : status and prospects', *Annals of Operations Research*, **61**(1), 143–164, (1995).
- [12] Anand S. Rao and Michael P. Georgeff, 'BDI agents : From theory to practice', in *ICMAS*, eds., Victor R. Lesser and Les Gasser, pp. 312–319. The MIT Press, (1995).
- [13] Fariba Sadri, Francesca Toni, and Paulo Torroni, 'A multi-stage negotiation architecture for sharing resources amongst logic-based agents', in *UKMAS, 2002*, (2002).
- [14] Marius M. Solomon, 'Algorithms for the vehicle routing and scheduling problems with time window constraints', *Operations Research*, **35**, 254–265, (1987).
- [15] W. Vickrey, 'Counterspeculation, auctions and competitive sealed bids', *Journal of Finance*, **16**, 8–37, (1961).
- [16] Makoto Yokoo, *Distributed constraint satisfaction: foundations of co-operation in multi-agent systems*, Springer-Verlag, London, UK, 2001.



# Bringing Agents into Application: Intelligent Products in Autonomous Logistics

Farideh Ganji and Ernesto Morales Kluge and Bernd Scholz-Reiter<sup>1</sup>

**Abstract.** Autonomous logistics is a new research domain that works on the decentralization of decision-making processes. To reach the research objectives of this domain, intelligent products are becoming the focal point. The implementation of autonomous systems leads to the deployment of Multi-Agent-Systems which manage the complex decision making processes. A key finding of the autonomous logistics research is the demand for positive emergence that should arise out of the global system when applying decentralized intelligent products. This paper concentrates on the first integration attempts of intelligent products in the context of autonomous logistics processes and presents a demonstration platform which assembles automotive tail-lights with autonomous logistics methods.

## 1 INTRODUCTION

Reaching the well-known requirements for logistics - having the right product at the right time at the right place - is becoming more and more difficult with traditional planning and control methods. The current research considers, therefore, the concepts of decentralization and autonomy on the logistics decision-making processes and reflects aspects such as flexibility, proactivity and adaptability. The idea of autonomous cooperating logistic processes is characterized by shifting decision competencies to autonomous logistics objects for decentralized and heterarchical planning and control. This is deemed to be an answer to the mentioned demands. Applying this concept new properties of a larger system may emerge by local interaction of subsystems. This key characteristic is called emergence whose effects are hard to anticipate due to complexity, resulting from subsystem interaction. Emergence may concern organizational structures or even problem solutions. Emergent organizations are evolving and thus able to adapt themselves to modifications in the environment and their members' goals. Positive emergence means that subsystem interaction leads to a better achievement of objectives of the total system than it is explicable by considering the behavior of every single system element. In the context of autonomous logistics, these effects are incorporated by implementing logistic objects (e.g., means of transport, freight, parts) as decentralized subsystems that dynamically coordinate with other subsystems to manage logistic processes and reach their respective goals (e.g., on-time delivery or minimization of delivery times).

This paper reflects an ongoing work on implementing autonomous control methods on logistics systems, specifically in production logistics, where the *Intelligent Product* plays a central role. This work is being performed in a technical subproject which develops also an

application and demonstration platform within the Collaborative Research Centre 637 "Autonomous Cooperating Logistic Processes-A Paradigm Shift and its Limitations" (CRC 637). Through the course of the paper a production scenario will be presented designed to investigate the applicability in the domain of production logistics. The scenario illustrates an autonomous assembly system for an automotive tail-light.

The assembly scenario is taken from a flow shop system that does not allow any flexibility within the sequence of processes. Today automotive tail-lights are manufactured with variant types in order to meet the customer demands. Thus variant flow shop systems evolved from the inflexible systems. Since these systems are still controlled centrally with a limited and predefined space of variants that are determined and scheduled beforehand, this realistic scenario was taken as a starting point to derive the introduced scenario with Autonomous Control by implementing variant types of the finished product which have to be chosen by the product itself.

## 2 RELATED WORK

### 2.1 Internet of Things

The concept of "Internet of Things"(IoT) is mainly driven by technologies and concepts like pervasive and ubiquitous computing. The vision of IoT describes the strongly growing interconnectivity not only between people, but also between "things" and has become a new paradigm in the recent years [10]. Today several research institutions and universities are working on this topic and even authorities are funding this research topic. There are also associations like EPC-Global<sup>2</sup> that work on industry driven standards on electronic product code with the perspective on implementing RFID<sup>3</sup> in the supply chain. IoT can be understood as an enabling framework for the interaction between a bundle of heterogeneous objects and also as a convergence of technologies. There are some required key functionalities to enable the interaction between "things" [3, 1]:

- Identification: Objects in the IoT are precisely identifiable by a defined scheme.
- Communication and Cooperation: Objects are capable to interact with each other or with resources in the net.
- Sensor: Objects can collect information about their environment.
- Storage: The object has an information storage that stores information about the object's history or/and its future.

<sup>1</sup> BIBA, Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Germany, email: gan, mer, bsr@biba.uni-bremen.de

<sup>2</sup> EPC-Global: Electronic Product Code, <http://www.epcglobalinc.org/home/>

<sup>3</sup> RFID: Radio Frequency Identification

- Actuating elements: IoT Objects are capable to act on their own without having a superordinate entity.
- User Interface: Adapted metaphors of usage have to be made available by the object.

## 2.2 Intelligent Products

An *Intelligent Product* can be understood as today's products enriched with competencies and abilities for decision-making and interaction with its environment. High level requirements of intelligent products are mentioned by several authors who reflect the demand on autonomous products. McFarlane [11] and Wong [20] describe the *Intelligent Product* as a physical and information based representation of an item which:

- possesses a unique identification.
- is capable of communication effectively with its environment.
- can retain or store data about itself.
- deploys a language to display its features, production, requirements, etc.
- is capable of participating in or making decisions relevant to its own destiny.

There are also very similar definitions of the properties of an Intelligent product from Kärkkäinen [9] and Ventä [17] but different in the perspective from which they look on the *Intelligent Product*. The focus of Kärkkäinen's description of *Intelligent Product* are logistics aspects in a supply chain.

## 2.3 Autonomous Logistics

In the field of *Autonomous Logistics*, cooperation and interaction are general requirements for complex systems where a high number of logistics objects are supposed to interact. Windt and Hülsmann [6] define autonomous cooperation and control as follows:

*Autonomous Control describes processes of decentralized decision-making in heterarchical structures. It presumes interacting elements in non-deterministic systems, which possess the capability and possibility to render decisions.*

*The objective of Autonomous Control is the achievement of increased robustness and positive emergence of the total system due to distributed and flexible coping with dynamics and complexity.*

One of the key elements of this definition is derived from the concept of positive emergence [8]. Emergence can be understood as the development of new structures or characteristics by the concurrence of simple elements in a complex system. As a consequence, positive emergence means that the concurrence of single elements leads to a better achievement of objectives of the total system than it is explicable by considering the behaviour of every single system element [8], [16]. Positive is meant to be an emergence that acts positive in the sense of the logistics system.

## 2.4 Multi-Agent-Systems

Intelligent autonomous objects require the integration of software agents and multi-agent-systems (MAS) which are a state-of-the-art

approach in implementing autonomous and interacting software systems. The autonomous decision maker are implemented and situated as software programs in a multi-agent environment and act on behalf of the real-world entities. The deployments of MAS imply capable agents as well as simple agents for distributed control in logistics systems and are one of the basic principles of our research. The ability of agents in MAS to communicate and coordinate with other agents enables them to solve complex tasks in cooperation (or competition) depending on their respective goals and abilities in distributed way. The decisions of an intelligent agent depend on its internal or "mental" state [14]. The presence of the current state and the knowledge on the current state of the world is a minimal requirement for the goal-oriented behaviour of an agent. Furthermore, the intelligent agent should be able to infer new knowledge from present knowledge by logical reasoning. Agents in a multi-agent environment interact in a standard way defined by FIPA<sup>4</sup>[4] which is a subsection of IEEE since 2005. In particular, the format and semantics of messages (Agent Communication language, ACL) sent between agents are defined by the FIPA standards [4]. Also the protocols for certain interaction processes based on speech act theory [15] are standardized by FIPA. In production logistics agents may be representatives of different logistic entities, e.g., products, assembly machines or hardware control items.

## 3 BUILDING BLOCKS

We present an ongoing implementation of a bundle of methods which are reflected in a material flow system with an applied production scenario. The material flow system is also being introduced into the research to ensure industrial conditions.

### 3.1 Hardware Abstraction Layer

The most relevant requirement of autonomous control is the ability of individual logistics entities to access context and environment data. Thus the ability to understand and process the data from data sources is the condition to build local decision-making systems [5]. For this purpose we used a "Hardware Abstraction Layer", which was developed for having a structured access to nearly any hardware of the system. The Hardware Abstraction Layer considered the findings from the point of view of data integration. Hans et al. [5] and also Hribernik [7] examined which aspects have to be considered from the point of view of data-integration in autonomous logistics networks. This gives freedom in terms of future extensions of the system.

### 3.2 Metal Cast RFID

One of the important steps towards autonomous parts is the unique identification possibility of autonomous objects. This can be attained by tagging or embedding auto-ID<sup>5</sup> technologies such as RFID. There are first prototypical integration of RFID tags at 125 kHz in the metal parts [2]. The implementation-scenario uses an automotive tail-light as Intelligent Product. The RFID tag was inserted while casting the tail light. This approach has the focus on enabling the products to be exactly identifiable and also autonomous from begin of their birth. Pille describes how to solve related challenges of this engineering process [12].

<sup>4</sup> FIPA: Foundation for Intelligent Physical Agents

<sup>5</sup> auto-ID: Automatic Identification

### 3.3 Decision Algorithms

To act autonomously, software agents are used as representatives of intelligent logistic objects, thus decision-making algorithms have to be implemented within the agents. For operating intelligent products in the mentioned assembly system, an algorithm basing on the “Product Type Corridor” is used. This algorithm is developed for assembly systems which produce more than one variant of products. The product moves along the product type corridor during the manufacturing and assembly process [19]. This means that there are variants to choose along the assembly process that allow the product to re-decide which product variant to target. This concept describes the available type variants that are currently possible considering the progress of production. The product is using the introduced method of autonomous product construction cycle for assembly systems. By deciding for a final type variant, the next possible production steps are identified. Therefore it is necessary to analyze the all-up situation, which calls for evaluation of every operation alternative [18]. This concept is a prerequisite for going into decision-making that is done with a model, which is capable to evaluate multicriterial status. This approach makes a multicriterial mathematical evaluation possible and is based on the fuzzy hierarchical aggregation [13]. The algorithm calculates one from several alternatives considering given criteria. Criteria are for instance waiting time at potential assembly stations, material in stocks of the stations and current customer orders.

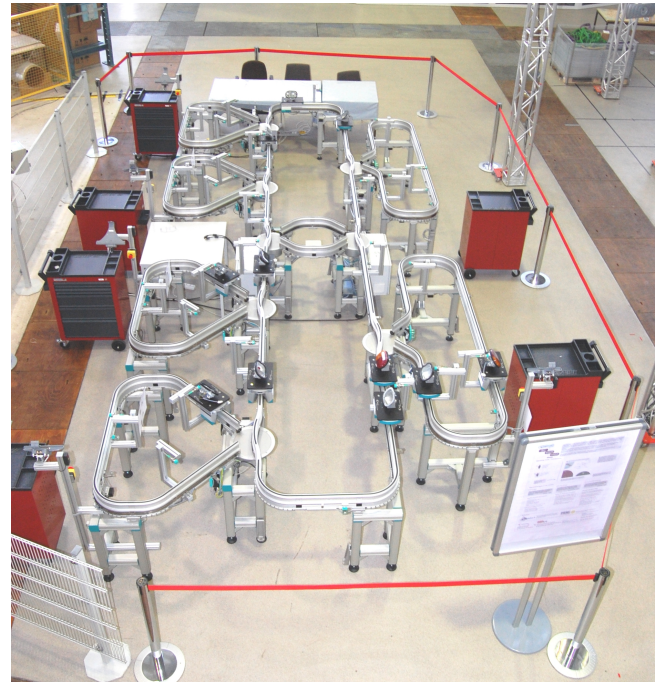
### 3.4 Hardware

For setting up a scenario that is comparable to real life machinery, a monorail conveying system is being introduced that works with self-propelling shuttles with a work piece holder capable to carry loads of up to 12kg (Figure 2). It is a modular system and gives the freedom of future extensions. The actual set-up of the monorail conveying system at the shop-floor of the BIBA<sup>6</sup> Institute allows the products to act flexible and to change the planned route by using the system integrated monorail-switches that offer multiple paths (Figure 1).

For the product it is then possible to remain on the main line or to deviate to a bypass. The implemented 125 kHz RFID technology was customized for our purposes to work with the casted RFID tags. RFID technology that is used in metals or even nearby is characterized by a low performance. The oscillating circuits of RFID tags detune in these environments, which has to be considered when tuning an antenna to work with metals.

## 4 FIRST IMPLEMENTATION

Induced by the amount of results coming out of the CRC 637 and the necessity to evaluate them, it was required to develop a platform that incorporates hardware such as RFID readers, a material flow system and on the top a software framework in order to integrate different methods and research results in a flexible manner. In this paper we describe the implementation of an autonomous control for manufacturing systems with the help of this developed framework. It is designed to have a user friendly interface. Different scenarios can be defined and edited by using the operator interface which is finally stored in a XML based configuration file. Editing a scenario includes the definition of final products to be manufactured, the manufacturing steps to be processed respectively the corresponding assembly



**Figure 1.** Mono-rail System



**Figure 2.** Shuttles with Intelligent products

<sup>6</sup> BIBA: Bremer Institut für Produktion und Logistik GmbH

stations, needed parts, type of hardware (e.g. auto-ID Systems) and finally the material flow layout.

#### 4.1 System Architecture

The described platform consists of different software packages that can be started in a distributed way on different machines. A scenario can be designed and started by using the operator interface. This action affects the starting of the multi-agent platform and creates the workshop agent that is equipped with the context information. The challenge of this agent is to create other relevant agents such as station agents depending on the available information. Furthermore the unique product agents will be created by the workshop agent at the first identification on the material flow system. Figure 3 shows the design of the system.

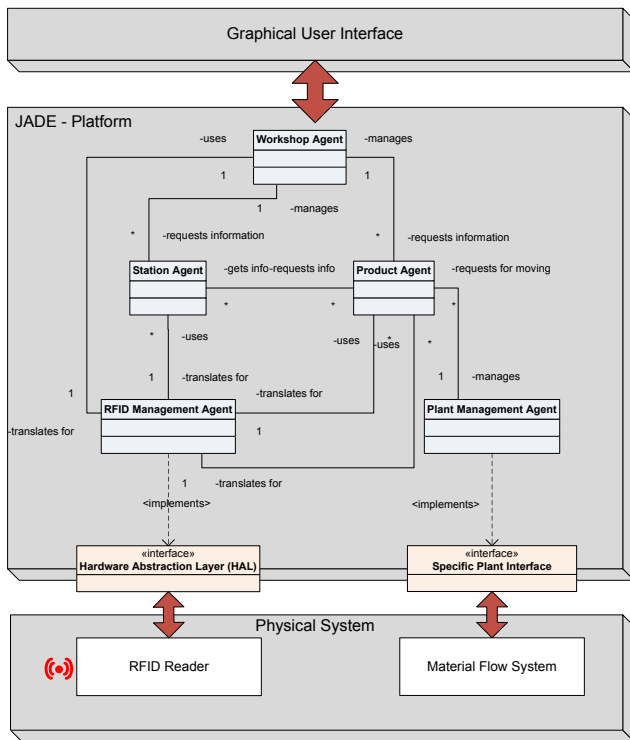


Figure 3. System Design

Based on MAS it seems to be appropriate to implement also the control software for hardware in form of software agents that pick up the signals, process them and communicate the information to other relevant agents. Up to now there are two hardware manager agents created, firstly for the RFID-Reader and secondly for the material flow system. These agents will be created automatically by starting the scenario. For supplying all created agents with context information, the configuration data is being send to all agents. The Hardware agents are triggered by real world signals (e.g. sensor signals) which induce them to broadcast information to the according product agents or station agents. As depicted in figure 3 product agents send requests to station agents as well as to workshop agent in order to gather necessary information for decision-making. The product agents also request the plant manager agent for any moving in the material flow system. The RFID management agent holds the con-

nection to the RFID Hardware over the Hardware Abstraction Layer (HAL) and delivers important information to other agents. This information contains the ID of the product as well as the the geographical position of the product.

#### 4.2 The Scenario

By using the developed framework we implemented a production scenario for investigating the applicability in the domain of production logistics. The scenario illustrates an autonomous assembly system for an automotive tail-light whereby the assembly itself is still designed to be a manual task. The autonomy refers to the decision-making of the specific products. The scenario has six stations; the starting station is implemented as the input/output for the material flow system where the semi-finished parts (metal cast part with integrated RFID) enter the assembly system. It is also used to take out the assembled/finished products. The other five stations are designed as assembly stations. The implemented assembly stations correspond to the assembly process and are designed to assemble bulbs (coloured and clear), seals and three types of diffusers.

The assembly process consists of five stages which are depicted in figure 4. The process starts with inserting the semi-finished metal-cast part into the material-flow system. The products, represented by software agents, are targeting a type variant (colored, clear or dark diffuser), which they choose on their own by considering customer orders. Orders can be edited in a separate user interface at any time, which affects then the behaviour of the products. All related processes of transport of the work piece and decisions are made by the products respectively their corresponding agents. The three variants require specific parts during the production process, which are then scheduled and chosen by the products as well.

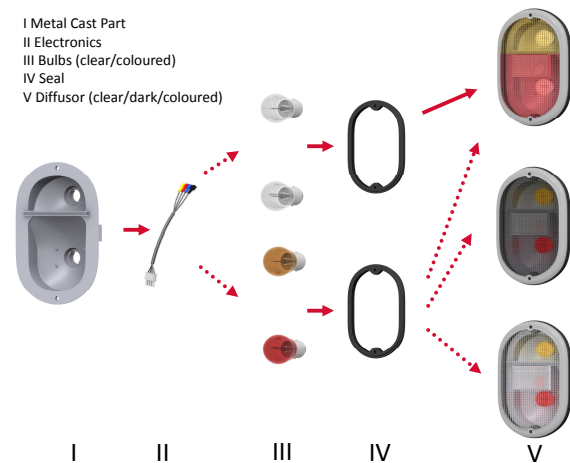


Figure 4. Assembly scenario

There are several possibilities to exert influence on the behaviour of the assembly scenario. Applying intended malfunctions or failures to the system or changing the customer orders force products to react autonomously to the new situation and make a new decision.

### 4.3 The intelligent product software agent

This implemented scenario is completely based on the multi-agent structure of Java Agent Development Toolkit JADE which is considered as the leading open-source agent platform in academia. So every real object such as casting parts or assembly stations are represented by distributed software agents which interact with each other. In general the product agent is mainly dedicated to decision-making and uses autonomous methods to route its real-world part through the assembly process using recent information. Depending on its current state it requests other agents for required information and thereupon makes new decision that may change its state. There are also other agents implemented to represent hardware which have a quite simple design and act without active decision making algorithms. The implementation of the product agents is based on the described method of the autonomous product construction cycle for assembly systems [19]. The agents have the challenge to decide for the optimal product type considering different context factors.

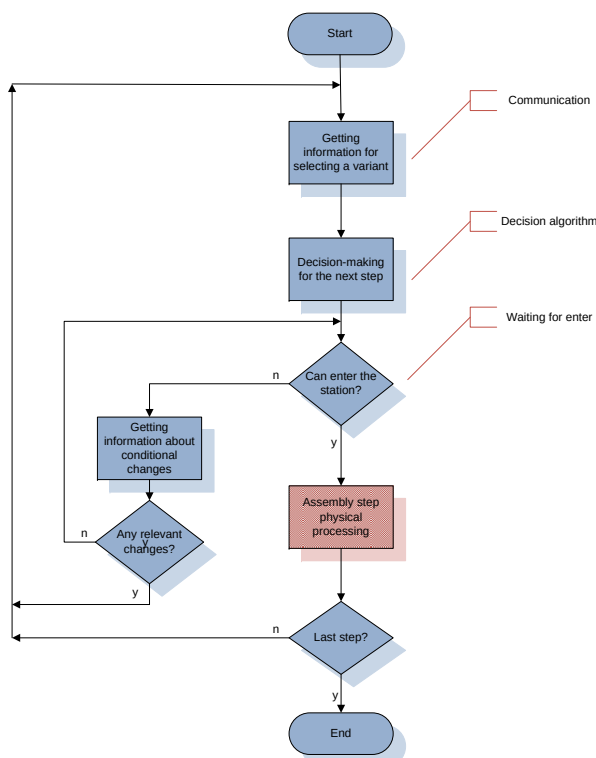


Figure 5. Scenario processing

The decision-making is in fact focused on choosing between the three product variants, which directly affects the next targeted production step respectively assembly station. These agents have to re-run the decision making process after each manufacturing step.

The product agents need important information for the decision-making such as current customer orders, waiting time at a potential assembly station and material in stocks of the stations. The needed information is gathered by communicating to other relevant agents such as the station agents, workshop agents and all other existing product agents. After having collected relevant data the product agent starts the decision-making process. There do exist logical (and physical) constraints that forbid products to choose the next production

processes randomly. The currently possible variant (constraint) and the scheduling to the next production step is determined by the implemented decision methods. The used decision-making algorithm is described in an own section. Figure 5 shows the process flow within the product agent.

## 5 CONCLUSIONS

In this paper we presented the first implementation of a decentralized control of an industrial material flow system with autonomous control methods through a multi-agent-system. The intelligent product agent becomes the centre of attraction and is enabled to make own decisions. The implementation in this demonstration platform shows that when having a product centric approach and not only having the control over the product, positive effects can be observed. It becomes obvious that the basic technology fundamentals for intelligent products do already exist. We believe that an emergence arises out of the decentralized approach. This becomes evident when applying intended malfunctions to the system or when conditions (e.g. customer orders) change. The products are able to react to the new situation without a central re-planning. We can state qualitatively that an increased robustness can be observed. Future will show more quantitative results, when metrics and operating figures, such as cycle times, will be elicited with the system.

## ACKNOWLEDGEMENTS

This research is funded by the German Research Foundation (DFG) as the Collaborative Research Centre 637 “Autonomous Cooperating Logistic Processes-A Paradigm Shift and its Limitations” (SFB 637).

## REFERENCES

- [1] Internet of things strategic research roadmap. online, 2009. [http://www.grifs-project.eu/data/File/CERP-IoT%20SRA\\_IoT.v11.pdf](http://www.grifs-project.eu/data/File/CERP-IoT%20SRA_IoT.v11.pdf) visited on May 31st 2010.
- [2] M. Busse, F.-J. Wöstmann, Th. Müller, T. Melz, and P. Spies, ‘Intelligente druckgussteile’, in *Metall11*, pp. 738–741, (2006).
- [3] E. Fleisch and F. Thiesse, ‘Internet der dinge’, in *Kurbel, K. and Becker, J. and Gronau, N and Sinz, E. and Suhl, L (eds.) and Enzyklopädie der Wirtschaftsinformatik - Online Lexikon*, <http://www.enzyklopaedie-der-wirtschaftsinformatik.de>.
- [4] Foundation for Intelligent Physical Agents. Fipa standard status specifications. online, 2002. <http://www.fipa.org/repository/standardspecs.html>.
- [5] C. Hans, K.A. Hribernik, and K.-D. Thoben, ‘An approach for the integration of data within complex logistics systems’, in *Dynamics in Logistics. First International Conference. LDIC 2007*, eds., H. D. Haasis, H.-J. Kreowski, and B. Scholz-Reiter, pp. 381–389, Heidelberg, (2008). Springer.
- [6] M. Hülsmann and K. Windt, *Understanding Autonomous Cooperation & Control - The Impact of Autonomy on Management, Information, Communication, and Material Flow*, Springer, Berlin, 2007.
- [7] K. Hribernik, C. Hans, and K.-D. Thoben, ‘The application of the epcglobal framework architecture to autonomous control in logistics’, in *2nd International Conference on Dynamics in Logistics*. Springer, (2009).
- [8] G. Küppers and W. Krohn, ‘Selbstorganisation. zum stand einer theorie in den wissenschaften’, in *Krohn, W., Küppers, G (Ed.) "Emergenz: Die Entstehung von Ordnung, Organisation und Bedeutung"*, 2. Aufl., pp. 7–26.
- [9] M. Kärkkäinen, J. Holmström, K. Främling, and K. Artto, ‘Intelligent products - a step towards a more effective project delivery chain’, in *Computers in Industry 50* (2), pp. 141–151, (2003).
- [10] F. Mattern and C. Flörkemeier, ‘Vom internet der computer zum internet der dinge’, in *Informatik-Spektrum, Vol. 33, Nr. 2*, pp. 107–121, (2010).

- [11] D. McFarlane, S. Sarma, J. L. Chirn, C. Y. Wong, and K. Ashton, 'Auto id systems and intelligent manufacturing control', in *Engineering Applications of Artificial Intelligence* 16 (4), pp. 365–376, (2003).
- [12] C. Pille, 'Produktidentifikation, intralogistik und plagiatenschutz - rfid-integration in gussbauteile', in *BDG-Fachtagung Gussteilkennzeichnung. Methoden und Datenmanagement - Praxisberichte*, VDG-Akademie, pp. V/1–V/4, Essen, Germany.
- [13] H. Rekersbrink, B. Ludwig, and B. Scholz-Reiter, 'Entscheidungen selbststeuernder logistischer objekte', *Industrie Management*, 23(4), 25–30, (2007).
- [14] S. J. Russell and P. Norvig, 'Artificial intelligence. a modern approach', 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2003, (2003).
- [15] J.R. Searle, 'Speech acts: An essay in the philosophy of language', Cambridge University Press, (1969).
- [16] K. Ueda, A. Lengyel, and L. Hatano, 'Emergent synthesis approaches to control and planning in make to order manufacturing environments', in *Annals of the CIRP* 53(2004)1, pp. 385–388, (2004).
- [17] O Ventä, 'Intelligent products and systems', in *Technology Theme - Final Report*. VTT, Espoo: VTT Publications, p. 304, (2007).
- [18] B.-L. Wenning, 'Methoden zur modelbildung in der technikkbewertung', in *CUTEC-Schriftenreihe Nr. 18*, Clausthal-Zellerfeld 1995, (1995).
- [19] K. Windt and O. Jeken, 'Allocation flexibility - a new flexibility type as an enabler for autonomous control in production logistics', in *42nd CIRP Conference on Manufacturing Systems*, (2009).
- [20] C. Y. Wong, Duncan McFarlane, Ahmad Zaharudin, and V. Agarwal, 'The intelligent product driven supply chain', in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2002, Tunisia: IEEE, (2002).

# Context-based Learning of Adaptive Vehicle Dispatching Strategies

Gerd Heiserich<sup>1</sup> and Ludger Overmeyer<sup>2</sup>

**Abstract.** The reinforcement learning framework has been used for finding and improving solutions to complex scheduling and resource allocation problems. However, in many real-world applications, state or action sets become very large due to the curse of dimensionality. This leads to approaches involving an approximation of the value function rather than a complete representation in a table. This contribution shows an application of reinforcement learning for a scenario in internal logistics involving multiple transportation vehicles operating in a common workspace. We show a solution for learning an adaptive vehicle dispatching strategy, which is suitable for a wide range of environmental conditions. This solution uses a context-based state representation that is scalable and allows a tabular representation of the value function even for large problem instances. The performance of the learned behaviour is evaluated by using a simulation model and is compared to common heuristic dispatching strategies.

## 1 INTRODUCTION

The ongoing trend towards mass customisation with an increasing number of product variants leads to the objective of agile and transformable systems in both production and transportation technology [17]. Quickly changing requirements have an impact on internal and external transportation processes, because production sites and machinery are generally served using just-in-time and just-in-sequence concepts, which hardly allow any temporal tolerances. For this reason, future logistic systems have to offer short reaction times and must react autonomously to unplanned scenarios.

In a supply network, numerous unanticipated events can occur, which potentially affect the entire system. Examples include failure of production equipment, changing amounts of demanded goods or uncertain transportation times due to traffic conditions. Finding optimal schedules for a complex structure with multiple participants such as a supply network is most often intractable due to the high number of degrees of freedom. Some authors even claim that a totally integrated supply chain management as a paradigm is generally not a viable approach [3].

Scheduling problems in general are known to be NP-hard. For this reason, the algorithms used for large real-world applications do not try to find a global optimum, but rather look for feasible solutions. Recently, methods from artificial intelligence have successfully been used for solving complex planning and scheduling problems in industrial applications. Examples include the stochastic lot scheduling

problem [18] and job sequencing for a given setup of production machinery [4].

Several research groups have specifically examined problems arising in transportation and logistics. The aim is to create transportation systems which are able to cope with variable environmental conditions without the need for reconfiguration or reprogramming. This leads to two major directions of research: On one hand, mechanical and control issues are covered. The aim is to construct flexible and recombining components capable of performing a variety of tasks. Examples for this direction of research are flexible conveyor modules as shown in [13], where the modules can be arbitrarily recombined and are able to autonomously set up the required configuration settings. An innovative mechanical concept using small-scaled modules for the transportation of packaged goods is introduced in [16]. A control architecture for transportation systems, which is based on hierarchical finite state machines with transition functions on different abstraction levels, is described in [15].

The second direction of research concentrates on the operational view of transportation systems in terms of a more abstract description of behaviour. Discussed issues include decision making and routing for vehicles in stochastic environments. This is closely related to the stochastic vehicle routing problem (SVRP), where a continuous appearance or cancellation of assignments is possible (see for example [5], [6]). In [20] a method for automatic adaptation of a decision-making algorithm for servicing stochastic customer requests is described. [19] shows a distributed routing algorithm for a fleet of vehicles which have to process a certain workload in a logistic net.

All the work mentioned above deals with external transportation processes with typical travel times in the range of several hours. In contrast, this article considers an internal transportation scenario, which is found at warehouses or dispatching points of shipping agencies. In these cases, transportation times between points of delivery are a few minutes at most. Moreover, dispatching points often have to deal with uncertain information about arrival times of inbound trucks. For both reasons, continuous re-planning on an operational level is not a feasible approach. The objective of this work is to use a learning approach to establish an adaptive dispatching strategy that makes online decisions based on the current state of the system; “adaptive” meaning that the system is able to cope with stochastic assignments at changing input rates.

[1] shows the portions of time of each partial process in a production and delivery chain. According to this, waiting times represent as much as 75 % of the total time and thus have by far the largest influence. This implies that there is a considerable potential for optimisation of transportation operation.

In the following section, the fundamental ideas of reinforcement learning are briefly reviewed. Section 3 gives a detailed explanation

<sup>1</sup> Institute of Transport and Automation Technology (ITA), Leibniz University of Hannover, Germany, email: gerd.heiserich@ita.uni-hannover.de

<sup>2</sup> Institute of Transport and Automation Technology (ITA), Leibniz University of Hannover, Germany, email: ludger.overmeyer@ita.uni-hannover.de

of the considered application scenario. After that, the applied learning algorithm for finding a dispatching strategy is discussed. The main contribution of this article is the development of a general state representation for the application, which makes the approach scalable to large problem instances. Finally, simulation results for the learned dispatching strategy in comparison to heuristic strategies are presented.

## 2 REINFORCEMENT LEARNING

The reinforcement learning framework deals with a class of problems in which a system - often called an agent - is operating in an environment while aiming at achieving a desired goal. At this point, we will just introduce the basic formula symbols and definitions that will be used throughout this article. Excellent introductions to theoretical foundations, methods and applications can be found in [7] and [21].

Reinforcement learning problems are modelled as Markov Decision Processes (MDP). This means the agent can reside in one state from a set of possible states. In each of these states, it can choose from a number of possible actions, which take it to another state. Actions are associated with rewards which contain information about their utility for the agent. The state transitions possess the Markov property: In a known system state, all the transition properties to following states depend only on the current state, and not on any previous states or chosen actions. Formally, an MDP is described by a tuple  $(S, A, p, r)$  with

- $S$  : set of system states
- $A$  : set of actions
- $p : S \times A \times S \rightarrow [0, 1]$ : transition probabilities
- $r : S \times A \rightarrow \mathbb{R}$ : reward function

If the agent is in a state  $s \in S$  and executes an action  $a \in A$ , it will proceed to state  $s'$  with the probability  $p(s, a, s') = P(s' | s, a)$  and will receive the reward  $r(s, a)$ .

The aim is to find a policy  $\pi : S \rightarrow A$  which is a function that maps a system state to an action. If a policy  $\pi^*$  maximizes the total reward over a long period of time, it is called optimal. There can be more than one policy to meet this condition. For evaluation of policies, the majority of approaches estimate a value function to describe the value of states or state-action pairs. This value function is denoted by

$$Q : S \times A \rightarrow \mathbb{R}$$

and returns the expected value of the future reward for taking an action  $a \in A$  in a state  $s \in S$  and following the policy  $\pi$  thereafter. For small state and action sets, the value function can be represented as a complete table with one entry for each state-action pair. For many real applications, states and actions are expressed as  $n$ -dimensional vectors of some features. This has the consequence that state and action sets grow exponentially in the number of components  $n$ . This effect is often called the curse of dimensionality and is a considerable challenge when solving real-world problems.

A possible approach to find the value function  $Q(s, a)$  is to estimate it from experience. In this case, the learning system keeps an average value of all the received rewards for each state-action pair. This average will converge to the true value if each state-action pair is visited often enough. Estimation methods involving averaging over samples are referred to as Monte Carlo methods. Other approaches are based on learning new estimates based on existing estimates.

There is also a considerable amount of research concerning transferring these methods to the multi-agent case, i.e. to learn a behaviour for a group of several cooperating learning systems (see [8], [9]). If all agents can observe their own and all other agents' actions, the problem can be reduced to the single-agent case. In our work, we will explicitly allow communication between subsystems. This leads to a global improvement of the behaviour without the need for a central planning strategy.

## 3 APPLICATION SCENARIO

### 3.1 Principle of Cross-docking

As an application scenario for the investigations presented here, a cross-docking distribution centre is considered. Cross-docking denotes the handling of transport units at a dispatching point with little or no storage of goods. The units are instead delivered pre-sorted and labelled by the suppliers and then recombined and directly forwarded after arrival. Consequently, all processes of transfer and removal of storage or stock rotation can be omitted. This leads to a reduction of operating costs in comparison to a traditional warehouse. The operating principle is illustrated in figure 1.

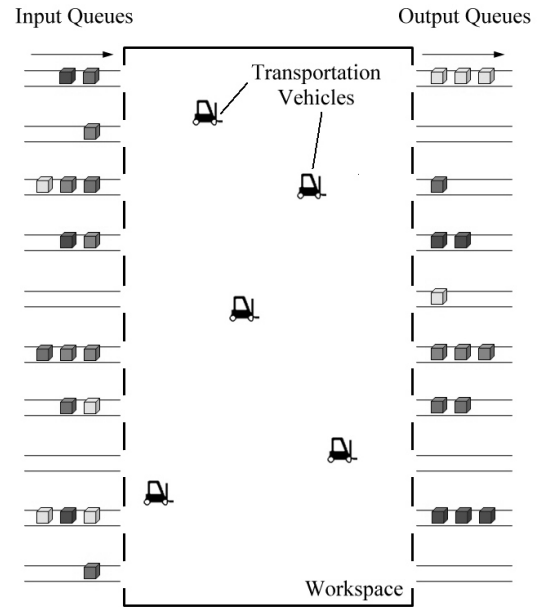


Figure 1. Cross-docking application scenario

Cross-docking is a distribution strategy which combines the advantages of supply via intermediate storage facilities on the one hand and direct supply on the other hand. Operators of cross-docking facilities are shipping agencies and large retail store chains [11]. Research has focused on several aspects such as the optimal position of a cross-docking facility in a supply network, the geometry of the workspace and scheduling and door assignment of inbound and outbound trucks [10].

During operation, transport units are being received at the input queues (pickup points) and have to be assigned to corresponding output queues (delivery points). The input and output queues correspond to loading ramps or doors where trucks dock from the outside to be

loaded and unloaded. Transportation within the distribution centre is typically performed by forklift trucks due to their capability of handling a large variety of goods.

In practical applications, the inbound and outbound doors are each assigned to a fixed destination to be served by the corresponding trucks. This assignment is usually persistent over a longer period of time such as several months. An adequate door assignment can help reducing the average distances for deliveries within the workspace. However, optimizing the assignments in short intervals of time is unfeasible for two major reasons: First, the employees can work more efficiently when being used to the assignments. Second, continuous re-planning would require precise and complete data about amount, type and destination of the transport units to be transported in the future. Such data is generally not available in practice.

Arrival times of trucks are only approximately known in advance in terms of time windows. Frequently, trucks will not be able to adhere to their itinerary, which leads to short-term changes of planned sequences and door allocations. These effects depend on external influences and are random processes from the cross-docking system's point of view. For this reason, the arrival of internal transportation requests will be considered an external factor that cannot be controlled. The task of the cross-docking system as a whole is then to react to these short-term changes and to find a favourable internal workflow in spite of unknown environmental conditions.

The application scenario is distinguished by quickly changing requirements: the arrival rate of new transportation requests is very unsteady during a given observation period. There are typical peak times during a day, a week or a year, but these are only significant in terms of statistic mean values and are not well suited for forecasting single events on the operational level.

For modelling the scenario, the transportation is assumed to take place in a rectangular workspace with  $N_p$  pickup points on one side and  $N_d$  delivery points on the opposite side, both equally spaced. Each pickup point corresponds to an input queue which may be occupied by a number of transport units. Within the workspace,  $N_v$  vehicles can freely move, i.e. there are no fixed lanes to drive on. Transport units are labelled by their destination, which is always exactly one delivery point, i.e. there is no choice from a set of possible delivery points. Furthermore, each vehicle can transport one transport unit at a time, so there are continuously alternating loaded and unloaded trips.

The presented application is related to dynamic vehicle routing or pickup and delivery problems on the one hand and to scheduling problems on the other hand, but shows some specific differences with respect to all of these: Unlike the cross-docking scenario, vehicle routing problems assume that service trips start and end at a central common depot [5]. One-to-one pickup and delivery problems assume transportation assignments from a single pickup point to a single delivery point, but are typically located in urban environments with large distances. In this case, re-optimisation of an existing plan using exact or heuristic methods is feasible for the given time constraints. In contrast, the cross-docking scenario has much shorter transportation times and a much higher arrival rate of new transportation requests. Furthermore, it is much more challenging to maintain a consistent representation of the global system state in real-time than for slowly changing environments.

Scheduling and resource allocation problems generally consider temporal constraints between activities, whereas in vehicle routing and cross-docking problems the visits at pickup points are independent of each other [2]. Moreover, vehicle routing problems aim at minimizing the total traveled distance. In scheduling applications the

processing times are known, and the objective is to optimize an average waiting time or the overall throughput. These criteria are also relevant for cross-docking.

To sum this up, vehicle routing problems focus primarily on spatial aspects, whereas scheduling problems focus on temporal relationships. Cross-docking is closer related to vehicle routing than to scheduling with regard to additional constraints. On the other hand, cross-docking is closer related to scheduling with regard to the relevant optimisation criteria.

### 3.2 Heuristic Dispatching Strategies

The problem of assigning transportation vehicles to requests in manufacturing environments and internal logistics is often solved by using simple heuristic rules [14]. These rules are usually designed to fulfill a single optimisation criterion and are optimal only for special cases. This statement will be illustrated for the cross-docking vehicle dispatching task by means of 3 heuristic dispatching strategies. These same strategies will also be used for evaluation and benchmarking later in this contribution.

**Shortest Distance** After completion of a transportation assignment at a delivery point, an unloaded trip will be carried out to the nearest pickup point with a vehicle demand greater than zero. This heuristic minimizes the total length of unloaded trips, but requires waiting transport units being kept at each pickup point. This leads to an increase of waiting times before pickup. This strategy maximizes the throughput on the condition that all input queues contain at least one transport unit, which cannot be guaranteed in the general case.

**Highest Demand** After completion of an assignment at a delivery point, an unloaded trip will be carried out to the pickup point with the highest vehicle demand, even if this pickup point is at a farther distance than other pickup points. This heuristic minimizes the length of the input queues, but leads to longer distances for the unloaded trips and to a decrease of throughput due to lower capacity utilisation.

**Remaining processing time** New transportation requests are immediately assigned to the vehicle with the minimum sum of remaining processing times, i.e. to the vehicle which can serve the request first. A disadvantage of this strategy is that an established plan might not be optimal any more at the time of execution, because new transportation requests can have arrived in the meantime. To overcome this problem, continuous re-planning would have to be introduced, which was already pointed out to be unfeasible in this scenario. The strategy also requires an estimation of all the remaining processing times, i.e. precise information about the position of all vehicles in the workspace. This information is usually not available in real systems today.

A consequence of these considerations is that the optimisation criteria *capacity utilisation* and *length of input queues* are not compatible with each other. The *average length of input queues* and the *average waiting time* are equivalent due to Little's Law [12].

An approach to optimize the behaviour beyond these heuristic approaches results from the unsteady arrival of new transportation requests, which can only be described by an average rate over a given interval of time. In fact, even this rate is not a constant, but typically depends on the time of day, the day of the week and the calendar month. In practice, the necessary number of vehicles is calculated using statistic mean values of input rate, transportation cycle times and

availability of equipment. For a given period of time such as a day or a shift, there is a certain amount of staff available according to prior manpower planning. A short-term adjustment of the available technical and human resources is hardly possible. As a consequence, there will always be phases of lower workload (or overcapacities respectively), during which the system could for instance process longer input queues at the cost of a sub-optimal capacity utilisation. This gives a frame for optimisation of vehicle assignment on the operational level. The question is then how to develop a vehicle dispatching strategy, which can efficiently handle progressions of incoming requests that are not precisely known in advance.

## 4 LEARNING A STRATEGY

### 4.1 Episodic Structure of the Task

The task is modelled as a decision process with states and actions according to section 2. From a transportation vehicle's point of view, the operation process consists of cyclic episodes of alternating loaded and unloaded trips. Each episode starts with picking up the first waiting transport unit from an input queue. At this point, there are no alternatives to choose from, since the destination point is defined by the transport unit. The unit is then transferred to a destination point. After delivery, a new destination (i.e. pickup point) has to be chosen for the following unloaded trip. After arriving at the chosen pickup point and pickup of a transport unit, the next episode begins. Hence, actions are chosen depending on the current system state each time a loaded trip was completed. The usefulness of this decision turns out when arriving at the chosen pickup point.

We define negative rewards for each time step of an unloaded trip and positive rewards for picking up transport units at a pickup point. These positive rewards are weighted by the length of the corresponding queue, in order to make a reduction of long queues more attractive. The value function is updated at the end of each episode with the accumulated reward. A special property of the application considered here is that it is an asynchronous decision process: Transportation vehicles are normally busy for longer periods of time during which they do not have to make any decisions. Decision events take place at randomly distributed intervals over time.

In phases of low workloads, vehicles might idle, because all waiting assignments are being served already. In this case it seems to be more appropriate to choose an arbitrary pickup point as a waiting position than to stay at the delivery point where the last transport was completed, in order to be able to serve a new transportation assignment more quickly. In a real system, the number of vehicles  $N_v$  is typically smaller than the number of pickup points  $N_p$ , so even if all vehicles were waiting at pickup points, the next assignment might still appear at a position with no waiting vehicle. For this reason, changing between waiting positions must also be coordinated between the vehicles. This behaviour can be understood as another dispatching strategy, for which a value function is required. As it turns out, the same learned value function can be successfully used for both dispatching strategies.

### 4.2 Learning Algorithm

The aim of the method which is presented here is to automatically find a set of decision rules for vehicle dispatching by using a simulated environment. In this environment, properties such as the geometry of the workspace, the number of vehicles, and the number of pickup and delivery points can be chosen by the user. The result of

the simulation is a behaviour which can be transferred to a real-world system.

The idea of the learning algorithm is to estimate the value  $Q(s, a)$  of all state-action pairs by averaging the obtained rewards. This is known as Monte Carlo Sampling [21]. Monte Carlo methods have the advantage that a learning system can learn a behaviour based on experience by interaction with the environment; no model of the environment with explicitly known probability distributions for state transitions is required.

For the following investigations, we are interested in the system behaviour under the condition of unknown arrival times of transportation requests. For this reason, we assume no *a priori* knowledge, but rather model the arrival of new requests as a Poisson process with the input rate  $\lambda$ . This corresponds to the average number of new transportation requests per time step. This random process is used both for training and for evaluation.

In order to guarantee convergence of the algorithm, each state-action pair must be visited sufficiently often. For this reason, occasionally actions have to be chosen that are not considered optimal. This is known as the exploration-exploitation dilemma. For our investigations, actions are chosen by  $\varepsilon$ -greedy action selection, i.e.

$$\pi(s) = \begin{cases} \arg \max_a Q(s, a) & \text{with probability } 1 - \varepsilon \\ \text{random } a \in A & \text{with probability } \varepsilon \end{cases}$$

We chose  $\varepsilon = 0.1$  for learning. For evaluation of the learned behaviour,  $\varepsilon$  was set to zero to prevent further exploration.

### 4.3 State and Action Representation

The value function is to be represented as a table, in which the mean values of the observed rewards for all visited state-action pairs are stored along with the number of visits. This table has a maximum of  $L(Q) = |S| \cdot |A|$  entries. In the given application scenario, there are as many possible actions as there are pickup points, i.e.  $|A| = N_p$ . The actions can be easily represented by an index number with respect to a sorted list of pickup points.

In contrast, the state description must contain information about both the occupancy of input queues and about the state of all vehicles. Therefore, a straightforward approach for a state representation is the tuple

$$s = (v_1, v_2, \dots, v_{N_v}, q_1, q_2, \dots, q_{N_p})$$

with  $v_i$  the state of the vehicle  $i$  and  $q_j$  the occupancy of the input queue  $j$ . The occupancy  $q_j$  is an integer number from the set  $S_p = \{0, \dots, \hat{q}\}$  where  $\hat{q}$  denotes the maximum buffer capacity. The vehicle state is given by the current position, orientation and loading state. The position and orientation are continuous variables, leading to an infinite number of possible states. This is a considerable difficulty for handling the decision process. To overcome this problem, the set of vehicle states can be transformed by discretisation to a finite set  $S_v$ . Then the total state set has the finite cardinality

$$|S| = |S_v|^{N_v} \cdot |S_p|^{N_p}$$

This grows exponentially with both the number of vehicles and the number of pickup points and is therefore only feasible for very small problem instances.

For this reason, the objective of this work is to develop a scalable state representation by reducing the dimensionality. To achieve this, we take advantage of two symmetric properties of the problem. The first of these properties is that all vehicles have the same features and can be interchanged. The second property is that from a vehicle's

point of view, the characteristics of a pickup point are its vehicle demand and its distance from the vehicle's current position. Therefore, the pickup points are also interchangeable with respect to the value function, i.e. two pickup points with the same demand and distance should have the same estimated value.

According to this, any vehicle  $i$  that has to choose an action can as well use the following parametrisation of the value function:

$$Q(s, a) = Q(\text{demand}(a), \text{distance}(v_i, a), \mathbf{v}_{rem}(s), \mathbf{q}_{rem}(s))$$

where the vehicle states of all remaining vehicles  $i' \neq i$  are concentrated in the vector  $\mathbf{v}_{rem} \in S_v^{N_v-1}$ . The properties of all remaining pickup points  $j \neq a$  are concentrated in a vector  $\mathbf{q}_{rem} \in S_p^{N_p-1}$ . We call the vectors  $\mathbf{v}_{rem}$  and  $\mathbf{q}_{rem}$  the "context" of a decision.

So far, this is only an alternative parametrisation and does not yet lead to a reduction of the number of dimensions. The idea is to find a function

$$f_c : S_v^{N_v-1} \times S_p^{N_p-1} \rightarrow \mathbb{R}$$

where  $f_c(\mathbf{v}_{rem}(s), \mathbf{q}_{rem}(s)) = g_c(s)$  is an indicator or a characteristic number, representing an abstraction of the actual context. This function is a projection of the context part of the state vector  $s$  to a subspace and provides the desired reduction of dimensionality. Therefore, the value function can be expressed as

$$Q(s, a) = Q(\text{demand}(a), \text{distance}(v_i, a), g_c(s))$$

From the theoretical considerations shown in section 3.2 and from experiences with the described heuristics it follows that the vehicles' behaviour should depend on the current degree of workload. A possible measure for this is the **total occupancy** of the input queues. This represents the amount of work to do within a certain time horizon. Another measure is the loading state of the vehicles. This can be represented by the number of currently **busy vehicles**, which will not be available for another transportation assignment until some future point in time. Both measures can also be combined to the **total demand**, which is the difference of the total occupancy and the number of vehicles currently executing an unloaded trip to a pickup point.

In all three cases, the context function  $g_c$  maps the entire context to a single integer number. If for instance the number of busy vehicles is used as the context,  $g_c$  will yield values from the set  $\{0, \dots, N_v - 1\}$  because the deciding vehicle is never busy. From the parametrisation given above follows that the state set with respect to  $g_c$  has the cardinality

$$|S_{g_c}| = (\hat{q} + N_v + 1) \cdot N_p \cdot N_v$$

which has a linear growth with the number of pickup points and a quadratic growth with the number of vehicles. We assume symmetry of pickup and destination points, so there are  $N_p$  possible distances from a destination point to a pickup point. For non-symmetric cases the growth is still linear, however with a different second factor.

The major questions arising are if this is a valid simplification and which approach to chose for representing the degree of workload. These questions were investigated using a simulation model. The results are shown in the next section.

## 5 SIMULATION RESULTS

Policies were learned according to the method described in section 4 during a learning phase of  $10^7$  simulation steps at an input rate of  $\lambda = 0.06$ . This corresponds to an expected value of 600000 transportation operations being carried out during this period of time. As

an evaluation criterion we use the average waiting time of new assignments from their appearance in an input queue until pickup by a transportation vehicle. Minimizing the average waiting time also minimizes the passage time through the dispatching point. This is favourable for serving the adjacent external transportation processes and also for minimizing the duration of stay of inbound and outbound trucks.

The throughput is less suitable for benchmarking, because for the steady state, the throughput will be equal to the input rate  $\lambda$  if the cross-docking system is properly dimensioned. An input rate higher than the transportation capacity will lead to a capacity overload. In this case, all input queues will be filled to a maximum, and it is already known that the *shortest distance* heuristic is best suitable for this case.

Table 1 shows a comparison of the three approaches to represent the context for an example instance of the problem with  $N_p = N_d = 10$  and  $N_v = 5$ . The number of busy vehicles clearly yields the best results for both the average waiting time at the end of the learning phase and for the steady state. Surprisingly, this is also the approach with the shortest decision table.

**Table 1.** Results for different context functions

| Context         | $\bar{t}_{w,0}$ | $L(Q)$ | $\bar{t}_w$ |
|-----------------|-----------------|--------|-------------|
| total occupancy | 54.5            | 3083   | 49.5        |
| busy vehicles   | <b>49.8</b>     | 700    | <b>43.9</b> |
| total demand    | 57.2            | 4257   | 47.2        |

$\bar{t}_{w,0}$ : average waiting time after learning phase

$L(Q)$ : length of value table

$\bar{t}_w$ : average waiting time (steady state)

The number of resulting table entries after the learning phase is shown in table 2 for different problem instances. Investigations were successfully carried out up to a size of  $N_p = N_d = 30$  (denominated "30 x 30" in the table) with 10 vehicles. The growth of  $L(Q)$  is very moderate, so even for large problem instances, a tabular representation of the value function is well tractable.

**Table 2.** Length of value table  $L(Q)$  for different problem instances

| $N_p \times N_d$ | Number of vehicles $N_v$ |      |      |      |
|------------------|--------------------------|------|------|------|
|                  | 4                        | 6    | 8    | 10   |
| 10 x 10          | 588                      | 857  | 1308 | 1789 |
| 15 x 15          | 764                      | 1116 | 1589 | 1974 |
| 20 x 20          | 901                      | 1387 | 1973 | 2573 |
| 25 x 25          | 970                      | 1582 | 2222 | 2919 |
| 30 x 30          | 986                      | 1592 | 2270 | 2945 |

As our final result, figure 2 shows a comparison of the average waiting times for the 3 heuristic strategies and the strategy obtained by reinforcement learning. Each point represents the resulting average waiting time after 200000 simulations steps, averaged over 20 simulation runs. The diagram shows that the *shortest distance* heuristic achieves a better performance than the *remaining processing time* heuristic for medium and high input rates, whereas for low input rates it is vice versa. The learned strategy is superior to the heuristics for all input rates except for a very small range near the maximum workload. The *highest demand* heuristic is only favourable for very low input rates or for problem instances with a very small number of

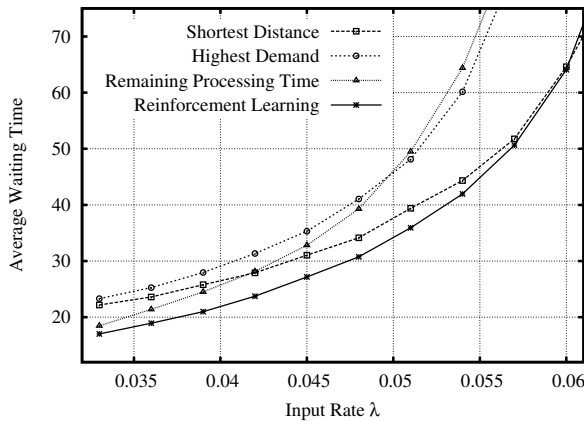


Figure 2. Performance of dispatching strategies for different input rates

pickup points (not shown in the figure). On average over the examined range of input rates, the learned strategy leads to a reduction of the average waiting time of 8.3 % with respect to the corresponding best heuristic strategies.

## 6 CONCLUSION

We have shown how reinforcement learning can be applied to find efficient dispatching strategies for a number of transportation vehicles in a cross-docking application scenario. The method is able to derive a behaviour from simulated experience and does not require an explicit representation of state transition probabilities of the environment. For a system configuration, whose properties can be specified by the user, the simulation model will generate a corresponding decision table.

We addressed the exponential growth of the state set by introducing a projection of the context description to a subspace. This method reduces the dimensionality of the state set. Using this context-based state representation, which estimates the general value of state-action pairs, our approach scales very well to problem instances of a realistic size. The performance of the adaptive strategy obtained by reinforcement learning was shown to yield a better performance than commonly used heuristic dispatching strategies on random conditions over a wide range of input rates.

This is certainly a rather specific approach which is tailored to the considered application domain. In particular, we took advantage of both the symmetry of the workspace and of the interchangeability of the individual transportation systems. However, the idea of using context descriptions might be extended to more general problems. An objective of our future research is to include the process of finding an appropriate state representation in the learning algorithm in order to automatically generate an appropriate context description for a given problem domain.

## ACKNOWLEDGEMENTS

This contribution was prepared within the research project “Cognitive Logistic Networks (CogniLog)” and supported by the German federal state of Lower Saxony with funds of the European Regional Development Fund (ERDF).

## References

- [1] Dieter Arnold and Kai Furmans, *Materialfluss in Logistiksystemen*, Springer Berlin Heidelberg, 5th edn., 2007.
- [2] J. Christopher Beck, Patrick Prosser, and Evgeny Selensky, ‘Vehicle routing and job shop scheduling: What’s the difference?’, in *Proc. 13th Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 267–276, Trento (Italy), (2003).
- [3] Wolf-Rüdiger Bretzke, ‘Supply chain management: notes on the capability and the limitations of a modern logistic paradigm’, *Logistics Research*, **1**(2), 71–82, (2009).
- [4] Balázs Csanád Csáji, László Monostori, and Botond Kádár, ‘Reinforcement learning in a distributed market-based production control system’, *Advanced Engineering Informatics*, **20**, 279–288, (2006).
- [5] Michel Gendreau, François Guertin, Jean-Yves Potvin, and René Séguin, ‘Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries’, *Transportation Research Part C*, **14**(3), 157–174, (2006).
- [6] Soumia Ichoua, Michel Gendreau, and Jean-Yves Potvin, ‘Vehicle dispatching with time-dependent travel times’, *European Journal of Operational Research*, **144**, 379–396, (2003).
- [7] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore, ‘Reinforcement learning: A survey’, *Journal of Artificial Intelligence Research*, **4**, 237–285, (1996).
- [8] Martin Lauer and Martin Riedmiller, ‘An algorithm for distributed reinforcement learning in cooperative multi-agent systems’, in *Proc. 17th Int. Conference on Machine Learning*, pp. 535–542, Stanford (CA), (2000).
- [9] Martin Lauer and Martin Riedmiller, ‘Reinforcement learning for stochastic cooperative multi-agent-systems’, in *Proc. 3rd Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, pp. 1516–1517, (2004).
- [10] Shayla Ley and Sherif Elfayoumi, ‘Cross dock scheduling using genetic algorithms’, in *Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, pp. 416–420, Jacksonville (FL), (2007).
- [11] Y. Li, A. Lim, and B. Rodrigues, ‘Crossdocking - JIT scheduling with time windows’, *Journal of the Operational Research Society*, **55**(12), 1342–1351, (2004).
- [12] John D. C. Little, ‘A proof for the queuing formula:  $L = \lambda w$ ’, *Operations Research*, **9**(3), 383–387, (1961).
- [13] Stephan H. Mayer, *Development of a completely decentralized control system for modular continuous conveyors*, PhD thesis, University of Karlsruhe (TH), Faculty of Mechanical Engineering, 2009.
- [14] David Naso and Biagio Turchiano, ‘Multicriteria meta-heuristics for agv dispatching control based on computational intelligence’, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, **35**(2), 208–226, (2005).
- [15] Ludger Overmeyer, Gerd Heiserich, Sascha Falkenberg, and Andreas Jungk, ‘Automatische Konfiguration und Optimierung von Materialflusssystemen durch kognitive Logistikmodule’, in *VDI-Berichte Band 2066, 18. Deutscher Materialfluss-Kongress*, pp. 197–208, Munich (Germany), (2009).
- [16] Ludger Overmeyer, Kai Ventz, and Sascha Falkenberg, ‘Interfaced multidirectional small-scaled modules for intralogistic operation’, *WGTL Logistics Journal*, (2009).
- [17] H. Van Dyke Parunak, ‘Applications of distributed artificial intelligence in industry’, in *Foundations of Distributed Artificial Intelligence*, eds., G.M.P. O’Hare and N.R. Jennings, chapter 4, 139–164, Wiley-Interscience, (1996).
- [18] Carlos D. Paternina-Arboleda and Tapas K. Das, ‘A multi-agent reinforcement learning approach to obtaining dynamic control policies for stochastic lot scheduling problem’, *Simulation Modelling Practice and Theory*, **13**, 389–406, (2005).
- [19] Henning Rekersbrink, Thomas Makuschewitz, and Bernd Scholz-Reiter, ‘A distributed routing concept for vehicle routing problems’, *Logistics Research*, **1**(1), 45–52, (2009).
- [20] Jörn Schönberger and Herbert Kopfer, ‘On decision model adaptation in online optimization of a transport system’, in *Management logistischer Netzwerke - Entscheidungsunterstützung, Informationssysteme und OR-Tools*, eds., Hans-Otto Günther, Dirk C. Mattfeld, and Leena Suhl, 361–381, Physica-Verlag, (2007).
- [21] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge (MA), 1998.

# Modeling a Neural Network Based Control for Autonomous Production Systems

Daniel Rippel<sup>1</sup> and Florian Harjes<sup>1</sup> and Bernd Scholz-Reiter<sup>1</sup>

**Abstract.** The increasing complexity of production logistic systems has lead to an emergence of new decentralized control concepts. The Collaborative Research Center 637 (CRC 637) investigates the advantages and limitations of autonomous control as one of these concepts. This research mainly focuses on control strategies consisting of precise descriptions of decision-making processes. In addition, it is developing a modeling framework for autonomously controlled logistic systems called ALEM.

As the derivation of precise behavioral descriptions from real world scenarios is difficult, artificial neural networks has come into focus. They provide a suitable method to imitate a system's behavior without a formal definition. The architecture for adaptive relocation control (ARC) uses neural networks for control purposes on shop floors. This article proposes a concept of modeling to include ARC's decentralised decision strategy into the ALEM modeling framework.

## 1 INTRODUCTION

Autonomous control aims to increase the flexibility and performance of logistic systems by allocating planning and control abilities to individual system elements. Therefore, different methods for decision making are of central interest. Most of these methods base on precise mathematical or procedural descriptions of the decision process. To model and simulate autonomously controlled logistic systems, the Autonomous Logistic Engineering Methodology (ALEM) has been developed. This methodology provides tools to model and simulate autonomous logistic systems for logistic experts. ALEM relies on a strict definition of the system elements' decision processes, represented by UML-Activity diagrams and UML-State Machines. In particular, creating models of already existing systems leads to difficulties with the derivation of diagrams which formally describe the system's decision strategies.

To cope with this difficulty, artificial neural networks provide a suitable solution to recreate the behavior of an existing logistic system, the decision processes of which cannot be formally described [16]. The networks can be trained to imitate the original system's behavior. Further, neural networks have been applied successfully to several production logistic scenarios, for example to predict delivery dates or manufacturing costs, for quality control or for sequence optimization [3, 6, 16, 21].

Although artificial neural networks provide a convenient technique for decentralized decision making [17], ALEM is incapable to include neural networks into its models at the moment. Therefore, this article investigates the requirements for the integration with ALEM. In addition, the article depicts the advantages of this Integration both

in context of autonomous control and with regard to the application of neural networks to production logistic scenarios.

The analyses of the requirements on ALEM refers to an example shop floor scenario. The scenario applies a decentralized control strategy using artificial neural networks for decision making. ALEM is used to model the scenario; thereby identifying missing notational concepts. To close this gap, the article proposes concepts and notational elements to represent the neural networks within ALEM models.

The first section introduces autonomous control and neural networks in detail. The second section depicts the shop floor example and explains the neural control strategy applied in the scenario. The third section depicts the most important aspects of the corresponding ALEM model as well as concepts to represent neural networks within the model. The article closes with sketching the advantages of integrating neural networks into ALEM.

## 1.1 AUTONOMOUS CONTROL

The term autonomous control is used in various scientific disciplines like physics, biology, artificial intelligence, control theory, and engineering sciences. In the context of logistic systems, Hülsmann and Windt define autonomous control as the "processes of decentralized decision-making in heterarchical structures. It presumes interacting elements in non-deterministic systems, which possess the capability and possibility to render decisions independently. The objective of Autonomous Control is the achievement of increased robustness and positive emergence of the total system due to distributed and flexible coping with dynamics and complexity" [9].

The application of autonomous control delegates the planning capabilities to work pieces or workstations, and disconnects commodities from the customer's orders. The objects dynamically create their own production plans. For example, once an intelligent logistic object enters production, it autonomously requests manufacturing from suitable workstations on the shop floor. Several relocation strategies have been investigated. Examples are the "Queue Length Estimator" an ant colony algorithm or the Dynamic Logistic Routing Protocol. In addition, the objects apply individual objectives. In case of a malfunction or a changing production situation, the work pieces are capable of reacting flexible. Once they are aware of a disturbance, they request manufacturing from another workstation of the same type. With regard to their product structure, they can shift the order of manufacturing steps. This allows postponing the problematic production step and can help resolving bottle neck situations [19]. Due to the dynamic allocation of work pieces to orders, they may shift their target end product. Thus, they can flexibly react to changes in the production situation, if for example new orders arrive or active

<sup>1</sup> BIBA - Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Germany, email: {rippel, harjes, bsr}@biba.uni-bremen.de

ones are canceled.

The ALEM framework is developed to model and simulate autonomously controlled systems. The methodology is designed to enable logistic experts to create models. The methodology consists of three components. First, a procedure model (ALEM-P) which guides the user through the process of modeling. Second, the ALEM-Notation (ALEM-N) which provides model elements and diagrams to represent the system, and third, a software tool (ALEM-T), which implements the procedure as well as the notation. The notation uses diagrams and elements of the Unified Modeling Language (UML) and adds several domain specific diagrams [18].

As process and system models usually imply a high degree of complexity [13], ALEM applies a view concept (Figure 1) proposed by Scholz-Reiter et al. [18]. The views focus on single aspects of the overall model and enables editing of lesser complex segments of the model [14].

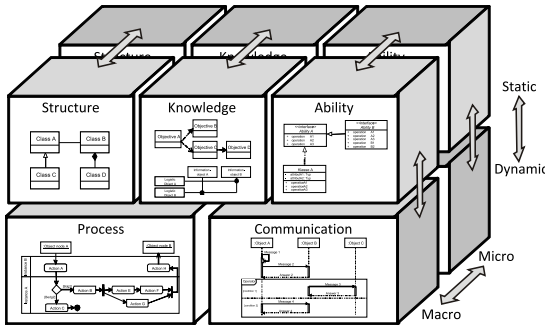


Figure 1. ALEM View Concept [18]

The views use default UML diagrams to capture different aspects of the overall model. For example, the structure view uses a UML-Class Diagram to depict the elements present in the logistic system. Those elements' abilities or knowledge are mapped using class diagrams, situated in the respective views of the model.

Symbolic modeling methods, like for example ALEM, require that knowledge, procedures or decision making strategies are modelled explicitly. In contrast, connectionist modeling methods, like artificial neural networks, capture knowledge within their structure. Usually knowledge is learned and not encoded directly. The next section introduces neural networks in more detail.

## 1.2 ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are derived from the biological concept of neural systems in nature [20]. Their strengths are their ability to learn from experience gained during operation, their fast data processing and their small modelling effort. Further, they have the ability to approximate complex mathematical functions which are completely unknown or cannot be described exactly. The knowledge of human experts is a typical example. The decision making of human experts is based on individual experience which cannot be mapped to an exact mathematical description. In this case, neural networks approximate the mathematical coherences in a "black-box" manner.

An artificial neural network basically consists of neurons which are structured in layers and connected via weighted links. There are at least two layers, the input and the output layer. Between these layers, an unlimited number of hidden layers is possible [5].

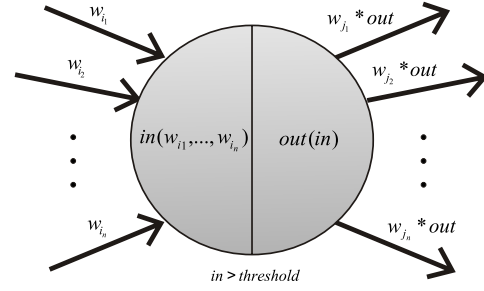


Figure 2. Artificial Neuron (own depiction)

Data processing within artificial neural networks is massively parallel. If a node receives an input signal, the value is transformed according to the node's activation function and is checked against the node's threshold. If the value exceeds the threshold, the result either proceeds directly to succeeding nodes or an output-function calculates the output (Figure 2). By this means, data is processed parallel through the nodes of the different layers. The final result of the calculation is presented at the output nodes.

Neural networks can learn from experience gained through operation. Hereto, sets of training data are repeatedly presented to the network until it computes the favored output. The learned knowledge is stored within the edges' weights. Correspondingly, a continuous adjusting of the internal weights takes place during the training process. This leads to an approximation of the mathematical function, which maps the input values to correct output values. In order to verify the generalization of the represented approximation, further sets of validation data are used. In addition, this avoids a phenomenon called *Overfitting* [8] which denotes a simple memorising of the presented training data.

In general, three types of learning processes can be distinguished. *Supervised Learning* denotes a procedure, where the input data as well as the expected output is presented to the neural network [2]. In the case of *Reinforcement Learning*, only the input data is presented. Instead of the desired output, the neural network obtains feedback whether the calculated values are correct or not [12]. Finally, *Unsupervised or self-organised learning* encompasses the presentation of input data without any feedback. In this case, the neural network tries to recognize patterns within the input data autonomously [10].

## 2 Adaptive Relocation Control (ARC)

Scholz-Reiter et al. introduced an intelligent production control concept for a customer oriented shop floor production using artificial neural networks [15]. Neural network based controllers relocate work pieces between machines and workstations on different production levels.

In production environments as well as in general, neural networks running over long time periods face the risk of *Catastrophic Forgetting* [4]. That is, that useful knowledge is overwritten with new information during the learning process. Due to this behavior, neural networks are mostly trained for only one specific task or situation. If the situation changes, the networks are either retrained or replaced. To face this problem, Scholz-Reiter et al. propose an additional software architecture (ARC) to structure the learning processes [17]. The software architecture supervise the performance of the neural networks and simultaneously train alternative networks. In case of de-

creasing performance, the active network is exchanged with an alternative one. This section first introduces the shop floor environment. Afterwards, the setup of the additional software architecture and its top level functions are described.

## 2.1 ARC SETUP

The neural control concept focuses on a shop floor environment [15]. The production facility is divided into several specialized departments, like a sawmill or a turnery. Work pieces can pass workstations and machines in any order. Shop floor production is characterised by a high degree of customization, caused by the production of only single pieces, prototypes and small series. Furthermore, customization leads to high variances concerning the order of production steps for every work piece.

The control concept following Scholz-Reiter establishes an inventory based control by means of artificial neural networks to this environment. It consists of closed control loops located between the different production stages of the shop floor. The embedded neural networks are responsible for the relocation of work pieces. Each time, a work piece is finished on a machine or workstation, a neural controller determines the subsequent facility the work piece is relocated to (see Figure 3). For each type of work pieces an appropriate neural controller is established between the stages of the shop floor.

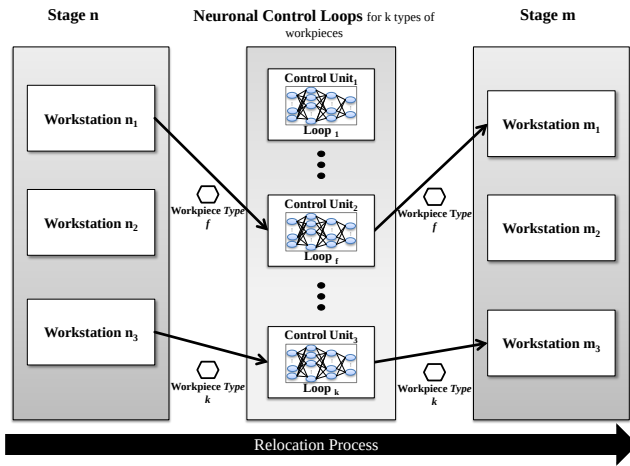


Figure 3. Structure of the control concept (own depiction)

A neural controller consists of two neural networks, one for the relocation decisions and one for prediction proposes. The controller is completed by an optimizer. The decisional networks determine the optimal machine or workstation on the subsequent production stage. Therefore, setup and processing time of the regarded work piece, as well as the current inventory levels of possible successors are taken into account.

The networks for decision-making are feed-forward networks with two hidden layers. The number of output and input nodes depends on the count of alternative workstations on the next production stage. The networks are trained using the resilient-propagation algorithm [1] in a supervised procedure. The initial training was conducted offline with training sets modelling simple priority rules. The following

online training bases on manually chosen examples gained through operation.

The predictive neural networks deliver forecasting values of future inventory levels influenced by the current relocation decision. Due to the comparatively high training effort of recurrent neural networks, the predictive networks are also conducted as feed-forward networks with two hidden layers [7]. To achieve a comparable performance, the prediction horizon was set to 7 days. For their supervised training process, the backpropagation algorithm [12] was used both for offline and online training. The training data for the initial offline training was derived from the inventory data logged during simulation runs with a rule based control.

Finally, an optimizer computes the optimal inventory level using an adaption of the simulated annealing algorithm. In this way, the inventory levels of all machines and workstations are optimized in respect of a maximal adherence to due dates for the shop floor.

## 2.2 ARC RELOCATION CONTROLLER

To cope with the problem of Catastrophic Forgetting, the control concept described in the previous section is extended by a software architecture [17]. The architecture for ARC encloses the networks (see Figure 4), and evaluates the performance of the employed neural network.

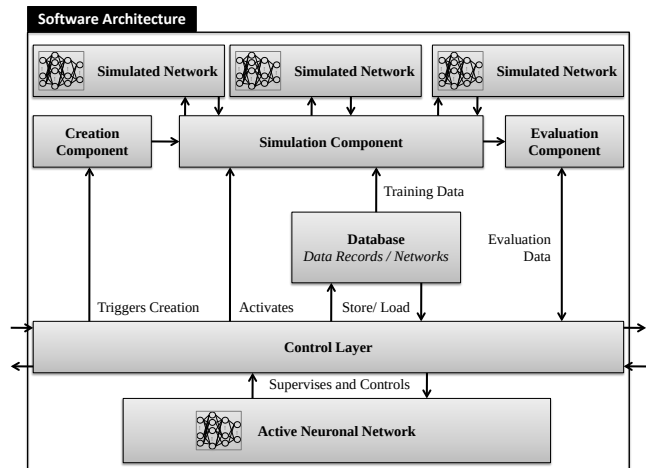


Figure 4. Structure of the Adaptive Relocation Control (ARC) [17]

If the quality of decisions made by a neural network decreases, alternative networks are generated and trained. Employing internal simulations, the performance of the active neural network and the possible alternatives is compared. If one of the alternative networks achieves a better performance, an exchange takes place. The architecture further collects training data for the simulated alternative networks and supervises the current production situation.

The relocation controller can be interpreted as a three layered system containing the execution layer with the currently active network, the data storage as a mid-layer and a top layer containing control, evaluation and simulation. The functions covered by the controller are now described.

**Evaluation** The controller continuously measures the performance of every active and simulated neural network. In addition, it measures the global adherence to due dates. The continuous evaluation ensures that the most efficient neural networks are applied.

**Creation** each controller creates alternative networks, depending on the current production situation or on trends becoming apparent. Therefore, statistical a-priori knowledge is used to determine network architectures.

**Simulation** The training and simulation of alternative networks is carried out using an internal material flow simulation. The input data is collected during operation, which allows comparing the simulated and the running neural network's performances. The continuous simulation offers alternatives to the actual operating control.

**Database** The database associates replaced neural networks and production situations in which specific networks or network architectures performed well. In addition, it stores equally classified sets of training data. In cases where changes in the production situation become apparent, situation specific information can be retrieved.

**Control** The control component manages the interaction of single functions. The results of the evaluation are processed and, if necessary, the creation and simulation of alternative networks is triggered. In addition, the active network is exchanged if an alternative network performs better. Further, the component collects training data for the simulation as well as the required data to evaluate the controllers performance.

### 3 SHOP FLOOR SCENARIO IN ALEM

This section depicts the most important aspects of an ALEM model, corresponding to the ARC job shop example explained. Thus, the scenario's features are characterized first. Afterwards, those ALEM diagrams, related to the ARC controller are presented and a concept of integrating the neural networks into the model is proposed.

#### 3.1 SCENARIO'S CHARACTERISTICS

ARC delegates the decision, to which workstation a work piece is relocated, to decentralized neural controllers. These controllers are set in between the production stages. This section investigates the scenario in terms of an autonomous control architecture.

Mainly, the system is characterized by three components: The work pieces, the workstations and the relocation controllers.

**The workstations** offer their ability to perform various manufacturing steps to work pieces. Each machine or workstation is associated to an input and to an output buffer storing work pieces. The workstation is aware of its current buffer status and can communicate the status on request.

**The work pieces** are manufactured in accordance to their type. Each product type is defined by a predefined sequence of production steps. In contrast to classic autonomous scenarios, work pieces do not select the next workstation/machine on their own, but request a decision from the corresponding relocation controller. Finally, the work piece executes the relocation controller's decision and proceeds to the selected workstation.

**The relocation controllers** The controller assumes the relocation of one certain work piece between two fixed production stages. If a work piece finishes manufacturing at a workstation, the relocation controller decides where on the next production stage the work piece shall be processed.

Furthermore, the relocation controllers perform additional activities, like data acquisition, network evaluation and replacement as well as they collect training data. These activities implement another dimension of a decision strategy: The controller monitors itself, and adjusts its own decisions in accordance to global objectives.

The ARC control processes can be interpreted as autonomous control processes. Thus, the scenario is representable using ALEM. As ALEM focuses on procedural descriptions, model elements are introduced to capture the neural control strategy into ALEM.

Despite of the relocation controllers' decisional strategy, the scenario conforms to classical shop floor scenarios. Thus, following the reference models proposed by Scholz-Reiter [19] and Kolditz [11], an ALEM model can be built. The relocation controller consists of several top level functions, as well as of a set of neural networks. The next sections first examines which aspects of the controllers can be expressed using the actual ALEM notation, and which aspects require future work. Requirements for the integration of external neural control are stated and a possible interface is proposed.

#### 3.2 MODEL OF THE CONTROLLER

The relocation controller is integrated into the structure view's class diagram [19] as specialization of the logistic resource (on the right of Figure 5). This allows the assignment of abilities, knowledge and behaviors to the controllers. The structure view's class diagram models all elements present in the logistic scenario as well as the relations among them. As the relocation controllers are not part of the work pieces or workstations, they act as intelligent logistic objects and have to be included into this diagram.

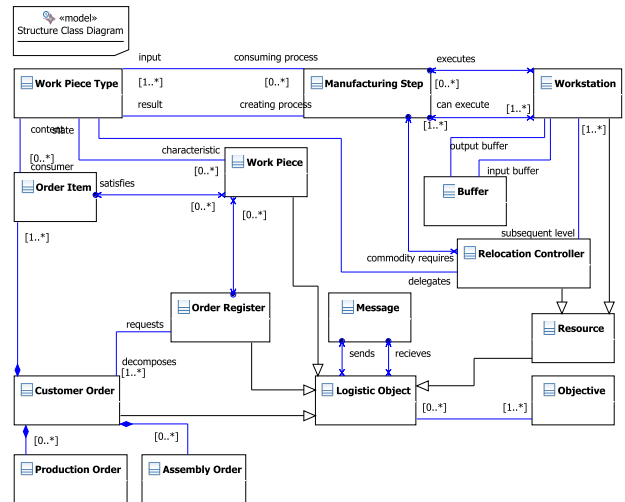


Figure 5. Adopted structure view's class diagram

The relocation controllers possess a set of basic abilities, which are incorporated into the ability view's class diagram. These abilities are essential to perform the top level functions provided. The corresponding interfaces are depicted in Figure 6.

On top of these basic abilities, more complex actions, as well as the controllers life cycle are established. The controller only adapts

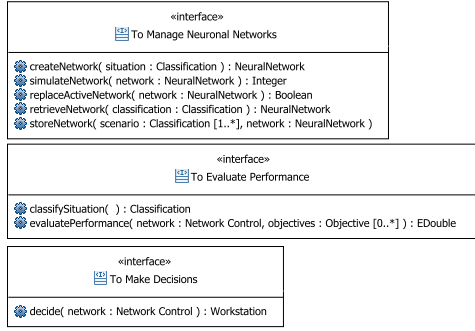


Figure 6. Abilities of the relocation controller

to two states: “Idle” and “Relocating”. While relocating the *Make Decision* activity is applied. While the controller is idle, it constantly executes the *Maintain Performance* activity. The corresponding diagrams are depicted in Figure 7. These activities make use of the abilities defined for the controller.

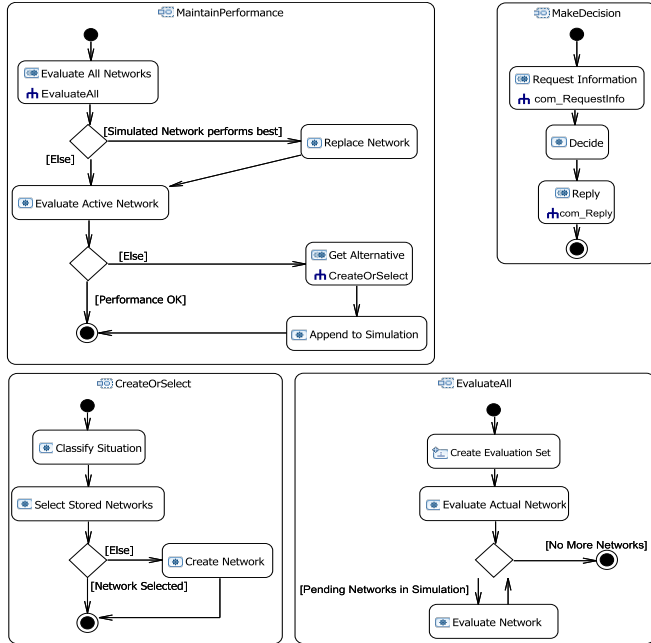


Figure 7. Relocation controller's UML-activity diagrams (simplified)

### 3.3 REQUIREMENTS TOWARDS THE INTEGRATION

The ALEM-Tool is designed to model and simulate ALEM models. Thus, the networks have to be integrated in a way, that they can be used within the model's simulation. Therefore, the networks are represented as a part of the model. Consequently, the corresponding

model elements have to include all information which is necessary to address the networks within the simulation.

The representation and execution of the neural networks are the only aspects of the relocation controller, which cannot be captured by ALEM in a straight forward way. This section investigates these aspects.

The execution of these neural networks depends on the software used to implement them. The networks used in this example are implemented using the SNNS environment<sup>2</sup>. This environment provides TCP/IP interfaces to interact with the networks.

**Representation:** The external information source is interpreted as knowledge. Thus, a visual representative is integrated into the ALEM knowledge view. In this case, it includes the network's TCP/IP address and port, as well as the parameters which are required to execute the network. The graphical elements, representing the neural networks are depicted in Figure 8.a.

**Execution:** Two additional abilities must be created to execute the networks. First, an ability to create the connection and to execute the network and second, an ability to interpret the results. In this case, the networks return a number corresponding to one of the machines on the next level. This value must be mapped to the concrete machine used in the ALEM model. The abilities are modelled within ALEM's ability view (Figure 8.b)

Based on the knowledge element which represents the external information source, the executional ability can be derived. Furthermore, the knowledge object references both, the network's return type, as well the ALEM type it corresponds to. This allows the derivation of the interpretation ability's signature.

To enable an execution of these abilities in an arbitrary simulation platform, both functions must be integrated into ALEM's reference libraries. Thus, templates can be created to transform the abilities' notational elements into an executable form.

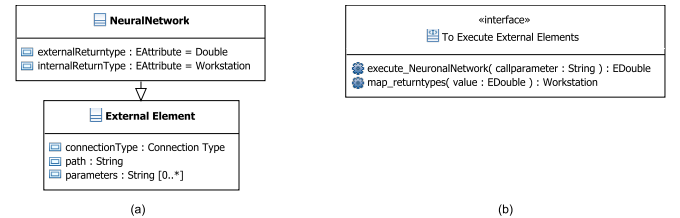


Figure 8. (a) Required knowledge elements; (b) Additional abilities

Another important requirement towards the simulation of models involving external information sources is the selection of the simulation environment. In accordance to the type of external source, the environment must be able to establish a connection, for example via TCP/IP or via call operation for external files. It must provide means to send commands and receive responses from the data source. Therefore, the inclusion of such black-box elements requires the selection of specific simulation platforms.

<sup>2</sup> Stuttgart Neural Network Simulator, <http://www.ra.cs.uni-tuebingen.de/SNNS/>

## 4 CONCLUSION AND OUTLOOK

The article provides a concept of integrating neural networks into the ALEM modeling framework. Therefore, a shop floor example employing the ARC control approach is introduced. The scenario's neural controllers are interpreted as resources, to enable modelling in ALEM. Further, notational elements to depict external, neural decision units are proposed.

As a major drawback, the presented way of integrating neural networks includes the addition of external software components. The interface to these components cannot be modelled in ALEM directly. Therefore, it has to be implemented within the simulation environment. This requires the user to possess at least basic experience in simulation programming. Furthermore, the inclusion of external decision elements, like in this case of neural networks, reduces the completeness of the ALEM models. Thus, it is rendered impossible to trace the decisional behavior made in the modelled system only by examining the model itself.

Integrating neural controllers into the ALEM framework provides the opportunity to recreate an existing system's behavior, without formally describing the decision processes. Therefore, the real-world system's decisions can be used as training data for the networks. This eases the derivation of a model of the original system. The proposed way of representing neural networks in ALEM enables an exchange of those networks by other decision strategies. This supports experimenting with autonomous control strategies. Furthermore, the simulation of the original system's model provides data, suitable to compare the performance of different control strategies within the same scenario.

In addition, exchangeable decision strategies simplify the development of scenarios applying neural networks. Data gained through simulation runs using default control strategies can be used as a base for the initial training of neural networks. Further adaptations to a desired decision strategy can be obtained by online training. The notational elements proposed for ALEM are kept general. Thus, it is possible to create templates to include other non-procedural information sources. It is necessary to investigate extensions of these elements and to define clear structures and processes to generate and maintain connections to other external information sources. With focus on a possible simulation, the ALEM reference libraries must be adopted to provide simulation specific implementations of the related operations and data structures. Therefore, suitable simulation environments must be investigated.

## ACKNOWLEDGEMENTS

This research is funded by the German Research Foundation (DFG) as part of the Collaborative Research Center (CRC) 637 "Autonomous Cooperating Logistic Processes - A Paradigm Shift and its Limitations", subproject B2, as well as part of the project "Automation of continuous learning and examination of the long-run behavior of artificial neural networks for production control", index SCHO 540/16-1.

## REFERENCES

- [1] Aristoklis D. Anastasiadis, George D. Magoulas, and Michael N. Vrahatis, 'New globally convergent training scheme based on the resilient propagation algorithm', *Neurocomput.*, **64**, 253–270, (2005).
- [2] D.K. Chaturvedi, 'Artificial neural network and supervised learning', in *Soft Computing Techniques and its Applications in Electrical Engineering*, 23 – 50, Springer Verlag, Berlin Heidelberg, (2008).
- [3] S.F. Crone, 'Forecasting in inventory management using artificial neuronal networks - a novel approach through asymmetric cost functions', in *Einsatz von Fuzzy-Sets, Neuronalen Netzen und Künstlicher Intelligenz in industrieller Produktion und Umweltforschung*, 59 – 69, VDI - Verlag, Düsseldorf, (2003).
- [4] Akira Date and Koji Kurata, 'A property of neural networks of associative memory with replacing units', *Artificial Life and Robotics*, **12**(1-2), 291 – 294, (2008).
- [5] G. Dreyfus, *Neural Networks Methodology and Applications*, Springer-Verlag Berlin Heidelberg, Berlin Heidelberg, 2005.
- [6] Marco Garetti and Marco Taisch, 'Neuronal networks in production planning and control', *Production Planning and Control*, **10**, 324 – 339, (1999).
- [7] T. Hamann, *Lernfähige intelligente Produktionsregelung*, Gito Verlag, Berlin, 2008.
- [8] S. Haykin, *Neural Networks and Learning Machines (3rd Edition)*, Prentice Hall, New Jersey, USA, 2008.
- [9] *Understanding of Autonomous Cooperation and Control in Logistics The Impact of Autonomy on Management, Information, Communication and Material Flow*, eds., M. Hülsmann and K. Windt, Springer Verlag, Berlin, 2007.
- [10] *Self-Organizing Maps*, eds., T. Kohonen, M. R. Schroeder, and T. S. Huang, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [11] Jan Kolditz, *Fachkonzeption für selbststeuernde logistische Prozesse*, Ph.D. dissertation, Universität Bremen, Berlin, 2009.
- [12] Wolfram-Manfred Lippe, *Soft-Computing: mit Neuronalen Netzen, Fuzzy-Logic und Evolutionären Algorithmen (eXamen.press)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
- [13] A.-W. Scheer, *Business Process Engineering - Reference Models for Industrial Enterprises*, Springer, Telos, 1994.
- [14] A.-W. Scheer, *ARIS - Modellierungsmethoden, Metamodelle, Anwendungen*, Springer Verlag, Berlin, 2001.
- [15] B. Scholz-Reiter and T. Hamann, 'The behaviour of learning production control', *CIRP Annals - Manufacturing Technology*, **57**(1), 459 – 462, (2008).
- [16] B. Scholz-Reiter, T. Hamann, H. Höhns, and G. Middelberg, 'Decentral closed loop control of production systems by means of artificial neural networks', in *Proceedings of the 37 th CIRP-International Seminar on Manufacturing Systems*, pp. 199 – 203, (2004).
- [17] B. Scholz-Reiter, F. Harjes, and T. Hamann, 'Automatisierung des Lernens neuronaler Netze in der Produktionssteuerung', *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, **1-2**, 101–104, (2010).
- [18] B. Scholz-Reiter, J. Kolditz, and T. Hildebrandt, 'Engineering autonomously controlled logistic systems', *International Journal of Production Research*, **47**(6), 1449–1468, (2009).
- [19] B. Scholz-Reiter, S. Sowade, T. Hildebrandt, and D. Rippel, 'Modelling of autonomous control explicitly considering orders as a kind of immaterial logistic objects in manufacturing processes', in *Proceedings of the 3rd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2009)*, München, Deutschland, (2009).
- [20] W.-H. Steeb, *The Nonlinear Workbook: Chaos, Fractals, Neural Networks, Genetic Algorithms, Gene Expression Programming, Support Vector Machine, Wavelets, Hidden Markov Models, Fuzzy Logic with C++, Java and SymbolicC++ Programs*, World Scientific Publishing Co. Pte. Ltd, Singapore, 2008.
- [21] N. Wang and J. Yu, 'Neuron based nonlinear pid control', *PRICAI 2006: Trends in Artificial Intelligence*, **4099**, 1089–1093, (2006).

# Towards intelligent material handling planning systems – Status quo and steps to be taken –

Tobias Sommer and Daniel Neuhäuser and Dirk Marrenbach and Karl-Heinz Wehking<sup>1</sup>

**Abstract.** The process of planning material handling systems is highly complex. By applying procedure models, methods, and expert knowledge, a material handling system is designed according to market requirements. The result of this design phase is highly dependent on the expertise and valuation of the human planner. In this respect, software-based decision support systems aim to enable reasonable and comprehensible design decisions. This article presents the vision of intelligent planning systems that are able to transform user requirements into efficient and effective material handling systems. In this respect, two steps that have to be taken in developing intelligent planning systems are discussed.

## 1 INTRODUCTION

Optimal material handling systems are elementary to ensure economical, high-quality, and time efficient operations in distribution structures.

The configuration is developed in a design process, in which market and company requirements are transformed into a material handling system. In this respect, decisions are made on the basis of planning data that take effect on the performance of the designed system.

With these decisions made by human planners, personal opinions and restricted knowledge may lead to a sub-optimal solution. Thus, the demand for intelligent software-based decision support systems arises, which may help to reduce planning uncertainties and enable reasonable results.

## 2 MATERIAL HANDLING SYSTEM DESIGN

*Material handling systems design* aims to create an optimal material handling system for a specified purpose. The design is influenced by technical, economic, and organizational restrictions that may be either internally or externally determined.

The design process is carried out in different stages. In each design step, alternative solutions are created and evaluated according to specified criteria. Then, the superior alternative is chosen and considered in consecutive design phases.

Having analysed relevant literature on procedure models for material handling system planning [1, 2, 3], the following planning steps can be extracted:

At first, planning goals are determined and the planning data basis is set up and analysed. The data basis includes performance requirements, assumptions, expected sales, etc.

After that, the material handling processes are designed. In this respect, different process alternatives are generated, evaluated, and

reduced to the most convincing alternative. The selected process alternative highly determines the structure of the material flow.

This material flow structure is developed in the layout planning stage. It includes the calculation of required space per area (e.g. stock receipt, order picking zone), the allocation of processes to these areas as well as locating the necessary equipment according to one or more specified target function(s), e.g. minimum transportation costs, minimum space costs or minimum shape irregularities.

The layout is clearly linked to the *material handling equipment selection (MHES)* that includes the choice, dimensioning, and evaluation of proper equipment as well as assessing the resulting material handling system. The choice between conveyors and forklifts or between floor storage and high rack warehouses illustrates different design alternatives.

All developed alternatives have to be assessed according to specific criteria. The most promising alternative is finally planned in detail.

This procedure model also includes the bid invitation of the chosen design alternative and its execution before starting the operating phase of the material handling system.

## 3 MATERIAL HANDLING EQUIPMENT SELECTION (MHES)

The selection of material handling equipment requires a comparison between user requirements and equipment capabilities. According to [4] three different approaches in MHES can be defined:

- *Analytical approaches* incorporate quantitative selection procedures, e.g. on basis of integer programming. They require a mathematical model of the selection problem, commonly formulated as one or more target function(s) that has/have to be optimised according to one or more constraint(s). The necessary quantification of qualitative variables is often highly speculative and may therefore lead to pseudo accurate results that can easily be misinterpreted. Realised analytical approaches for MHES are provided by [5, 6, 7].
- *Knowledge-based approaches* rely on expert knowledge on equipment selection. This knowledge is then transformed into decision rules that compare user requirements with equipment capabilities to select proper material handling resources, e.g. “In case the product mass exceeds 15 kg, employees need to use lifting equipment to handle the product.” Realised knowledge-based approaches for MHES are provided by [8, 9, 10, 11].
- *Hybrid approaches* combine knowledge-based and analytical approaches in different steps of the decision process. Therefore, they are not addressed in detail in the following. Realised hybrid ap-

<sup>1</sup> University of Stuttgart, Germany, email: sommer@ift.uni-stuttgart.de

proaches for MHES are provided by [4, 12]. Now, focus is on the applicability of knowledge-based approaches.

The relevance of all three approaches in equipment selection faces limitations due to the high complexity of inbound logistics: Interrelations between product, process parameters, and layout must be taken into account when selecting a material handling equipment, as they all determine the efficiency and reliability of the material handling system.

Following [11], knowledge-based MHES is performed in several decision steps: The selection process starts with sorting out equipment classes by basic criteria, e.g. the required usage flexibility. After that, static properties of equipment classes, like loading capacities, have to be analysed. This step is followed by using decision trees to incorporate expert knowledge to the decision process.

A detailed cost comparison of the remaining alternatives as well as sensitivity analyses may help to assess the remaining alternatives in different business scenarios. Finally, a detailed simulation study of the chosen system in different situations, e.g. including component failures, is proposed.

As material handling equipment has to be assigned to each process, e.g. truck unloading is to be carried out by forklifts, a process based categorization of equipment is necessary. Thus, relevant material handling reference processes have to be defined, e.g. storing, packaging, moving.

The number and scope of these reference processes is clearly linked to the relevant planning task. This means that reference processes for distribution centres may differ from those for production supply systems.

After setting up the reference process basis, specific decision rules have to be developed for each process. These rules represent expert knowledge in comparing user requirements with equipment capabilities.

Combining each reference process with only a limited number of available equipment types may help to reduce the complexity of the rule development process: Instead of setting up rules that reject forklifts as possible packaging equipment, only explicitly compatible packaging equipment types (e.g. stretching machines) are linked to the packaging reference process.

By applying decision rules to material handling selection problems, equipment types can be reduced to a number of requirement compatible alternatives. This means that the application of each decision rule leads to incompatible devices being dismissed from the decision process.

Finally, only compatible devices remain in a virtual alternatives list. Deciding for only one alternative nevertheless may require detailed analytical or simulative investigation of both the remaining devices as well as the analysis of interrelations in the whole material handling system. In case the system being planned includes  $n$  processes and each process can be carried out by  $m$  alternative compatible devices, the number of alternative system configurations is up to  $m^n$ , which may exceed the capabilities of both human planners and computer equipment. Thus, the demand for more intelligent planning procedures arises.

## 4 VISION OF INTELLIGENT MATERIAL HANDLING PLANNING SYSTEMS

Although planning procedures and knowledge-based decision support systems may help to structure the planning process, the planning results are still highly dependent on the expertise of the planner.

Intelligent algorithms may help to increase both planning efficiency and effectiveness. With this intelligence being software-based, artificial intelligence would be applied in the planning process.

Following the considerations on a definition of artificial intelligence presented in [13], artificially intelligent planning systems (IPS) would be able to generate reasonable and comprehensible decisions on the basis of structured, but a priori unknown data. They could apply selection and compatibility rules, communication protocols, and decision logics to decide which equipment to use for specific processes and to combine in order to form an efficient and effective material handling system.

This also includes the proper combination of devices by analysing compatibilities and overlapping operational areas. Thus, possible configurations of distribution systems can be generated and assessed autonomously on the basis of planning goals and data.

The human planner remains the sole authority to finally choose one alternative to be realised. Table 1 indicates the role of an intelligent planning system in different steps of the material handling planning process.

**Table 1.** Proposed role of intelligent planning systems (IPS)

| Planning step | Description                   | Role of IPS           |
|---------------|-------------------------------|-----------------------|
| 1             | Planning goals definition     | Step supported by IPS |
| 2             | Data basis setup and analysis | Step supported by IPS |
| 3             | Process design                | Step processed by IPS |
| 4             | Layout planning               | Step processed by IPS |
| 5             | MHES                          | Step processed by IPS |
| 6             | Detailed planning             | -                     |
| 7             | Bid invitation                | -                     |
| 8             | Start of operations           | -                     |

## 5 STEPS TOWARDS INTELLIGENT PLANNING SYSTEMS

In the remainder of this article, two important issues in overcoming problems in intelligent planning systems are to be addressed:

This includes the setup and analysis of the planning data basis (5.1) as well as the material handling equipment selection and system evaluation (5.2).

### 5.1 Planning data basis setup and analysis

With the quality of the data basis taking high influence on the quality of the planning results, all aspects of creating the planning data basis have to be taken seriously. In order to automatically process input data to a material handling system configuration, reliable and formally described data is necessary. Therefore, different data specific problems require methodical processing:

- Different parts of the planning data basis vary in their duration of validity. Laws or market sizes serve as examples for long-term valid data, whereas products, stock sizes and sales reflect dynamically changing short-term counterparts.
- As planning results highly depend on input data, incorrect data may lead to poor material handling system design. The quality aspect of planning data is furthermore interrelated with the problem of data interpretation and processing: Despite the underlying data quality, missing knowledge, motivation or methodical support may induce the planner to incorrect interpretation and processing. Table 2 serves to highlight the different categories of

planning data quality and its interpretation: If the – according to specified criteria – correct data is correctly interpreted or the incorrectness of data is correctly identified, specific measures can be taken to process the data to promising planning results. In case correct data is incorrectly interpreted or incorrect data is perceived as correct, design shortcomings are likely.

**Table 2.** Data quality and interpretation

| data    |           | interpretation                          |                                           |
|---------|-----------|-----------------------------------------|-------------------------------------------|
|         |           | correct                                 | incorrect                                 |
| correct | correct   | correct data is correctly interpreted   | correct data is incorrectly interpreted   |
|         | incorrect | incorrect data is correctly interpreted | incorrect data is incorrectly interpreted |

Due to the complexity of the planning process, the relationship between input data and planning results cannot be described analytically. New methods have to be developed and applied to ensure an adequate description and processing of input data in the planning process:

- Formal descriptions of the planning process as well as formal descriptions of planning data have to be developed. Thus, vague and qualitative requirements like “flexibility” or “adaptability” can be quantified and automatically processed. Existing approaches that aim to assess qualitative performance of already realised systems (e.g. “agility” in [14]) have to be transformed to *a priori* design methodologies which ensure proper design of systems according to specified qualitative performance requirements. Furthermore, the frequency of usage of different planning data as well as its effect on planning outcome has to be examined.
- As the relationship between data input and planning result cannot be analytically determined, Artificial Neural Networks (ANN) may help to identify data patterns that take high influence on the planning results. Therefore, a new measure has to be developed stating the importance of specific data in the planning process (importance factor). These insights can consequently lead to the derivation of a formal input vector that demands input of all relevant planning data and that can be processed in all consecutive planning phases.

By applying both measures, the frequency of usage as well as the importance of specific planning data can be determined. Table 3 summarises all possible combinations of frequency of usage and importance. Each combination leads to specific actions that have to be taken in the planning process.

Basically, important data has to be analysed and processed more carefully than low-importance-data. In case of limited resources, the frequency of usage may indicate priorities in data handling: This means that important data that is often used has to be analysed and processed with high priority.

To sum it up, shortcomings in data understanding often reject automated data processing. The discussed measures may lead to a better understanding of data relevance and an improved data processing. As a result, the selection of equipment and its integration to an efficient material handling system can be operated by a computer-based design system.

Enabling software planning tools to automatically create and evaluate system configurations nevertheless requires a paradigm shift to-

**Table 3.** Data usage and importance

| importance |      | frequency of usage                                                            |                                                                                |
|------------|------|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
|            |      | high                                                                          | low                                                                            |
| high       | high | Data is <i>often</i> used and has <i>high</i> influence over planning results | Data is <i>rarely</i> used but has <i>high</i> influence over planning results |
|            | low  | Data is <i>often</i> used but has <i>low</i> influence over planning results  | Data is <i>rarely</i> used and has <i>low</i> influence over planning results  |

wards intelligent planning objects, as described in the remainder of this article.

## 5.2 MHES and system evaluation applying Intelligent Planning Objects (IPOs)

Instead of a hierarchic top down selection of equipment, the development of a bottom-up approach is proposed: This approach incorporates Intelligent Planning Objects (IPOs) that mimic properties and behaviour of handling objects, resources, and processes.

Therefore, different IPOs have to be developed: Each IPO represents a specific class of material handling technology (automated guided vehicles (AGVs), order picking technology, conveyors ...) and is equipped with adequate cost and performance assessment methods as well as with relevant strategies. Therefore, resource allocation to specified material handling processes is enabled.

Furthermore, one super-IPO is required to assess the final material handling system configurations as well as to transform user requirements and therefore serves as an interface to the human planner. In a specific planning process, instances of IPO-classes are created that build up a virtual material handling system.

IPOs are able to organise themselves autonomously using defined interfaces and exchange protocols according to a global ontology. The process of material handling systems planning requires a decomposition of user requirements to deviate functions to be carried out in the resulting material handling system, e.g. transshipment of incoming pallets.

Furthermore, the required functions can be organised in different processes: For example, the function “order picking” consists of processes like goods movement, storage, identification, etc.

The concept of IPOs is mainly based on the idea of holons, which have been applied to various industrial problems (e.g. in [15]). Nevertheless, IPOs require a higher level of independence and intelligence than holons, as they have to assess their own role and performance in the desired system on their own.

Our vision is that required functions are processed in a virtual intelligent planning system to deviate processes that can be related to material handling equipment. Then, each IPO evaluates its own compatibility to these processes. In case an IPO is able to execute the required process, the IPO can “block” this process in a virtual IPO marketplace. For example, a conveyor is able to execute a transport process of a transshipment function.

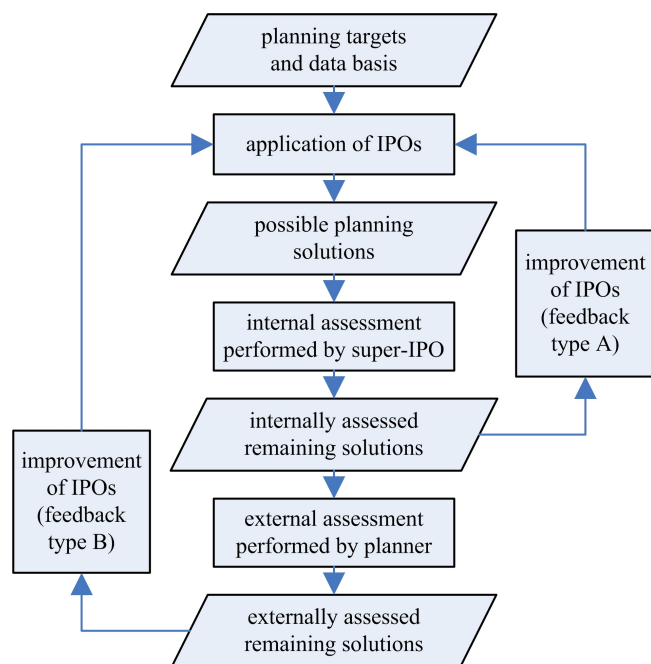
The following three cases may occur:

- *Case 1:* Every deviated process is performed by exactly one allocated IPO. In this case, the next step is generating the connections between all processes.
- *Case 2:* Suppose that  $n > 1$  IPOs are able to execute a specific process, different system alternatives are created, e.g. one alternative including conveyors whereas another might include AGVs. Both internal as well as external assessment of the alternatives may help

to reduce the quantity of alternatives. This evaluation requires an *a priori* connection of all processes.

- **Case 3:** A process cannot be executed by one single IPO, but several IPOs contribute to execute the process. Thus, a cluster of these IPO has to be build. For example, the requirements for storing multiple products (light and small vs. large and heavy) cannot be fulfilled by one type of rack. But different types of racks can be combined to store the whole product range. The clusters are built by exchanging structured and formalized messages between the IPO: These messages describe possible combinations of IPOs as well as their specific conditions.

After every process has been assigned to one or more IPO(s), the next step is to combine these IPOs to a material flow system. This combination is moderated by the super-IPO. This IPO builds up and evaluates the connections between the IPOs. As a result, a specific list of possible material handling system configurations is created and presented to the user by the super-IPO.



**Figure 1.** Network of transputers and the structure of individual processes

The resulting alternative system configurations are to be assessed according to specific criteria. Intelligence is incorporated by installing two logical feedback types (see Figure 1):

**Feedback type A:** By internally assessing different possible alternative system configurations, inferior ones are analysed automatically to avoid them being proposed in further planning projects.

**Feedback type B:** Proposed configurations are externally assessed by the human planner. In case a generated solution is inferior according to the planner's criteria, his assessment has to be translated into a language processable by the IPO system. The planner has to describe his reasons in a structured message.

By implementing analysis rules into the super-IPO, this information can be incorporated to deduce new IPO internal configuration rules. It is important to reduce the number of alternatives presented to the human planner in order to increase the applicability of this feedback type.

As a third feedback type, benchmarking of realised material handling systems (e.g. on the basis of operating data) may help to validate and improve the theoretical alternative assessment method.

## 6 CONCLUSION AND OUTLOOK

To enable efficient material handling systems planning, different methods and tools aim to support the human planner in finding superior alternatives. Incorporating them into more powerful planning systems may help to reduce uncertainty in the planning process.

In this article, two steps towards intelligent planning systems were discussed: Methods to ensure high quality planning data as well as Intelligent Planning Objects (IPOs) that propose promising configurations to the planner. IPOs may be applied to both initial designs as well as reconfigurations of material handling systems.

These approaches require significant progress in different research fields: First of all, the dependencies between input data and planning results have to be investigated and formalised. This is especially important, as the applicability of IPOs depends on relevant and reliable planning data. Moreover, the sensitivity of planning results can be determined according to their planning input data.

Another field of research is the complex MHES problem. Although isolated methodologies for equipment selection have been developed, a holistic approach that covers interactions between different equipment types is not known yet. In this respect, an ontology for material handling systems is required to enable a communication between digital entities of equipment types.

The most complex part of research is the required intelligence of the IPOs. Both feedback types illustrated in Figure 1 increase the requirements for the ontology. Every feedback of the logistics planner has to be translated into the IPO communication language (ontology) as IPO have to interpret the information contained in each feedback and adapt it to the specified system evaluation methods.

To sum it up, the vision of intelligent planning systems for material handling is promising in finding superior system configurations. The two discussed approaches indicate required progress in different research fields. The human planner benefits in every case from being supported in complex decisions by learning systems.

## References

- [1] J. M. Apple, *Material Handling Systems Design*. John Wiley & Sons, New York/ Chichester/ Brisbane et al., 1977.
- [2] P. Baker and M. Canessa, 'Warehouse Design - A structured approach', *European Journal of Operational Research*, **193**, 425-436, (2009).
- [3] B. Rouwenhorst, B. Reuter and V. Stockrahm et al., 'Warehouse design and control - Framework and literature review', *European Journal of Operational Research*, **122**, 515-533, (2000).
- [4] P. S. Welgama and P. R. Gibson, 'A hybrid knowledge based/optimization system for automated selection of materials handling system', *Computers and Industrial Engineering*, **28**, 205-217, (1995).
- [5] A. Atmani and S. P. Dutta, 'Product design and material handling system selection - A concurrent engineering approach', *International Journal of Industrial Engineering - Theory, Applications and Practices*, **3**, 40-50, (1996).
- [6] R. S. Lashkari, R. Boparai and J. Paulo, 'Towards an integrated model of operation allocation and material handling selection in cellular manufacturing systems', *International Journal of Production Economics*, **87**, 115-139, (2004).
- [7] J. Velury and W. J. Kennedy, 'A systematic procedure for the selection of bulk material handling equipment', *International Journal of Production Economics*, **27**, 233-240, (1992).
- [8] C. Cho and P. J. Egbelu, 'Design of a Web-based integrated material handling system for manufacturing applications', *International Journal of Production Research*, **43**, 375-403, (2005).

- [9] E. L. Fisher, J. B. Farber and M. G. Kay, 'MATHES – An expert system for material handling equipment selection', *Engineering Costs and Production Economics*, **14**, 297-310, (1988).
- [10] O. Kulak, 'A decision support system for fuzzy multi-attribute selection of material handling equipments', *Expert Systems with Applications*, **29**, 310-319, (2005).
- [11] S. Naish and P. Baker, 'Materials handling – Fulfilling the promises', *Logistics and Transport Focus*, **6**, 18-26, (2004).
- [12] S. H. L. Mirhosseini and P. Webb, 'A Hybrid Fuzzy Knowledge-Based Expert System and Genetic Algorithm for efficient selection and assignment of Material Handling Equipment', *Expert Systems with Applications*, **36**, 11875-11887, (2009).
- [13] S. Russel and P. Norvig, *Artificial Intelligence – A Modern approach*, Pearson, Upper Saddle River, 2010.
- [14] C. L. Heger, *Bewertung der Wandlungsfähigkeit von Fabrikobjekten*, PZH Hanover, PhD Thesis 2007 [in German].
- [15] A. Giret and V. Botti, 'Engineering Holonic Manufacturing Systems', *Computers in Industry*, **60**, 428-440, (2009).



# Reasoning About Geometry: An Approach Using Spatial-Descriptive Ontologies

Markus Eich and Frank Kirchner<sup>1</sup>

**Abstract.** Autonomous robots perceive their environment usually through exteroceptive sensors which generate features in the spatial domain. Finding the corresponding semantic or symbolic description can be referred to as the *anchoring* or *grounding* problem. Some of the latest research in robotics is dedicated to the generation of semantic maps. This includes labeling of metric maps which are provided by 3D point clouds. Usually, a model database has to be generated beforehand in order to classify objects in the spatial domain. In our approach, we propose a semantic classification based on object primitives and their spatial 3D relationship. We introduce spatial feature descriptors which can be mapped directly to a symbolic level. By looking at the relationships in the spatial domain, we are able to describe and classify known objects without model learning. The spatial entities can be defined directly using domain knowledge and ontologies. We apply our approach to a mobile dual-manipulator robot with application to logistic scenarios. We propose an ontology-based description of an indoor environment and a probabilistic reasoning approach based on spatial feature descriptions. The paper will give an overview of our current work on applying AI methods to logistics scenarios.

## 1 INTRODUCTION

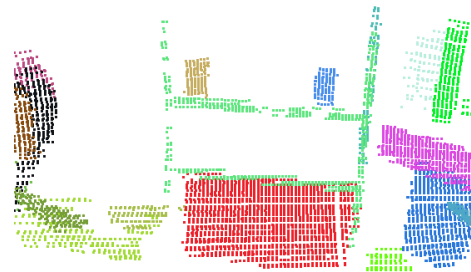
Mobile robots are usually moving in and interacting with a 3D environment, so 3D perception is mandatory for such systems. Besides path-planning and map-building approaches, which have been thoroughly investigated over the last decade, robots need an understanding about their environment.

As more processing power and more sophisticated 3D range sensors become available, an increasing number of approaches is dealing with the generation of coherent, metric 3D models. Nevertheless, it becomes clear that simple metric information about the environment is not sufficient to establish real autonomy in terms of interacting with and reasoning about the robot's environment. For intelligent behavior, it is preferable to send the robot high-level commands, like "Move to the table in the office!" or "Take the box from the table in the office!" instead of sending the robot to pure metric coordinates. The problem is always to anchor the spatially detectable features to a semantic level. A typical indoor environment is shown in Figure 1 which was generated by our robot in a logistic setup. The task of the robot is to reach the shelf and grab a certain object. While in our experiments the object detection is based on vision and SIFT-features, the shelf itself is detected using a tilting 3D laser scanner. Learning a geometric model of a shelf would include learning different types of shelves, with different numbers of boards and with different sizes.

Our key idea is to separate the model from the physical appearance and include a divide and conquer strategy. On a semantic level a typical shelf can be described by the following description:

- All rectangular shapes have an elongation of less than two meters.
- The number of rectangular shapes is more than one.
- All rectangular shapes covered an area less than two square meters.
- All rectangular shapes are parallel to each other and to the floor.
- All rectangular shapes have the same size.
- All rectangular shapes have the same vertical alignment.

This intuitive semantic description will hold for most of the shelves which are used in our lab for experiments. An important feature of the description above is that all semantic relationships and features are measurable in the spatial domain. This means, by analyzing the extracted shape information and their relationship to each other, the bridging between the spatial and semantic representation can be achieved for some shape primitives. In the remainder of this work we will describe how objects can be detected using spatial/semantic feature descriptors. The process of analyzing the environment using 3D



**Figure 1.** A point cloud taken by the robot in our lab. It shows a shelf and some objects which are contained in the shelf.

laser range finders is basically performed in two consecutive steps. In a first step, laser data is acquired using a tilting laser setup or 3D LIDAR (light detection and ranging) system and matched to an existing point cloud model. In a second step of the scene-recovery process, geometric information is extracted from the merged point cloud data. This can be achieved by using 2D plane extraction [16] or the direct extraction of 3D primitives [13]. Some common surface reconstruction methods include the ball pivoting algorithm [1] and

<sup>1</sup> Robotics Innovation Center, DFKI Bremen, Germany, email: first\_name.last\_name@dfki.de

the Delaunay triangulation [8].

Most of the described algorithms are aimed at reconstructing accurate surfaces and finding their application in reverse engineering of 3D structures. The accuracy of these algorithms results in high computational costs. They are thus not suited for robotic applications because the surface reconstruction can take up to several minutes.

The plane extraction algorithm described in [16] works well on noisy sensor data and uses an iterative region-growing algorithm. The drawback of this approach is that it needs organized point clouds (i.e. the neighbor of each point is known). This is common for 3D LIDAR systems but not true for merged or registered scans. The approach we present in this paper will allow scene recovery from unorganized point sets and will also extract features from the spatial domain. The approach described in [9] provides a fuzzy clustering algorithm in order to segment a 3D scenery into cluster subsets without model learning. An approach of iterative triangulation of unorganized point clouds is described in [10]. All the described algorithms above are dealing with the spatial domain and are usable for identifying regions in LIDAR generated 3D point cloud data.

Coming from the semantic side, [6] and [7] describe how semantic maps are used for high-level planning and spatial reasoning. The authors describe in their work the bridging between the spatial domain and the semantic domain which they call S-Box (spatial box) and T-Box (taxonomy box). The semantic interpretation of physical objects is carried out by optical marker identification but not directly on spatial interpretation of point cloud data.

In the work described in [11], a constraint network is chosen in order to identify spatial entities such as wall, floor, and doors. That work shows how an environment can be described efficiently by using only the two constraints “parallel to” and “orthogonal to”. We will extend this idea by adding additional spatial features which can be directly extracted using shape recovery on segmented regions.

## 2 SHAPE RECONSTRUCTION

3D data of the scenery is generated using a tilting laser scanner. From this, we generate a data set  $\Psi$  with vector elements  $\psi_i = (x_i, y_i, z_i)^T$ , representing points in space. We process the raw data in two consecutive steps. First, we apply a fast region-growing algorithm to segment the point cloud data into regions which belong to a coherent plane. In a second step, the geometric data of the detected planes are extracted. The shape of the planes are of major interest. Therefore, the segmented point regions are polygonized using alpha shapes. Our region-growing approach follows partly the algorithm described in [12], with the extension that our approach is able to process unorganized point clouds by efficiently integrating K-nearest neighbor (KNN) search into the growing process. Computationally, the most expensive function in the algorithm is the KNN search which can be approximated with a runtime of  $\mathcal{O}(n \log(n))$  [15]. Due to the requirement of being able to process unorganized point clouds which occur in registered scans, composed by using ICP [3], we have to optimize the KNN search during the region-growing process.

Our key idea is to perform the cost-intensive KNN search at the beginning and store each point separately with a pointer to its KNNs. The growing plane keeps track of its growing frontier, i.e. each plane has its assigned nearest neighbors which are the next candidate for the growing process. During the region-growing process, the KNNs of each point, which is grown into the plane, are merged with the region frontier. We briefly summarize Rabbani’s approach in Algorithm 1 and state our extension to this algorithm.



**Figure 2.** The semi-humanoid robot Aila, which was developed for dual manipulation tasks in a logistic scenario. The robot uses a tilting 3D LIDAR system for precise point cloud generation in 3D space. A second PMD LIDAR is used for obstacle avoidance.

The problem concerning runtime in the base algorithm is that the

---

### Algorithm 1 Region Growing Algorithm

---

```

1: Input : Pointset  $\Psi$ 
2: Output : Subsets  $R_1, \dots, R_n$  of regions belonging to one plane
3: while  $\Psi$  not empty do
4:   Select random seed point  $\psi$  from  $\Psi$ 
5:   Find nearest neighbor of  $\psi$ 
6:   if  $MeanSquareError(R_i \cup \psi) < \delta$  and
       $\|BestFittingPlane(R_i) - \psi\|_{\perp} < \gamma$  then
7:     Add new point  $\psi$  to  $R_i$ 
8:   end if
9: end while

```

---

nearest neighbor search has to be performed each time a new seed point is selected. Searching the nearest neighbor can be done in constant time if a single scan line is taken from a tilting laser or a 3D LIDAR system. This does not hold for merged, registered or filtered scans because the information of the nearest neighbor is lost.

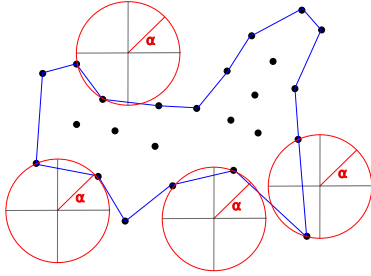
In order to extend Rabbani’s algorithm to be efficient for unorganized point clouds, we modify the algorithm by performing a KNN search only once at the beginning.

Each KNN of each point is then stored using a priority queue, ordered according to the distance to the seed points. Whenever a new point is added to the current region, the priority queue of the new added point is merged into the current region-growing front. By using pointer arithmetic we reduce memory usage.

After the regions have been segmented using Algorithm 1, the shape of each segmented point region is extracted. Because the appearance of each shape is important for later scene analysis, we extract the concave hull, i.e. the polygon which approximates the shape of the original object.

We choose the alpha shape approach, which is described more detailed in [14] and [2], in order to extract the concave polygon of the

detected regions. Figure 3 gives an idea of how alpha shapes are computed. For the analytic description of shapes the reader is pointed to [14]. The alpha shapes are a special case of a Delaunay triangulation



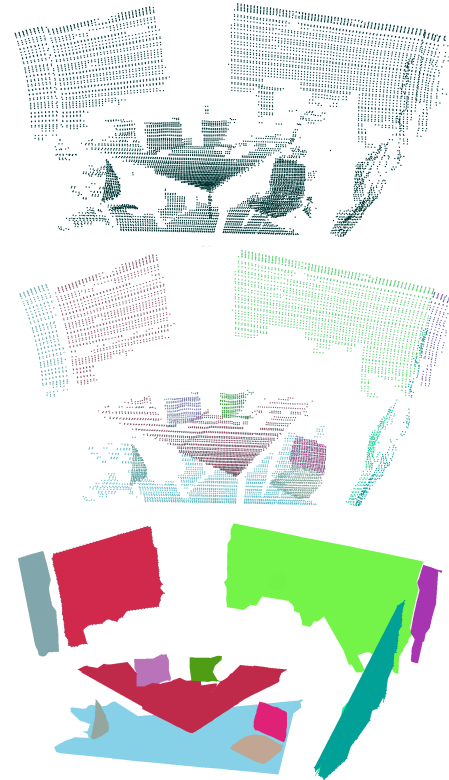
**Figure 3.** Alpha shape example in 2D. Alpha shapes can be described as the shape which is created if a disk (or a sphere in 3D) with the radius  $\alpha$  is rolled along the point cloud, starting with a border point. Each time a point is hit by the disk, it becomes the new pivoting point and the previous and the new pivoting point are connected.

and are mathematically well defined. For each point in a segmented region  $R$ , a vertex is created. Two vertices  $p_i$  and  $p_j$  are connected if there exists a 3D sphere with radius  $\alpha$  which has  $p_i$  and  $p_j$  on the boundary and does not contain any other point in  $R$ . In our implementation, we use the alpha shape reconstruction provided by [4]. Every alpha shape is then projected to a 2D plane for later shape analysis, e.g. eccentricity, size, shape, center of gravity. Note the dependency of the recovered shape on the radius  $\alpha$ . If  $\alpha \rightarrow 0$ , each point in  $R$  forms a separate shape, if  $\alpha \rightarrow \infty$  the convex hull is calculated. Figure 4 shows our lab scenery and the segmented planes and structures. For our approach we exploit the hypothesis that most of man-made indoor structure is of a rectangular shape. Examples for such structures are, for instance, tables, shelves, doors, walls, but also usable common objects of every-day use, such as monitors, TV-sets, microwave ovens, refrigerators, etc. In order to detect such objects, it is important to detect rectangular shapes from the set of alpha shapes calculated in the prior steps.

The detection of the rectangular shapes is done using Hough transformation. The shapes are therefore projected in a discrete 2D bitmap with fixed cell size. On this 2D image plane, the shapes can be analyzed straight forward. Based on the line segmentation results, the detected polygons are checked whether they intersect in four points and their dihedral angle is around  $90^\circ$ .

The detection of rectangular shapes is one of the important features needed to form the spatial feature vector described in the next section. In addition to the “rectangularness” of the shape, important features like the area covered and the expansion can be extracted. Before we introduce the spatial feature descriptors in the next chapter, we summarize the processing steps for unorganized point clouds.

- KNN search and storage in a priority queue for each point.
- Application of region-growing algorithm using the KNN-priority queues as growing frontier.
- Extraction of concave polygons using alpha shape recovery
- Projection of alpha shapes into discrete 2D bitmap.
- Detection of lines using Hough transformation.
- Detection of rectangular shapes using line analysis.
- Extraction of form features from the 2D bitmap.



**Figure 4.** A lab scene and the corresponding segmentation steps. After the point cloud has been segmented and after the generation of the alpha shapes, some 2D planes are left for further feature analysis. As a human it is easy to detect the floor, the table, and the chair based only on the knowledge of spatial relationships between the objects.

### 3 SPATIAL FEATURE DESCRIPTION

Once the shapes have been recovered from the unorganized point cloud, the goal is to classify the structure the robot perceives and to label the structure with semantics. To make semantic labeling possible in indoor environments, we make use of some basic assumptions. If we look around in a typical indoor environment like a household environment or an office environment, it is clear that most structures are of rectangular shape and mostly parallel or orthogonal to each other.

We will explain our semantic labeling approach using a simple example: Think of two persons who are able to communicate with each other. One person is able to perceive his environment, the other is not but has a perfect knowledge about the environment. One can think of the first person being in the spatial domain (without any understanding) and the other person being in the semantic domain.

Now think of how the person in the spatial domain would describe a *table* without knowing what a table is. A few spatial features would be sufficient until the person in the semantic domain can guess what is meant by the description (e.g. rectangular shape, parallel to the floor (or ceiling), height less than one meter from the floor, etc.). What happens is that the person in the semantic domain matches the available information to its internal model. Similar to the processing in a decision tree, every additional information given will increase the likelihood for a certain entity in the semantic model space.

Similar to the example above, the robot has to extract a vector of feature descriptors of the spatial entities in order to compare them with the semantic knowledge database. In a first approach, we define a set of vectors which are able to describe spatial entities of an environment. The feature vector  $\Phi$  is defined as

$$\Phi = (A, E, R, \Theta)^T,$$

where  $A \in \mathbb{R}^2$  is the area covered by the shape,  $E \in \mathbb{R}$  describes the maximum extension of the detected entity, and  $R \in [0, 1]$  describes the “rectangularness” of the entity, with  $R = 1$  describing a perfect rectangle. In our current approach, we only consider perfect rectangles as a binary decision. In later implementations, we also want to consider similarities to rectangles in order to increase the robustness of our approach. The reason to look for a rectangular structure is given by the observation that most artificial objects have a rectangular plane in their structure (e.g. doors, shelves, closets, walls, monitors, fridges).

$\Theta \in [0, 1]$  describes the relationship to other detected entities and is given by

$$\Theta = P(\omega, \sigma)$$

where  $\omega \in \Omega$  is the target of the relationship and  $\sigma \in \Sigma$  is the definition of spatial relationship.  $\Omega$  is a set of targets, i.e. labels in the semantic domain;  $\Sigma$  holds the attribute space which maps semantic attributes to spatial attributes. The labels in the semantic domain are defined in a model data base and include entities like *desk*, *table*, *door*, etc.

The attributes describe the spatial relationship between the detected entities (i.e. the parent of the relationship) and the target entity. An example for an attribute is

$$\begin{aligned} \sigma &\rightarrow [0, 1] : \\ \text{above} &\rightarrow (Pos\_Z_{parent} - Pos\_Z_{target}) < 0, \end{aligned}$$

which means that the target is below the parent entity. In our current implementation, we again consider a likelihood function in order to deal with uncertainties. For instance, two shapes can be parallel with

the certainty of 0.9 due to noise and rounding differences in the extraction process.

$P \in [0, 1]$  maps the relationship between the parent entities and the target entity, where 1 is the maximum likelihood. Another advantage of our approach is that it can be chained forward. For instance, an object on the table can be identified as a monitor with likelihood 0.8. Because the likelihood of the table is 0.5, the likelihood of being a monitor for the object is reduced to 0.4

Mapping semantic attributes to spatial relationships is the main contribution in our approach. A mapping between attribute *A above B* and a geometric expression would include that the z-value of the spatial entity A is higher than the z-value of B.

Simply speaking, we solve the semantic classification by recursively calling the relationship function  $\Theta$  until a **spatial axiom** is reached. The spatial axioms are defined by entities which do not depend on a relationship with other entities. They are defined as shapes having the spatial feature vector

$$\Phi = (A, E, R, 1)^T,$$

implying that there is no relationship needed in order to put semantic labels on spatial axioms, so the likelihood is set to 1.

An example of a spatial axiom is, for instance, a floor which is the lowest horizontal entity in reference to the robot coordinate system. So the floor is not identified by a relationship to other entities.

From the spatial axioms, the relationships  $\Theta$  are resolved until the root of the classification tree is reached. The function  $\Theta$  is a recursive call to the semantic domain space. In order to define a match between a model entity and an extracted spatial entity, we define the following similarity equation. The spatial feature descriptor of a model  $\Phi_{model}$  and an extracted spatial entity  $\Phi_{entity}$  are similar if

$$\Phi_{model} \odot \Phi_{entity} < \delta,$$

where  $\delta$  is the similarity threshold. The feature disparity function  $\odot$  between  $\Phi_{model}$  and  $\Phi_{entity}$  is defined as:

$$\begin{aligned} \Phi_{model} \odot \Phi_{entity} := & \|A_{model} - A_{entity}\| \\ & + \|E_{model} - E_{entity}\| \\ & + \|R_{model} - R_{entity}\| \\ & + \|\Theta_{model} - \Theta_{entity}\| \end{aligned}$$

When classifying structures, we are not dealing with full 3D perception but with a projection of 2D shapes in 3D space which is typical for a LIDAR recovered structure. Considering shape analysis, all detected shapes are projected onto a 2D plane. In order to analyze the recovered shapes (cf. Section 2), the planes are quantized by a projection in a 2D occupancy grid with a fixed grid of 1cm per pixel.

In our first approach, we focus on rectangular shapes, which will cover most of the objects found in indoor environment. In order to detect rectangular shapes in 2D, a Hough transformation is used [5]. To achieve this, all detected alpha shapes are projected into a discrete 2D bitmap. In order to detect a rectangular shape, all lines are analyzed with regard to their intersection in four points and whether the angles between the lines are  $\sim 90^\circ$ .

We now summarize our approach for the semantic labeling of spatial entities.

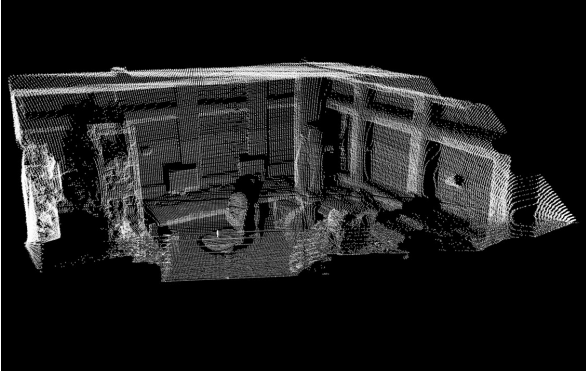
- A spatial database with labels and spatial description is set up. Each entity is represented by a spatial feature descriptor  $SFD_{model}$ . Each element is described by a spatial feature descriptor  $\Phi_{model} = (A, E, R, \Theta)^T$ . Some entities must be spatial axioms and not depending on the relationship to any other entity (i.e.  $\Theta = 1$ ).

- The parameters A,E,R of  $\Phi$  are extracted from the detected shapes in the spatial domain using the rectangle detection.  $\Theta$  is evaluated until a spatial axiom is found.
- The disparity function  $\Phi_{model} \odot \Phi_{entity}$  is evaluated. If the spatial feature descriptors are similar, the detected entity is matched with the model.

#### 4 PRELIMINARY RESULTS

We tested our algorithm in a typical indoor environment using the robot setup pictured in Figure 5. For the scan, we chose 100.000 points at an opening angle of  $180^\circ$  horizontally and  $90^\circ$  vertically. The segmentation results are shown in Figure 6.

It is obvious that important spatial entities are segmented correctly



**Figure 5.** A point cloud of an office environment including 100.000 points

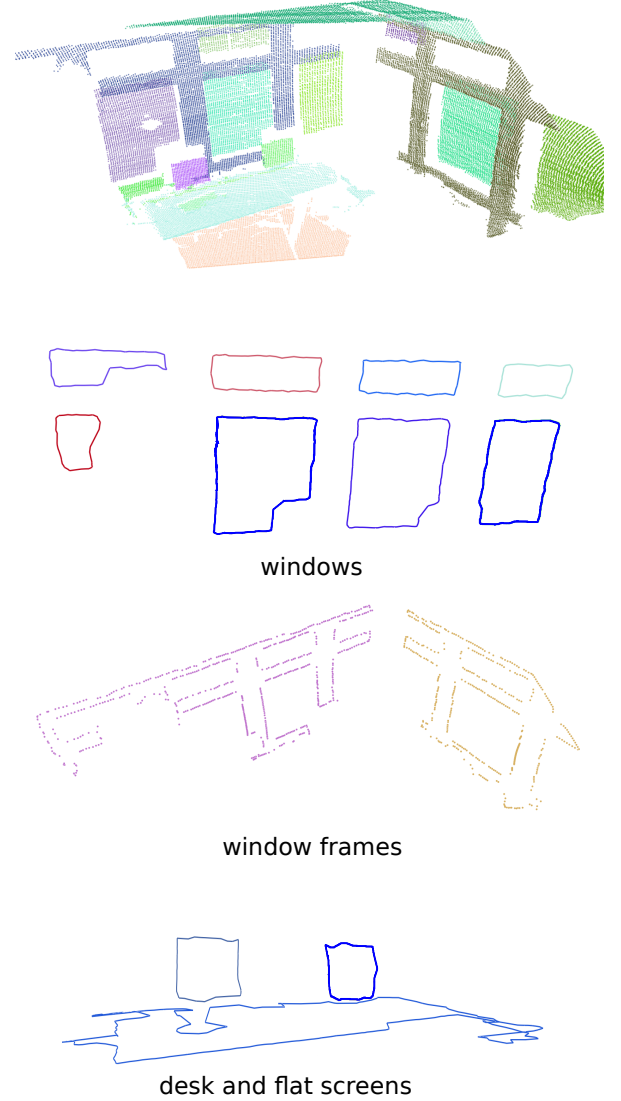
and the shape is recovered correctly. Besides the correct outer shape, important spatial information can be derived from the shapes, e.g. their appearance, the plane normal, and the metric extension. By this means, the 3D scene can be interpreted using constraints based on pre-knowledge about the environment. In a typical office environment, a table can be described as a rectangular shape parallel to the floor. The floor itself may be identified as being the largest horizontal plane with the lowest z-value, while the perpendicular, rectangular shape above the desk may be classified as flat screen monitors. Figure 7 shows an example in order to extract the features A,E,R from the shape polygons. In order to classify the shapes recursively, we define three semantic entities, i.e. *desk*, *screen*, *floor* with their spatial relationships.

$$\begin{aligned}
 \Phi_{desk} &= (1.28, 1.60, 1, \Theta(\text{floor}, \text{parallel}))^T \\
 \Phi_{screen} &= (0.24, 0.57, 1, \Theta(\text{desk}, \text{orthogonal}) \wedge \Theta(\text{desk}, \text{above}))^T \\
 \Phi_{floor} &= (2.0, 2, 0, 1)^T
 \end{aligned} \quad (1)$$

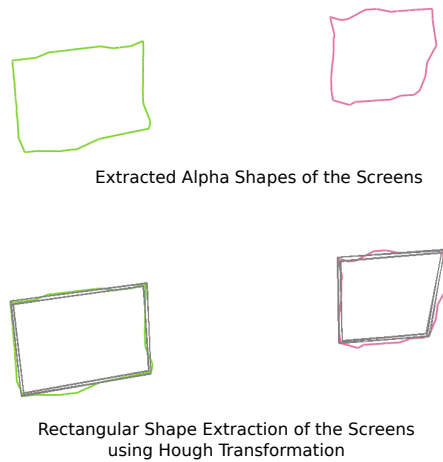
The relationships *parallel*, *orthogonal*, and *above* map the corresponding entities to the spatial relationships

$$\begin{aligned}
 \sigma &\rightarrow [0, 1] : \\
 \text{parallel} &\rightarrow (N_{parent} \cdot N_{target}) - (|N_{parent}| \cdot |N_{target}|) < \epsilon \\
 \text{orthogonal} &\rightarrow N_{parent} \cdot N_{target} < \epsilon \\
 \text{above} &\rightarrow (Pos\_Z_{parent} - Pos\_Z_{target}) > 0
 \end{aligned} \quad (2)$$

In the relations above,  $N$  denotes the normal vector of the extracted regions belonging to the regarded shape (cf. Section 2).  $Pos\_Z$  denotes the position of the shape (represented by the center of mass of the corresponding shape) in the vertical direction. Note that alpha



**Figure 6.** The results of our segmentation algorithm. The planes in the point cloud are segmented correctly and assigned to different colors. Some selected alpha shapes are presented. The window shapes, the shape of the window frame, and the shape of the flat screens on the desk are clearly recognizable after the polygonization.



**Figure 7.** The two flat screen monitors from our scenery in Figure 6. First, the alpha shapes are extracted. Using our rectangle detection algorithm, the shape is clearly defined as rectangular. From the shape, the parameters  $A$  (area) and  $E$  (extension) can be easily extracted. The rectangularness ( $R$ ) is set to 1 for each of the shapes. The relation function  $\Theta$  is omitted in this example because no other entities are related to the shapes.

shape or Hough transformed structures do not contain any relationship to each other. The extracted relationship features are processed during the region-growing process, e.g. center of gravity, or normal vector.

Currently, we are able to extract all the spatial features mentioned in this paper and chain the extraction of the relationship function  $\Theta$ . What is still missing to finally prove our concept is the implementation of the search algorithm that is able to match the spatial feature descriptors of detected entities to existing model feature descriptors. First experiments are promising considering only three simple entities, such as floor, table and monitor standing on the table. The next step is to match the features described in this paper to a semantic database of objects.

## 5 CONCLUSION

In this paper, we combined a method for recovering structure from unorganized 3D point clouds in the robotics domain. We presented two algorithms: The first transforms the point cloud into independent plane segments. The planes are then processed by computing the concave hull using an alpha-shape algorithm. By this means, the shapes of objects can be recovered efficiently. We showed how rectangular structures can be detected from the extracted shapes after the polygonization and which features can be extracted in order to apply semantic labeling to spatial entities. We introduced a spatial feature description together with a spatial relationship mapping, allowing to find labels for detected entities. We finally provided first results of the geometrical extraction process.

Future work focuses on the implicit mapping between semantic and spatial entities. For the time being, we are able to fill the feature vectors for known entities and define a similarity function. The last step, i.e., the automatic classification of detected objects is still to be realized. Another research problem is to build a descriptive ontology

of the semantic space, allowing spatial reasoning in the semantic space and using our approach for bridging the gap between the semantic and the spatial domain. Our main research goal is to have a descriptive language for spatial entities which can be searched for object classification based on similarities in the feature space.

## ACKNOWLEDGEMENTS

The work is sponsored by the German Federal Ministry of Education and Research (BMBF) within the “SEMPROM” project under contract number 01IA08002.

## REFERENCES

- [1] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, ‘The Ball-Pivoting Algorithm for Surface Reconstruction’, in *IEEE Transactions on Visualization and Computer Graphics*, volume 5, pp. 349–359, Los Alamitos, CA, USA, (1999). IEEE Computer Society.
- [2] Fausto Bernardini and Chandrajit L. Bajaj, ‘Sampling and reconstructing manifolds using alpha-shapes’, in *CCCG*. IEEE Computer Society, (1997).
- [3] P. J. Besl and H. D. McKay, ‘A method for registration of 3-d shapes’, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **14**(2), 239–256, (1992).
- [4] CGAL Editorial Board, *CGAL User and Reference Manual*, 3.5 edn., 2009.
- [5] R.O. Duda and P.E. Hart, ‘Use of the Hough transformation to detect lines and curves in pictures’, *Communication of the ACM*, **15**, (January 1972).
- [6] C. Galindo, J.A. Fernández-Madrigrá, J. González, and A. Saffiotti, ‘Robot task planning using semantic maps’, in *Robotics and Autonomous Systems*, volume 56, p. 955966. Elsevier, (2008).
- [7] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernandez-Madrigrá, and J. Gonzalez, ‘Multi-hierarchical semantic maps for mobile robotics’, in *Proc. IROS*, p. 34923497. Citeseer, (2005).
- [8] C Kuo and H Yau, ‘A Delaunay-based region-growing approach to surface reconstruction from unorganized points’, *Computer-Aided Design*, **37**(8), 825–835, (2005).
- [9] N. Lomenie, ‘A generic methodology for partitioning unorganised 3D point clouds for robotic vision’, in *First Canadian Conference on Computer and Robot Vision, 2004. Proceedings.*, pp. 64–71. Ieee, (2004).
- [10] Z.C. Marton, R.B. Rusu, and M. Beetz, ‘On Fast Surface Reconstruction Methods for Large and Noisy Datasets’, in *Proceedings of the IEEE International Conference on Robotics*, (2009).
- [11] A. Nüchter and J. Hertzberg, ‘Towards semantic maps for mobile robots’, *Robotics and Autonomous Systems*, **56**(11), 915926, (2008).
- [12] T. Rabbani, F. van Den Heuvel, and G. Vosselmann, ‘Segmentation of point clouds using smoothness constraint’, in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 36, p. 248253. Citeseer, (2006).
- [13] R. Schnabel, R. Wahl, and R. Klein, ‘Efficient RANSAC for Point-Cloud Shape Detection’, *Computer Graphics Forum*, **26**(2), 214–226, (Juni 2007).
- [14] Wei Shen, ‘Building boundary extraction based on lidar point clouds data’, in *ISPRS08*, p. 157. ISPRS, (2008).
- [15] Pravin M. Vaidya, ‘An optimal algorithm for the all-nearest-neighbors problem’, in *SFCS ’86: Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pp. 117–122, Washington, DC, USA, (1986). IEEE Computer Society.
- [16] N. Vaskevicius, A. Birk, K. Pathak, and J. Poppinga, ‘Fast Detection of Polygons in 3D Point Clouds from Noise-Prone Range Sensors’, in *IEEE International Workshop on Safety, Security and Rescue Robotics, 2007. SSRR 2007*, p. 16. IEEE Press, (2007).

# Working Simulations with a Foundational Ontology

Robert Porzel and Tobias Warden<sup>1</sup>

**Abstract.** Simulation is a commonly used method, for example in economics, engineering or computer science. In many cases simulations become difficult to design and control, especially when the simulated actors need to possess, acquire or process knowledge in some form. Today there exist simulation systems that are based on the multi-agent paradigm and use formal ontologies for representing the world, internal states of an agent or domain knowledge. However, such systems frequently feature individual non-congruent models that are not aligned with state-of-the-art foundational ontology frameworks. In this work we outline a foundational simulation ontology (FSO) that models simulations themselves so that not only the domain-specific models can be aligned to a standardized upper level, but also scenario design for ontology-based simulations can be performed by applying state-of-the-art ontology engineering principles. Our showcase scenario is that of autonomous logistic processes, where we apply our model for the domain of transport logistics.

## 1 Introduction

Simulation constitutes a well-established method for the examination of specific properties of the individual approaches that are being simulated. Early examples of this method can be found in architectural engineering *simulations*, where small scale models were constructed to test the static properties of the real entity to be constructed. Today, simulations do not require *in vivo* physical models, but can be set up and examined *in silico*. Numerous techniques have been employed in these software-based simulations. In this field of study some well-established paradigms have emerged, such as equation-based approaches or ones using PetriNets [4]. More recently, also simulations that are based on the multi-agent paradigm have been examined and applied [30].

Some examples for such multiagent-based simulations are found for system analysis and evaluation in a variety of domains ranging from models of bee recruitment [22] to simulating complex business processes, such as for supply networks [3]. Generally speaking, the agent-based simulation paradigm lends itself well to the simulation of complex systems on the micro-level where individual decision makers are modeled explicitly as autonomous agents embedded in dynamic environments. More specifically, and in contrast to alternatives such as equation-based modeling, the agent-centered approach to modeling facilitates the design of complex technical systems [30] due to:

- the opportunity for task decomposition,
- a natural mapping from real-world actors or entities to agents,
- a focus on modeling of individual behavior,
- the availability of multiagent system development frameworks.

Still, the concrete decision to embrace multiagent-based simulation for evaluation purposes when starting from a blank slate, is often perceived as a mixed blessing as the effort required to design particular multiagent-based simulations, especially when increasing the number of involved agents and/or with higher environmental complexity, often exceeds that of comparable simulation approaches, such as PetriNets or queuing networks [22]. Off-the-shelf agent frameworks are typically not designed to consider simulation-specific issues, such as synchronization, for which solutions exist in standard simulation approaches [4]. In addition, the design of the simulation environment itself in which the multiagent system can be placed, requires significant development resources. Thus, there is still an engineering challenge for multiagent-based modeling and simulation to be addressed [34]. More recently, efforts have been undertaken to employ formal and explicit knowledge models, i.e. formal ontologies, for describing the simulation world model [15]. Despite the inferential benefits gained from employing ontologically modeled knowledge for multiagent-based simulation, we have identified several remaining issues that need to be addressed for creating portable and (re-)usable knowledge models simulation systems:

- there still remains a substantial hurdle for the engineer to design and configure a simulation,
- currently employed domain models provide little extensibility and do not scale well to other application domains and simulation systems,
- the need for contextual reification is not addressed so that agents only poorly cope with dynamic changes in the environment that need to be reflected in revisions of their individual conceptualizations of the (simulated) world.

In this work, we firstly present the state of the art in ontology-based multiagent simulation systems and show how the aforementioned challenges for creating scalable simulations<sup>2</sup> in complex domains can be approached while - at the same time - lowering the hurdle for the non-ontologist to design the simulations accordingly. To that end we propose a foundational simulation ontology which models simulations *per se* and thereby provides a set of ontological patterns for *plugging in* the domain-specific ground knowledge of what is being simulated. On that basis the proposed framework additionally allows for a context-dependent reification of *ground* domain entities by means of including an additional set of descriptive patterns.

<sup>1</sup> University of Bremen, Germany, email: {porzel,warden}@informatik.uni-bremen.de

<sup>2</sup> Please note that *scalable* as it is employed here does not refer to scaling the number of agents engaged in a simulation, but rather means going from comparatively simple simulations to more complex ones.

## 2 State of the Art

As stated above, simulations are an appropriate and widely used means to evaluate the performance and adaptation abilities of systems in dynamic environments. For many domains, such as logistics or communication networks, discrete event simulation [4] has been the predominant simulation technique. However, in the case of more complex models, sequential simulation will exhibit poor runtime performance. For this, parallel distributed simulation systems have been developed [7] that enable the integration of simulation hardware as well as different simulation systems. Therefore, a so-called *High Level Architecture* [20] has been proposed defining services and interfaces to couple multiple simulation systems with potentially different simulation purposes. An alternative approach to distributed simulation of complex systems is constituted by multiagent-based simulation (MABS). In this case, the simulation model comprises multiple autonomous interacting software agents [37] that may run concurrently and distributed over multiple computers. The fundamental motivation behind multiagent-based simulations is to model agents as a direct and natural mapping of the simulated real-world entities acting in the simulation. This is particularly useful if the simulated world consists of multiple technical systems, organizations, and/or humans. Consequently, a major application domain has been social simulation [5, 30]. Beyond simulation distribution, the model decomposition to agents in multiagent-based simulation allows for extension, integration, and substitution of agents. Researchers or other system users that agree on an environment and interaction model for the domain of interest can add their own agents in a common scenario. The respective agents could be evaluated in separate simulation experiments or even concurrently in a competitive manner. The latter case, for instance, is applied in the RoboCup soccer simulation league [25].

Still, the research challenge of how best to design a simulation and create the corresponding agents' world models remains an open issue. For approaching these questions, we concentrate in particular on situation-aware agents which extend and substantiate the classical rational agent definition. For this category of agents, their respective belief about the world is no longer taken for granted. It is rather actively controlled by information acquisition as an additional meta-level reasoning process [14, 23]. Modeled human and artificial *decision makers* in our interpretation include both human controllers in the simulated processes as well as inanimate entities participating agentively in the simulation.

With regard to the representation of ground domain knowledge in the field of logistics, domain-specific ontologies have been compiled which formalize and explicate the domain at hand, e.g., transport networks, transport and production logistics, or physical goods in the logistics domain [34]. Despite these forerunners in building explicit formal models for simulating the properties of autonomous agents, there are several leaps remaining to be made in order to pass the aforementioned hurdles regarding the simulation design processes, the scalability and re-usability of the ensuing systems and the agent's respective ontological flexibility in terms of possessing the descriptive powers to construe their world dynamically and independently.

In the following, we will discuss how the research challenges, listed in Section 1, have been approached in the knowledge management sub-project of the Collaborative Research Center 637 on autonomous logistic processes and their limitations<sup>3</sup>. In this research effort, rational agents and multi-agent systems have been identified as a vantage point for conceptual modeling and simulation

of cooperating logistic decision makers, with a particular focus on situation-aware agents which extend and substantiate the classical rational agent definition [36]. As mentioned above, for this category of agents, their respective belief about the world is no longer taken for granted.

Next, we will, therefore, discuss how a firm ontological grounding can be established that satisfies the requirements of simulating autonomous logistic processes with multi-agent systems.

## 3 Adding Foundational Knowledge

In case an ontology is to be used solely for capturing and representing knowledge for a specific resource within a given community and if the intended meaning of the terms used within the respective community is generally understood and agreed upon in advance by all its members, then little is to be gained by the employment of a foundational ontology, such as DOLCE or SUMO [24, 27]. If, however, an ontology is to be extended or re-used in different settings or even ported to new domains and applications, then a foundational ontology becomes indispensable, as previous attempts on building scalable knowledge models have shown [12, 11, 28]. While these matters have been discussed before and are by now widely accepted in the ontology engineering community a new level of complexity is added for the case of simulating logistic processes with autonomously acting and learning agents as we will discuss in Section 3.2 below.

Generally speaking, a foundational ontology constitutes an axiomatic theory about the high-level domain-independent categories in the real world [17], such as object, attribute, event, spatial and temporal connections and the like. The purpose of a foundational ontology is to act as a modeling basis for building individual domain ontologies, e.g., an ontology of logistic processes and objects as in our showcase application. Equally important is that foundational ontologies provide ontology design patterns that prescribe best practice modeling, avoid ontological idiosyncrasies and save a substantial amount of modeling effort [9]. In short, the benefits of using a foundational framework are that:

- foundational ontologies provide a reference point for comparing different possible ontological approaches, and are useful for analyzing, harmonizing, and integrating existing domain ontologies and specific metadata standards, such as electronic product catalogs of existing logistic IT systems;
- foundational ontologies also provide a starting point for the design of new domain ontologies - rather than having to begin from scratch a foundational framework provides a predefined set of ontological entities that can be extended for the specific domain ontologies;
- ideally, the foundational ontology also provides applicable ontology design patterns for handling re-occurring modeling needs, such as the location of an entity in space and time.

In this work we will highlight the latter feature in our description concerning the reification of ground objects, where several design patterns can be applied. In this respect also our choice of employing DOLCE as foundational ontology was motivated by the need to enable agents to adapt their conceptualizations and create new ones *on the fly* because several dedicated DOLCE-modules provide a corresponding framework, as we will discuss in greater detail below. DOLCE belongs to the WonderWeb library of formal ontologies [24].<sup>4</sup> It is intended to act as a starting point for comparing and

<sup>3</sup> Web site: [www.sfb637.uni-bremen.de](http://www.sfb637.uni-bremen.de)

<sup>4</sup> More detailed comparisons of DOLCE to other foundational frameworks

elucidating the relationships with other ontologies of the library - so-called *modules*. DOLCE is based on the fundamental distinction between enduring entities (i.e., objects or substances) and perduring entities (i.e., events or processes). The central relation between *Endurants* and *Perdurants* is that of participation. For example an *Endurant*, such as a transportation vehicle, participates in a set of possible activities, such as driving or parking, which in their nature are *Perdurants*. DOLCE also introduces *Qualities* as another category that can be seen as the basic entities one can perceive or measure: shapes, colors, sizes, sounds, smells, as well as weights, lengths or electrical charges. Spatial locations constitute a special kind of physical quality and temporal qualities encode the spatio-temporal attributes of *Endurants* and *Perdurants*. Finally, *Abstracts* do not have spatial or temporal qualities and they are not qualities themselves. An important example for the domain of logistics are so-called *Regions* that are used to encode the measurement of qualities as conventionalized in some metric or conceptual space.

Furthermore, DOLCE features a rich reference axiomatization in modal logic (S5), which captures basic ontology design patterns, such as location in space and time, dependence or parthood. The ontology core is minimal in that it only includes the most general concepts and patterns. This, as pointed out above, makes it well-suited for modularization. In fact, there is a wealth of additional theories that can be included on demand, which we will take as our starting point for modeling simulations themselves as part of a foundational system of ontologies.

### 3.1 Separating the Real from the “Not So Real”

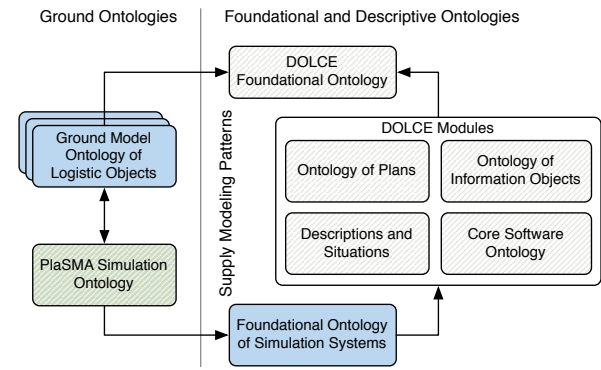
Before starting to model domain-specific entities and the corresponding logistic processes, we had to ask ourselves whether the existing ontological infrastructure provides suitable starting points for our modeling purposes. As a result of this, we have chosen to employ a set of ontological modules which have been designed along with the DOLCE framework, namely:

- the Ontology of Information Objects [16];
- the Ontology of Plans [10];
- the Core Software Ontology [29];
- the Descriptions and Situations module [13].

Before exemplifying the usage of these ontological modules, we provide an overview of this infrastructure and how our foundational simulation ontology is fitted into it in Figure 1.

As it is clear that a software agent that is simulating a logistic object - or any other agentively acting entity for that matter - is not the real thing itself, a proper ontological place for rational software agents needs to be established. Hereby the Core Software Ontology provides a fitting design pattern. Therein a differentiation is made between a *ComputationalActivity* and a *ComputationalObject*. While *ComputationalObjects* are actually modeled as physical objects, i.e. *Endurants*, in DOLCE - because they physically exist in the hardware and memory of a running computer system - a *ComputationalActivity* is a sub-class of *Perdurant* - because they refer to events or processes in which *ComputationalObjects* participate. It, therefore, becomes almost straightforward to model a software agent as an *AgentivePhysicalObject*. This is not because it might *control* a physical object, but because it itself is an agentive object constituted in the physical matter of a *ComputationalObject*. Consequently, a new

also reflect the individual ontological commitments that underly each modeling framework [28].



**Figure 1.** Framework for modeling multiagent-based simulation systems, grounded on DOLCE, using the showcase of simulating autonomous logistic processes with the PlaSMA Simulation Framework [34].

class of *SimulatedAgentivePhysicalObjects* was added as a sub-class of *AgentivePhysicalObject*. This means that the entity simulating a logistic object is both a physical object and an agent itself and assumes the relation *internally-represents* which ranges over the class *Plan*. Thereby an explicit connection to the Ontology of Plans is established.

This example also constitutes a textbook showcase of the advantage of employing foundational systems and their modeling patterns: by *simply* finding the right place for the entity to be modeled, as shown above for the case of a simulation agent, one obtains a whole ensemble of relations and ranges that come from the corresponding super-classes that are part of the employed foundational module. These relations and ranges, in turn, actually serve the desired needs of the modeler. Of course, this is not a coincidence at all but a result of careful ontology engineering performed by the designers of the foundational framework. However, in many communities the employment of a foundational framework is still regarded as a cumbersome and often futile exercise, while - in our minds - rather the opposite is true.

Analogously, once a corresponding *SimulationObject* is inserted as a sub-class of *ComputationalObject*, we can express that:

- a *SimulationObject* realizes **some** *InformationObject*<sup>5</sup>
- a *SimulationObject* participates-in **some** *ComputationalActivity*

Also, by using the basic DOLCE design patterns to attribute spatial and other qualities to physical objects, we were able to model qualities - such as simulated traffic flow or simulated speed limits - of our simulated world by re-using the patterns established for real objects and their qualities.

For example, we also pointed to the additional challenge of allowing our software agents to learn in their environments and form new conceptualizations based on their respective body of experience in the simulated world, which will be described in the following section. Before that, however, we want to point at the emergent possibility to employ the Core Software Ontology even further, as the entire class hierarchy of the simulation software can now be described ontologically as well. This makes it feasible to interact - before and during a running simulation - not only with the simulation itself, but also with the underlying model. This, of course, presents additional challenges in the domain of human-computer interaction, but it puts the design of both the simulation and its software on an equal footing.

<sup>5</sup> Since *Software* is modeled as an *InformationObject* this solution expresses exactly what is needed for a foundational simulation model.

### 3.2 Reification of the Ground Domain

In recent work the notion of ontological design patterns has quickly become a central issue in ontology engineering research [8]. In its original form the first type of patterns, so called *logical patterns*, specifies ways of solving standard ontology modeling problems, such as how to model n-ary relations or how to employ subsumption macros [9]. The second type, so called *content patterns*, features applications of logical patterns, which means that instances of content patterns are composed of logical patterns and combinations thereof. Content patterns are concerned with specifying ways of representing everything that is not given by the logical vocabulary itself. For example, *is-a* relations come with the logical inventory, but *part-of* relations do not and are, consequently, part of the domain-specific content of the ontology. While the foundational layer provides the basic ontological distinctions, axiomatizations and design patterns for the development of further domain-independent and domain-specific layers of *ground* ontologies as well as additional layers of *descriptive* ontologies, which we will discuss below, it is important to note that this important distinction is primarily motivated as an ontological separation enabling an ontology engineer to express *reified* contexts [32] at the level of concepts or relations.

As a consequence of employing the Descriptions and Situations module [13], our simulation framework allows for the distinction between ground and descriptive models. Thus, we can provide a ground ontology of the logistic domain, featuring objects (such as trucks or containers), places (such as manufacturing and storage sites), as well as the transportation networks connecting them. However, these ground objects do not seek to model the roles that these entities can play - both in the real world or a simulated one.

For example, a truck can play the role of a means of transport. Nevertheless, it can also play the role of a freight object, if it is itself being transported. In an extreme case one can even consider that, for example, an entire storage facility is moved from one place to another and becomes logistic freight as a result and needs to be stored and transported itself.

The descriptive branch of the foundational ontology can then be applied for explicating such context-dependent reifications of real and simulated entities. For example, our proposed ontology framework consequently provides the logical and content patterns to express that a ground logistic class, such as a *Truck*, can be ontologically reified as a *MeansOfTransportation*, a *FreightObject*, or a *TrafficObstacle* depending on the context at hand.

As a result one can employ the same modeling instruments, including logical and content patterns, for descriptive entities as one employs for modeling ground entities. This, in turn, circumvents the need to resort to other logical instruments for describing these entities, such as to formulate so-called *theories* about the ground model, including theories about possible worlds or - in a weakened form - modal propositions about possible *situations* [2] that require universal algebra to express the semantics of the logical forms [35].

Therefore, the approach taken herein provides the possibility to employ the logical patterns, i.e. the specific logical vocabulary, of the given foundational ontology, for modeling descriptive entities in the same way as one does for the ground part. Again, this approach would be impossible without a foundational layer linking the ground and descriptive branches of the integrated ontology.

The DOLCE module *Descriptions and Situations* provides specific logical and content patterns for representing reified contexts and states of affairs [13]. In contrast to ground entities, such as physical objects or events, the extension of a descriptive ontology to include

different conceptualizations that an autonomously learning logistic agent may derive of these entities poses a challenge to the ontology engineer. The reason for this circumstance is the fact that these conceptualizations are taken to assume meaning only in combination with some other entity. Accordingly, as discussed above, their logical representation is generally set at the level of theories or models and not at the level of concepts or relations.

In order to avoid potential terminological confluences and express it in natural language a descriptive statement is *about* something, e.g. it represents the *meaning* of something in a given context, while a ground statement is about the thing itself, e.g. for classifying instances within a given domain. In this modeling framework a situation is, consequently, clearly defined as a set of instances from the ground domain.

For example, a situation could be constituted by the instances of a specific object, e.g. *Truck:HB-IV-42*, a specific individual container and a specific road, e.g. the A27 in Germany, at a specific time, e.g. a certain day when the traffic flow was at a specific viscosity level. When seeking to describe this situation one would somehow like to express that the truck played the role of a means of transport, the container played the role of the freight object and the path was played by the specific road that the truck took under good conditions. In some other context however, e.g. that of an accident - one might seek to describe the truck as responsible for the accident and the capsized container as an obstacle.

In any case one would like to seek to refrain from simply multiplying the ground *isa* relations to express these states of affairs, as this is problematic both from a modeling as well as from a practical point of view. For avoiding such a taxonomic explosion, a dedicated descriptive pattern for a context-dependent reification of ground entities can be employed.

The employment of the foundational logical patterns described herein yields additional advantages. For example, the employment of such patterns - even when used with hindsight by means of *refactoring* - existing ontologies into pattern-based ones, has been shown to be beneficial for ontology quality when measured in terms of performance on a given task, e.g. ontology alignment [33]. Most importantly, however, a descriptive pattern for context-dependent reification, i.e. a coding of the functional meaning of some thing expressed in the terminology, introduced above, is to represent a pragmatically analyzed situation. It, therefore, enables the ontology engineer or an artificial agent to express that, using the example provided above, some ground entity is playing the *functional role* of a *MeansOfTransport*, which is transporting some other entity on a road, that plays the role of the path on a day where the actual traffic flow makes this possible.

The specified logical and content patterns of the *Descriptions and Situations* module feature three core descriptive entities, i.e. the classes *Courses of Events*, *Functional Roles* and *Parameters* [13]. These classes are linked by means of relations, which specify that:

- *Parameters* are *requisite-for* their *functional roles* and *Courses of Events*;
- *Functional Roles* are the *modality-targets* in the conceptualized *Courses of Events*.

Finally, the classes can be linked to the ground entities they describe, via the following relations:

- *Courses of Events* are *sequenced-by* *Perdurants*, i.e. processes within the ground ontology, such as *Locomotion*;

- *Functional Roles* are *played-by Endurants*, i.e. objects within the ground ontology, such as of type *Vehicle*;
- *Parameters* are *valued-by Regions*, i.e. phenomena that are sensed on scales, such as *TrafficFlow*.

Elaborating the example presented above in the domain of simulating logistic actors and processes, we can insert a class *LogisticActorRole* as a subclass of *AgentDrivenRole* that is already part of the Descriptions and Situations module. As individual subclasses of the class *LogisticActorRole* describe the functional roles of endurant entities, we proceeded by inserting specific logistic roles as further subclasses, such as *FreightActorRole*, *StorageActorRole* or *MeansOfTransportRole*. As defined by the Descriptions and Situations module all of these roles can only be played something that is an *Agent*, which is satisfied by our foundational simulation ontology, since *SimulatedAgentivePhysicalObjects* are *AgentivePhysicalObjects* which itself is a subclass of *Agent*. Additionally, these functional roles inherit further important relations, such as for the case of our class *LogisticActorRole* (*LAR*):

- *LAR* generically-dependent-on **some** *PhysicalEndurant*;
- *LAR* defined-by **some** *Description*;
- *LAR* participant-in **some** *Perdurant*.

This enabled both immediate relations to the corresponding physical objects as well as to the processes in which they participate. As a next logical step - for endowing the ground ontologies with a dynamic layer of agent-specific descriptions - these elementary patterns were employed to construct an underlying model of simulation patterns that will be exemplified in the following section.

### 3.3 Simulation Patterns

In our minds, the central determining factor in the construal of objects for their context-specific reification is constituted by their activities. In a sense one can regard the simulated world to afford the agents with an array of possible actions, which constrain and determine the roles which can be played by whom. Therefore, we take the concept of a *CourseOfEvents* as a starting point for our model of logistic tasks - whether they are simulated or real. Being a descriptive entity a *LogisticTask* is sequenced by perduring events, which means we need to specify the real world or simulated processes that can be construed, for example, as a *TransportationTask*.

Within this description, we can specify the kind of objects that can play the role of *MeansOfTransport* or *FreightObject*, which are taken from the ground domain ontology. Let us note, once more, that to say a truck can **play** the role of a *MeansOfTransport* in a given *TransportationTask* is quite different from saying that it **is** a *MeansOfTransport* in the ground ontology. Analogously, we can employ ground models of qualities and regions, e.g. for specifying traffic flow or speed limits, to supply the parameters that need to be valued for making a segment of the street network suitable as a path in a *TransportationTask*.

The employment of such patterns for modeling simulations based on the multiagent paradigm becomes extremely significant in the light of allowing an individual agent to form novel conceptualizations autonomously. While the ontology engineer still has the chance to constrain the range of possible construals, e.g. what kind of ground objects could be construed to play obstacles, means of transportation or else, it is still up to the agent to base these conceptualizations upon individual *experiences*. Hereby, it should be noted that, our foundational simulation ontology does not constrain which learning approaches or heuristics are to be employed for the context-dependent

reifications. Here one can employ a multitude of possible approaches ranging from the more traditional machine learning [26] to newer ones based on so-called *echo state* networks [21, 6] that feature a build-in short term memory. What the foundational simulation ontology does provide, however, is a target representation and modeling instrument to express the results of the learning processes in a formal and explicit manner, which automatically becomes part of the domain-specific simulation model.

## 4 Conclusion and Future Work

The work presented above assumes that simulations constitute a useful and important technique in many scientific branches, but that they become difficult to design and control, especially when the simulated actors need to handle non-trivial knowledge in some form or another. While the employment of ontology-based multiagent systems for those simulations where the simulated actors need to possess, acquire or process knowledge has been shown to be working well [34], we also found that an ontology that models simulations themselves is still missing. In our minds such a foundational simulation ontology offers potential advantages in several respects.

**Design Phase:** In the design phase of setting up a multiagent-based simulation the employment of a foundational simulation ontology makes it possible to exploit the ontological patterns provided by the foundational framework, which not only eases the task of scenario engineering, but also creates more congruent models across individual simulations and thereby facilitated their reusability.

**Runtime Phase:** During the runtime phase we anticipate the advantage that complex simulations can be rendered (more) accessible for human-computer interaction due to the more homogeneous explicit and formal representation of the involved agents and their tasks, as any presentation ultimately hinges on the underlying representation.

**Analysis Phase:** In the analysis phase easier and well-grounded analysis and evaluation of recorded simulation runs ought to be possible, as the logical instruments provided by the foundation simulation ontology come from the standard logical inventory of formal ontologies and can be employed for additional validation and inferential reasoning using standard reasoning engines [19, 18, 1].

Generally speaking, in this paper we have outlined that basic motivation and approach for employing foundational ontologies and a corresponding general simulation model together with some concrete examples and modeling choices. While some benefits of this approach can be shown qualitatively, e.g. the re-definition of simulation design as an ontology engineering task, others await a quantitative evaluation. This is especially true for evaluating how the corresponding augmentation of an agent's ontological knowledge influences their performance, for example, in the case of self-organized logistic processes.

Therefore, our future work will be on implementation and evaluation using task-based ontology evaluations methods [31]. In addition, we are currently exploring the question how the domain application in the domain of transport logistics can serve as a testbed for evaluation the interaction with complex simulations from a human-computer interaction perspective.

## Acknowledgments

This research has been funded by the German Research Foundation (DFG) within the Collaborative Research Centre (CRC) 637 Autonomous Cooperating Logistic Processes A Paradigm Shift and its Limitations at the Universität Bremen, Germany.

## REFERENCES

- [1] F. Baader, C. Lutz, and B. Suntisrivaraporn, 'Cel: A polynomial-time reasoner for life science ontologies', in *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06)*, eds., U. Furbach and N. Shankar, volume 4130 of *Lecture Notes in Artificial Intelligence*, pp. 287–291. Springer-Verlag, (2006).
- [2] Jon Barwise and John R. Perry, *Situations and Attitudes*, MIT Press, Cambridge, Mass., 1983.
- [3] M. Bloos, J. Schönberger, and H. Kopfer, 'Supporting cooperative demand fulfillment in supply networks using autonomous control and multi-agent-systems', in *INFORMATIK 2009. Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, (2009).
- [4] Carmen-Veronica Bobeanu, Eugene J. H. Kerckhoffs, and Hendrik Van Landeghem, 'Modeling of Discrete Event Systems: A Holistic and Incremental Approach using Petri Nets', *ACM Trans. Model. Comput. Simul.*, **14**(4), 389–423, (2004).
- [5] P. Davidsson, J. Holmgren, H. Kuhlbaeck, D. Mengistu, and M. Persson, 'Applications of agent based simulation', in *Multi-Agent-Based Simulation VII, International Workshop (MABS 2006)*, pp. 15–27. Springer, LNAI, vol. 4442, (2006).
- [6] Lutz Dickmann, Tobias Lensing, Robert Porzel, Christoph Lischka, and Rainer Malaka, 'Sketch-based interfaces: Exploiting spatio-temporal context for automatic stroke grouping', in *10th International Symposium on Smart Graphics*, (2010).
- [7] R. Fujimoto, *Parallel and Distributed Simulation Systems*, Wiley & Sons, New York, NY, USA, 2000.
- [8] A. Gangemi and V. Presutti, *Handbook of Ontologies (2nd edition)*, chapter Ontology Design Patterns, 221–244, Springer, 2008.
- [9] Aldo Gangemi, 'Ontology design patterns for semantic web content', in *M. Musen et al. (eds.): Proceedings of the Fourth International Semantic Web Conference*, Berlin, Springer, (2005).
- [10] Aldo Gangemi, Stefano Borgo, Carola Catenacci, and Jos Lehmann. Task taxonomies for knowledge content. metokis deliverable d07, 2004.
- [11] Aldo Gangemi, Nicola Guarino, Claudio Masolo, and Oltramari, 'Sweetening WordNet with DOLCE', *AI Magazine*, **24**(3), 13–22, (2003).
- [12] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with dolce, 2002.
- [13] Aldo Gangemi and Peter Mika, 'Understanding the Semantic Web through Descriptions and Situations', in *Proceedings of the ODBASE Conference*, pp. 689–706. Springer, (2003).
- [14] Jan D. Gehrke, 'Evaluating Situation Awareness of Autonomous Systems', in *Performance Metrics for Intelligent Systems Workshop*, pp. 206–213. NIST, (2008).
- [15] Jan D. Gehrke and C. Ober-Blöbaum, 'Multiagent-based Logistics Simulation with PlaSMA', in *Informatik 2007 - Informatik trifft Logistik, Band 1.*, Bremen, Germany, (2007). Springer.
- [16] N. Guarino. Deliverable d2: Ontology of information objects, 2006.
- [17] Nicola Guarino and Roberto Poli, 'Formal ontology in conceptual analysis and knowledge representation', *Special issue of the International Journal of Human and Computer Studies*, **43**, (1995).
- [18] V. Haarslev and R. Möller, 'Racer: A core inference engine for the semantic web', in *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), located at the 2nd International Semantic Web Conference ISWC 2003, Sanibel Island, Florida, USA, October 20*, pp. 27–36, (2003).
- [19] Ian Horrocks, 'The FaCT system', in *Proc. of the 2nd Int. Conf. on Analytic Tableaux and Related Methods (TABLEAUX'98)*, ed., Harrie de Swart, volume 1397 of *Lecture Notes in Artificial Intelligence*, pp. 307–312. Springer, (1998).
- [20] *Standard for Modeling and Simulation High Level Architecture. IEEE Standard 1516-2000*, ed., IEEE, IEEE Computer Society, 2000.
- [21] Herbert Jaeger, 'The echo state approach to analysing and training recurrent neural networks', Technical report, Technical Report 148, German National Research Institute for Computer Science, (2001).
- [22] Franziska Klügl, Christoph Oechslein, Frank Puppe, and Anna Dornhaus, 'Multi-Agent Modelling in Comparison to Standard Modelling', in *Artificial Intelligence, Simulation and Planning in High Autonomy Systems (AIS 2002)*, pp. 105–110. SCS Publishing House, (2002).
- [23] Hagen Langer, Jan D. Gehrke, Joachim Hammer, Martin Lorenz, Ingo J. Timm, and Otthein Herzog, 'A Framework for Distributed Knowledge Management in Autonomous Logistic Processes', *International Journal of Knowledge-Based and Intelligent Engineering Systems*, **10**(4), 277–290, (2006).
- [24] Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari, 'D18: Ontology Library (final)', Project deliverable, WonderWeb: Ontology Infrastructure for the Semantic Web, (2003).
- [25] N. Mayer, J. Boedecker, R. da Silva Guerra, O. Obst, and M. Asada, '3D2Real: Simulation league finals in real robots', in *RoboCup 2006: Robot Soccer World Cup X*, pp. 25–34. Springer, LNAI, vol. 4434, (2008).
- [26] Tom M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
- [27] Ian Niles and Adam Pease, 'Towards a Standard Upper Ontology', in *Proceedings of the Int. Conference on Formal Ontology in Information Systems*, pp. 2–9, (2001).
- [28] Daniel Oberle, Anupriya Ankolekar, Pascal Hitzler, Philipp Cimiano, Michael Sintek, Malte Kiesel, B. Mougouie, S. Vembu, S. Baumann, Massimo Romanelli, Paul Buitelaar, R. Engel, D. Sonntag, N. Reithinger, Berenike Loos, R. Porzel, H.-P. Zorn, V. Micelli, C. Schmidt, Moritz Weiten, F. Burkhardt, and J. Zhou, 'Dolce ergo sumo: On foundational and domain models in swinto (smartweb integrated ontology)', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, **5**(3), 156–174, (2007).
- [29] Daniel Oberle, S. Lamparter, Andreas Eberhart, and Steffen Staab, 'Semantic Management of Web Services', in *Service-Oriented Computing - ICSOC 2005*, volume 3826 of *LNCS*, pp. 514–519. Springer, (12 2005).
- [30] H. V. D. Parunak, R. Savit, and R. L. Riolo, 'Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide', in *Multi-Agent Systems and Agent-Based Simulation, First International Workshop*, volume 1534 of *LNCS*, pp. 10–25, Paris, France, (July 4–6 1998). Springer.
- [31] Robert Porzel and Rainer Malaka, 'A task-based framework for ontology learning, population and evaluation', in *Ontology Learning from Text: Methods, Evaluation and Applications*, eds., Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini, volume 123 of *Frontiers in Artificial Intelligence*, 107–122, IOS Press, (JUL 2005).
- [32] Erich Herrmann Rast, *Reference and Indexicality*, number 17 in *Logische Philosophie*, Logos Verlag, 2007.
- [33] Ondrej Světlík, Vojtěch Světek, Christian Meilicke, and Heiner Stuckenschmidt, 'Testing the impact of pattern-based ontology refactoring on ontology matching results', in *CEUR Workshop*, eds., Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Heiner Stuckenschmidt, volume 431. CEUR-WS.org, (2008).
- [34] Tobias Warden, Robert Porzel, Jan Gehrke, Otthein Herzog, Hagen Langer, and Rainer Malaka, 'Towards Ontology-Based Multiagent Simulations: The Plasma Approach', in *24th European Conference on Modelling and Simulation*, Kuala Lumpur, Malaysia, (2010).
- [35] Alfred North Whitehead, *A Treatise on Universal Algebra with Applications*, Cambridge University Press (Reprint 1960), Cambridge, UK, 1898.
- [36] Michael Wooldridge, *Reasoning about Rational Agents*, The MIT Press, 2000.
- [37] Michael Wooldridge and N. R. Jennings, 'Intelligent agents: Theory and practice', *Knowledge Engineering Review*, **10**(2), 115–152, (1995).

# An Approach to Formal and Semantic Representation of Logistics Services

Julia Hoxha<sup>1</sup> and Andreas Scheuermann<sup>2</sup> and Stephan Bloehdorn<sup>3</sup>

**Abstract.** Modern logistics systems are characterized by an increasing structural complexity and dynamicity, which arises from trends in the global economy, decomposition of supply chains and individual customer demands. These trends have amplified the need for greater flexibility in the (re-)configuration of supply chains and, additionally, the demand to move from centralized to decentralized IT solutions for planning and control.

This paper presents an approach to achieve flexibility and decentralization in supply chain configuration and management. The approach combines loosely-coupled logistics services with semantic technologies for a unified representation of diverse logistics data and service functionalities. The contribution is a formal knowledge model of the information in the logistics domain using ontologies. The paper further discusses the semantic representation of logistic services in a framework that enables automated and intelligent techniques for discovery, ranking, execution and efficient composition of services into more complex and flexible logistics processes.

## 1 INTRODUCTION

Globalization of markets, division of labor, decomposition of supply chains and individual customer demands lead to increased structural complexity and dynamicity of modern logistics systems. Formerly rigid supply chain structures with long-term contracts have diminished for the benefit of short-term contracts and event-driven supply chain configuration. The need to (i) flexibly assemble or adapt supply chains and corresponding control systems, and to (ii) integrate heterogeneous logistics data and exchange logistics information between globally dispersed actors have become key success factors.

Conventional IT solutions, mostly Enterprise Resource Planning (ERP) systems, are mainly based on centralized planning and control mechanisms, concentrating on services within a single organization. Consequently, these solutions insufficiently comply with the current demands and challenges, increasing the need for alternative flexible and decentralized approaches.

Meanwhile, from an IT viewpoint, current technologies, most notably Web Services, significantly facilitate the provision of services over computer networks, changing the way distributed computing systems are being architected. More and more software systems are designed as service-oriented computing architectures with loosely-coupled software components and data resources, which are easily accessible using standardized technologies [7, 10].

However, these service-oriented solutions are not fully adapted along supply chains. First, the external integration of logistics processes, i.e. the integration across organizational boundaries, still remains an ongoing and challenging issue. This is generally due to syntactical, structural, and semantic mismatches between enterprise systems, organizational disturbances, and constant reconfiguration of supply chains based on developments in the global economy [1, 14]. Therefore, we believe that the fundamental problems mainly lie in the lack of appropriate approaches, which support the efficient integration of the plethora of heterogeneous representations of logistics information, as well as the engineering of modular logistics services with flexible interaction. Second, we can assist in handling the complexity of making logistics decisions, which has constantly increased because of wider product variety, smaller lot sizes, more tiers, geographically dispersed actors and less vertical integration [13]. Significant advances in dealing with this complexity are needed, providing value-added services to the end-customers and increasing the flexibility of the involved supply chains. The added-value lies in finding those logistic services that optimally fulfill customer-specific requirements. This can only be realized if at least formal and correct descriptions of the services are available.

The objective of this work is to introduce an approach to formal and semantic representation of logistics services, adopting the concepts of service-orientation and Semantic Web technologies. We apply Semantic Technologies to formally model the information of the logistics domain. We further provide a semantic representation of modular real-world logistics services (e.g., transportation, storage, handling), so that they can be composed into more complex logistics processes and dynamic supply chains.

Our contribution is a principled approach that provides semantic representation of logistics information and services. We present (i) a motivating logistics application scenario, (ii) a formal knowledge model that captures the overall aspects of the logistics domain and (iii) an approach for the semantic description of logistics services in a service-oriented application framework.

The remainder is structured as follows: Section 2 illustrates an application scenario, which motivates the need for formal semantic representations. In Section 3, the formal knowledge model and the corresponding ontologies are presented, while Section 4 proposes a framework for semantic representation of logistics services. Section 5 discusses the benefits of using this framework in selected scenarios. In Section 6, we review related work, presenting afterwards in Section 7 conclusions and an outlook on future work.

<sup>1</sup> Karlsruhe Institute of Technology (KIT), Germany, email: julia.hoxha@kit.edu

<sup>2</sup> University of Hohenheim, Germany, email: andreas.scheuermann@uni-hohenheim.de

<sup>3</sup> Karlsruhe Institute of Technology (KIT), Germany, email: stephan.bloehdorn@kit.edu

## 2 USE CASE

To illustrate the situation of the external integration of logistics services, we take a deeper look at the integration of transportation services from the perspective of a fourth-party logistics (4PL) provider. A 4PL provider is regarded as a non asset-based integrator of logistics services along supply chains. In the following, we contrast common decision situations of the 4PL provider, as they are typically handled now, with the desired target situation, thus motivating our solution approach.

### 2.1 Current Situation

Given the end-customer requirements, e.g., amongst others, delivery time, service levels, characteristics of goods, as well as source and destination, the 4PL's ultimate goal is to integrate transport services conforming to these needs.

Typically, large catalogs containing contact details of potential transport service providers are inspected manually - mostly based on keyword search. A lack of common terminology to expose the capabilities of a provider leads to ambiguous service classifications favoring suboptimal search results. When there are no appropriate transport services available, additional manual effort has to be invested in decomposing the transport process into two or more transport services, which in combination match the final customer requirements.

In a next step, inquiries about the service provisioning are carried out by phone, fax, email, or sometimes through proprietary IT interfaces, which in each case strongly depend on the providers IT infrastructure and applications. Subsequently, the selection of a transport service provider is performed in a two-staged process. First, the different offers received in heterogeneous data formats have to be transformed in a standardized representation format. Second, before final decision making, the customer requirements and the offered transportation services have to be matched because not all transport offers equally comply with the original request.

Integration of external logistics services is primarily characterized by manual effort and lots of human interaction. Therefore, errors occur increasing costs and leading to inefficient supply chains. The absence of formal semantics and service-oriented computing solutions not only prevents enhanced data processing, but also intelligent and automatic service integration.

### 2.2 Target Situation

Filling the afore-mentioned gap and overcoming the challenges necessitates an approach, which applies semantic techniques and adapts them to logistics systems. The deployment of a semantic service application framework for the logistics domain provides an explicit and formal representation of all information enclosed in logistics processes. The framework also enables a semantic and detailed description of logistics services, which are easily accessible, loosely-coupled, and open to combination or composition into more complex logistics processes, such as supply chain configuration. This semantic approach leads to automated and more intelligent external process integration.

The application scope of semantic service descriptions is significantly extended by the techniques that we may apply on top of these annotated services, such as automatic discovery, ranking and composition of all available transportation services. The formally and semantically described services, being published from a wide range of logistics providers, may be searched by a particular 4PL based on

some desired criteria. Automated discovery and ranking of the services that optimally match these criteria is performed and final solutions are proposed to the 4PL.

The proposed solution enables the 4PL to configure the logistics system more efficiently. Errors caused by manual data integration efforts and unambiguous information are avoided. Decision-making is less concerned by handling structural complexity and dynamic behaviour of supply chains.

## 3 MODELING THE LOGISTICS ONTOLOGY

In this section, we propose an ontology that offers a unified representation of the logistics domain and facilitates the formal, semantic description of logistics services. Studer et al. [15] define an ontology as "a formal, explicit specification of a shared conceptualization." Based on this definition, a logistics ontology is an abstract and formal model of consensual knowledge about the logistics domain. Using an ontology, we are able to formally describe the semantics of terms representing an area of knowledge and give explicit meaning to the information. This enables automated reasoning, information integration and application of intelligent approaches such as semantic service description, decision support, semantic search, knowledge management and the like.

### 3.1 Ontology Domain and Scope

To model the logistics ontology, we have followed the ontology engineering method proposed by Noy et al. [9]. We have thus defined the scope of the ontology in terms of competency questions (CQ). We describe below a set of such questions, which our target ontology should be able to answer.

**CQ 1.** Which actors are involved in providing logistics services?

**CQ 2.** Which logistics services are required to realize the flow of goods?

**CQ 3.** What aspects are critical to selecting a logistics service?

**CQ 4.** Which logistics service providers are capable of providing a requested logistics service?

**CQ 5.** Which information is required to provide adequate logistics services?

**CQ 6.** What characteristics constitute a transportation service?

**CQ 7.** What kind of resources are used by logistics services?

**CQ 8.** What are relevant metrics to measure service performance?

In the following section, we present the logistics ontology describing in detail its most relevant entities.

### 3.2 Ontology of the Logistics Domain

The objective of the logistics ontology is to capture the essence of the logistics domain. Typically, an ontology contains the following constructs: concepts, relations, axioms, individuals and assertions. To encode the ontology we use OWL DL, a Web-based ontology language based on Description Logics (DL) that provides high expressiveness while maintaining favorable computational properties for reasoning [6].<sup>4</sup>

The ontology is modeled using Protégé,<sup>5</sup> a platform that offers support for ontology creation, visualization and manipulation. Specifically, we used the Protégé-OWL Plug-in to model the classes

<sup>4</sup> <http://www.w3.org/TR/owl-guide/>

<sup>5</sup> <http://protege.stanford.edu/>

and their relations. In the following, we explain the entities that compose the modeled logistics ontology.

**Classes.** The top-level basic classes (concepts) are *Process*, *Service*, *Resource*, *Service Level Parameter*, *Actor*. The concepts *Process* and *Service* are further specialized as *LogisticsProcess* and *LogisticsService* respectively. A logistics process may be atomic, composed of only one logistics service, or it may be a composite process, containing a series of services that together form a workflow.

**Properties.** In OWL DL, relations between classes are defined by the *properties* constructs, distinguishing between two main categories: *Object properties* and *Datatype properties*. The *Object properties* link individuals to individuals, whereas the *Datatype properties* link individuals to data values. In our context, an object property *usesResources* links the individuals of the class *Service* to those of the class *Resource*.

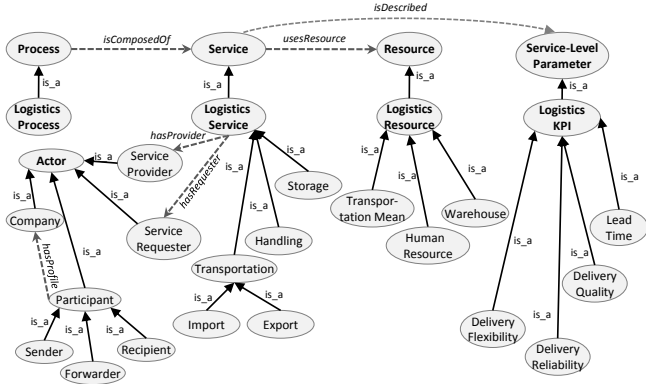


Figure 1. Top-level Concepts of the Logistics Ontology

A snapshot of the constructed ontology is illustrated in Figure 1. This part focuses on modeling the logistics services of a supply chain, which may often be linked together to form a process with resources, actors and performance metrics.

The concept *Logistics Process* is a specialization of the concept *Process*. We allow for composition and decomposition of logistics services regarding various levels of abstraction. Therefore, linked to the concept *Process* via the object property *isComposedOf* is the class *Service*, which is further specialized with the subclass *Logistics Service*. *Logistics Service* acts as an umbrella to model more specific services like transportation, handling and storage.

To provide complete service descriptions for each logistics service, a profile is specified comprising functional and non-functional characteristics. We adopted the upper level concepts of *Resource* and *Service Level Parameter* from the Web Service domain as basis for modeling logistics resource and logistics KPI (Key Performance Indicator) in our service-oriented approach. A logistics service uses *Logistics Resource*, a concept further specialized into more specific concepts: transportation mean, warehouse and human resource. To model non-functional properties, we derived the subclass *Logistics KPI*, specialized, among others, into subconcepts *Delivery Flexibility*, *Delivery Reliability*, etc.

To fully capture the information of the domain, we modeled many more concepts, varying from the different types of carriers, transportation, goods, logistics documents and logistics standards. Another important part is the specification of actors involved in the various logistics processes.

ious logistics processes.

In our ontology, an *Actor* may be a service provider, a service requester or a participant in the logistics chain, who may play the role of recipient, sender or forwarder. Moreover, every actor has a particular company profile, ranging from logistics company to business enterprise, manufacturing company or a virtual organization. This entire information is captured and modeled in our domain ontology, another part of which is further shown in Figure 2.

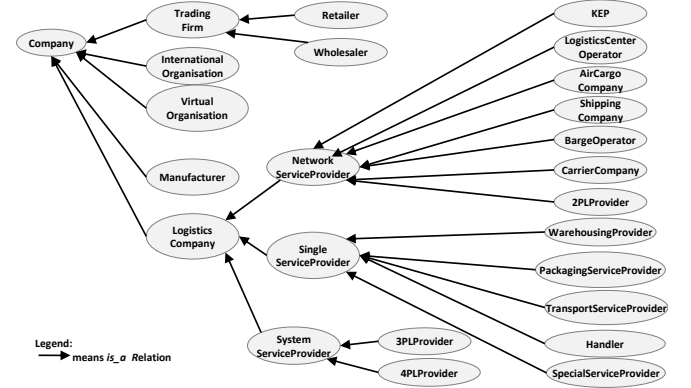


Figure 2. Modeling Logistics Actors

**Axioms.** OWL is provided with formal model-theoretic semantics corresponding to Description Logics. Beyond basic class subsumption relationships, the interpretation of the classes and relations of the logistics ontology can be constrained with additional DL axioms, examples of which we present below.

**Axiom Set 1.** One of the subclasses of the class *Actor* is the *LogisticsCompany*. A logistics company may be either a single-service, network-service or system-service provider, which are mutually disjoint concepts. A 2PL is a subclass of a network-service provider. A system-service provider is classified as a 3PL or 4PL company.

$$\text{LogisticCompany} \sqsubseteq \text{SingleServiceProvider} \sqcup$$

$$\text{SystemServiceProvider} \sqcup \text{NetworkServiceProvider}$$

$$\text{SingleServiceProvider} \sqcap \text{SystemServiceProvider} \sqcap$$

$$\text{NetworkServiceProvider} \sqsubseteq \perp$$

$$2\text{PLProvider} \sqsubseteq \text{NetworkServiceProvider}$$

$$\text{SystemServiceProvider} \sqsubseteq 3\text{PLProvider} \sqcup 4\text{PLProvider}$$

**Axiom Set 2.** A logistics process is composed of one or more services, representing respectively either an atomic or a composite process. A service has a provider and a requester, each of them with a particular company profile. Moreover, a service uses resources, which are classified as person, transportation mean or warehouse.

$$\text{LogisticsProcess} \sqsubseteq \text{Process} \sqcap (\text{AtomicProcess} \sqcup$$

$$\text{CompositeProcess}) \sqcap \forall \text{isComposedOf. LogisticsService}$$

$$\text{AtomicProcess} \sqcap \text{CompositeProcess} \sqsubseteq \perp$$

$$\text{LogisticsService} \sqsubseteq \text{Service} \sqcap \forall \text{isFollowedBy. LogisticsService}$$

$\square \exists \text{hasProvider.ServiceProvider} \square \exists \text{hasRequester.ServiceRequester}$   
 $\square \forall \text{isDescribed.LogisticsKPI} \square \forall \text{usesResource.Resource}$

**Axiom Set 3.** The performance of a service is measured by various metrics, such as service level, confirmation time, delivery time, order processing time, punctuality, reachability, etc.

*Performance*  $\sqsubseteq \forall \text{isMeasuredBy} . (\text{ServiceLevel} \sqcup \text{Punctuality} \sqcup \text{Reachability})$

**Individuals and Assertion.** The classes of an ontology are instantiated, specifying the concrete objects or individuals that belong to a particular class. For example, we instantiate the class *Service Provider* with the individual *Cargoliner GmbH*. Specific properties between individuals are instantiated in assertions.

**Reasoning.** We may automatically perform reasoning on the developed ontology, such as consistency checking, subsumption testing and instance classification. The ontology is presented in an OWL document, which is available online.<sup>6</sup>

The ontology is used as basis for the semantic representation of the logistics services. Some of its core concepts are adopted from the service modeling approach, which we use for the semantic service descriptions.

## 4 SEMANTIC REPRESENTATION OF LOGISTICS SERVICES

In this section, we introduce a framework for representing and implementing the different functionalities conducted in the logistics domain using Web Services standards and Semantic Web representations. Web Services, on one side, provide standard means for the interoperation among the different software applications of various platforms. They encapsulate software functionalities that are distributed, but programmatically accessible over internet protocols [11]. A service-oriented framework allows not only flexible interaction, but also greater reusability of the deployed functions.

On the other side, the Semantic Web markup languages used for the representation of these services provide the means for their computer-interpretable description and easier access to these services. Web Service standards yield a basic language for describing service functionalities, the Web Service Description Language (WSDL). It only provides syntactic elements with no formal semantics, i.e. no means for formalized reasoning about the description statements.

The approach we use for the semantic description of logistics services is OWL for Services (OWL-S), an ontology that enables the description of what the services provide and how they can be used. This representation supports the automated discovery, execution, monitoring, as well as composition and interoperability of the services.

### 4.1 Semantic Description of Logistic Services

We provide a semantic representation of logistics services using the three main clusters of constructs of OWL-S 1.1,<sup>7</sup> which are *Service Profile*, *Service Model* and *Service Grounding*. Respectively, they describe the capabilities that the service provides, the ways the service works internally, as well as details on how the service can be accessed and executed via linking to the appropriate WSDL file.

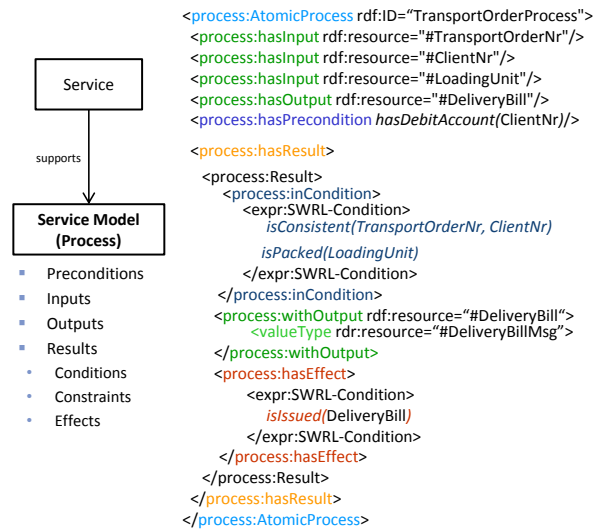


Figure 3. OWL-S Service Model of a Transportation Service

Service Profile describes the functionality in terms of the basic sets of inputs, outputs, preconditions and effects (IOPEs) of the service for automated discovery. Service Model provides the details on its internal processes, describing how the service works. Hereby, the service is viewed as a process, which may be classified as atomic or composite. A composite process is a combination of atomic processes with a particular data flow and control flow defined using constructs like sequence, choice, iteration, etc. The conditions and effects are specified as logic statements, necessary for inference engines, utilizing Semantic Web Rule Language (SWRL).<sup>8</sup>

In Figure 3 we illustrate the semantic description of an atomic process representing a transport service. It is to be always considered that all the resources defined in the service descriptions are specified as concepts in the logistics ontology of Section 3. The process takes as input an identifying *transport order number*, *client number* and *loading unit* to be transported. The output is a *delivery bill*. If the process has a precondition, then it cannot be performed successfully unless the precondition is true. In our case, we have specified that the client should have an existing debit account.

The *inCondition* property specifies the condition under which the result occurs. We have specified that there should be a match between the input *ClientNr* and the *TransportOrderNr*. Effects determine changes in the state of the world upon service execution, e.g. a *DeliveryBill* is really issued after process invocation.

### 4.2 Semantic Services Application Framework

The formal and semantic description of logistics services supports the application of more intelligent and automated functionalities, which we present in a framework shown in Figure 4.

Initially, the services should be made available via the different *Service Providers*, who use the appropriate tools for semantic *Annotation* and *Publishing* of their services. An implementation of the service, represented in WSDL description, should exist in their platforms to enable future invocation. The semantically annotated services, precisely the generated service descriptions, are saved in a *Repository* together with the ontologies that these descriptions use.

<sup>6</sup> <http://www.interloggrid.org/dl34>

<sup>7</sup> <http://www.w3.org/Submission/OWL-S/>

<sup>8</sup> <http://www.w3.org/Submission/SWRL/>

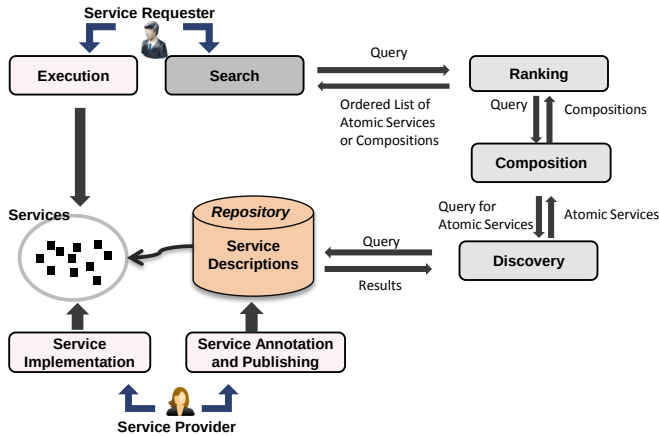


Figure 4. Semantic Services Application Framework

A *Service Requester* may search for services via a Web-based search interface, specifying the desired preconditions, inputs, outputs, and effects. The user's query is processed, using complex reasoning techniques, to discover in the repository those descriptions, whose IOPEs optimally match those defined in the user's query. The returned result may be a list of atomic services or even compositions (composite processes) of services. The list is ranked and displayed to the user, who is then able to execute, according to the implementation made available by the provider, the service that best matches his needs.

The power of semantics in logistics services lies in the automation of functionalities like discovery, composition, monitoring, or execution as shown in this framework. It allows the logistics clients to intelligently discover and invoke the processes that optimally match their needs from a wide range of services made available by the autonomous logistics providers.

## 5 APPLICATIONS OF THE LOGISTICS SEMANTIC APPROACH

In the following, we sketch approaches for the practical application of our framework. We first illustrate the benefits of using our ontology and semantic service description framework for the logistics domain conceptually in the following evaluations scenarios, then move on to potential ontology queries.

### 5.1 Evaluation of Logistics Models

**Use Case 1 - Check for consistency and conflicts.** We check whether concepts, relations and properties between the elements of two or more actors are specified in accordance to each-other. For example, seamless integration of a transport service, provided to an original equipment manufacturer, requires both lifting data models and mapping them to the proposed logistics ontology. Thus, we are able to identify conflicts and ensure consistency.

**Use Case 2 - Check for consistency across process models.** Information relevant to logistics services is captured across various actors and autonomous models, leading to inconsistencies. Our framework not only provides a vocabulary of consensual logistics knowledge, but also serves for detecting inconsistencies from incorrect

model usage, e.g. the ontology assures that logistics services are linked correctly to the required resources, adequate metrics and actors roles.

**Use Case 3 - Behavioral checks on process models.** The provided representation of logistics processes, specifically using the control flow constructs of OWL-S, allows us to perform checks on their correct behavior, regarding the execution order of the activities. Behavioral monitoring of the process model checks e.g. activity *Issue Transportation Order* is performed before *Issue Delivery Bill*.

### 5.2 Querying the Logistics Ontology

We perform queries and gather results from distributed and autonomous systems, whose data may be highly heterogeneous. The logistics ontology provides a unique view of diverse data, allowing us to pose queries upon this model and retrieve integrated results.

In order to illustrate the benefits provided by such an integrated view, we list potential example queries that could be answered using our ontology framework. We formulated queries in SPARQL<sup>9</sup> and, due to space limitations, present the code for the third query only. Originally designed as a query language for graph patterns in Resource Description Format (RDF), SPARQL is practically also used to encode queries against OWL knowledge bases, interpreting the basic graph-matching capabilities using the semantics of the ontology language.

**Query 1.** Show the contact details and description of transportation services offered by providers satisfying delivery quality above 99%.

**Query 2.** List the logistics providers that offer storage and handling services.

**Query 3.** Show the name, description and website of the logistics companies that offer a transport service with delivery size *Full-Truck-Loaded*.

```
PREFIX pr: <http://www.example.com/Logistics.owl#>
SELECT ?name ?description ?website
WHERE {
  &?subject pr:hasWebsite ?website .
  &?subject pr:hasDescription ?description .
  &?subject pr:hasName ?name .
  &?subject pr:hasTransportService ?object .
  &?subject pr:hasDeliverySize pr:Full-Truck-Loaded }
```

We have formulated and posed these, as well as similar queries, in the context of the initial case study analysis.

## 6 RELATED WORK

Related work of logistics ontologies can be grouped into three parts: (1) logistics ontologies freely available on the web, (2) logistics ontologies developed by companies, and (3) logistics ontologies presented in scientific publications.

On the Web, dedicated ontology search engines like Swoogle,<sup>10</sup> DAML Ontology Library<sup>11</sup> and SchemaWeb<sup>12</sup> reveal that the number of available logistics ontologies is very small. They can either be assigned to the domain of manufacturing or exclusively describe special logistics terms, such as e.g. capacity, aircraft types or hazardous cargo. All these ontologies only provide a taxonomy, lacking formal axioms or relations and can, therefore, be regarded as lightweight-ontologies. In companies, there is a widespread usage of information models. These models (e.g. SAP information models) are not regarded as ontologies because they are constructed by means of data modeling techniques and do not provide formal semantics.

<sup>9</sup> <http://www.w3.org/TR/rdf-sparql-query/>

<sup>10</sup> <http://swoogle.umbc.edu/>

<sup>11</sup> <http://www.daml.org/ontologies/>

<sup>12</sup> <http://www.schemaweb.info/>

Considering scientific publications, Haugen and McCarthy [5] describe an extension of the REA Ontology concerning internal accounting to support logistics. The work of Wendt et al. [17] focuses on how to derive, by merging to domain specific ontologies, common logistics concepts for scheduling. Pawlaszczyk et al. [12] considered the Enterprise Ontology [16] as a foundation to describe the role of logistics ontologies in mass customization. The works in this group provide logistics ontologies, which only represent some basic concepts compared to real world complexity of logistics.

Another group aims at applying ontology languages (“ontologizing”) to existing logistics models. For supply chain simulation, Fayez et al. [3] propose an OWL representation of the SCOR model without providing details on its implementation. Leukel and Kirn [8] develop a logistics ontology based on the SCOR model for describing activities in logistics, their properties and relations. Haller et al. [4] propose a methodology to ontologize the *RosettaNet* specification, in order to resolve heterogeneities in dynamic supply chain settings. While these works provide rich logistics ontologies, they either focus on enhanced communication or on modeling logistics elements and interrelations. Additionally, Ye et al. [18] propose an architecture for web-based integration of supply chains, adopting the Enterprise Ontology. However, none of these logistics ontologies has a grounding in top-level ontologies originating from service-oriented computing.

A greater deficiency can be observed in the area of semantic representation of logistics services. One relevant approach is presented by Cuadrado et al. [2], which applies semantic Web Services for integration in logistics. This work mostly keeps a case-study perspective focusing on freight logistics, defining aligned requirements for service description.

## 7 CONCLUSIONS

In this paper, we have presented an approach for the semantic representation of the logistics domain, which offers solutions to the integration challenges among heterogeneous data and interoperability of logistics services from different providers. We have developed an ontology model using OWL DL for the formal, unified representation of logistics data and a service-oriented approach for the semantic description of logistics processes using OWL-S.

Introducing semantics in the logistics domain and combining them with Web Service standards support automated discovery and composition of services from heterogeneous and external logistics providers. It helps both dynamic and optimal configuration of supply chains and facilitates logistics decision-making. We are currently extending the logistics ontology with additional relations and axioms, to enable more complex reasoning and inferencing, making it fully applicable to a broader range of logistics scenarios. We also investigate the mapping to existing modular sub-ontologies.

We will extend the repository of service descriptions with more real-world logistics services. Based on this, we will start to implement and integrate service discovery and composition techniques as part of an intelligent logistics services portal. Extending the repository of logistics service descriptions will allow us to test the alignment of services into more complex processes and evaluate their interoperability.

## ACKNOWLEDGEMENTS

The work presented in this paper was partly funded by the German Federal Ministry of Education and Research under the project InterLogGrid (BMBF 01IG0910E/G).

## REFERENCES

- [1] N.P. Archer, ‘Supply Chains and the Enterprise’, *Journal of Enterprise Information Management*, **19** (3), 241–245, (2006).
- [2] J.E. Cuadrado, Ch. Preist, and S. Williams, *Integration of B2B Logistics Using Semantic Web Services*, volume 3192 of *Artificial Intelligence: Methodology, Systems, and Applications*, 96–105, Springer Berlin / Heidelberg, 2004.
- [3] F. Fayez, L. Rabelo, and M. Mollaghasemi, ‘Ontologies for supply chain simulation modeling’, in *Winter Simulation Conference*, Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A., (pp. 2364–2370, 2005).
- [4] A. Haller, J. Gontarczyk, and P. Kotinurmi, ‘Towards a Complete SCM Ontology: the Case of Ontologising RosettaNet’, in *Proceedings of the 2008 ACM symposium on Applied computing*, (2008).
- [5] R. Haugen and W.E. McCarthy, ‘REA, a Semantic model for Internet Supply Chain Collaboration’, *OOPSALA 2000 Business Objects and Component Design and Implementation Workshop VI: Enterprise Application Integration*, (2000).
- [6] I. Horrocks, P.F. Patel-Schneider, and F.V. Harmelen, ‘From SHIQ and RDF to OWL: The Making of a Web Ontology Language’, *Journal of Web Semantics*, **1**, 7–26, (2003).
- [7] M.N. Huhns and M.P. Singh, ‘Service-oriented computing: key concepts and principles’, *IEEE Internet Computing*, **9**(1), 75–81, (Jan-Feb 2005).
- [8] J. Leukel and S. Kirn, *A Supply Chain Management Approach to Logistics Ontologies in Information Systems*, volume 7 of *Lecture Notes in Business Information Processing*, 95–105, Springer Berlin Heidelberg, 2008.
- [9] N.F. Noy and D.L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology*. Online, 2001.
- [10] Mike Papazoglou and Willem-Jan van den Heuvel, ‘Service oriented architectures: approaches, technologies and research issue’, *The VLDB Journal*, **16**(3), 389–415, (July 2007).
- [11] M.P. Papazoglou, *Web Services: Principles and Technology*, Prentice Hall, 2007.
- [12] D. Pawlaszczyk, A.J. Dietrich, I.J. Timm, S. Otto, and S. Kirn, ‘Ontologies Supporting Cooperation in Mass Customization - A Pragmatic Approach’, *MCP-CE 04*, (2004).
- [13] M. Perona and G. Miragliotta, ‘Complexity management and supply chain performance assessment. A field study and a conceptual framework’, *International Journal of Production Economics*, **90** (1), 103–115, (2004).
- [14] N. Singh, ‘Emerging Technologies to Support Supply Chain Management’, *Communications of the ACM*, **46** (9), 243–247, (2003).
- [15] Rudi Studer, V. Richard Benjamins, and Dieter Fensel, ‘Knowledge engineering: Principles and methods’, *Data and Knowledge Engineering*, **25**(1-2), 161–197, (March 1998).
- [16] M. Uschold, M. King, S. Moralee, and Y. Zorgios, ‘The Enterprise Ontology’, *Knowledge Engineering Review*, **13**, 32–98, (1998).
- [17] O. Wendt, T. Stockheim, S. Grolik, and M. Schwind, ‘Distributed Ontology Management Prospects and Pitfalls on Our Way Towards a Web of Ontologies’, *Dagstuhl Workshop*, (2002).
- [18] Y. Ye, D. Yang, Z. Jiang, and L. Tong, ‘An Ontology-based Architecture for Implementing Semantic Integration of Supply Chain Management’, *International Journal of Computer Integrated Manufacturing*, **21**, 1–18, (2008).

## AUTHOR INDEX

|                                 |    |                              |           |
|---------------------------------|----|------------------------------|-----------|
| Adam, Emmanuel .....            | 31 | Marrenbach, Dirk .....       | 55        |
| Bloehdorn, Stephan .....        | 73 | Morales Kluge, Ernesto ..... | 37        |
| Cohn, Anthony G. ....           | 3  | Neuhäuser, Daniel .....      | 55        |
| Eich, Markus .....              | 61 | Nimis, Jens .....            | 13        |
| Estivie, Sylvia .....           | 31 | Ottens, Brammert .....       | 25        |
| Faltings, Boi .....             | 25 | Overmeyer, Ludger .....      | 43        |
| Frommberger, Lutz .....         | 1  | Porzel, Robert .....         | 67        |
| Ganji, Farideh .....            | 37 | Pulter, Natalja .....        | 13        |
| Harjes, Florian .....           | 49 | Rippel, Daniel .....         | 49        |
| Heiserich, Gerd .....           | 43 | Schill, Kerstin .....        | 1         |
| Hette, Gaël .....               | 31 | Scheuermann, Andreas .....   | 73        |
| Hoxha, Julia .....              | 73 | Scholz-Reiter, Bernd .....   | 1, 37, 49 |
| Karaenke, Paul .....            | 7  | Schuele, Michael .....       | 7         |
| Kirchner, Frank .....           | 61 | Sichman, Jaime S. ....       | 19        |
| Léauté, Thomas .....            | 25 | Sommer, Tobias .....         | 55        |
| Lockemann, Peter C. ....        | 13 | Warden, Tobias .....         | 67        |
| Mandiau, René .....             | 31 | Wehking, Karl-Heinz .....    | 55        |
| Marcellino, Fernando J. M. .... | 19 |                              |           |