Qualitative spatial reasoning with topological information in BDI agents

Michael Schuele[1] and **Paul Karaenke**[1]

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.

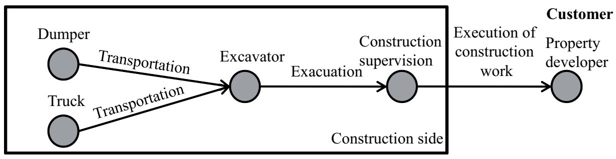

Figure 1. Service chain model example of a construction side

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,

[1] 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

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

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 is it *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^{-1}	PO	EQ	NTPP	$NTPP^{-1}$
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.

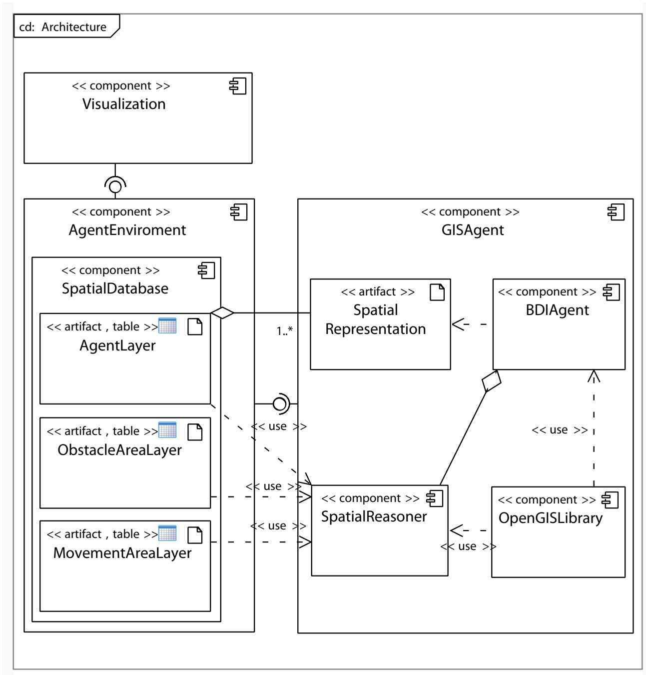

Figure 2. UML component diagram of system architecture

4 EVALUATION

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

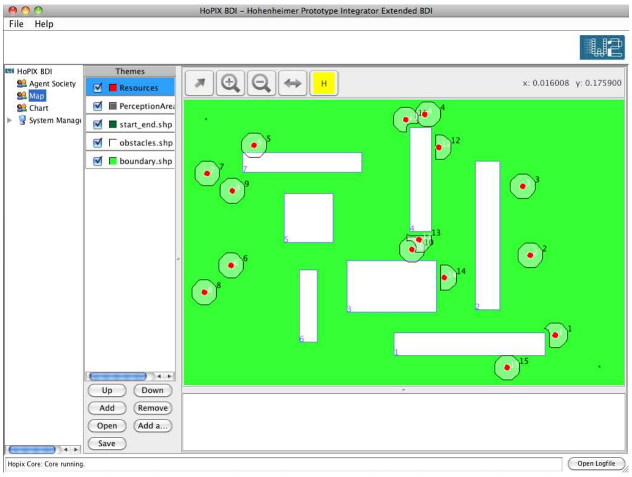

Figure 3. Screenshot of spatial visualization

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.

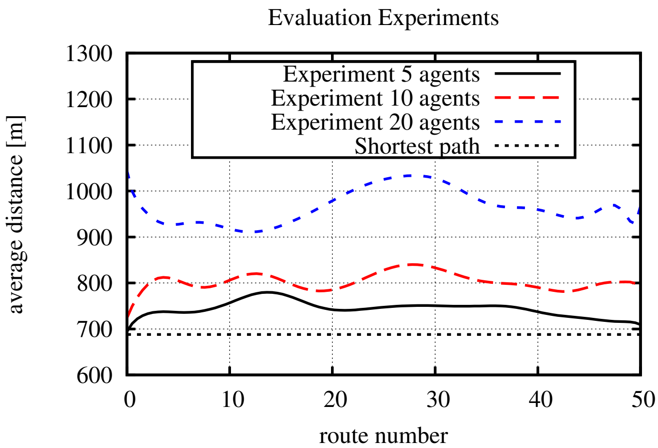

Figure 4. Analyze of mean passed distances of agents in evaluation experiments

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 populated 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 routeing', *International Journal of Production Research*, **29**(12), 2377–2391, (1991).

[10] S. Kirn 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.refractions.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. Schuldt 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. Wölfl, 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).