

Distributed Clustering of Autonomous Shipping Containers by Concept, Location, and Time

Arne Schuldt and Sven Werner

Centre for Computing Technologies (TZI)
University of Bremen, Am Fallturm 1, D-28359 Bremen
{as,sw}@tzi.de

Abstract. Recent developments in logistics show an increasing trend towards autonomous control. Intelligent software agents, that represent logistic objects like shipping containers, plan and schedule their way through a logistic network. This paper addresses the aspect of cooperation. A special focus lies on the second step of the model for cooperation, namely team formation. The question is by which criteria shipping containers, or logistic objects and agents in general, can form clusters. Starting from the particular demands of the logistics domain this paper argues in favour of conceptual, spatial, and temporal properties. A framework that takes concept, location, and time into account is introduced and demonstrated by an example application.

1 Introduction

Logistics plays an important role in a globalised economy. Trade between different parts of the world demands reliable transport networks. The challenge when dealing with logistic processes is to cope with the complexity and the dynamics that are inherent in these processes. Thus, recent developments show an increasing trend towards autonomous logistics. From the artificial intelligence point of view, such processes can be modelled by software agents which act on behalf of the represented logistic entities in a multiagent system [15].

Today, shipping containers form one of the most important logistic objects as they handle most of the intercontinental transport of packaged goods [7]. In the history of logistics, the concept of containerisation as a medium for intermodal transport is a relatively new one. It has been established since 1956 when Malcom Purcell McLean (1913 – 2001) started to employ containers on a large scale [10]. Containers revolutionised the area of transport logistics as they allow the transfer of goods between different means of transport without the necessity for repacking. In particular, this massively decreases the time required for loading goods onto ships. An important property of shipping containers is that they exhibit standardised sizes which makes them highly interchangeable. Hence, they provide a high degree of abstraction from the concrete goods carried.

Represented by software agents, containers are capable of autonomously planning the transport from their source to a sink. In some situations it might be

useful or even necessary to cooperate with other containers in order to meet this objective. A common approach addressing this purpose is the model for cooperation [16]. The question remains what are the criteria by which shipping containers can be grouped together. This paper argues in favour of conceptual, spatial, and temporal properties. The same properties have, for instance, also been applied for information retrieval from the semantic web (e. g. [8, 14]).

The remainder of this paper is structured as follows: Section 2 discusses the addressed scenario and Section 3 presents related work. The identified clustering criteria are introduced in Section 4. A compatible clustering protocol is presented in Section 5 and its implementation is dealt with in Section 6. As a case study, it is subsequently applied to an example problem in Section 7. Eventually, a conclusion follows in Section 8.

2 Scenario

This paper addresses the area of forward logistics which deals with procuring products from all over the world. These goods are packed into shipping containers in order to be loaded onto container vessels. The particular focus of this paper lies in an onward carriage scenario, that begins as soon as the containers arrive at their scheduled port of discharge (Fig. 1). The task is then to find an appropriate warehouse in which the contained goods can be received. This is challenging due to the underlying complexity and dynamics. The complexity is caused by the high throughput of containers as well as the parameters influencing the decision, e. g., date of arrival, port of discharge, properties of the goods, and capacities, just to name a few. Containers are sometimes delayed or even lost. This prevents static planning and leads to highly dynamic processes. Approaching this challenge with centralised methods (such as [6]) is, therefore, only efficient to a minor degree.

As discussed in Section 1 the idea is therefore to apply an autonomous approach in which the planning and scheduling is carried out locally by the containers themselves. In the process of solving this task the containers' content plays an important role. With regard to the subsequent distribution it is desirable for a company to concentrate similar goods as early as possible. As an example, multiple containers with T-shirts that are scheduled to be sold together should be jointly received in the same warehouse. Proceeding this way allows similar products to be delivered together by truck to the distribution centres. Otherwise, the goods would have to be shipped from multiple warehouses which increases the probability of empty space on the trucks. While this is an argument for the coordination of multiple containers of the same company, there exist also reasons for preventing containers from being clustered. Consider, for instance, fresh fruits. Due to their perishability, they can only be received in a refrigerated warehouse. A second example is that of jewellery which can only be stored in secured warehouses. Furthermore, damaged goods cannot be received in any conventional warehouse; they have to be delivered to a selector who repairs them. These examples illustrate that some goods cannot be received arbitrarily

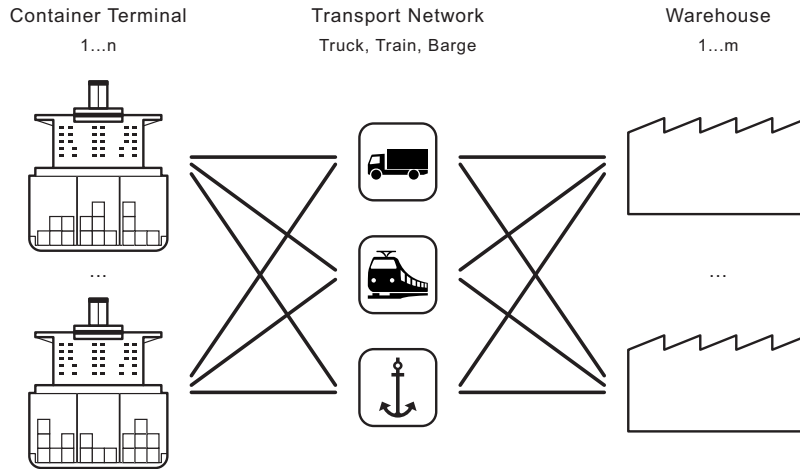


Fig. 1. Containers arriving at a container terminal have to find an appropriate warehouse. Thereby, they also have to choose an available transport network

together with others in the same warehouse. The above considerations on the content of containers shall be summarised as their *concept*.

It is clear that the content of containers alone is not sufficient to decide whether they should be clustered. Furthermore, the *location* of containers is relevant. As an example, transporting containers by train is much cheaper than loading them onto trucks. Nevertheless, this is only the case if a certain number of containers is transported together on the same train. Thus, a shipping container has to find partners that share a common location and destination. This directly leads to the third criterion, namely *time*. The containers should additionally be scheduled for a similar date of delivery. Otherwise, some of them are forced to wait for the others; possibly longer than their own scheduled date of delivery.

The above applications indicate the necessity for the clustering of similar shipping containers belonging to the same company. The clustering criteria that can be derived from these examples are concept, location, and time.

3 Related Work

As discussed above, the task is to establish meaningful groups of shipping containers. Clustering algorithms partition a given set of objects into distinct groups. The objective is to achieve a maximal distance between different clusters and a minimal distance between the members of each cluster. Previous algorithms such as k-means [12] generally have a centralised perspective on the data to be clustered. Hence, they cannot be applied to the distributed problem at hand.

Distributed clustering approaches can be found in the area of wireless sensor networks [1]. Like the logistics domain, the application of sensor networks is highly distributed. In general, sensors are spread within a certain area in order

to observe specific phenomena. Their primary tasks are acquiring data from the environment as well as transmitting the collected data to a base station. Since the individual sensors act independently of a central power supply, saving energy resources is a crucial task. Therefore, clustering algorithms for sensor networks aim at optimising energy consumption and communication costs. A common approach is to cluster sensors by spatial proximity. The routing of data messages is then organised in a hierarchical way [2]. Each cluster collects the sensor data of its region, aggregates it, and transmits it to the base station. A prominent approach is LEACH, which stands for low-energy adaptive clustering hierarchy [9]. In this method some sensor nodes choose to be cluster-heads. The remaining nodes join the cluster that requires minimum communication energy (generally the spatially closest cluster-head). In order to equally distribute the power consumption for long-range communication the cluster-heads are regularly changed, which also leads to a new cluster partitioning.

In order to judge an application to logistic object agents, it is necessary to discuss differences between sensor networks and the logistics domain. While it seems suitable for sensor networks to form clusters just by spatial proximity and residual energy, this is not the case for shipping containers. By contrast, as described in Section 2, further criteria apart from location have to be taken into account. Even w. r. t. location, it is not always reasonable to consider only quantitative spatial distances. In contrast, qualitative relationships between shipping containers (e. g., same destination) are of particular interest. Moreover, the continuous re-clustering in order to distribute power consumption conflicts with the objective of long-term cooperation. Finally, in a sensor network, generally all nodes exist right from the start. In the logistics domain, such a predetermined date does not exist as shipping containers might also join the network later on.

4 Clustering Criteria

The approach presented in this paper is based on the so-called model for cooperation [16]. This model defines a formalisation for cooperative problem solving (CPS) approaches. The CPS process is divided into four steps: recognition of potential for cooperation, team formation, plan formation, and team action. The particular focus of this paper is on forming a distinct team that is capable of realising a certain goal, i. e., the second step. The detection of a potential for cooperation is considered to be agreed upon in the first step in which an agent concludes that some of its goals could be reached by a group of agents. Furthermore, the third and fourth step are carried out after applying the presented method and are also out of the scope of this paper.

In order to address the task of team formation, it is necessary to compare the properties of the respective agents. As described in Section 2 the presented approach takes concept, location, and time constraints of agent properties into account. These properties are considered relevant to agents that have to deal with transport problems and that may take advantage of forming clusters. After applying the proposed team formation step in terms of the model for cooperation,

the formed cluster of agents is supposed to be adequate for realising the desired goal. However, it is also possible that this phase may not succeed, since failure is also a possible outcome of the model for cooperation.

4.1 Conceptual and Spatial Constraints

The knowledge concerning the content of shipping containers is represented by an ontological approach based on description logics [4]. Description logic is a decidable fragment of first-order logic. The modelling consists of concepts representing sets of objects and roles as relationships between these objects. Figure 2 shows an example ontology that consists of three major parts: article, property, and location. The first part arranges the transported goods by their type into a taxonomical hierarchy. The taxonomy allows the recognition of more general classes of goods; for instance, it can be concluded that T-shirts are textiles. A second part of the ontology comprises properties that goods may have. By introducing further roles it is possible to model that, say, fruits are perishable or jewellery is valuable. In contrast to the type of goods, which generally never changes, the properties may change (although rather seldom). As an example, goods might be damaged during transport. Furthermore, the location of goods can be integrated as an ontological concept, which is the third part. Thus, further restrictions can be realised, e. g., a class comprising all containers with valuable jewellery that are currently at a container terminal. Changes in this part of the description are most likely as goods are transported between different locations.

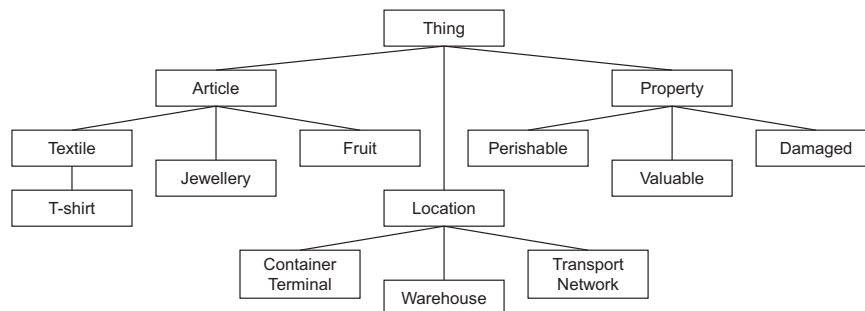


Fig. 2. An example ontology for goods with three major parts: article, property, and location. The lines connecting concepts indicate taxonomical relations

As discussed above, the ontological descriptions of logistic objects change over time. The time-span in which an ontological concept holds can be characterised with the help of the following definition of temporal intervals:

Definition 1 (Temporal Interval) *A temporal interval τ is defined by a start point t_s and an end point t_e , where $t_s < t_e$:*

$$\tau = (t_s, t_e); t_s, t_e \in \mathbb{N}, t_s < t_e$$

Based on this definition, the ontological concept of a container agent during a given time-span can be defined as a subclass of the universal concept \top :

Definition 2 (Agent Concept) *Let α be an agent. The ontological concept of α during τ is represented by*

$$c(\alpha, \tau) \sqsubseteq \top$$

4.2 Agent Clusters

In order to achieve a common goal, container agents are capable of clustering. The lifetime of the whole cluster can thereby exceed the membership of a single agent. Furthermore, each agent can leave a cluster and rejoin later. Therefore, the membership of an agent cannot be characterised by a single temporal interval. By contrast, a set of intervals has to be applied (which might also be empty):

Definition 3 (Cluster Membership) *Let α be an agent, let γ be a cluster. The membership of α in γ is defined by a set of temporal intervals:*

$$m(\alpha, \gamma) = \{\tau_1, \dots, \tau_n\}, n \in \mathbb{N}$$

Before joining a cluster, a container has to determine its similarity to the cluster. This procedure is a special case of the so-called matchmaking problem [13]. It denotes the decision whether advertisements offered by business entities or agents match requests issued by other agents. An approach proposed by [11] addresses this task by considering the formal semantics of ontology-based knowledge representation techniques. They propose five different degrees of match between two concepts c_1 and c_2 : exact $c_1 \equiv c_2$, plug-in $c_1 \sqsupseteq c_2$, subsume $c_1 \sqsubseteq c_2$, intersection $\neg(c_1 \sqcap c_2 \sqsubseteq \perp)$, and disjoint $c_1 \sqcap c_2 \sqsubseteq \perp$.

Like agents, clusters are also characterised by concepts that hold within given temporal intervals. During its membership, the concept of each agent is supposed to be subsumed by the cluster concept:

Definition 4 (Cluster Concept) *Let γ be a cluster. The ontological concept of γ during τ is represented by*

$$\forall \alpha \in \gamma \forall \tau_i \in \text{int}(\tau, m(\alpha, \gamma)) c(\alpha, \tau_i) \sqsubseteq c(\gamma, \tau) \sqsubseteq \top$$

Apart from subsumption, Definition 4 also uses the auxiliary function int , that computes for a given set of temporal intervals their intersection with a reference interval. This ensures that the concept of the cluster members is only compared for those times that are covered by τ :

$$\text{int}(\tau, \{\tau_1, \dots, \tau_n\}) = \{(\tau \cap \tau_i) \mid i \in \{1, \dots, n\} \wedge (\tau \cap \tau_i) \neq \emptyset\} \quad (1)$$

Besides matching containers and clusters, the approach can also be applied in order to find appropriate warehouses. As an example, it can be deduced from the ontology that a cluster of containers carrying damaged T-shirts can be received in a warehouse that is capable of receiving damaged textiles.

4.3 Temporal Constraints

As discussed in the previous section, the concept (including location) of container agents plays an important role during cluster formation. So far, the application of time has been limited to checking whether agents meet a cluster concept during a given temporal interval. Besides, some applications also demand certain temporal relationships between the agents within a cluster. Figure 3 illustrates three shipping container examples. First, if containers plan to be transported by the same truck, one after another, their temporal intervals have to be disconnected (i. e., without overlap). Second, if one container conducts the planning for the whole group, its membership has to subsume the memberships of all other agents. Finally, if containers plan to share a train, their expected dates of arrival at the station should correspond, i. e., the respective intervals must share a common end.

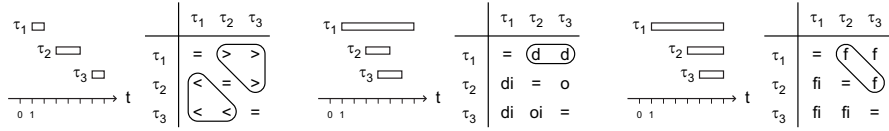


Fig. 3. Three examples for relations between sets of temporal intervals within an agent cluster and the respective temporal matrices

A classical approach for temporal reasoning is formed by Allen’s qualitative relational system [3] (qualitative in the sense that it abstracts quantitative data to a semantically defined representation) which is depicted in Fig. 4:

Definition 5 (Temporal Relation) Let τ_i, τ_j be temporal intervals. The position $\tau_i \tau_j$ of τ_j w. r. t. τ_i is then characterised as

$$\tau_i \tau_j \in \{<, >, m, mi, o, oi, s, si, d, di, f, fi, =\}$$

However, these 13 relations characterise pairs of temporal intervals. Hence, this paper proposes a generalised approach of predicates about arbitrary sets of temporal intervals in order to meet the above requirements.

Relation:	$\tau_i \tau_j$	before	meets	overlaps	starts	during	finishes	equal
		<	m	o	s	d	f	=
Example:	τ_i τ_j							
Inverse:	$\tau_j \tau_i$	>	mi	oi	si	di	fi	=
		after	met by	overlapped by	started by	contains	finished by	equal

Fig. 4. The 13 relations that have been proposed by Allen [3] in order to characterise the qualitative relationship between two temporal intervals

In order to reason about temporal intervals within a cluster, it is first necessary to obtain all these intervals. This is achieved by the following auxiliary function:

$$\text{allIntervals}(\gamma) = \bigcup_{\alpha \in \gamma} m(\alpha, \gamma) \quad (2)$$

Since the above set is not necessarily sorted (accordingly to start and end points), it is transferred into an ordered list by a further auxiliary function:

$$\begin{aligned} \text{orderedIntervals}(\gamma) &= \langle \tau_1, \dots, \tau_n \rangle, \tau_i \in \text{allIntervals}(\gamma), \\ \forall_{\tau_i=(t_{s_i}, t_{e_i})} &\begin{cases} t_{e_i} \leq t_{e_{i+1}} & \text{iff } t_{s_i} = t_{s_{i+1}} \\ t_{s_i} < t_{s_{i+1}} & \text{else} \end{cases} \quad (3) \end{aligned}$$

Based on these auxiliary functions, it is then possible to define the matrix M_γ that comprises the relations for all pairs of temporal intervals of a cluster:

Definition 6 (Temporal Matrix) *Let γ be an agent cluster. The matrix*

$$M_\gamma = \langle \tau_1, \dots, \tau_n \rangle \times \langle \tau_1, \dots, \tau_n \rangle, \tau_i \in \text{orderedIntervals}(\gamma)$$

characterises the relations $\tau_i \tau_j$ between all temporal intervals in γ with $\tau_i \tau_j$ being the entry in row τ_i and column τ_j .

Figure 3 gives these matrices for the depicted configurations. Restrictions like those discussed in the introductory paragraph of this section can be defined as predicates on the temporal matrices. For instance, in the first example the only relations that are allowed to occur in the whole matrix are $\{<, >\}$. Thereby, the matrix' main diagonal is excluded as it relates each interval to itself, which always results in $=$. For the second example, it has to be ensured that there exists one row comprising at most the relations $\{s, d, f, =\}$. In order to determine the third example it is sufficient to examine pairs of subsequent intervals (which is a diagonal of the matrix). The occurring relations are restricted to $\{f, =\}$. The general definition for these three types of predicates is as follows:

Definition 7 (Temporal Matrix Restriction) *Let γ be an agent cluster. Restrictions on its temporal matrix M_γ can be described by a set ρ of temporal relations and one of the following predicates:*

$$\begin{aligned} \text{wholeMatrix}(M_\gamma, \rho) &= \forall_i \forall_j (\tau_i \tau_j \in M_\gamma \wedge i \neq j) \rightarrow \tau_i \tau_j \in \rho \\ \text{matrixRow}(M_\gamma, \rho) &= \exists_i \forall_j (\tau_i \tau_j \in M_\gamma \wedge i \neq j) \rightarrow \tau_i \tau_j \in \rho \\ \text{matrixDiagonal}(M_\gamma, \rho) &= \forall_i \tau_i \tau_{i+1} \in M_\gamma \rightarrow \tau_i \tau_{i+1} \in \rho \end{aligned}$$

5 Clustering Protocol

The clustering criteria presented in Section 4 are based on semantic knowledge and qualitative relations. Although they are not restricted to specific clustering

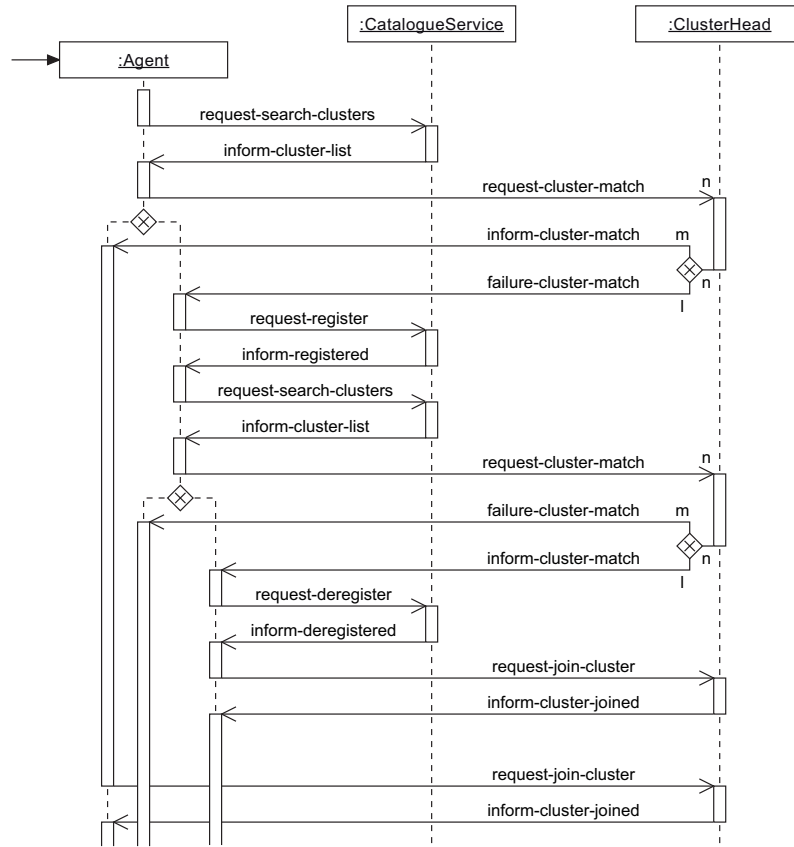


Fig. 5. The protocol flow of the distributed clustering approach for autonomous agents described by an AUML sequence diagram (exceptional messages are omitted for the sake of readability)

algorithms, they cannot be applied with arbitrary methods. As an example, the algorithms discussed in Section 3 are not applicable since they demand quantitative distances. This section presents a clustering protocol (Fig. 5) for autonomous agents that is capable of considering the criteria introduced.

As soon as an agent is created, it queries a catalogue service for existing clusters. Subsequently, it communicates its properties to all cluster-heads. Clusters matching the properties of the agent send positive answers. If the requesting agent receives a positive answer, it joins the respective cluster. Otherwise, the agent chooses to register as a cluster-head itself.

The above method suffers from a potential problem: concurrency. Querying the catalogue and registering oneself as a cluster-head is not an atomic operation. Hence, other agents with the same properties can register in between. In order to address this issue, the agent has to send its properties to all cluster-heads

that have been registered in between as soon as its registration is finished. If the agent finds another cluster-head exhibiting the same properties and an earlier registration time-stamp, it deregisters and joins the earliest cluster found.

The asymptotic communication complexity of the protocol is $O(mn)$. This can be explained by the fact that all n agents contact at most all m clusters, whereby $m \leq n$. For most applications, m is considered to be even much smaller than n , which means $m \ll n$. The main benefit of this protocol, however, is the high degree of autonomy that is left to the individual agent, since the catalogue service does not make any decisions for the agents.

6 Implementation

The protocol is implemented within the Java-based agent-framework JADE [5]. The simulation is synchronised; it is based on time-slots of one day each. The implementation contains three types of agents. The first one is the shipping container creator agent. It uses real-world data from existing containers and defines agents representing the containers accordingly. The creation is conducted at the respective time-slots. The second agent type represents the shipping container itself. Having determined its respective values for concept, location, and time, this agent type proceeds following the protocol presented in Section 5. All communicative actions are conducted according to the FIPA request protocol. They apply the standardised achieve rational effect classes offered by the JADE framework. Due to long response delays of JADE's standard directory facilitator (DF) implementation, a third agent type is defined. This agent administers a list of existing clusters and is called a catalogue service agent: the agent holds a list comprising the agent ids of all cluster-heads. Additionally, it features three basic operations: adding clusters, removing clusters, and questioning the cluster list.

7 Case Study

In order to test the clustering criteria introduced and the performance of the proposed algorithm, a case study is conducted. The goal is to demonstrate that the approach can overcome the limitations of other approaches regarding distribution and semantic knowledge. Furthermore, its applicability to real-world problems has to be examined. As an example application, similar containers located anywhere are expected to form clusters in order to be jointly received in a common warehouse. The purpose is to reduce the subsequent distribution costs as discussed in Section 2. The lifetimes of the agents are thereby expected to overlap pairwise without a gap in between.

The case study is carried out with real-world data from about 2,400 containers that were actually in operation during three months in 2006. The containers join the network distributed over this time-span. A manual data inspection reveals that there exist 215 clusters; so this is the expected outcome of the experiment. Due to the asymptotic complexity, it is expected that the number of conversations and the computation time will develop almost quadratically.

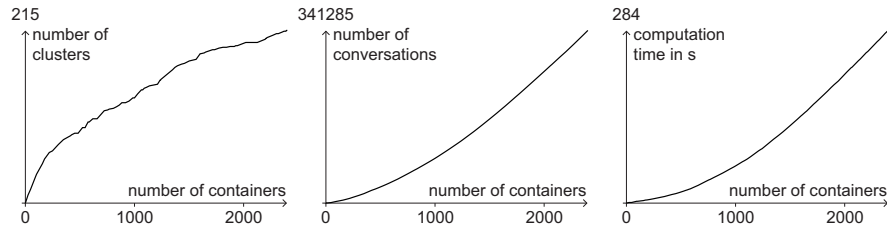


Fig. 6. The number of clusters (left) and conversations (centre) and the time needed for computing the simulation (right) in relation to the number of containers.

The average results of 50 test runs are given in Figure 6. The number of clusters in relation to the total number of agents is depicted on the left hand side. The exact shape of the curve strongly depends on the underlying data. As a result it can, however, be discovered that the final number of clusters is 215 for all test runs, i. e., the algorithm is capable of solving the addressed problem. The total number of conversations (Figure 6 centre) is 341,285. This is below the expected asymptotic complexity. This result can be explained by the fact that not all clusters exist right from the start. Nevertheless, the plotted part of the curve indicates the expected quadratic shape. This observation also corresponds with the computation time that is depicted on the right hand side of Figure 6. On a computer with Windows XP and an Intel Centrino Duo processor with 2.16 Gigahertz, the total time for clustering is 284 seconds.

8 Conclusion

This paper introduces a framework that allows agents representing autonomous shipping containers to form clusters by concept, location, and time. The conceptual, spatial, and temporal properties are thereby represented as semantic knowledge and qualitative relations. The case study conducted demonstrates the applicability to problems in the shipping container domain. The method presented does not make special assumptions about the objects represented. Hence, it is not limited to containers and can also be applied in order to realise clustering of other logistic objects (e. g., autonomous packages). As concept, location, and time are general properties, it is also possible to apply the approach in multiagent systems that model domains not related to logistics.

The clustering algorithm presented exhibits two main features. First, it is capable of handling the above semantic features. Second, it is highly distributed and leaves all decision-making to the autonomous agents. A question for future research is how to further reduce communication efforts. Shipping containers are often located within greater logistic entities, such as container vessels or terminals. Therefore, it seems promising to install them as proxies that cache and bundle communication with the catalogue service and the clusters.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38:393–422, 2002.
2. J. N. Al-Karaki and A. E. Kamal. Routing Techniques in Wireless Sensor Networks: A Survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.
3. J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.
4. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, Cambridge, UK, 2003.
5. F. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE*. John Wiley & Sons, Chichester, UK, 2007.
6. G. B. Dantzig. Application of the Simplex Method to a Transportation Problem. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 359–373. John Wiley & Sons, New York, NY, USA, 1951.
7. H.-O. Günther and K. H. Kim, editors. *Container Terminals and Automated Transport Systems*. Springer-Verlag, Berlin, Germany, 2005.
8. S. Hübner, R. Spittel, U. Visser, and T. J. Vögele. Ontology-Based Search for Interactive Digital Maps. *IEEE Intelligent Systems*, 19(3):80–86, 2004.
9. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *33rd Hawaii International Conference on System Sciences*, volume 8, pages 8020–8029, 2000.
10. M. Levinson. *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger*. Princeton University Press, Princeton, NJ, USA, 2006.
11. L. Li and I. Horrocks. A Software Framework for Matchmaking Based on Semantic Web Technology. *International Journal of Electronic Commerce*, 8(4):39–60, 2004.
12. J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–296, 1967.
13. P. Shvaiko and J. Euzenat. A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics*, 4:146–171, 2005.
14. U. Visser. *Intelligent Information Integration for the Semantic Web*, volume 3159 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Germany, 2004.
15. G. Weiss, editor. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1999.
16. M. Wooldridge and N. R. Jennings. The Cooperative Problem Solving Process. *Journal of Logic & Computation*, 9(4):563–592, 1999.