
Towards Autonomous Logistics: Conceptual, Spatial, and Temporal Criteria for Container Cooperation

Arne Schuldt and Sven Werner

Centre for Computing Technologies (TZI)
University of Bremen, Am Fallturm 1, D-28359 Bremen

Summary. Shipping containers handle most of today's intercontinental transport of packaged goods. Managing them in terms of planning and scheduling is a challenging task due to the complexity and dynamics of the involved processes. Hence, recent developments show an increasing trend towards autonomous control with software agents acting on behalf of the logistic objects. Despite of the high degree of autonomy it is still necessary to cooperate in order to achieve certain goals. This paper argues in favour of conceptual, spatial, and temporal properties on which shipping containers can form groups in order to jointly achieve certain goals. A distributed clustering method based on concept, location, and time is introduced. Subsequently, a case study demonstrates its applicability to a problem in the shipping container domain.

1 Introduction

Shipping containers play an important role in intercontinental transport. They have been established since 1956 when Malcom Purcell McLean started applying intermodal transport on a large scale [8]. The first containers in Germany were unloaded in Bremen in 1966 [11]. Today, shipping containers handle most of the intercontinental transport of packaged goods [7].

Planning and scheduling the route of a container from its source to its sink is a complex task. This is due to the high number of containers in service as well as the task of finding an optimal utilisation of the available transport and storage facilities. Furthermore, containers are sometimes delayed or even lost. This prevents static planning and leads to highly dynamic processes. Centralised approaches, such as [6], exhibit only a limited degree of efficiency in addressing this issue. By contrast, there is an increasing trend towards autonomous logistics. In this paradigm logistic objects are represented by autonomous software agents which act on their behalf in a multiagent system [13]. In terms of shipping containers this massively reduces the complexity since only the parameters of each single container have to be taken into con-

sideration. Furthermore, replanning (e. g., due to delays) can be done locally without rescheduling the whole network.

According to the above discussion it is desirable to achieve a high degree of autonomy in order to cope with the underlying complexity and dynamics. Nevertheless, it still makes sense to cooperate in achieving certain goals [14]. For instance, transporting containers by train is much cheaper than transporting them by truck. This, however, is only true if several containers choose to share the same train. By contrast, it is not profitable to employ a train in order to transport a single container. As another example, it is preferable to receive containers comprising the same content in the same warehouse. This allows distribution centres to be centrally supplied; the probability for empty space on the employed trucks is reduced. Regarding autonomous logistics containers have to coordinate and group themselves accordingly to their content.

Forming clusters is a possibility for shipping containers to succeed in these goals. Previous clustering methods like k-means [10] are not applicable for this task as they conflict the autonomy by their centralised perspective on the data to be clustered. Approaches from the wireless sensor network domain [1, 2] seem to be more promising as they work in a distributed way. However, they are generally limited to quantitative spatial distances. The problems outlined above can therefore not be covered as they require to take semantic knowledge into consideration. In particular, conceptual, spatial, and temporal aspects can be deduced as clustering criteria from the above examples.

The remainder is structured as follows: Section 2 introduces the clustering criteria identified for cooperating containers and gives a formal definition. Subsequently, a case study with an example application is presented in Sect. 3. Finally, Sect. 4 discusses the work presented.

2 Criteria for Cooperation

The abstract model for cooperation [14] divides cooperative problem solving into four steps: recognition of potential for cooperation, team formation, plan formation, and team action. The focus of this paper is a concrete approach for the second step, i. e., forming a distinct team that can reach a certain goal. In order to address this task it is necessary to compare the properties of the respective containers. In particular, the presented approach takes concept, location, and time constraints into account. These properties are considered relevant to containers that may take advantage of forming clusters. Having conducted the team formation step the formed cluster is supposed to be adequate for realising the desired goal.

2.1 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 1 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.

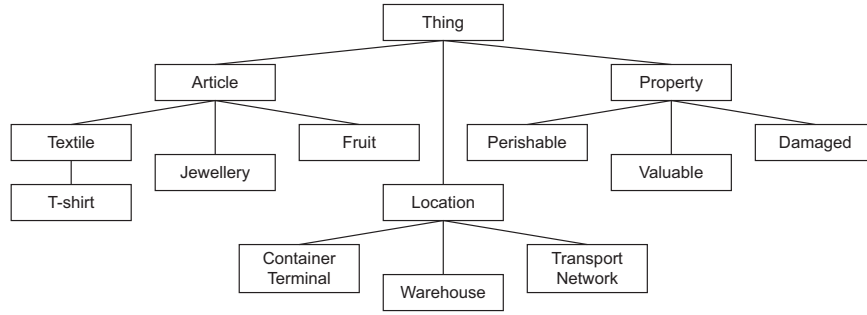


Fig. 1. An example ontology for goods with three major parts: article, property, and location. The lines connecting concepts indicate taxonomical relations

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*

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

2.2 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 [12]. It denotes the decision whether advertisements offered by business entities or agents match requests issued by other agents. An approach proposed by [9] 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.

2.3 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 2 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.

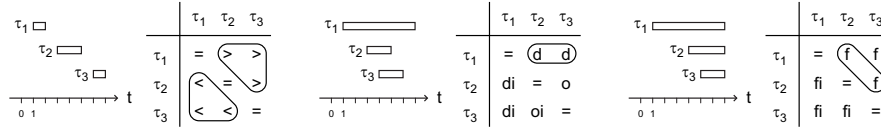


Fig. 2. Three examples for relations between sets of temporal intervals within an agent cluster and the respective temporal matrices

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. 3:

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.

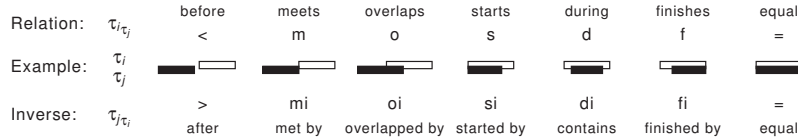


Fig. 3. The 13 relations that have been proposed by Allen [3] in order to characterise the qualitative relationship between two temporal intervals

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:

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

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 2 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:*

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

3 Case Study

In order to test the clustering approach it is implemented within the Java-based agent-framework JADE [5]. The clustering is thereby conducted as follows: as soon as a container joins the network it queries a catalogue service for existing clusters. Subsequently, it communicates its properties to all cluster-heads. Clusters matching the properties of the container 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.

As a case study the approach is applied to data of about 2,400 containers which were in operation during three months in 2006. The goal is to demonstrate that the approach overcomes 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 (see Sect. 1). The lifetimes of the agents are thereby expected to be overlap pairwise without a gap in between.

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. Since each of the n containers contacts all m clusters at most once it is expected that the asymptotic development of agent conversations and computation time is $O(nm) = O(n^2)$.

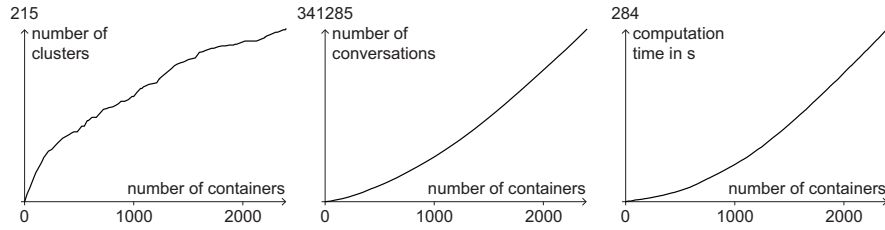


Fig. 4. The number of clusters (left) and agent conversations (centre) and the computation time (right) in relation to the number of containers

The average results of 50 test runs are given in Fig. 4. The number of clusters in relation to the total number of containers 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 (Fig. 4 centre) is 341,285, which is below the expected complexity. That is because not all clusters exist right from the start. Nevertheless, as expected the plotted part of the curve indicates an increase that is faster than linear. This observation also corresponds with the computation time that is depicted on the right hand side of Fig. 4. On a computer with Windows XP and an Intel Centrino Duo processor with 2.16 GHz the total time for clustering is only 284 seconds.

4 Discussion

In autonomous logistics, it is a challenging task for the participating logistic objects to cooperate in achieving their goals. The framework introduced

allows similar 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 on the represented objects. 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.

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. M. Levinson. *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger*. Princeton University Press, Princeton, NJ, USA, 2006.
9. 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.
10. J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–296, 1967.
11. H. Schwerdtfeger, T. von Zabern, and H.-J. Pölking-Eiken. *Bremen/Bremerhaven Container Story*. BLG, Bremen, Germany, 1991.
12. P. Shvaiko and J. Euzenat. A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics*, 4:146–171, 2005.
13. G. Weiss, editor. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1999.
14. M. Wooldridge and N. R. Jennings. The Cooperative Problem Solving Process. *Journal of Logic & Computation*, 9(4):563–592, 1999.