Plug and Play Transport Chain Management: Agent-Based Support to the Planning and Execution of Transports

Paul Davidsson*[1,2], Johan Holmgren[1], Jan A. Persson[1] Į, Andreas Jacobsson[2]

1. Blekinge Institute of Technology, Sweden
2. Malmö University, Sweden
*SE-205 06, Malmö, Sweden. E-mail: paul.davidsson@mah.se

Abstract. A novel approach to efficiently plan and execute effective transport solutions is presented. It provides agent-based support for key tasks, such as, finding the best sequence of transport services for a particular goods transport, monitoring the execution of the transport, as well as the interaction between the involved actors. The approach is based on the FREIGHTWISE framework in which a minimal set of information packages is defined. The purpose is to capture all the information that needs to be communicated between the actors involved in a transport, such as, transport users, transport providers, and infrastructure managers, during the complete process from planning to termination. The approach is inspired by the concepts of virtual enterprises and breeding environments. We analyse the requirements of such an approach and describe a multi-agent system architecture meeting these requirements.

1 Introduction

There is a need for increased collaboration between transport chain actors, e.g. transport users, transport coordinators, transport operators and terminal operators, in order to achieve efficient, and adaptive intermodal transportation. We propose an approach, called *Plug and Play Transport Chain Management* (PnP TCM), which focuses on collaboration for developing transport solutions, for activity coordination, and for administrative information transactions.

PnP TCM is based on the FREIGHTWISE framework (FWF) [1] whose main purpose is to simplify the phases of planning, executing and following up transport services between users and providers of such services. An important aim is to do this without interfering with the internal processes and systems in the organisations. The FWF identifies four transport chain roles. A *Transport User* (TU) is anyone (e.g. a 3PL provider) who wants to transport cargo. A *Transport Service Provider* (TSP) carries out the transport of the cargo, including the management of the transport services and the operation of the transport means and handling equipment. The *Transport Network Manager* (TNM) is responsible for providing information regarding the infrastructure related to planning and executing transports. Finally, the *Transport Regulator* (TR) monitors that all transport services are completed according to existing regulations. All interaction between these actors makes use of a small set of well-defined *information packages*, for which the responsibilities and requirements for use are specified. The PnP TCM approach can be seen as an instantiation of the more general *Plug and Play Business* concept [2] and aims at:

- making information about the available transport services easily accessible,
- providing support for finding the "best" set of transport services for a particular transport, including a match-making functionality that makes it easier for potential transport chain actors to find each other and negotiate with potential collaborators,
- supporting the negotiation and collaboration between the actors in a transport chain,
- lowering the entry barriers for small-sized companies to participate in highly integrated transport chains by providing low cost and easy-to-use software tools.

Ideally, Pnp TCM should be implemented in a completely distributed fashion, both with respect to control and information, and it should be seamlessly interoperable with relevant legacy ERP (Enterprise Resource Planning) systems of the participating actors. Fig. 1 illustrates the PnP TCM approach where each actor uses the PnP TCM software to set up and participate in a transport chain. The software has two types of user interfaces, one type interacts directly with legacy ERP systems, or other relevant information systems, and the other is a simple web browser interface, which may be particularly useful for small enterprises.

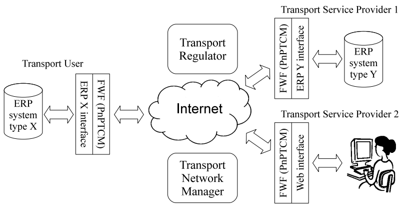

Fig. 1: An illustration of the PnP TCM approach with five participating transport chain actors.

In PnP TCM, transport chains are viewed as *virtual enterprises*, which may be defined as temporary alliances of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks [3]. Another important concept for PnP TCM is *Internet community*. Potential transport chain actors join a PnP TCM community by (installing and) executing the PnP TCM software. To enter the community, an actor needs to declare its address and other formalities, as well as, agreeing on a user license agreement. In addition, a TSP needs to provide a Transport Service Description (TSD) for each transport service it provides. The TSDs should be updated continuously so as to mirror the current situation. The community is dynamic in the sense that enterprises may join and leave it at any time. PnP TCM can be seen as a breeding environment [3] for transport chains, i.e., an association of enterprises that have both the potential and the ambition to collaborate with each other through the establishment of long-term cooperation agreements and an interoperable infrastructure.

2 The FREIGHTWISE Framework

The interaction between the transport chain actors is illustrated in Fig. 2. With exception for the first step, where the Transport Provider plan which Transport Services it should offer, the figure shows what happen during the planning, execution and completion of a transport concerning a certain cargo. Initially, the Transport Provider interacts with the Transport Network Managers using Transportation Network Status (TNS) information packages in order to take into account information about the transportation network.

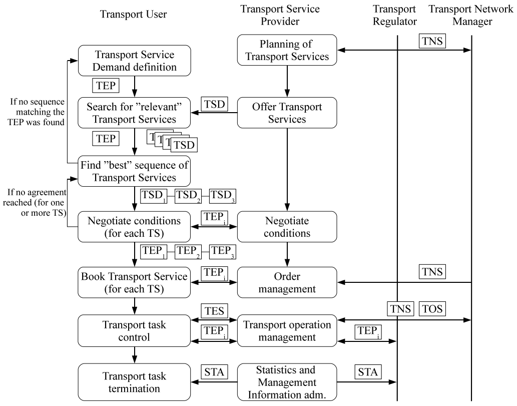
Transport User
Transport Service Provider
Transport Regulator
Transport Network Manager
Transport Service Demand definition
Planning of Transport Services
TNS
TEP
If no sequence matching the TEP was found
Search for "relevant" Transport Services
TSD
Offer Transport Services
TEP
TSD
Find "best" sequence of Transport Services
If no agreement reached (for one or more TS)
TSD1
TSD2
TSD3
Negotiate conditions (for each TS)
TEPi
Negotiate conditions
TEP1
TEP2
TEP3
Book Transport Service (for each TS)
TEPi
Order management
TNS
TES
TNS
TOS
Transport task control
TEPi
Transport operation management
TEPi
Transport task termination
STA
Statistics and Management Information adm.
STA

Fig. 2: The interaction between transport chain actors.

At the beginning of the planning process for a specific goods transport, the TU specifies its transport demands, resulting in an initial Transport Execution Plan (TEP) covering the complete transport from origin to destination. The TEP is initial, or preliminary in the sense that it specifies the items that are to be transported, the desired time and date of collection and delivery, the origin and destination, and the condition requirements of the items during the transport (e.g., ambient temperature conditions), but not which Transport Service(s) to use. The next step is to find relevant transport services that potentially can meet the transport demands specified in the initial TEP. This is done by searching among the TSDs published by the TSPs and selecting those that potentially can be used for meeting the requirements of the TEP.

This set of TSDs then provides the input to the next step, which is to find the sequence of TSDs that satisfies the requirements of the TEP. If more than one such sequence is found, the one that is "best" according to one or more criteria, such as, cost, reliability, environmental impact, etc, is chosen. If no sequence that satisfies the requirements is found, the TEP needs to be revised, e.g. with respect to desired time of delivery (or, additional TSDs need to be found). When a sequence is found, it is time to negotiate the detailed conditions for each of the TSD with the corresponding TSP. This is done in terms of a TEP corresponding to that part of the transport, i.e., for each TSD in the sequence a TEP will be produced, and the sequence of TEPs will correspond to the initial TEP covering the complete transport from origin to destination. If no agreement is reached for one (or more) of the TSDs the process of finding the best sequence will be resumed. The final step of the planning phase is to book the transport services (one for each TEP in the sequence).

During the execution phase, the transport is monitored by the TSP through keeping track of the driver, the transport means, load units etc. The Transportation Network Manager is providing information about the transport network, the traffic as well as individual Transport Operation Status (TOS). The Transport Regulator also receives the Transport Execution Plan and is using this information as part of hazardous goods management, and tax and customs management. The information from the monitoring functions is put together and made available to the TU by means of the Transport Execution Status (TES) information package. When the transport is completed, the required statistic information (STA) is collected, prepared and sent to the TU and the relevant Transport Regulators.

3 The PNP TCM Software

The set of installed PnP TCM software clients, which reside on the users' computers in a community, can be viewed to form an *artificial society*. Davidsson lists four types of artificial societies: open, closed, semi-closed, and semi-open [4]. These categories balance the trade-off between important society properties, such as, openness, flexibility, robustness, and trustfulness. In open societies there are no restrictions at all for joining the society, which makes it very open and flexible, but not robust and trustworthy. In an FWF setting, this may correspond to that all interaction is performed in an ad-hoc fashion, e.g., TSDs are published openly on the WWW and the Transport Users need to find the offers through the use of general WWW search engines, etc. The opposite is true for closed societies where all members must be known when the society is initiated. In an FWF setting, a closed society solution may be a completely centralized system in which all information is stored and through which all interaction between the transport chain actors is mediated. In many situations, such as in breeding environments for transport chains, there is a need for societies that balance the trade-off between the society properties. We will therefore limit our discussion to the two intermediate categories.

An important actor in the context of artificial societies is the "owner" of the society, i.e., the person or organization that have the power to decide which software entities may enter, which roles they are allowed to occupy, what communication language should be used, the set of norms and rules that are valid within the society, etc.

In semi-closed artificial societies, external software agents are not allowed to enter. However, actors have the possibility to initiate new software agents in the society, which will act on behalf of the actor. In semi-closed societies, there is a (central) physical environment, in which the agents (representing their owners) execute and communicate with other agents. This requires that the actors' agents can access some level of mutual communication properties, which are included in the breeding environment. Semi-closed societies convey almost the same degree of openness as semi-open societies, but are less flexible. From a FWF perspective, they fail to meet the requirement of a distributed solution. On the other hand, they have a larger potential for implementing important society attributes, such as, security and trustfulness.

The main difference to semi-closed artificial societies is that, in semi-open societies, agents execute locally on the clients individual computer systems. However, another distinction is that the environment owner is no longer in control of the agents even though the environment owner still has the power to, e.g., dictate the rules of engagement within the society. In order to meet security-related requirements, semi-open societies are equipped with a *Gatekeeper*, to which every agent needs to connect before entering the society. In addition, the gatekeeper, being a trusted third party, may also mediate the payment of transport services, provide contract validation, etc.

When analyzing semi-open societies, it is useful to make a distinction between two types of such societies, those with a centralized and those with a decentralized communication architecture. In the case of centralized communication, the agents run on the members' own individual computer systems, but all communication between the agents is routed via a central server. In decentralized communication, the agents run on the members' individual computer systems and all communication is conducted directly between these end nodes. As the vision of FWF consists of a distributed solution, a semi-open agent society with decentralized communication seems desirable.

3.1 Software Architecture

To arrive at an appropriate architectural design for the PnP TCM system, we used the GAIA methodology [5], in which three main steps (analysis, architectural design, and detailed design) are defined. In GAIA, the designer develops a number of different models, and we here provide a short summary of how we applied the GAIA methodology. The models that are most relevant to discuss here are the Organizational Structure, Preliminary and Final Role models, as well as the Agent Model.

We first describe the first activity in GAIA, which concerns dividing the system into sub-organizations. In PnP TCM, each actor (TU, TSP, TNM and TR) that is able to connect to the PnP TCM community, as well as the gatekeeper of the PnP TCM community defines a separate sub-organization with a corresponding preliminary role. We have identified four categories of activities (community connection management, TSD management, building/negotiating/booking transport solutions, and information handling), in order to identify important activities and responsibilities and to divide the preliminary roles into more specialized roles.

When deciding upon the structure of the Multi-Agent System (MAS), a trade-off usually has to be made between several factors; workload of the members of the organization, computational complexity, organizational simplicity, organizational rules, and the structure of the real world system, etc [5]. We have chosen structures for the sub-organizations, which support the design of generic clients, which are able to represent different types of enterprises (e.g. varying in size and structure). Moreover, we have defined different sub-organizational roles in order to achieve a reasonable workload distribution among the members of a sub-organization.

On an overall PnP TCM organizational level, we have chosen a distributed approach, where agents interact directly with each other in a peer-to-peer fashion. However, the community is controlled by a gatekeeper, who decides who should be allowed to enter the PnP TCM community, provides information about connected actors, as well as, managing a reputation system. A distributed approach makes the system more robust, and it helps distributing the workload among the different members. Also, it enables privacy assurance of the participating organizations since it enables local control of information.

A number of roles are modeled in the clients that connect to the PnP TCM community; a (Community) Connection Manager who is responsible for managing the connection to the PnP TCM Community, a Human User Interface that allows the human operators to interact with the client MAS, and an Information Manager who is responsible for providing and processing information that is communicated between clients. A connection to the local ERP system should be possible. Hence, an ERP Adapter role is explicitly modeled.

A significant characteristic of a TU is that it performs several sequences of activities in parallel in order to find, negotiate, and book a solution for a transport task (TEP), at the same time as it manages the connection to the PnP TCM community. Therefore, we propose a hierarchical structure where a TEP Manager role is responsible for coordinating all the parallel sequences of activities (multiple TEPs should be processed in parallel). To deal with the computational bottleneck of combining TSDs into end-to-end transport solutions (i.e. optimizing), we propose that the TU contains multiple Optimizers (described by an Optimizer role) and an Optimization Manager who is responsible for initiating new Optimizers to which optimization tasks are delegated. For maintaining a local repository of TSDs, (subscribing for TSDs,...), a TSD Finder role is defined. To further distribute the workload of the TU, responsibilities related to negotiating and booking transport services is assigned to a Negotiation and Booking Manager role.

The major characteristic of a TSP is that it perform several parallel, however typically not very resource demanding, activities, such as managing TSD subscriptions, negotiations, and bookings. A TSD Announcer is responsible for managing subscription information, sending TSDs to subscribers, etc, and a Negotiation Manager and Order Manager is given the responsibilities to negotiate and manage bookings of transport services. It should be noted that fleet management is performed externally, which is why the ERP Adapter serves as the interface towards the planning system.

The purpose of the agent model is to map the roles that were defined in the role model into agents (an agent is defined as a set of roles). There are many ways in which this mapping can be performed, and we propose a one-to-one mapping between roles and agent classes. However, there are a few exceptions; in a TU client there are several Optimizer agents and the TEP Manager agent plays the role of both the TEP Manager and that of the Optimization Manager. Each organization that joins the PnP TCM community has a software client that is implemented as a MAS. In Fig. 3, the agent systems for TUs and TSPs are illustrated.

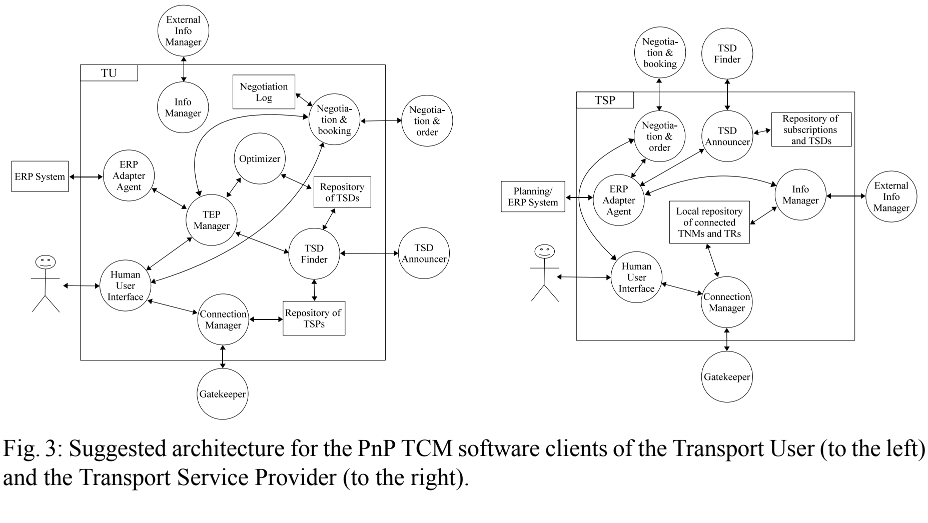

Fig. 3: Suggested architecture for the PnP TCM software clients of the Transport User (to the left) and the Transport Service Provider (to the right).

3.2 The TSD Announcers and Finders

The task of the Finder is to provide the Optimizer with all TSDs currently relevant for the corresponding TU. The task of the Announcer is to distribute the TSDs of the corresponding TSP to those TUs who finds them relevant (and to notify the TUs if a TSD is not valid anymore).

There are a number of important quality attributes that should be taken into account when designing the Announcer and the Finder. Assuming that there may be hundreds of thousands transport services, or TSDs, available at any point in time, and that thousands of new TSDs may be made available each day, this places some strong requirements with respect to *scalability*. That is, the system should be able to handle large numbers of users without causing long response times, etc. The scalability is related both to telecommunication, computer processing, and computer memory usage. Another important quality attribute is *robustness*, by which we here mainly mean the availability of system critical information (e.g., the TSDs) at any point in time.

Distributing and Storing the TSDs The main design issue is where to store the TSDs and how to distribute them. A simple solution would be to store all TSDs in a central repository. The task of the TSD Announcer is then just to send new TSDs to this repository, and the TSD Finder would interact directly with this repository asking for relevant TSDs. However, there are several problems with this solution. For instance, it is not robust as the repository will constitute a single point of failure and, moreover, scalability can be questioned, as the central repository will constitute both a communication and processing bottleneck. An extreme solution would be to let the task of finding the best sequence of TSDs be handled centrally, but such an approach would put even harder requirements on the central processing capacity. However, it may be a back-up solution for situations when the TU interacts with PnP TCM using equipment with limited processing capabilities, such as, some mobile devices. There are in principle two different completely distributed extreme solutions:

1. The TSDs are stored only at the Transport Service Provider offering it. This is fairly robust, but will cause a huge amount of communication overhead. A TU would need to ask all TSPs about their TDSs for each transport to be planned.
2. All TSDs are stored in a local repository at all TUs. This is very robust as all TSDs will always be available for the TUs. However, the scalability is weak with respect to the number of TSDs as the distribution and storage will require a large amount of memory and communication.

An attractive feature of the second distributed approach, where each TU has all relevant TSDs stored in a local repository, is that there is no need for communication between the Announcers and the Finder during the planning of a particular transport, thus minimizing response time and adding to the system's robustness. However, as mentioned above, scalability could be perceived as a problem. Broadcasting all TSDs to all TUs will probably lead to unnecessary communication and storage of irrelevant TSDs at the TUs. A slightly more sophisticated approach, which may remove a large number of the irrelevant TSDs and thus reduce the communication and storage needs, would be to let the TUs *subscribe* to TSDs with specific characteristics, e.g., based on geographical areas. (Note that the subscription information does not need to be stored and distributed by any central entity, it is only known by the TSP and the TU involved.) A TU does not need to subscribe to all TSDs of a TSP, it could be limited to transport services within a geographical area, and/or concerning certain type of goods. However, the TU then needs, in order to know which TSP a TU should subscribe, some general characteristics of what type of transport services each TSP provides. This information does not need to include information about timetables, costs, etc., but instead focus on “from-to” information (i.e., shipping between port A and B, or the geographical areas covered) and goods type. This info is assumed to be updated rather infrequently. The TU (Finder) then subscribes to the TSDs of the TSPs it finds relevant, by sending a subscription message to those TSPs (Announcers). The TU (Finder) will then immediately receive the TSDs currently offered by the TSP. When a TSP has a new transport service to offer, the corresponding TSD will be sent (by the Announcer) to those TUs (Finders) subscribing to it. There is also a possibility for the TSP to inform the TUs that a TSD is no longer available, e.g., due to that all capacity of a timetabled transport is already allocated by sending a *retract* message.

Analysis of the Scalability As mentioned above, this approach may be weak with respect to scalability. Therefore, this aspect is analyzed in more detail. In the proposed approach, each TU needs to store the (IP) address and characterization of each TSP. Assuming that this information can be stored in 1 kB, the information for 100.000 TSPs could be stored in 100 MB which is negligible for modern computer devices and would take just a minute or two to distribute via a normal broadband connection.

We estimate that the memory requirement of a TSD in average is approximately 20 kB. Thus, 100.000 TSDs would then correspond to 2 GB. As the cost of memory of relevant sizes is negligible (e.g., 8 GB flash memory costs below €10), we can conclude that storing 400.000 TSDs locally would be no problem for any type of device. Thus, the main issue concerns the distribution of the TSDs. In Table 1, it is presented how long time it would take to send X TSDs using a communication channel with capacity Y (20kB corresponds to 20*1024*8 = 163.840 bits).

Table 1: A presentation of the timeslots involved in TSD distribution for three types of network communication.

	1 TSD	100 TSDs	10.000 TSDs	1.000.000 TSDs
144 kbps*	1.14s	114s	190 min	316 h
2 Mbps**	0.08s	8.19s	13.6 min	22.8 h
10 Mbps***	0.02s	1.64s	2.73 min	4.55 h

* IMT-2000 (the global 3G wireless communication standard) specifies a minimum data rate of 144 kB/second in high-mobility (vehicular) applications of 3G
** IMT-2000 specifies a minimum data rate of 2 MB/second for indoor (stationary) applications of 3G
*** Normal broadband connection (ADSL, WiFi, etc.)

As can be seen, using a normal broadband connection, the initial downloading of one million TSDs would take a couple of hours and daily updates of ten thousands TSDs would take a few minutes. These figures seem acceptable from a users point of view. For highly mobile (vehicular) devices, on the other hand, such an approach seems not viable using current communication technologies. However, it would probably not be necessary to have a complete and updated set of TSDs while on the move.

Interaction Protocols A TU may at any time decide to start subscribing for TSDs of

TSPs to which it previously is not subscribing. This could be useful when a transport to or from an unforeseen geographical region is planned. Similarly, it could unsubscribe from TSDs of a particular TSP. A TSP may at any time revise its general characterization of the transport services it provides. The revised characterization is sent to the Gatekeeper, who then distributes it to all TUs. The Gatekeeper also distributes the characterizations of new TSDs entering the PnP TCM community immediately to all TUs. The interaction between the Gatekeeper, TUs (only one is shown) and TSPs (only one is shown) focusing on the Finder-Announcer interaction can be seen in Fig. 4.

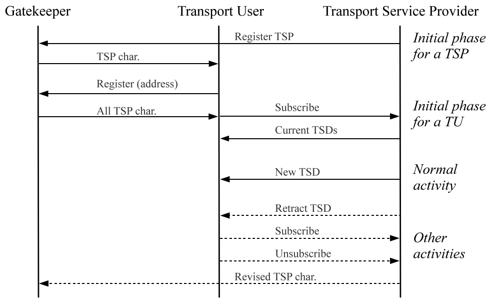

Fig. 4: The interaction between the Gatekeeper, TUs and TSPs.

In addition, there should be interaction protocols for leaving the community. Moreover, the Gatekeeper may be responsible for a reputation system concerning the TUs and TSPs. For instance, if a TSP (or TU) has refused to act according to the agreement between two parties, the other part can inform the Gatekeeper, who keeps track of the complaints, and if necessary takes an appropriate action. It could also be the case that a TSP (or TU) refuses to follow the rules of the community, e.g., not replying to requests.

Related Solutions There are a number of varieties of the proposed approach, e.g., it can be made even more distributed by letting the Gatekeeper only know about the addresses of the TSPs (and thus not their characteristics). A disadvantage with this approach is that a TU must ask all the TSPs about their characteristics, causing a lot of unnecessary communication and processing. In this case, the distribution of revised TSP characteristics becomes more complex in that the TSP needs to know the addresses to all TUs.

On the other hand, the approach can be made slightly more centralized and thereby lowering the communication needs. The distribution of new TSDs does not have to be done continuously (at announcement); it could also be done in batches, e.g., once a day. In this case the distribution could be handled by the Gatekeeper. However, this makes the system more centralized, as the central entity needs to store and distribute the TSDs.

3.3 Optimizer

From the set of locally stored TSDs, the task of the Optimizer is to find the “best” sequence of TSDs that satisfies the requirements of a TEP, according to one or more criteria, such as, cost, reliability, environmental impact, etc. A large number of optimization algorithms have been developed to solve different types of shortest path problems. However, the type of optimization problem that needs to be addressed by the Optimizer has a number of non-standard characteristics, e.g., the mixing of scheduled and non-scheduled transports, restricted vehicle capacities, and time windows for pick-ups and deliveries. According to our knowledge, there is no available off-the-shelf algorithm that can be used.

Solution Approach For the problem of generating a set of transport solutions (sequences of TSDs), we propose a heuristic solution approach that consists of three steps that are performed in sequence. An important reason for suggesting a heuristic approach is that there might exist millions of TDSs that can be combined into an immense number of transport solutions. This makes an exact model too complex even for a smaller number of available TSDs. For the transport task at hand, the Optimizer first filters out a set of relevant TSDs (defining a network of TSDs) from a possible huge amount of available TSDs that the TU has at hand by its rules of subscription. Next, it uses the selected TSDs to create a number of candidate end-to-end paths based on approximated average characteristics of TSDs, which in the final step are considered in more detail to find the best sequences of TSDs (exact characteristics of the TSDs are considered).

Step 1: Transport Service Network Generation (finding relevant TSDs) The main task of the first step is to find a relevant set of TSDs, in order to reduce the set of TSDs to be used in the subsequent steps of the solution approach. As mentioned above, the TU keeps a local repository of TSDs (managed by the TSD Finder), corresponding to some specific characteristics, such as geographical area, cargo type, etc. Since the local repository of TSDs may contain a huge amount of TSDs, the optimizer needs to have some kind of filtering mechanism that enables it to find a set of relevant TSDs for the particular transport at hand. Filtering based on geographic areas, makes the filtering non-trivial. For instance, how can it be determined whether a TSD with origin X and destination Y is relevant for a TEP with origin A and destination B? It should be noted that filtering can be made in many ways, and here we present one possible way of doing it, i.e., more filtering rules can be added if needed.

The first step of the solution approach requires the following input (i) definition of a transport task, i.e., origin, destination, time-windows for pickup and delivery, as well as, information about the goods to be transported. This information is available in a preliminary TEP and (ii) the set of all locally stored TSDs (including cost, timing, etc) that describe the available transport services. A scheduled TSD specifies the origin, destination of the transport service, departure/arrival times, transport cost (potentially including cost for loading and unloading), as well as, time for loading and unloading goods. For a demand-driven TSD, a service location (area or point), cost and time for loading and unloading cargo are specified, as well as, information that allows the transport cost and time between two specific locations to be estimated by the Optimizer. It produces a set of relevant artificial TSDs (ATSDs) including their origin and destination nodes/terminals. The idea of introducing ATSDs is that an ATSD will always have the information of a specific origin, a specific destination, a travel time and a cost. This is in contrast to a TSD, which may be specified for operation within a service area without specific origin and destination. Further, a TSD may be representing multiple transport options between different pairs of service locations or possibly represent the transport legs of a route, and hence, will be substituted by a number of ATSDs.

Step 1.1 (initial filtering) A first activity is to filter out non-relevant TSDs by disregarding all TSDs that are incompatible with the specified type of goods or have irrelevant timing, i.e., scheduled TSDs with departure earlier than the pickup time window or arrival later than the delivery time window.

Step 1.2 (geographic limitation of transport services) Next, we propose to use a geographical limitation of the transport service network. An obvious approach that can be used to reduce the size of the transport service network is to focus on all scheduled transport services within a specified geographical area. We propose to use an ellipse that contains both the origin and destination of the TEP to disregard those transport services that do not support any service (both pickup and delivery) in the elliptic area. In most cases, an appropriately chosen geographical area should work satisfactory, but there are also examples of when the characteristics of the geography (for instance, when there is a geographical obstacle in between the origin and destination) might make this approach somewhat inefficient. This problem can be dealt with by using pregenerated paths between main terminals, but this is, however, outside the scope of this paper. In order to further limit the scheduled transport services within a geographical area, we propose to use a sequence of 2-dimensional circles with the center on a straight line between the origin and destination of the TEP. We suggest that these circles are numbered with higher numbers for areas closer to the destination. The approach with sequenced circles enables filtering out transports going in the “wrong” direction.

Step 1.3 (creating scheduled ATSDs) The aim of this step is to generate a network of scheduled transport services represented as ATSDs. Before creating ATSDs, the Optimizer should select a set of terminals or nodes, where the considered type of goods can be reloaded. A minimum requirement is that these terminals should be located inside the geographical area that was selected in step 1.2 and be served by scheduled transports.

The idea is to iterate over all TSDs that were used to select relevant terminals, and for each of them create a relevant set of ATSDs (one or more). We suggest to only create ATSDs that have both the origin and destination within the geographical limiting ellipse. We also suggest to filter out (i.e. not considering) ATSDs that departs from a circle with a higher number and arriving to a circle with smaller number. An example of how TSDs can be transformed into ATDs is given in Fig. 5.

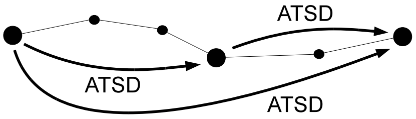

Fig. 5: Example of how artificial TSDs for a timetabled route are generated.

Step 1.4 (connecting origin and destination) In this step non-scheduled ATSDs are added in order to connect the origin and destination (of the TEP) with the previously generated network of scheduled transport services. We propose to follow an approach similar to Caramia and Guerriero [6]. In our context, their approach translates into:

1. First, generate ATSDs that represent non-scheduled (typically road) transports directly from the origin to the destination of the TEP. These transports somehow represent backup solutions that can be used in case no better solution can be found.

2. Next, generate ATSDs, corresponding to non-scheduled TSDs that connect the origin and destination, respectively, with the n closest terminals.

Note that these ATSDs require that transport costs and transport times can be estimated despite the fact that non time-tabled TSDs often are specified for service areas and not for point to point locations.

Step 1.5 (connecting transport networks) When using the proposed strategy, there is a risk that there exist clusters of connected nodes (i.e., groups of terminals) that are not connected (i.e., there is no ATSD connecting them). We suggest to fill these gaps in the service transport network by adding nonscheduled ATSDs between terminals that otherwise cannot be connected. One could potentially look for non-scheduled TSDs between all pairs of nodes belonging to different sets of connected nodes. The considered pairs could be reduced by using their geographical location (within the group in relation to origin and destination of the TEP). However, exactly how this could be done efficiently is not obvious and we leave this issue to future work.

Step 2: Path Selection The aim of this step is to create candidate paths and subsequently also to reduce the set of ATSDs by disregarding ATSDs that do not belong to any of the candidate paths that connect the origin with the destination of the TEP. We propose to solve k-shortest paths problems to obtain a set of candidate paths. Since the k-shortest paths problem cannot handle time-dependencies (i.e., timetables) explicitly, the network of ATSDs that was generated in step 1 cannot be used directly in this step:

- Two nodes may be connected by multiple scheduled ATSDs, with different departure times, traveling times, costs, etc.
- The departure and arrival times of the transports that might be used in a path from the origin to the destination of the TEP cannot be assumed to be well synchronized, and it is difficult to explicitly account for this, i.e., to estimate the total of waiting and travel time.

To deal with these complicating factors, this step includes the computation of approximated costs and traveling times of the scheduled ATSDs. We suggest to do the approximation by computing the average cost and travel time per operator between two nodes. Obviously there are alternatives to consider when approximating cost and travel time, e.g., such as using the minimum cost and travel time offered for an operator.

To compensate for lack of synchronization and terminal processing time, we propose to add some extra time to the approximated (averaged) transport time and subsequently to the cost of the scheduled ATDS. We suggest that, in order to produce potentially relevant paths, the k-shortest paths problem should be solved at least three times with different link attributes: cheapest, fastest, and weighted time and cost.

Step 3: Optimizing the Selected Paths (finding sequences of TSDs) The final step in the solution approach is to iterate over the paths that were generated in the previous step, and for each of them find the best combination of TSDs that fulfills the requirements of the TEP. The output is a set (possibly empty) of feasible sequences of ATSDs, which can be sent to the "negotiation and booking" agent for further processing.

Holmgren et al. [7] proposed an optimizing algorithm for a problem that is similar to the problem that needs to be solved in this step of the solution approach. They considered a problem with time-based costs, and allowed pickup and delivery outside the time windows with a penalty cost. These minor differences should not cause any problems for reapplying the approach in this context. The approach has been incorporated in a multi-agent-based transport chain simulator, where simulated transport planners used it to propose intermodal door-to-door transport solutions. The approach has shown to work well in simulated scenarios [8] for a reasonable amount of alternative transport services. It is based on using a search tree with nodes representing ATSDs with departure times, arrival times and (cumulative) costs. The search tree is gradually expanded with new child nodes as the search progresses, and since the search begins in the destination of the TEP, the child node represents an ATSD one step (in the considered path) closer to the origin than the parent node. It uses a number of techniques to make the search tree more efficient, i.e., reduce the nodes investigated: (i) identify and remove branches which will lead to infeasibility, (ii) depth first based search for finding potential feasible solutions, (iii) choose branches representing earlier transports first, and (iv) prune on costs (possible since costs are cumulative from the root).

Sometimes, the TEP defines strict time windows with earliest and latest times both at the origin and destination. A relevant issue that needs to be considered in this step of the solution approach is how the Optimizer should act when an optimal transport alternative is too fast to obey the time windows of the TEP. The proposed solution to this problem is to reapply the algorithm with relaxed time windows to at least be able to produce a set of solutions that can be sent to the negotiator for further processing.

Fault Recovery (infeasibility handling) An important issue that needs to be considered is that the Optimizer may fail to generate a feasible solution (or the best solution may not fulfill the required quality). Two reasons might be that: (i) no connection exists between origin and destination given the considered TSDs, and (ii) no existing connection obeys the time windows. Hence, the Optimizer should provide information to the TEP Manager about the results: suggest one or more solutions, or a failure and its reason. Hence, the TEP Manager could accept one or more solutions or suggest appropriate actions in case of failure (i.e., expand the considered geographical area or the set of subscription of TSD), and potentially involve a human planner. Thus, searching for a transport solution can be seen as an algorithm with an inner loop containing the three steps performed by the Optimizer, and the outer loop contains the analysis of the feedback and decision taken by the TEP Manager.

Alternative Approaches for the Optimizer For the problem of generating a set of sequences of TSDs that fulfill the requirements of a TEP, we have proposed a solution approach were the Optimizer solves the problem in three sequential steps. It is possible to integrate these steps in a number of ways; for example, the steps of path selection and optimizing the selected paths could be performed together. In the literature, we have not found any model that deals with all of the three steps, either integrated nor sequentially as in our solution approach. However, there exist a few models (cf. [6, 9–11]) that deal with, and integrate, the second and third steps of the solution approach, but none of them solves exactly the same versions of the problem that we are dealing with. Nevertheless, some of the approaches show promising results which could be explored further.

3.4 Negotiation, Booking and Order Manager

The task of the Negotiation and Booking Manager of a TU is to book a transport service for each of the TSDs in the sequence of TSDs provided by the Optimizer (which has been converted to a sequence of TEPs by the TEP Manager) according to the preferences of the TU. Thus, the output is a sequence of TEPs that are agreed upon with the TSPs providing the transport services. The task of the Negotiation and Order Manager of a TSP is to secure agreements with TUs according to the preferences of the TSP.

There is a long tradition in the area of agent-based systems of studying the automation of reaching agreements. From work in the negotiation area (cf. Jennings et al. [12]), we identify four different components as relevant for the PnP TCM setting, namely:

- a negotiation set, representing the space of possible obligations agents can make,
- a protocol, which defines the legal obligations that the agents can make,
- a collection of strategies, one for each agent, which determines what obligations the enterprises will make, and
- a rule that determines when the negotiation is over and the deal has been closed.

In the negotiation between a TU and TSP the negotiation set consists of the different terms and conditions specified in the TEP (and the possible values they can assume).

The protocol has two components, the pre-booking agreement as specified by a TEP, and the actual order (or booking), which is also specified as a TEP. The negotiation results in an electronic contract, in this case a TEP, which govern the collaboration process. Electronic contracts are to be regarded as virtual representations of traditional contracts, i.e., “formalizations of the behaviour of a group of agents that jointly agree on a specific business activity” [13]. Electronic contracts usually have a set of identified roles, obligations or prohibitions to be fulfilled by the parties involved in the relation. The FREIGHTWISE framework focuses on obligations, i.e., that an agent’s role is defined by the obligation, which it has towards another agent to bring about a certain state of affairs before a certain deadline.

The general strategy of the Negotiation Manager of a TU is to first reach pre-booking agreements for all the TEPs in the sequence, and then place actual orders for each of the TEPs. If agreement is not reached for one or more of the TEPs (and the alternative choices that may have been provided by the Optimizer has been tried), this is reported to the TEP Manager who then asks the Optimizer to find a new sequence of TSDs (or possibly just replace the ones where negotiation failed).

The pre-booking phase starts when the Negotiation Manager of the TU sends the first version of the TEP to the Negotiation Manager of the TSP (for each of the TEPs in the sequence). The TSP’s Negotiator then has three options: (i) to accept the TEP as it is, thus conforming the pre-booking of the transport service, (ii) to reject the request, or (iii) to modify the TEP by changing one or more terms or condition, let us denote this revised version for TEP’. The two first options will end the negotiation (by sending an accept or reject message to the TU), whereas the third option will give back the initiative to the TU. The TU’s Negotiator now has the same three options as the TSP’s Negotiator had: to accept TEP’, to reject it, or to modify it. As before, the two first options will terminate the negotiation, whereas the third will provide a new bid (TEP”) to the TSP to

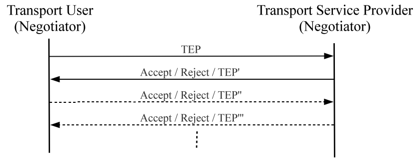

Fig. 6: An example of a negotiation process between a TU and a TSP.

consider. The negotiation will continue until either the TU or the TSP accepts or rejects the current TEP, as illustrated in Fig. 6.

The Negotiators may be given limits for what is acceptable by its “owner” (the human/organization on whose behalf it negotiate) regarding the terms and conditions that are subject for negotiation. This can make the negotiation automated to a large extent. However, when such directives are not available, it may be necessary for a Negotiator to check with its owner whether a certain bid is acceptable or not. Moreover, it may be necessary for the TU Negotiator to coordinate a number of negotiations. For instance, it may have been given a limit on the total cost of the transportation.

When the TU Negotiator has managed to negotiate agreements (pre-bookings) with all the involved TSP Negotiators, it should book (order) all the transport services. This is done by sending the agreed-upon, final version of the TEP for each of the transport services to the corresponding TSP Negotiator. They will then confirm the order or, if something unforeseen has happened which caused the TSP not to be able to carry out the service, reply with a reject message. If the TU Negotiator receives one or more reject message, it needs to report this to the TEP Manager, who then can ask the Optimizer to find the second best option for the missing transport service (which may be a completely new sequence of TEPs). The TU Negotiator may even report the reject to the Gatekeeper, who then can keep track on the number of rejects of different TSP. This could form a basis for the reputation system, where the number of rejects (in the booking/ordering phase) gives an indication on the reliability and trustworthiness of a certain TSP.

3.5 Interface Agents

From the users’ perspective, the complexity of the process of setting up the transport chain should be hidden by the PnP TCM software. As mentioned earlier, there are, at least, two types of interfaces:

- A web-based interface, which can be used by all types of users, independently of company size and IT maturity. One version for the TU, which consists of a number of different views specialized for each of the phases in the process, and one version for the Transport Service Provider, also with a number of different views.
- An adapter agent interface, which makes the PnP TCM software interoperable with the user’s ERP or other legacy system. An adapter may have to be developed for each ERP system, but once it is developed it can be re-used by other organizations using that ERP system. One approach to implement the adapters is the general wrapper agent solution based on open source freeware introduced by Davidsson et al. [14] that makes it possible for any business system to exchange (administrational) information with any other business system.

4 Conclusions

We have suggested a distributed approach to realize the FREIGHTWISE Framework vision, based on the concepts of Virtual Enterprises and Breeding Environments. Moreover, we have described agent-based software architectures for the PnP TCM software clients. The viability of the approach was analysed theoretically and has been partially validated in a number of business cases within the FREGHTWISE project.

References

1. Fjørtoft, K., Westerheim, H., Vannesland, A., Hagaseth, M.: WP13 FREIGHTWISE framework architecture, release 3 (2009)
2. Davidsson, P., Hederstierna, A., Jacobsson, A., Persson, J.A., Carlsson, B., Johansson, S.J., Nilsson, A., Ågren, G., Östholm, S.: The concept and technology of plug and play business. In Manolopoulos, Y., Filipe, J., Constantopoulos, P., Cordeiro, J., eds.: ICEIS (4). (2006) 213–217
3. Camarinha-Matos, L., Afsarmanesh, H.: Elements of a base VE infrastructure. Journal of Computers in Industry **51**(2) (2003) 139–163
4. Davidsson, P.: Categories of artificial societies. In Omicini, A., Petta, P., Tolksdorf, R., eds.: ESAW. Volume 2203 of Lecture Notes in Computer Science., Springer (2001) 1–9
5. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing multiagent systems: The gaia methodology. ACM Transactions on Software Engineering and Methodologies **12**(3) (2003) 317–370
6. M.Caramia, Guerriero, F.: A heuristic approach to long-haul freight transportation with multiple objective functions. Omega **37**(3) (2009) 600–614
7. Holmgren, J., Davidsson, P., Persson, J.A., Ramstedt, L.: An agent based simulator for production and transportation of products, The 11th World Conference on Transport Research, Berkeley, USA, 8 - 12 June 2007 (2007)
8. Davidsson, P., Holmgren, J., Persson, J.A., Ramstedt, L.: Multi agent based simulation of transport chains. In Padgham, L., Parkes, D.C., Müller, J., Parsons, S., eds.: AAMAS (2), IFAAMAS (2008) 1153–1160
9. A.Ziliaskopoulos, Wardell, W.: An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays. European Journal of Operational Research **125**(3) (2000) 486–502
10. Chang, T.: Best routes selection in international intermodal networks. Computers & Operations Research **35**(9) (2008) 2877–2891
11. Jansen, B., Swinkels, P., G.J.A., Teeuwen, B.d., Fleuren, H.: Operational planning of a large-scale multi-modal transportation system. European Journal of Operational Research **156**(1) (2004) 41–53
12. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: Prospects, methods and challenges. International Journal of Group Decision and Negotiation **10**(2) (2001) 199–215
13. Cardoso, H.L., Oliveira, E.C.: Virtual enterprise normative framework within electronic institutions. In Gleizes, M.P., Omicini, A., Zambonelli, F., eds.: ESAW. Volume 3451 of Lecture Notes in Computer Science., Springer (2004) 14–32
14. Davidsson, P., Ramstedt, L., Törnquist, J.: Inter-organization interoperability in transport chains using adapters based on open source freeware. In: Interoperability of Enterprise Software and Applications, Springer Verlag (2005) 35–43