

# Agent-based Multi-organizational Interaction Design: A Case Study of the Dutch Railway System

Jie Jiang\*, Bob Huisman\*<sup>§</sup> and Virginia Dignum\*  
\*<sup>§</sup>Delft University of Technology, Delft, The Netherlands  
Email: {jie.jiang, b.huisman, m.v.dignum}@tudelft.nl  
<sup>§</sup>NedTrain, The Netherlands

**Abstract**—In multi-organizational alliances, inter- and intra-organizational interactions are performed in dynamic environments towards the cooperative goals. To adapt to such dynamics, not only should organizational design provide sufficient representation of organizational requirements, but also provide control mechanisms to switch between different contextual states as a response to an external change. OperA+, as an agent-based organization modeling framework, on the one hand, inherits the advantages of describing and analyzing complex relations between entities, and on the other hand, supports design, analysis and communication of organizational knowledge at multiple levels of abstraction in multiple contexts. This paper aims at a demonstration of OperA+ for an analysis and design study of the Dutch railway system in the domain of train maintenance which is expecting an organization evolution from the current system to a new design.

**Keywords**-organizational design; multi-agent system; contextual behavior; railway system; condition-based maintenance.

## I. INTRODUCTION

In multi-organizational alliances, organizational models are the key instrument for designers and decision-makers to understand, analyze and communicate organizational knowledge, which in turn help to improve organization structures and interactions between people and other organizational resources. Given that organizations operate in dynamic environments, it is difficult to capture the continuously changing systems where inter- and intra-organizational interactions are performed at multiple levels towards the cooperative goals.

To reflect such dynamics, organizational design should, on the one hand, provide sufficient representation of the organizational requirements so that the overall system complies with its objectives, and on the other hand, provide control mechanisms to switch between different contextual states as a response to an external change. Furthermore, it is essential to have a clear and mutual understanding between designers and managers so that the whole system can be maintained in a consistent way.

Agent-based models have been increasingly developed and adopted to describe, analyze and simulate socio-technical systems and explore phenomena concerning complex relations between entities (e.g., [1], [2], [3]). OperA+ is an agent organization modeling framework that targets

multi-organizational collaboration in dynamic settings. It supports design, analysis and communication of organizational knowledge in such a way that processes and knowledge are integrated at different levels from abstract missions/values to operational details so that designers can decide the appropriate level of analysis relevant to their requirements, and stakeholders with different interests can have a better communication.

More importantly, OperA+ introduces a dynamic organizational structure as an adaptive response to environmental changes through contextual representations and refinements. Most of the agent organization frameworks (e.g., [4], [5]) support organizational changes as well, but rely on the actions of the agents themselves during operation. The question is *why* should system design merely “help” the agents creating an initial organizational structure while expecting them to react properly on sudden and significant changes in the environment afterwards. We posit that it makes sense to anticipate and guide structural adaptation at design time by defining a set of organizational structures to be invoked by environmental variables. This makes it easier to predict and verify system behavior, which facilitates the acceptance of self-organizing systems in serious business applications. Furthermore, the predefined organizational structures enable the designers to make full use of the existing knowledge on organization design in management science, which will improve system performance and reliability. On the other hand, adaptation at run time is more complex and couldn't always be foreseen. In this case, the internal adaptation of agents is thus an important aspect, which should be differentiated from the perspective of our approach.

To illustrate these features, this paper demonstrates the use of OperA+ by applying it to an analysis and design study of the Dutch railway system in the domain of train maintenance. The railway system includes a variety of entities (e.g., software, human) from several organizations that are largely autonomous, geographically distributed, and heterogeneous in terms of their operating environment, resources, and goals. In this system, agents represent real interests and real entities, i.e., different agents have different owners, goals, interests, and preconditions for collaboration. With the introduction of intelligent sensors and artificial agents,

the railway system is expecting a new design to improve both system efficiency and reliability.

The rest of this paper is organized as follows. In Section II, we describe the train maintenance scenario in the Dutch railway system, as a distributed decision making (DDM) process. In Section III, the design problem of the railway system is defined. In Section IV, the OperA+ framework for modeling agent organizations and interactions is presented, which is then used for analyzing organizational evolution of the current real life system to the future design respectively in Section V and Section VI. In Section VII, we discuss the effectiveness of OperA+ when it is applied to such a multi-agent coordination problem. The related work is briefly discussed in Section VIII. Finally, we conclude our work and raise directions for future work in Section IX.

## II. TRAIN MAINTENANCE SCENARIO

Currently, train maintenance occurs on a regular mileage or time basis. The question whether a job has to be executed, follows from inspection results obtained at the depot. To compensate for the uncertainty with respect to the execution of individual jobs, the volume of resource exceeds the average demand and slack is build into production plans. Furthermore, it is difficult to anticipate randomly occurring potential failures.

To improve the ratio of fleet availability and maintenance resources, as well as operational reliability, the maintenance company is planning to use information that is available from sensors and computers on board of trains. With this on-line information, it becomes possible to model the actual status of trains and to generate or predict orders for specific maintenance jobs and hence depot visits. Although sensor data might reduce uncertainty, the organizational principle will remain unchanged in the new design.

Within the scope of the problem, different organizations are involved, each having its own internal functional structure. We assume that each of them complies to some set of overall system objectives, so one can speak of an “Organizational Distributed Decision Making (DDM) System” [6]. In this DDM system, we consider three types of agents: human planners, software systems, and (parts of) organizations. Human planners make the final decisions, while the software systems support the process by communicating potential consequences of different plan options and by giving suggestions to solve conflicts and to optimize plans. That is, (automatically generated) schedules should be taken as suggestions [7], while the human planners might refer to specialized schedulers or team leaders on the maintenance depot.

## III. PROBLEM DEFINITION

All of the organizations involved, the processes, and the software products in the railway system are dynamic entities. Currently, most of train maintenance is executed in large

packages at some well equipped depots once every three months. In the future, the use of on-line status information of trains will increase steadily. Then schedules are the result of matching dynamically the need for maintenance jobs of individual trains, the expected use of resources and components, the status of the depots, and the status of transport operations. Meanwhile, software systems used by the organizations will be adapted and renewed. This continuously changing system can hardly be caught in a single-level technical design. To these ends, several requirements are needed for a design solution. On the one hand, agent interaction must be represented independently from their internal design. On the other hand, the interaction structure itself should leave room for organizational and technological developments. So the system design itself must be flexible and extensible over time. Such requirements are well supported by OperA+ which is selected for an analysis and design study of this DDM planning system.

## IV. OPERA+

OperA+ ([8]–[10]) is an agent-based, multi-context, multi-level framework for modeling multi-organizational collaborations. Targeting multi-organizational alliances in dynamic settings, OperA+ expands organizational modeling in two directions that enable analysis and decision making in different situations where interactions are performed at multiple levels towards collective goals. One direction adds details to organizational models from abstract (business values) to concrete (operational details). The other direction identifies the multiple application environments (i.e., contexts) of an organizational model and elaborates these into sets of contextual specifications according to the specific requirements of the refined contexts. The two directions together provide a contextual link between organizational values and implementation details. Figure 1 shows the modeling framework of OperA+ which consists of four abstraction levels.

At the strategy level, the top-level objective of a system refers to an application domain determined by relevant situational variables which concern but do not restrict to individuality, activity, location, time, and relation [11]. Such situational variables are mainly from two sources, i.e., external condition variables and agent feedback. A *context* is defined as a specific subset of the Cartesian product of the variables’ value sets. They indicate system responses (states) to various situational conditions by enabling different roles and associated objectives, which decompose the top-level objective into a set of sub domains. Specifically, a sub domain will expand if more information about the variables is achieved, shown as the area enclosed by the dotted line.

For each context, a set of interrelated *roles* representing different responsibilities and requiring different capabilities are identified so that a solution for the top-level objectives in a specific context is specified. Given the role specifications,

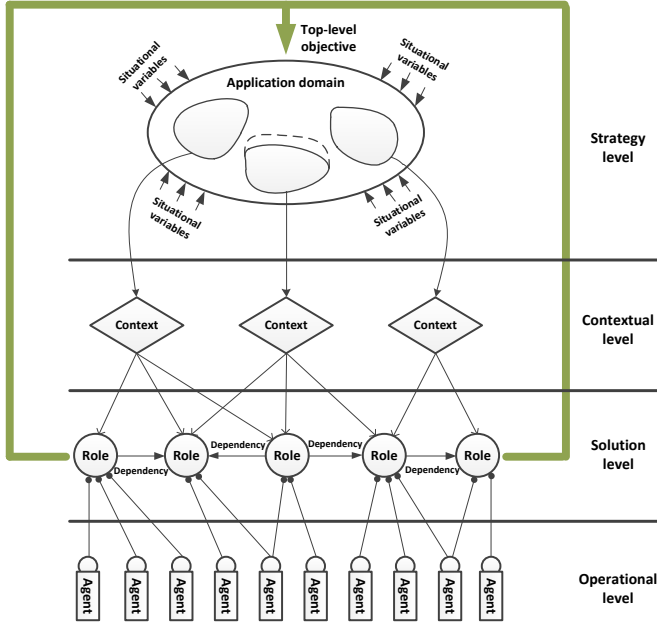


Figure 1. Modeling framework of OperA+

the solution level describes both the social relations and the interaction processes [8]. Moreover, an objective of a role at the solution level can again be decomposed into more specific components, which facilitates a modular way of organization modeling.

The purpose of the organizational design by OperA+ is to specify a mechanism to coordinate the behavior of the operational agents that enables the system to achieve its top-level objective in dynamic settings. Only by an exact match between organizational roles and operational agents, such a system behavior will be realized. Thus, at the operational level, eligible (external) agents will participate in the system and enact the roles that are assigned to them.

In OperA+, the organizational design of a system can be initiated at two abstraction levels. The first is the strategy level which specifies the top-level objective of the system and gradually identifies the required roles and their dependencies according to the expected functionalities of the system from abstract to concrete. The second is the operational level that depicts the operational details of a system from which related roles and contexts can be abstracted.

## V. CURRENT SITUATION ANALYSIS

### A. Operational Model

In the domain of train maintenance introduced in Section II, the operational model of the current railway system is depicted in Figure 2, which describes the involved entities and their information dependencies. The three main entities involved are the Train Operating Company (TOC), the

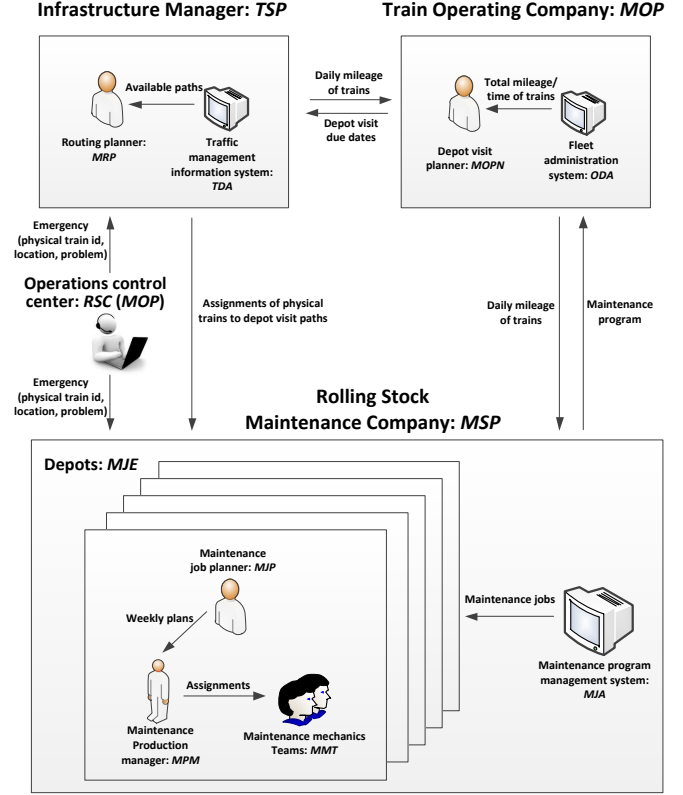


Figure 2. Operational model of the current railway system

Infrastructure Manager (IM) and the Rolling Stock Maintenance company (RSMC). TOC makes the time schedule for the train services provided, while IM controls the physical flow of trains across the railway network. To enable the transfer of a specific train to one of the RSMC depots, TOC requests IM to assign the physical train to a path planning which finally leads to the depot. The time horizon spans approximately one week. RSMC prescribes to TOC both the mileage and time interval that is applicable to a specific train type, defines the content of the maintenance program (i.e. the set of jobs to be executed), and manages the depots. Within these entities, different organizations of sub entities cooperate to realize corresponding functionalities.

### B. Role Identification

Using the OperA+ methodology, two kinds of roles are identified from the operational model of the current railway system, i.e., atomic roles, and composite roles (referring to lower-level organizations of roles), both of which are specified in terms of objectives, required capabilities, and dependencies. Figure 2 shows the link between the operational entities and the system roles, which is indicated by colons.

At the top level, we first consider the three main responsibilities captured by three roles. The Maintenance

Order Provider (MOP) has the objective to have the trains be maintained when the maintenance program or actual condition requires it. The Transfer Service Provider (TSP) has the objectives to provide information about the daily mileage of each operational train and to facilitate the transfer of trains to depots as requested by the MOP. Finally, the Maintenance Service Provider (MSP) has the objective to provide maintenance according to the maintenance program of trains sent to the depots, and to return them to operations as specified by the time table.

To achieve the objective, MOP enables an organization of three sub-roles, Operational Data Administrator (ODA), Maintenance Order Planner (MOPN) and Repair Service Coordinator (RSC). ODA has the objective to signal the need of specific trains for planned maintenance. It should have the capability to process the usage data delivered daily by TSP, to compare the cumulative data with the applicable thresholds, and to report on the remaining gaps. To guarantee safety, MOPN has the objective to let the trains be maintained before the critical mileage or time interval is exceeded and to assure fleet availability, while minimizing the life-cycle cost. MOPN requires the capabilities to generate, update, and communicate feasible depot visit plans, and to maximize the use of the trains based on the mileage and time interval allowed. RSC has the objective to have trains repaired when their condition requires immediate action. It should be able to identify the services required, to request for a suitable routing, and to tune the schedule with the depot involved.

TSP consists of two sub-roles, Traffic Data Administrator (TDA) and Maintenance Routing Planner (MRP). TDA has the objective to provide information about the availability of routing options, to keep track of the assignment of trains to planning paths, and to provide daily mileage data of trains. It should have the capability to store, update and query routing plans over time. MRP has the objective to facilitate the transfer of trains to depots as requested by MOP by making feasible routing plans for specific trains and by managing the plan execution.

MSP is enabled by a lower-level organization of the two sub-roles Maintenance Job Administrator (MJA) and Maintenance Job Executer (MJE). MJA has the objective to keep track of the maintenance jobs performed on trains and to signal the need for new jobs. Since MOP acts upon the usage data or urgent failures with the aim to have a train merely transferred to a depot, MJA has to identify the precise content of the work to be done, i.e. it has to specify the specific jobs to be scheduled. It requires the capabilities to keep track of the physical configuration of trains and the usage of the components, and to generate jobs regarding planned maintenance, component exchange, and modifications. MJE itself can be decomposed into three sub-roles, Maintenance Job Planner (MJP), Maintenance Production Manager (MPM), and Maintenance Mechanics Team

(MMT). MJP has the objective to assign feasible working plans to train depot visits. It should have the capability to group the jobs generated by MJA in such a way that the plan meets the time constraints and resource availability. MPM has the objective to assign jobs for execution to MMT, monitor the progress, and adjust the plan when necessary. It requires production management capabilities. Finally, MMT has the objective to actually perform the maintenance jobs as assigned to them, according to the work descriptions and safety regulations. Agents enacting this role should be trained and certified to do so.

### C. Context Abstraction

From the maintenance perspective, the top-level objective of the railway system is “efficient train maintenance”. Corresponding to this objective, a specific application domain concerning train maintenance is specified. In this application domain, the main situational variable is the states of the trains, which are of two kinds, i.e., with the threshold usage of mileage and time, and with unexpected failure. Corresponding to these two states, the railway system delivers two types of services representing its inner state. One is regular maintenance by planned depot visits. The other is emergent repair by unplanned depot visits.

Given the two states of the trains and the responding services delivered by the railway system, two sub domains that adhere to the top-level objective can be identified. Therefore, we abstract two refined contexts from the context of train maintenance  $c_0$ . The first context  $c_1$  is called planned maintenance which corresponds to the regular maintenance service for the trains with the threshold usage of mileage and time. The second context  $c_2$  is called unplanned maintenance which corresponds to the emergent repair service for the trains with unexpected failure. Since trains may require different maintenance services at one time, both contexts can be active simultaneously in the system. That is, the associated roles can coexist.

1) *Planned maintenance*: According to the maintenance program, trains are sent to depots for inspection and repair at regular intervals. TOC is responsible for checking the mileage and time usage of each train and delivering them to a depot according to the maintenance rules. Normally, the depot visit order is planned one week in advance so that IM can plan a physical route to transfer the train to a depot. At the same time, the depot will receive related information from IM about the expected arrival time of the train and from TOC the usage data of the train. When the train arrives at the depot, there are mainly two categories of maintenance work that will be done. The first category concerns ordinary inspection and repair maintenance. The second category includes additional work such as replacing main parts, small design changes and overhaul work, which usually cost more time and resources.

Therefore, the working process of the planned maintenance is in a sequential order, i.e., after the depot visit order is sent, the depot visit path can be scheduled, and finally the train specific maintenance jobs can be planned. The uncertainty mainly lies inside the maintenance plan since only after the train arrives at a depot the inspection results can be known and the detailed maintenance job schedule can be finalized.

2) *Unplanned maintenance*: Unfortunately, unexpected failures occur, resulting in requests for intermediate depot visits. IM has to plan an emergent route so that the malfunctioned train can be transferred to a nearby depot. When the train is assigned to a depot, the depot has to make a quick response and adapt its maintenance plan. In order to coordinate the actions towards this unexpected train failure, the Operations Control Center (OCC) is involved to communicate the emergent problems of the train to the IM and the RSMC.

Thus, the working process of the unplanned maintenance is in a concurrent order, i.e., after the problem announcement is sent, both depot visit path and maintenance jobs have to be scheduled in a quick response. Uncertainty lies in both depot visit path and repair plan.

#### D. Organization Model

According to the roles identified in Section V-B, the railway system in the domain of train maintenance is a result of multi-organizational interactions at multiple levels of abstraction. At an abstract level, three interrelated roles, i.e., MOP, TSP and MSP, build up an organization that achieves the overall objectives of the system. Given the refinement relation of the general train maintenance context, some of the roles at the abstract level are extended with finer-grained components that collaborate in such a way that the objectives of the roles at higher levels can be achieved in a specific context. From higher levels to lower levels, more detailed information is specified on how interactions between different entities are coordinated in a compositional manner. That is, a role in a higher level organization will be elaborated as a lower-level organization in which roles with finer-grained objectives and dependencies are specified according to the requirements of the specific context.

Therefore, given the refinement relation of the contexts, we build a multi-level and multi-context organization model of the railway system as shown in Figure 3. It can be seen that at an abstract level, the two contextual specifications share common roles and dependencies. However, since the two refined contexts have different requirements and working mechanisms, the lower-level specifications have different organizations of roles and dependencies. In this case, it is the organization of MOP that has different extensions since maintenance orders are from different sources in the two refined contexts.

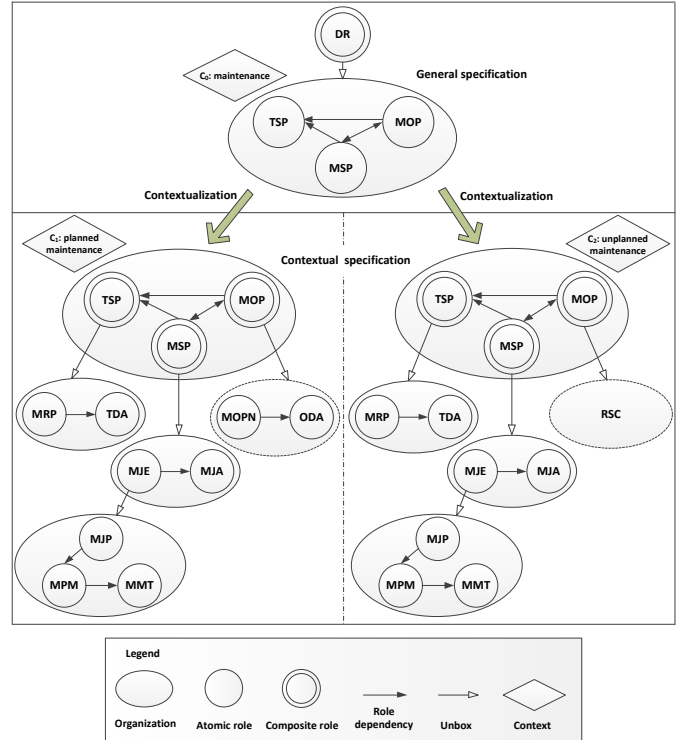


Figure 3. Organization model of the current railway system

Table I  
ARTIFICIAL AGENTS INTRODUCED IN THE NEW SYSTEM DESIGN

Artificial agents	Capabilities
Train Sensor (TS)	Detects or measures different status conditions of trains
Train Agent (TA)	Status analysis, depot visit planning, social behavior
Maintenance Routing Agent (MRA)	Automatic scheduling of depot visit path, social behavior
Maintenance Planning Agent (MPA)	Automatic scheduling of maintenance plan, social behavior
Depot Coordinating Agent (DCA)	Negotiating, conflict solving

## VI. FUTURE DESIGN

### A. Introducing Artificial Agents

With the development of artificial intelligence, not only can intelligent agents accomplish advanced functionalities but also assist decision making by their sophisticated communicating ability. Table I shows the detailed information of the artificial agents introduced in the new system design.

Each train will be installed with intelligent sensors (Train Sensor, TS) that can capture real-time status information of the train such as mileage, time, temperature, pressure, etc. With such information, the Train Agent (TA) that attached to the train can decide whether the train needs maintenance service or not. Moreover, TA is capable of social behavior so that it can communicate with other agents to improve

their performance. For example, TA may adjust its depot visit plan according to the observation of the workload of different depots.

Maintenance Routing Agent (MRA) and Maintenance Planning Agent (MPA) are respectively responsible for assisting human beings in making decisions on transfer paths and maintenance plans. To achieve these, MRA and MPA will communicate with TA to know their depot visit requirements and resource needs of the trains so that TSP and MSP can obtain a routing-feasible and resource-feasible schedule optimized in some sense.

Depot visit orders are distributed to several depots in the maintenance company. Since the resources of the depots are limited and their expertise is different, there might be conflicts when they receive maintenance orders from train agents. At this time, Depot Coordinating Agent (DCA) will be in charge of solving such conflicts by coordinating between MPAs and making optional solutions.

### B. Human and Artificial Agent Interaction

With the artificial agents, on the one hand, real-time information can be obtained and distributed through the whole system. On the other hand, interactions between human decision makers and artificial agents improve the communication between different organizations. Figure 4 shows the operational model of the future design in which interactions between human and artificial agents play an important role. It can be seen that train agents collect all the status information about trains and communicate to IM and RSMC about their depot visit requests and resource needs. Plan negotiation is finalized by a routing and job commitment.

### C. Context Transition

In the new system, due to the real-time monitoring ability of train sensors, and the advanced planning, communicating ability of train agents, depot visits of trains will be planned based on real needs and thus prevent unexpected failures to a large extent. Accordingly, a new context called adaptive maintenance marked as  $c_3$  emerges in the new system. It not only incorporates a part of the unplanned maintenance domain but also extends the planned maintenance domain, since more status information of trains can be achieved and therefore spreads the service of train maintenance to a wider domain. In this new context, train maintenance is based on real-time status information of trains from two sources. One is from intelligent train sensors and the other one is from the operational control center who collects emergent failure of trains and serves as a manual sensor.

In the new system, train maintenance will mainly be dealt with in the adaptive maintenance context which not only reduces the uncertainty of maintenance plans but also improves the flexibility of the scheduling task. However, the

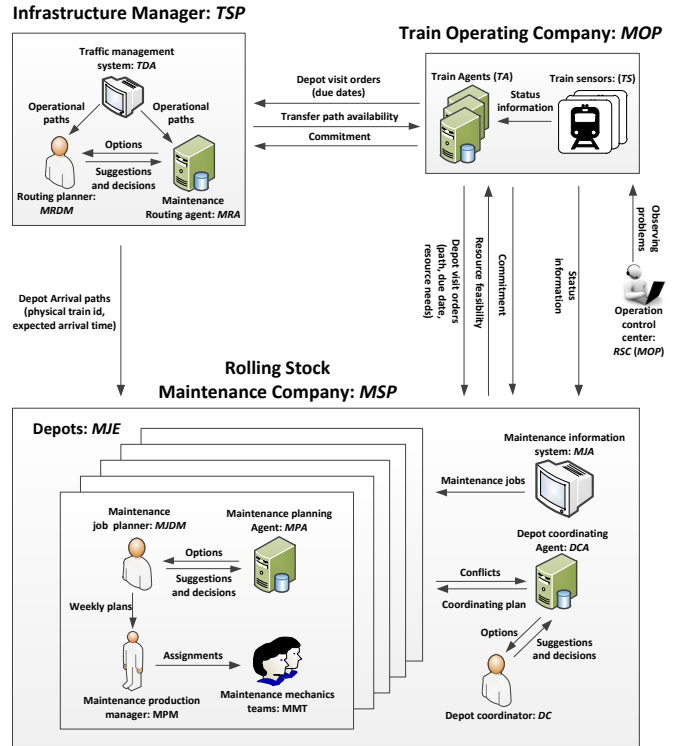


Figure 4. Operational model of the future railway system design

other two contexts still remain in the new design so that the railway system can gradually adapt to the changes.

### D. Organization Model Evolution

Given the context transition, the abstract organization model incorporates another contextual specification with respect to the adaptive maintenance context in the new design as shown in Figure 5. It can be seen that in this contextual specification more specific roles are added with the introduction of artificial agents. In MOP, TA takes the responsibility of planning maintenance orders based on the status information received from TS or RSC. Another important evolution is that the objective of MRP and MJP is distributed to two kinds of roles, i.e., Maintenance Routing Decision Maker (MRDM), Maintenance Job Decision Maker (MJDM) who make the final decisions, and MRA, MPA who support the process by communicating potential consequences of different plan options and by giving suggestions to solve conflicts and optimize plans. Besides, a role of Depot Coordinator (DC) is added in MSP so that conflicts between the actors of MJE can be managed with the help of DCA.

Comparing with the organization model of the current railway system, we can clearly see the evolution of the new organizational design in terms of the roles, dependencies and situational domains where interdependent roles are situated to achieve the overall objective. From the decision making

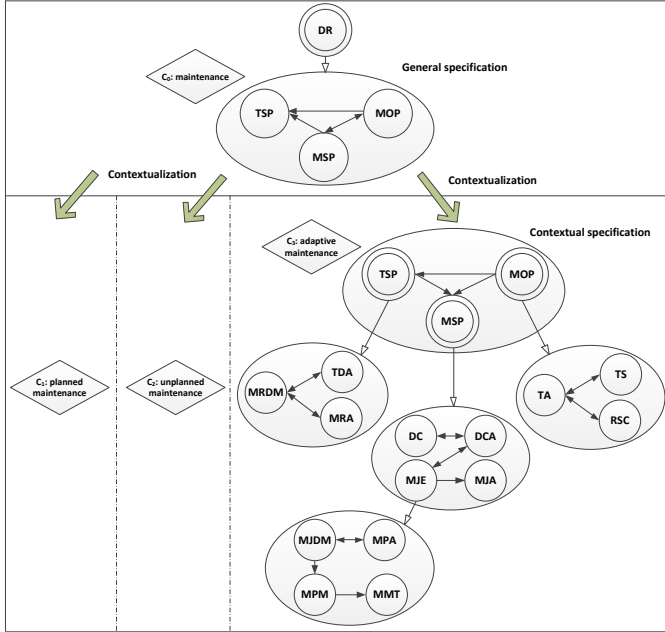


Figure 5. Organization model of the future railway system design

point of view, this model supports discussion and communication between people with different background knowledge, since it provides a comprehensive analysis of organizational interactions from abstract missions to concrete operations with respect to different contexts. On the other hand, with a structured description linked to operational agents, the model provides a basis for implementation of the new design.

#### E. Working process Adaptation

With the transition of context and the evolution of organizational model, the working process of the system also adapts to the new design, in such a way that before the depot visit order is sent, the depot visit path and the maintenance plan have already been scheduled. That is, train agents will firstly communicate the status information and resource needs of the trains to TSP and MSP. Only when TSP and MSP provide feasible options can train agents propose the final decision of depot visit order and make commitment to TSP and MSP.

### VII. DISCUSSION

From the analysis and modeling of the organizational evolution of the railway system, we can see that the features of OperA+ provide an efficient way of representing organizational interactions from three aspects.

(1) **Model reusability:** OperA+ models organizational interactions at multiple levels of abstraction. The higher levels model the systems in terms of coarser-grained components while the lower levels provide increasing details to the components designed and controlled by different entities, which facilitates modularity that can clearly describe distinctive

behavioral patterns with respect to different contexts. Corresponding to the railway system, we can see that the top-level organization is made up of three interrelated roles which remain the same during the life cycle of the organizational design while the lower-level organizations consist of more specific entities which adapt to different situational domains. This not only enables integrating different types of components in one model and provides necessary opacity since designers can decide how “deep” a model is constructed, but also ensures that components and groups of components can be easily reused at all levels of design.

(2) **Model flexibility:** An important feature of OperA+ is that it provides mechanisms for context sensitive design of organizational interactions. That is, it uses situational variables to abstract contexts that characterize conditional alternatives of achieving the system goals. These contexts make changes to the organization model at a contextual level, which in turn lead to the adaptation of other structures such as interaction structure (process model). As in the railway system, we abstract two refined contexts in the current system and a new context in the future design, each of which is determined by the value of situational variables and refers to a sub domain in the general maintenance domain. To this end, specific roles and role dependencies are identified for the requirements of each refined context. Such a way of structuring organizational interactions, not only adds meaningful building blocks in organizational models, but also improves model flexibility since current contexts can be easily adapted and new contexts can be easily integrated when changes occur.

(3) **Model evolution:** At an abstract level, organizational models link to missions and values that are fairly stable. Given context refinement relations, more specific roles are identified through lower-level organizations, which enables model adaptation to changing environments. As in the railways system, the top-level organizational model of both current system and future design stay the same, which represents the mission of the system in the train maintenance domain. With the introduction of artificial agents, the current system evolves to the future design by enabling a new context of adaptive maintenance in which both the social relations and interaction processes are adapted. In this way, we can clearly see the evolution of the organizational model from the commonalties and differences of the role/organization specifications and working processes in different contexts.

Furthermore, such a multi-level, multi-context organizational design reflects different levels of concerns from the perspectives of different stakeholders and makes it possible to achieve a clear and mutual understanding between designers and managers.

## VIII. RELATED WORK

In the research areas of agent organization modeling and agent-oriented software engineering, a number of frameworks relate to our work. AGR [5] describes an organization as a role-group structure imposed on agents and provides the basic foundational elements required in MASs to foster dynamic group formation and operation. Moise+ [4] is an organizational model that considers the structure, the functioning, and the deontic relation among them. MaSE methodology [12] is a full life cycle methodology for analyzing, designing, and developing heterogeneous MASs. Gaia [13] is an agent-oriented methodology based on the organizational concepts of roles, interactions, and organizations. Tropos [14] is a software development methodology, where concepts of agent paradigm are used along the whole software development process. INGENIAS [15] provides five meta-models (organization, environment, tasks/goals, agent and interaction) to guide the development of a MAS. ASPECS [16] provides a suite of refinement methods for modeling systems at different levels of details from requirements to code. Prometheus [17] is an AOSE methodology which covers a range of activities from requirements specification through to detailed design of agent internals.

Nevertheless, none of these approaches consider contextualization of organizational models at multiple levels of abstraction. That is, on the one hand, constructing a hierarchy of context refinement relations that reflect the dynamics of multi-organizational interactions, and on the other hand, with layers upon layers of detailed customization of organizational models from abstract values to operational details according to the requirements of the situated contexts.

## IX. CONCLUSION

In this paper, we apply the OperA+ methodology to an analysis and design study of organizational evolution of the Dutch railway system in the domain of train maintenance. It has been shown that organizational interactions in both current real life system and future design are systematically described. According to the value domains of the situational variables that influence the way the objective of train maintenance is achieved, three contexts are abstracted in the two systems, which makes the organizational models more meaningful and easier for adaptation.

More importantly, OperA+ enables flexible agent organizations that show adaptive behavior within dynamic environments and facilitates an evolutionary view on organizational design of socio-technical systems. These are reflected by the case study of the railway system, in which the dynamics of organizational interactions is fully captured and represented by the multi-level and multi-context modeling framework. With the explicit representation of contexts, organizational models are closely coupled with situational information, which makes it easier for actors to accommodate their actions in changing environments.

In future work, we intend to develop a control mechanism that dynamically assigns agents to roles and manages system behavior, while keeping record of the number of agents enacting a specific role. This will provide a basis to operationalize the organizational evolution of the railway system and implement agent-based simulations of the future system design.

## REFERENCES

- [1] K. H. V. Dam, *Capturing socio-technical systems with agent-based modelling*. TU Delft, 2009, PhD Thesis.
- [2] F. Dalpiaz, R. Ali, Y. Asnar, V. Bryl, and P. Giorgini, "Applying tropos to socio-technical system design and runtime configuration," in *Proceedings of the 9th Workshop on Objects and Agents (WOA08)*, 2008.
- [3] M. Menonides, B. Huisman, and V. Dignum, "Towards agent-based scenario development for strategic decision support," in *Agent-Oriented Information Systems IV*. Springer LNCS, 2008.
- [4] J. F. Hubner, J. S. Sichman, and O. Boissier, "Moise+: towards a structural, functional, and deontic model for mas organization," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, 2002.
- [5] J. Odell, M. Nodine, and R. Levy, "A metamodel for agents, roles, and groups," in *Agent-Oriented Software Engineering (AOSE)*, ser. LNCS 3382, 2005.
- [6] C. Schneeweiss, "Distributed decision making – a unified approach," *European Journal of Operational Research*, vol. 150, pp. 237–252, 2003.
- [7] V. Wiers, "The relationship between shop floor autonomy and aps implementation success: Evidence from two cases," *Production Planning and Control*, vol. 20, pp. 576–585, 2009.
- [8] V. Dignum, *A model for organizational interaction: based on agents, founded in logic*. Utrecht University, 2004.
- [9] J. Jiang, V. Dignum, Y. Tan, and S. Overbeek, "A context-aware inter-organizational collaboration model applied to international trade," in *Proc. EGOV*, 2011.
- [10] J. Jiang, V. Dignum, and Y. Tan, "An agent based inter-organizational collaboration framework: OperA+," in *COIN VII*. Springer LNAI, 2012.
- [11] A. Zimmermann, A. Lorenz, and R. Oppermann, "An operational definition of context," in *Modeling and Using Context*, ser. LNCS, 2007.
- [12] S. A. DeLoach, "The mase methodology," in *Methodologies and Software Engineering for Agent Systems: The Agent-oriented Software Engineering Handbook*, 2004.
- [13] F. Zambonelli, N. R. Jennings, and M. Wooldridge, "Multi-agent systems as computational organizations: The gaia methodology," in *Agent-oriented Methodologies*, 2005.
- [14] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An agent-oriented software development methodology," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 8, pp. 203–236, 2004.
- [15] J. Pavón, J. J. Gómez-Sanz, and R. Fuentes, "The ingenias methodology and tools," in *Agent-oriented Methodologies*, 2005.
- [16] M. Cossentino, N. Gaud, V. Hilaire, S. Galland, and A. Koukam, "Aspects: an agent-oriented software process for engineering complex systems," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 20, pp. 260–304, 2010.
- [17] L. Padgham and M. Winikoff, *Developing intelligent agent systems: a practical guide*. John Wiley and Sons, 2004.