Technology-independent Functionality Models for Road Traffic Systems

Boris Shishkov

IICREST
53 Ivan Susanin Str., 1618 Sofia, Bulgaria

Abstract. Improving road traffic management, by addressing concerns including safety, pollution, congestion, and travel time is important. Adequately specifying technology-independent functionality models for road traffic surveillance and management systems is the focus of this work, as well as the adequate specification of corresponding supportive information systems. Technology-independent functionality models are not only needed for better understanding the problems and discussing them of full value with both developers and users but also for establishing appropriate traceability that would allow updating the technology accordingly based on desired updates in the (real-life) business processes and regulations. The contribution of our work is thus in the following directions: (i) Methodological support that concerns the technology-independent functionality modeling for road traffic systems and (related to them) supportive information systems; (ii) Conceptualization of incorporating service-orientation and context-awareness in the functionality of specified systems.

1 Introduction

We observe overpopulation as well as increasing standards of living in many urban areas around the world and hence mobility (of people and goods) is becoming crucially important. In such areas, mobility is realized mainly by means of road transport and rail transport. Logically, road traffic is mostly contributing to transportation-related problems because of its being more complex (as flow) and less predictable (with very many human beings involved), compared to rail traffic. Further, road transport provides higher mobility of people and goods (compared to rail transport) with its door-to-door service [1]. We thus focus on road traffic as an important means of transport for people and goods. In particular, we are interested in finding solutions to the biggest concerns (in our view) related to road traffic, namely: safety, pollution, congestion, and travel time.

The development of traffic surveillance and management systems that go beyond the current state of the art and which can be made service-oriented and context-aware and which are capable of supporting both individual mobility and optimal network wide management is claimed to be crucial in this regard [24]. Adequately specifying technology-independent functionality models for such road traffic systems is the

Shishkov B. (2011).
Technology-independent Functionality Models for Road Traffic Systems.
In *Proceedings of the 1st International Workshop on Future Internet Applications for Traffic Surveillance and Management*, pages 3-18
DOI: 10.5220/0004472700030018

focus of this work as well as the adequate specification of corresponding supportive information systems. Technology-independent functionality models are not only needed for better understanding the problems and discussing them of full value with both developers and users but also for establishing appropriate traceability that would allow updating the technology accordingly based on desired updates in the (real-life) business processes and regulations [5]. Nevertheless, most current efforts in this direction have their underlying experimental applications which are specific to the type of sensors and other technological facilities they depend on; for example, there may be reference to image processing algorithms and related image feature extraction techniques that aim at capturing traffic dynamics along a road. Anyway, although such examples require specific technologies in their implementation, an abstraction could be made that generalizes the contemporary trends in traffic management approaches. Further, the traffic related parameters that need to be measured remain the same, i.e. traffic volume in cars per time unit, car speed, road lane occupancy, length of vehicle per lane, gap between successive cars etc, as they are essential in a following stage to feed the traffic prediction models that traffic management agencies apply in their daily routine operations [1]. The contribution of our work is thus in the following directions: (i) Methodological support that concerns the technology-independent functionality modeling for road traffic systems and (related to them) supportive information systems; (ii) Conceptualization of incorporating service-orientation and context-awareness in the functionality of specified systems.

The outline of the remaining of the current paper is as follows: Section 2 outlines our conceptual views concerning not only the stakeholders and processes related to road traffic but also the role of road traffic systems and (related to them) supportive information systems. Road traffic systems and their supportive information systems are further addressed in more detail in Section 3, where the specification of a road traffic system as a business system and the design of a supportive information system are discussed. Relating this to service-oriented and context-aware solutions is then addressed in Section 4. Further, Section 5 contains partial exemplification of the solution directions proposed in the previous sections. Section 6 contains a brief analysis of related work. Finally, Section 7 contains the conclusions.

2 Conceptual Views

Presenting our conceptual views concerning not only the stakeholders and processes related to road traffic but also the role of road traffic systems and (related to them) supportive information systems, is considered important because any adequate IT support should be consistent with stakeholders' activities and their facilitations [16].

In the first sub-section of the current section, our discussion addresses stakeholders and processes; in Sub-Section 2.2, we discuss the roles of road traffic systems and (related to them) supportive information systems.

2.1 Stakeholders and Processes

As already mentioned we will present our views on the main stakeholders and processes relevant to road traffic.

Considering roads, in general, one would logically observe different entities and processes, as shown on Figure 1. The most typical entity types are vehicles (for whom roads are built), human beings (whose pedestrian activities (as opposed to their activities as drivers of vehicles) are 'interrupted' by road traffic), and devices (supposed to help controlling road traffic); the most typical process types are driving (a vehicle), road crossing (by a human being), monitoring (by a human being and/or a device), and sending control commands (by a human being and/or a device). Further, some entity (e) /process (p) types can fall into several different categories:

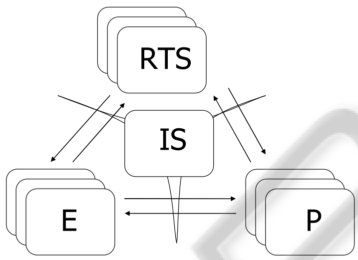

Fig. 1. Conceptual view on Entities (E), Processes (P), Road Traffic Systems (RTS) and (related to them) supportive Information Systems (IS).

(e-i) With regard to vehicles, they can be categorized based on their general purpose (cars, trucks, buses, and so on), based on their ‘institutionalization’ (private/municipal vehicle, police vehicle, ambulance, customs vehicle, and so on), and so on;

(e-ii) With regard to human beings, they can be categorized based on gender (male/female), based on age (child/grown-up), based on their function (pedestrian/driver/policeman/…), and so on;

(e-iii) With regard to devices, they can be categorized based on their general purpose (signal devices, such as traffic lights, monitoring devices, such as sensors, and so on) or based on their placement (fixed devices, for example attached to traffic lights, mobile devices, for example attached to cars, and so on), and so on;

(p-iv) With regard to driving, it can be categorized based on urgency (normal/emergency driving), based on lawfulness (lawful/illegal driving), and so on;

(p-v) With regard to road crossing, it can be categorized based on authority (pedestrian/policeman road crossing), based on lawfulness (lawful/illegal road crossing), and so on;

(p-vi) With regard to monitoring, it can be categorized based on the beneficiary (police/statistics/…), continuity (constant/upon-violation), and so on;

(p-vii) With regard to sending control commands, it can be categorized based on the location type (on-the-road/crossroad controlling), based on the signal type (traffic lights, speed instruction, and so on), and so on.

This is illustrated in Figure 2:

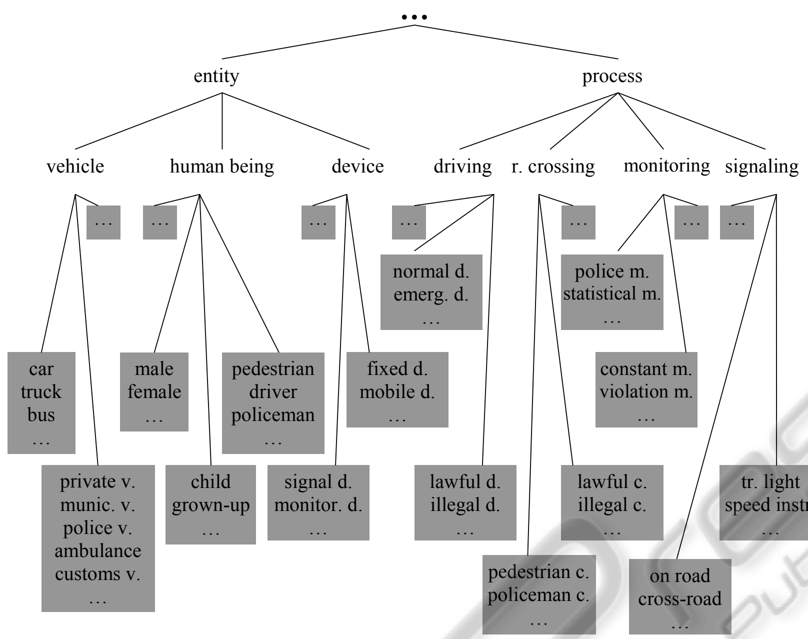

Fig. 2. View on main types and categories of entities and processes in road traffic.

Discussing so far just entities and processes, we have not discussed the roles of road traffic systems and (related to them) supportive information systems. We will do this in Sub-Section 2.2.

2.2 Road Traffic Systems and Information Systems

Road Traffic Systems and (related to them) supportive Information Systems, as exhibited in Figure 1, have great importance in managing road traffic, because such systems should guarantee that all involved stakeholders follow the regulations and are appropriately instructed [1].

Road Traffic Systems (or RTSs, for short) are referred to (in this paper) as all human beings, facilities and regulations realizing surveillance and management (including synchronization) with regard to stakeholders and processes that concern road traffic. These are typically the traffic lights, the instruction boards, the surveillance devices, and so on, as well as the corresponding regulations and supportive personnel.

Information Systems (or ISs, for short) are also needed, for supporting RTS and stakeholders with all that relates to information storage and information processing, and this mainly concerns sensor data – all raw data collected through sensors, needs to be processed and stored properly.

Hence, an RTS is responsible mainly for control and synchronization, and an IS – mainly for data storage and processing, as illustrated in Fig. 3:

Finally, the overall picture, as presented in Fig. 1, suggests that: there is link not only between the stakeholders (entities) involved in road traffic and corresponding

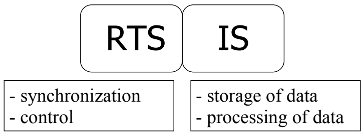

Fig. 3. Main responsibilities of Road Traffic Systems (RTS) and (related to them) supportive Information Systems (IS).

processes (for example, between a driver and a security check the driver is to pass through) but also between entities / processes and RTS (this is obvious since it is expected that RTSs establish synchronization and control); further, this all comes 'through' an IS involvement since all that takes place in managing road traffic is about (underlying) data flows.

It is thus concluded that both RTS and IS have great importance in road traffic management. Further, a technology-independent view on such crucial systems is essential, as already mentioned in the Introduction. Hence, we will especially address system design in the following section, from a technology-independent perspective, relating this particularly to the design of RTS and IS – for road traffic management.

3 System Design Views

We consider an RTS (Road Traffic System) and an IS (Information System) as a business system and an information system, respectively, following the terminology in [5]:

- A *system* is characterized by composition (it is composed of some entities), environment (there should be a delimitation between what is inside the system and what remains outside it), and structure (there are relations concerning the composition and the environment).

- A *business system* is composed of human beings collaborating among each other through actions which are driven by the goal of delivering products to entities belonging to the environment of the system; further, it may be that there are technical devices involved but it is the responsible human being 'behind' who 'counts', not the device itself.

- An *information system* is composed of human beings (often facilitated by ICT applications as well as technical and/or technological facilities) collaborating among each other driven by the goal of supporting informationally a corresponding business system.

Hence, we will discuss firstly the specification of an RTS and then information systems design.

3.1 Specifying an RTS

With regard to specifying a business system, in general, and an RTS in particular, we would follow principles of the Language-Action Perspective (LAP) and Organizational Semiotics [3] because of their proven strengths concerning technology-independent modeling of business systems [5]. Due to the limited scope of this paper, we will only briefly discuss some important issues; interested readers can find more about LAP in [6,23] and about Organizational Semiotics in [4].

The **first challenge** is specifying the RTS functionality in terms of actor roles and interactions that concern these roles as well as drawing the delimitation between what is inside the system and what remains outside it.

With regard to this challenge, we start by addressing *actor roles*. Here it is to be mentioned that in considering an action, we need to take into account the human responsibility behind rather than considering just the human/technical entity (partially) realizing the action [5]. Take for example a surveillance camera; taking it as a technical device on its own would bring little value because it is of crucial importance also what is the purpose behind - it is one thing if the camera is used to monitor traffic for statistical purposes and it is another thing if the camera is used to 'catch' violations. Take the latter example. In this case, whatever the technical facilitation, the responsible human being is of essential importance - the human being behind who is responsible for the inspection, including also the decision to impose sanction, for example. Thus, a typical actor role here would be VIOLATION INSPECTOR; fulfilling this role would include monitoring, violation identification, sanction determination, and so on. Each of these 'atomic' activities point to the responsibility of the violation inspector but it is possible that some of them are realized by a technical device (for example, the monitoring); still, this would be in the end responsibility of the violation inspector (a machine obviously cannot be kept responsible).

As *interactions* are concerned, they are considered always as occurring between two actors (we speak of an *actor* when we have an actor role fulfilled by a concrete entity) and if the setting is more complex, it is assumed that decomposition to sets of interactions involving exactly two actors is possible [6]. Further, we apply the *request-promise-state-accept* interaction pattern that is inspired by the Language-Action Perspective: according to this pattern, one of the actors would execute something (this execution would lead to the *production fact* characterizing the interaction), in response to a request by the other actor; once the production fact has occurred, this has to be announced to the requesting actor (state) and accepted by this actor (accept). This interaction pattern allows for modeling real-life business processes in a realistic way, as studied in [23]. For the sake of brevity, we will not discuss this further in the current paper, leaving interested readers to find more in the mentioned references.

Finally, it is of crucial importance to adequately draw *delimitation* between what (which actor roles) is inside the business system (in our case – the RTS) and what remains in its environment. We do not consider any particular guidelines for this, it just has to be done when identifying actor roles and interactions; then it could be that we identify: (i) interactions concerning only actor roles that are inside the business system; (ii) interactions concerning only actor roles that are in the environment of the business system; (iii) interactions 'crossing over' – one actor role is inside the business system and the other actor role is in the environment of the business system.

The **second challenge** is establishing the regulations underlying the interactions within an RTS and with regard to this, we consider as background Organizational Semiotics, in general, and NAN - the Norm Analysis Method [4], in particular.

Norms, which include formal and informal rules and regulations, define the dynamic conditions of the pattern of behavior existing in a community and govern how its members (agents) behave, think, make judgments and perceive the world.

Norms are developed through practical experiences of agents in a community, and in turn have functions of directing, coordinating and controlling their actions within the community. When modeling agents and their actions, which may reveal the repertoire of available behaviors of agents, norms will supply rationale for actions. Norms will also provide guidance for members to determine whether certain patterns of behavior are legal or acceptable within a given context. An individual member in the community, having learned the norms, will be able to use the knowledge to guide his or her actions, though he or she may decide to take either a norm-conforming or a norm-breaking action. When the norms of an organization are learned, it will be possible for one to expect and predict behavior and to collaborate with others in performing coordinated actions. Once the norms are understood, captured and represented in, for example, the form of deontic logic, it will serve as a basis for programming intelligent agents to perform many regular activities.

The long established classification of norms distinguishes between perceptual, evaluative, cognitive and behavioral norms; each governing human behavior from different aspects. However, in business process modeling, most rules and regulations fall into the category of behavioral norms. These norms prescribe what people must, may, and must not do, which are equivalent to three deontic operators of “obligation”, “permission”, and “prohibition”. Hence, the following format is considered suitable for specification of behavioral norms:

whenever <condition>
if <state>
then <agent>
is <deontic operator>
to <action>

We give also an example for a norm governing interest charges concerning a fine to be paid by a driver:

whenever an amount of outstanding fine
if more than 25 days after posting
then the driver
is obliged
to pay the interest

Thus, semiotic norms can be useful in elaborating actor roles and/or interactions. Nevertheless, even though identified when actor roles / interactions are considered, each norm needs to be analyzed further and positioned in a corresponding hierarchy of norms where higher level (‘constitutional’) norms govern lower level norms [3]. For the sake of brevity, we will not go in more detail regarding this.

In summary, business systems, such as RTS, are to be specified methodologically and applying the request-promise-state-accept LAP pattern can be helpful if actor roles and corresponding interactions are properly identified. Considering underlying regulations is as well necessary and can be usefully done through semiotic norms (in particular – behavioral norms). This all concerns the specification of a technology-independent model that may be further elaborated in terms of technical platforms and/or facilities/devices which may be (partially) fulfilling some actor roles; however, considering this is left beyond the scope of this paper, as already mentioned.

3.2 IS Design

Systems, such as RTS, are business systems in a sense that they are neither essentially automating something (even though there may be technical devices within them) nor they are there to informationally support particular processes. It is only that some tasks are performed by human beings and other tasks may be preformed by technical devices. Still, there are neither underlying IT architectures nor IT system requirements. Thus, with regard to RTS we only need to have a business process model, we need to identify actor roles and interactions, and also address the underlying regulations. This all holds also for the IS design but there are other issues as well to be taken into account when considering an information system. What is most important is IS's supportive role to what already exists, and this can be (as mentioned above) achieved through: (i) automation of processes or (ii) informational support with regard to processes. What we are addressing in the current paper, and also following what is outlined in Figure 1, is in line with (ii); hence, the role of the IS (as considered in the paper) is to support processes involving different stakeholders and RTS.

Hence, we are not reflecting 'real-life' level entities in IS entities but we are to specify new entities and new processes (for the IS). We firstly need to establish what is the desired support to be delivered (by the IS) and then go to the identification and specification of requirements.

Since requirements relate directly to the IS specification [2], they are to describe properly how the system-to-be should behave; application domain information is also needed as well as information on constraints on the system's operation. Considering all this, it could be distinguished between requirements that concern the IS fitting in its environment (*domain-imposed requirements*) and requirements that concern the desired (technical) characteristics of the system (*user-defined requirements*). For more information on requirements identification and specification, interested readers are referred to [5].

The requirements model is to be the basis for identifying actor roles and interactions, and also projecting important underlying regulations – as discussed in the previous sub-section.

We would need then the right-granularity functionality pieces, for furthering the information system design, and these can in principle correspond to any of the following: (i) a particular actor role can be reflected in a piece of functionality; (ii) an actor role can be reflected in several functionality pieces; (iii) several actor roles together can be reflected in a piece of functionality. This choice would depend on suitabilities with regard to the applied technology platform(s).

Such functionality pieces can be supported either by system components or by external IT services [18].

IT service solutions are more interesting for us not only because service-orientation becomes increasingly popular [9] but also because relying on (global) IT services is characterizing most current road traffic systems, as it is well known.

For this reason, we will continue the discussion in the following section, from a service perspective, suggesting possible benefits from applying distributed web services and (related to them) context-aware solutions. Then, in Section 5 we will provide partial exemplification of what has been proposed in the paper.

4 Towards Distributed and Adaptable IT Service Solutions for Traffic Management

As already mentioned, this section discusses service-orientation and adaptability (achieved through context awareness), on top of the system specification visions presented in the previous section.

4.1 Distributed Web Services

Most of the latest software technology innovations are centered around the service concept [8] and service-orientation in turn demands a heavy distribution [8,7]. We thus consider these together in the current sub-section, starting with a discussion on the service concept.

From an abstract point of view, a service represents a piece of well-defined functionality that is available at some network endpoint and is accessible via various transport protocols and specialization formats [21]. The functionalities provided by services cover a vast spectrum reaching from low level features like offering storage capabilities, over simple application functions like changing a customer address, to complex business processes like hiring a new employee [18].

The ability to create new ICT applications from existing services, independently on who provides these services, where they are provided, and how they are implemented, would mean usefully utilizing the service perspective in application development [16]. Such kind of application development is innovative not only because the application is not constructed from the scratch (actually, this is true also for component-based application development) but also because the development itself is fully centered around the desired end functionality to be consumed by users (this leads to service compositions and hence developers would no longer possess full control over all software components that play roles in delivering the application functionality). Hence, the application development task (as considered in general) might split into: (i) development of small software modules delivering generic adjustable services to whoever might be interested in using them, and (ii) composition of complex functionalities, by using available generic services. This all inspires new middleware developments also [17].

According to such latest middleware visions, on the basis of the specification of a needed functionality, the middleware would determine a composite service that would deliver the required functionality. This would be done either automatically or through a developer's intervention. This would obviously concern not only the application development but also its performance because services would require in most cases processing power of back-end server systems – hence both application creation and application execution would rely on support from the ‘Cloud’, especially when web services (services created and executed through the Web) are considered.

Furthermore, in order to be of actual use, such services would demand enabling technology standards and some recent views of Papazoglou [8] appear to be actual in this respect. Transportation protocols are to be mentioned firstly because logically, web services’ relying on a transportation protocol is crucial. Although not tied to any specific transportation protocol, web services build on ubiquitous Internet connectivity and infrastructure to ensure nearly universal reach and support. Hence, their mostly relying on HTTP (the connection protocol that is used by web services and browsers) and XML (a widely accepted format for all exchanging data and its corresponding semantics) looks logical. Having this as foundation, we have to briefly discuss three core web service standards, namely SOAP, WSDL, and UDDI: (i) SOAP (Simple Object Access Protocol) is a simple XML-based messaging protocol on which web services rely in exchanging among themselves information. SOAP implements a request/response model for communication between interacting web services. (ii) WSDL (Web Service Description Language) is a language that specifies the interface of a web service, providing to the requestors a description of the service in this way. (iii) UDDI (Universal Description, Discovery, and Integration) represents a public directory that not only provides the publication of online services but also facilitates their eventual discovery. And finally, as part of the web services composition, we need to introduce some orchestration defining their control flows [18], such as sequential, parallel, conditional, and so on, and to also determine complex processes that would usually span many parties. BPEL4WS (Business Process Execution Language for Web Services) can usefully support such composition activities [22]. Finally, as the collaboration among many parties (through their web services) is concerned, a common observable behavior (choreography) would often need to be defined. CDL4WS (Choreography Description Language for Web Services) can usefully support such collaboration descriptions.

4.2 Adaptable Context-aware Solutions

The utilization of a generic service for a specific user-related situation logically relates to acquiring knowledge on the context of the user and also exploiting this knowledge to provide the best possible service, which is labeled as context-awareness by Shishkov & Van Sinderen [19]. We hence claim that taking the end-user context into account is important in adequately delivering a service. Examples of end-user context are the location of the user, the user’s activity, the availability of the user, and so on. We do assume that the end-user is in different contexts over time, and as a consequence (s)he has changing preferences or needs with respect to services. Relating this to application creation and execution: the application is informed (for example through sensors) of the context (or of context changes), where the sensing is done as unobtrusively (and invisibly) for the end-user as possible. Sensors sample the user's environment and produce (primitive) context information, which is an approximation of the actual context, suitable for computer interpretation and processing. Higher level context information may be derived through inference and aggregation (using input from multiple sensors) before it is presented to applications which in turn can decide on the current context of the end-user and the corresponding service(s) that must be offered [20].

As according to previous studies, there are three challenges related to achieving adequate context aware support, namely: capturing, delivery, and response capacity [16]: (i) it appears to be non trivial establishing (through sensors) what change in the end user's situation has occurred – raw sensor data can easily be misinterpreted and also establishing adequate Quality-of-Context levels remains a challenge; (ii) even though deterministic service delivery approaches (when it is clear still at design time what behavior corresponds to what end user situation) should work OK, situations would occur which could not have been predicted at design time and thus require non-deterministic solutions – with respect to this, we doubt how much reliable such solutions would be, taking into account observed failures in many artificial-intelligence –related projects. We thus need better link to data analysis and more powerful run-time solution facilities. (iii) When delivering non-deterministic solutions, it may happen that such services are required that need unavailable resources; this would hence make it more difficult delivering timely the proper service(s) to the end-user.

5 Illustrating Example

As mentioned at the end of Section 3, we provide in the current section partial exemplification with regard to what has been proposed in this paper as solution directions. It needs to be emphasized however that the exemplification is just partial, due to the limited scope of the current paper, and the example is deliberately a 'toy example', adapted from another case study, since the purpose is just to briefly illustrate some important modeling steps discussed in the previous sections.

We take a monitoring-related example, putting things such that road traffic is monitored for 'catching' violations of traffic rules. There is a mediating system, called 'The Monitoring Mediator' (or 'Mediator', for short), this system is part of the Information System (IS), and is supporting stakeholders and the road traffic system, by informationally checking and advising on whether or not a violation has occurred. To make the example simpler, we abstract from the rest of the IS and also from the road traffic system and the stakeholders. We abstractly introduce the 'Customer' as the one who receives the advice that is delivered by the Mediator.

Thus, as it can be seen from Fig. 4, the Customer requests and the Mediator delivers an advice – if we put things as simple as that, we may have just these two actor roles, namely Customer and Mediator. We need however to 'peek' inside the Mediator, in order to be able to specify its functionality, and as seen from the figure, we go to another level of granularity where we can model actor roles which are internal with regard to the Mediator. Considering requirements and other issues (not mentioned for brevity), we arrive at a model where the following 4 actor roles are presented (these are 4 major roles and the model does not claim exhaustiveness): (i) Advisor (A) – delivering the advise to the Customer (C) (on behalf of the Mediator) on whether or not a violation has occurred; (ii) Match-maker (MM) – realizing match between what has been captured (through sensors) and analyzed (through reasoning), on one hand,

and what are the violations specifications (as it is according to the law), on the other hand; (iii) Request processor (R) – taking raw sensor data and realizing interpretation through reasoning techniques; (iv) Data searcher (D) – making a list of all relevant violation specifications as according to the law. Hence, the Mediator can deliver the advice (through A) to C, only after MM has delivered to A, which in turn needs to be preceded by the deliveries of R and D.

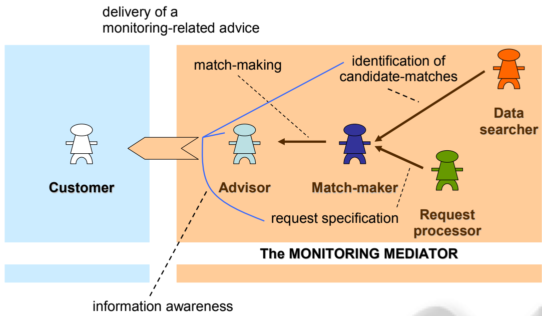

Fig. 4. General business process model for the monitoring mediation case.

We then go to the Entity model, as exhibited in Figure 5, where we have the dashed line delimiting what is inside the system from the rest, and we have there all actor roles discussed so far. Hence, the actor roles represent the entities in this model and there are some lines connecting two entities – this reveals the structure (how the entities are related to each other). These relations (representing the interactions) follow the 'deliveries' already discussed: A delivers to C, MM delivers to A, R delivers to MM, and D delivers to MM (the grey boxes indicate the one who is delivering).

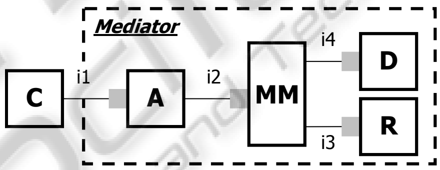

Fig. 5. Entity model for the monitoring mediation case.

The Entity model is then straightforwardly reflected in an Interaction model (Fig. 6) where the emphasis is on the interactions which are modeled using the ISDL technique [23]. As it can be seen, workflow details, such as sequence, synchronization, and so on, are reflected: as it can be seen: i3 and i4 should both be completed (synchronization) before i2 is completed, and i1 would be completed after i2 has been completed (sequence). On the top right corner of the figure is the higher-granularity-level model (as already discussed) where we look at all as one interaction (a delivery between the Mediator and the Customer).

Finally, we elaborate each interaction, by applying the request-promise-state-accept LAP pattern, as it is shown on Figure 7. The 4 grey ovals correspond to the 4 interactions' production facts and these are elaborated with corresponding coordination acts (such as request, promise, state, and accept); thus, the first 'line' from top to

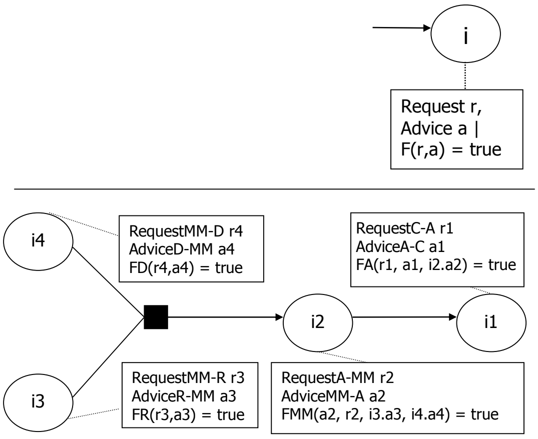

Fig. 6. Interaction model for the monitoring mediation case.

bottom is to be read as follows: Start, request by C, promise by A, production fact, state by A, accept by C, and so on. There are decision points – allowing not only for promise but also for decline (d), and these in turn allow for failure scenarios as well. We are not going to explain the model in more detail – interested readers can find more information on LAP in the references already mentioned.

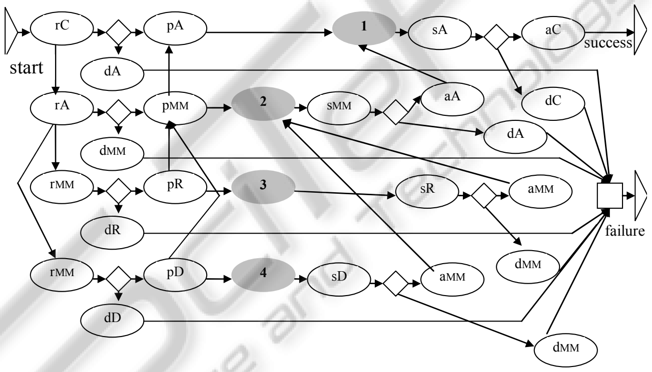

Fig. 7. Service-level model for the monitoring mediation case.

This model is considered as a 'service-level' model since on its basis one could usefully derive services.

Deriving service specifications, based on such a model, is nevertheless not trivial and requires resolving issues, such as tight coupling (possibly through introducing an orchestrating entity) and context-awareness, for instance, which would mean introducing new entities and interactions. Still, this way of progressing from very general real-life information to a well-structured and theory-rooted model is claimed to be beneficial. Such a model can be elaborated further, by adding semiotic norms.

However, for the sake of brevity, we are not going in further details with regard to this case. The purpose of this section (as also mentioned before) was to just give an impression on how part of what has been proposed can be realized through particular modeling techniques.

6 Related Work

Discussing road traffic, one could point to numerous relevant research activities, R&D projects, and initiatives. We do not claim exhaustiveness however in our brief analysis of related work, for two reasons: (i) the work reported in this paper is part of the FIATS-M initiative [1] and we find it most important to appropriately align to the works of other FIATS-M participants; (ii) for the sake of brevity, we cannot go in much detail in the current section, and we thus focus only on analyzing related work limited to the FIATS-M initiative, in general, and the work presented at FIATS-M'11, in particular.

The work presented by Brahmananda Sapkota [9] focuses on context-aware global IT services that interact among each other in delivering support to drivers and other stakeholders involved in road traffic. This work can be complementary with regard to our work since the focus there is mainly on utilizing services while in our work we address their specification rooted in methodological information system development. The work presented by Bjorn Kijl [10] focuses on assessing viability of the entire road traffic system both from a business sustainability perspective and an innovation perspective. This could link to our work since such an input may be relevant with regard to the technology-independent road traffic models that need constant analysis and updating, and this needs to be in turn reflected in the technology evolution. The work presented by Thomas Jackson [11] focuses on predicting situations on the road based on establishing a pattern whose occurrence has preceded the situation. We can relate to this in our work, by incorporating such 'mechanisms' at design time; then the run time support would adequately fit in the overall system functionality. The work presented by Apostolos Kotsialos [12] focuses on Internet based network control and this work's relation to our work is two-fold: (i) the network management and control are relevant to the functionalities of both the road traffic system and its supportive information system (as considered in our work); (ii) any Internet-based support should be realized through web services (addressed in our work). The work presented by Tom Thomas [13] focuses on monitoring and analyzing road traffic, for using this in possible re-designs. This again relates to our technology-independent models that may incorporate such 'mechanisms' at design time, for the sake of better run time support in the end. The work presented by Maria Virvou [14] focuses on user monitoring and modeling, in general, and on considering the emotional states of drivers, in particular; this is done in order to trigger such kind of support to a driver that would stimulate his or her good emotional state while on the road. If this would be applied in the context of FIATS-M, then again, it needs to be incorporates in the system specification, still at design time, and this relates the work on user monitoring and modeling to our work. Finally, the work presented by Jos Vrancken [15] focuses on network management and flows optimization for road traffic, considering not only the traffic inside the network under consideration but also on its border – different criteria may be applied in prioritizing the desired effects of traffic management.

Again, this is essential for supporting road traffic and needs to be incorporated in the systems functionalities at design time; this is how this work relates to our work.

7 Conclusions

In this paper, we have discussed the design of business/information systems and related to this specification of (context-aware) IT services, putting all this in the perspective of road traffic management, providing as basis conceptual views on road traffic. We have emphasized on the importance of properly specifying business/information systems in a technology-independent way, as a guarantee for the adequacy of the technology system to be developed on top of that, and we have provided methodological guidelines accordingly. Finally, we have partially exemplified our views.

Hence, the contribution of this paper is two-fold: (i) our having established and justified the role of technology-independent system modeling, especially with regard to the technical and technological facilitation of road traffic. (ii) the proposed solution directions and provided partial methodological guidelines (and exemplification) regarding technology-independent business/information system modeling for road traffic.

As further research, we plan projecting this work to a lower level, establishing adequate relations between technology-independent models (as focused in the current work) and technology-specific solutions.

References

1. FIATS-M (Home), http://www.fiats-m.org
2. Software Engineering Institute (Home), http://www.sei.cmu.edu
3. Shishkov, B., Dietz, J. L. G., Liu, K.: Bridging the Language-Action Perspective and Organizational Semiotics in SDBC. In ICEIS'06, 8th Int. Conf. on Enterpr. Inf. Systems (2006)
4. Liu, K.: Semiotics in Information Systems Engineering. Cambridge University Press, Cambridge (2000)
5. Shishkov, B.: Software Specification Based on Re-usable Business Components. Delft, The Netherlands: Sieca Repro (2005)
6. Dietz, J.: Enterprise Ontology, Theory and Methodology. Springer-Verlag, Berlin Heidelberg (2006)
7. Object Management Group (OMG): MDA – Guide, V1.0.1, omg/03-06-01 (2003)
8. Papazoglou, M. P.: Web Services: Principles and Technology. Pearson Pr. Hall (2008)
9. Sapkota, B. and Van Sinderen, M. J.: An Adaptive Service Platform for Traffic Management and Surveillance. In: FIATS-M'11, 1st International Workshop on Future Internet Applications for Traffic Surveillance and Management. SciTePress (2011)
10. Kijl, B., Nieuwenhuis, L. J. M., Meertens, L. O., Iacob, M. E.: Assessing the Viability of Service Innovations – a Structured Business Modeling Approach. In: FIATS-M'11, 1st International Workshop on Future Internet Applications for Traffic Surveillance and Management. SciTePress (2011)

11. Hodge, V.J., Jackson, T., Austin, J.: Intelligent Decision Support using Pattern Matching. In: FIATS-M'11, 1st International Workshop on Future Internet Applications for Traffic Surveillance and Management. SciTePress (2011)
12. Kotsialos, A.: Perspectives of Internet Based Road Network Traffic Flow Modelling and Control. In: FIATS-M'11, 1st International Workshop on Future Internet Applications for Traffic Surveillance and Management. SciTePress (2011)
13. Thomas, T. and Veensta, S.: Network Monitoring and Personalized Traffic Control: A Starting Point Based on Experiences from the Municipality of Enschede in The Netherlands. In: FIATS-M'11, 1st International Workshop on Future Internet Applications for Traffic Surveillance and Management. SciTePress (2011)
14. Virvou, M.: User Modelling and Emotion Recognition of Drivers through a Multi-Modal GPS Interface. In: FIATS-M'11, 1st International Workshop on Future Internet Applications for Traffic Surveillance and Management. SciTePress (2011)
15. Vrancken, J.: The Future of Road Traffic Management, Moving from Local to Network-wide, From Reactive to Proactive. In: FIATS-M'11, 1st International Workshop on Future Internet Applications for Traffic Surveillance and Management. SciTePress (2011)
16. Shishkov. B. and Van Sinderen, M. J.: Service-Oriented Coordination Platform for Technology-Enhanced Learning. In: I-WEST'09, 3rd International Workshop on Enterprise Systems and Technology. INSTICC Press (2009)
17. Leymann, F.: Combining Web Services and the Grid: Towards Adaptive Enterprise Applications. CAiSE Workshops (2005)
18. Van Sinderen, M. J.: From Service-Oriented Architecture to Service-Oriented Enterprise. In: I-WEST'09, 3rd International Workshop on Enterprise Systems and Technology (2009)
19. Shishkov. B. and Van Sinderen, M. J.: On the Design of Context-Aware Applications. In: IWEST'08, 2nd Int. Workshop on Enterprise Systems and Technology. INSTICC Press (2008)
20. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services, Concepts, Architectures and Applications. Springer-Verlag, Berlin-Heidelberg (2004)
21. Bosworth, A.: Developing Web Services. In: 17th Int. Conference on Data Engineering (2001)
22. Pasley, J.: How BPEL and SOA are Changing Web Services Development. Internet Computing, IEEE (2005)
23. Shishkov, B., Van Sinderen, M.J., Quartel, D.A.C.: SOA-Driven Business-Software Alignment. In: ICEBE'06, 2nd Int. Conference on e-Business Engineering. IEEE Computer Society (2006)
24. Vrancken, J. L. M., Schuppen, J. H. van, Soares, M. S., Ottenhof, F.: A Hierarchical Network Model for Road Traffic Control. In: Proceedings of the 2009 IEEE Int. Conference on Networking, Sensing and Control, Okayama, Japan (2009)