Road Transport Informatics - Conceptual Framework and Technological Components

Dr. Techn. Mads Nygård, The Norwegian Inst. of Techn.
Dr. Ing. Agnar Aamodt, The University of Trondheim

Sintef Delab
7034 Trondheim, Norway
Phone: +47 7 592979 Fax: +47 7 532586

Abstract

This paper addresses an advanced utilization of information technology within the road transport field. The main motivation for our investigation is the synergetic effect a genuine coupling of the two fields will create. A major objective is to build a conceptual framework for road transport informatics in order to establish a common basis for researchers from both fields. Another major objective is to identify and specify suitable mechanisms for implementation of key system components. The main paradigm for our work is always how mechanisms from the information technology field can optimally support needs within road transport.

1 Introduction

The paper has four separate parts, each one corresponding to a separate chapter. Chapter 2 discusses sets of requirements applicable to many RTI-systems. This chapter also identifies a set of questions appropriate for many RTI-services. Chapter 3 examines a generic architecture suitable for integration of many RTI-services into a common system. This chapter also raises a lot of interesting issues. Chapter 4 indicates how such an integrated service system could be exploited. The information processing is carried out in several steps, resulting in iterative processes being executed in a hierarchical fashion. Chapter 5 describes some key mechanisms for implementation of such an integrated service system. Our approach takes inspiration from how other advanced systems function, and based on this we come up with some new notions both suitable and necessary for our system realization.

2 System Requirements

This chapter will discuss sets of requirements applicable to many RTI-systems. It will also identify a set of questions appropriate for many RTI-services.

The *main motives* for utilization of IT-based services in the road traffic sector are:

- increased road safety
- more economically sound road management
- reduced environmental impact

Some desirable *additional effects* are:

- greater convenience and more information at hand for road users
- reduced costs for road authorities - and increased benefits for service providers

Current trends have already resulted in broad agreement on several sets of requirements for such services. See for example [4] and [5].

2.1 User Demands

User requirements appropriate for most IT-applications of this type are:

- *Availability*

  The system as a whole and each of the services should be accessible to any user at any time.
- *Reliability*

  The system as a whole and each of the services should have high tolerance against error situations and high capability towards error repair.
- *System Robustness*

  The system and the services should be adapted to the dynamic environments. For instance, one should be prepared for continuous changes in demand and climate.
- *User Friendliness*

  The system and the services should be simple and efficient to use. For instance, the man-machine interfaces must be advanced, and some decision support may be needed.
- *Information Integrity*

  The service information which is stored must be correct. Hence, each single data value must be a true mapping of the relationship that is represented, and the different data objects must correspond to consistent mappings.

0-7803-1235-X/93/$3.00 © 1993 IEEE

- *Information Security*

  The system information which is stored must be confidential. Hence, access to such data must be controlled according to authorization, and anonymity within such data must be guaranteed due to privacy reasons.

These user requirements will of course be emphasized differently by various IT-applications.

2.2 Functional Demands

Functional requirements imposed on most IT-systems of this kind imply an appropriate implementation of a suitable subset of services like the following:

- *Travel Information*

  An example is dispatching static tourist information.
- *Traffic Information*

  An example is dispatching dynamic route guidance information.
- *Traffic Control*

  This means optimization of traffic as a compound collection. Two examples are lights control of a road network within a city and entrance control to a road corridor between two cities.
- *Vehicle Control*

  This means registration and control of vehicles as separate elements. One example is intelligent cruise control through automatic control of inter-car distance.
- *Multimodal Management*

  This encompasses distribution of traffic demand on separate transport modes. It also includes administration of corresponding resources - such as parking facilities at mode changes, and booking of different services - such as hotels and restaurants.
- *Access Control* on arrival at different resources - and *Fee Collection* concerning use of different resources.
- *Optimization* of vehicle *Fleets* - such as lorries and taxies, and *Prioritization* of vehicle *Groups* - such as public transport.
- *Integrated Management* of vehicles, corresponding personnel, freight and corresponding documents.

What is really important is to *answer* the following *questions*:

- To what extent should we only register information about traffic conditions for analytical purposes - and to which degree should we also control traffic based on registered information?
- To what extent should we only send data to vehicles - and to which degree should we also allow drivers to act on received information with returned data?
- How should we make the separation between static and dynamic information to vehicles?
- How far should we go from surveillance via tutoring to policing of drivers?

There is *a unique trend* here:

- One sees more and more advanced choices along each of these dimensions.

This is probably due to the rather mature technological basis.

Which services to include and which services not to include, will necessarily vary from system to system.

3 System Architecture

This chapter will examine a generic architecture suitable for integration of many RTI-services into a common system. Such an integration must be based on standard principles for specification, design and implementation of open distributed systems. The different RTI-services may be associated with separate nodes in the open distributed system. This raises a lot of interesting issues.

3.1 System Integration

Current trends have already provided the motivation for integrated systems each encompassing several of the RTI-services mentioned in Chapter 2. See for example [3].

Such an integration must be based on standard principles for specification, design and implementation of open distributed systems.

The basic guidelines should be:

- *Modularity*

  One has to implement well-defined and clearly separable components. This relates specifically to operations and maintenance requirements.
- *Flexibility*

  One has to see to that different components may be plugged in and out corresponding to existing needs. This relates specifically to configuration and application requirements.
- *Compatibility*

  It must be ensured that both data communication and data storage are implemented according to adopted standards.
- *Interoperability*

  It must be ensured that the separate services work together as an integrated system.
- *Scalability*

  The system must at any time be adjustable to novel and changing needs. As road traffic is one of our most dynamic sectors, this is a crucial requirement.
- *Degradability*

  The system must be functioning regardless of which parts break down. As traffic management may never be suspended, there must be completely safe fall-back options.

3.2 System Components

The different RTI-services may be associated with separate nodes in the open distributed system.

The *data communication* among such system nodes must be based on standardized interfaces and protocols.

More generic issues that should be clarified are:

- *Heterogeneity*

  How different are the system components allowed to be (regarding both software and hardware)?

  This relates to how much adjustment the different service providers must expect in order to have their products integrated into the system.
- *Autonomy*

  How independent are the system components allowed to be with respect to local matters?

  This relates to how much coordination the different service providers must accept in order to have their products integrated into the system.
- *Real-Time Support*

  What response requirements must the system observe internally among the components and externally towards the users?

  Externally this concerns the time a user has to accept to wait until a request is answered. Internally this concerns the time the system may let elapse until a change somewhere in the system is known in the whole system.
- *Knowledge-Based Support*

  What advanced techniques must the system use internally among the components and externally towards the users to meet specific needs?

  Externally this concerns needs for advanced decision support to a user facing a very complex system. Internally this concerns needs for advanced representational and inferential support to the system handling a very large and dynamic amount of information.

More specific issues that should be clarified are:

- *Partially Distributed Data Storage*

  How should data fragmentation (i.e. dividing a logical item into several smaller objects), data replication (i.e. making several physical copies of a single object), and data allocation (i.e. placing each specific item/object/copy on separate nodes), be carried out?

  Efficiency and information security may, for example, induce very different choices.
- *Concurrency Control* and *Recovery*

  What correctness criteria should apply for the separate services and the integrated system with respect to parallel activities and erroneous behaviour?

  Efficiency and information integrity may, for example, induce very different choices.

There is *no unique trend* here:

- One still sees extremely diverse choices concerning each of these topics.

This is probably due to a rather immature application basis.

4 System Applications

This chapter will indicate how such an integrated service system could be exploited, given the requirements and architecture discussed and examined in Chapters 2 and 3. The information processing would be carried out in several steps. The output from some of these steps must be fed back as input to some other functions resulting in an iterative process. The execution of some of these functions may also take place more or less locally resulting in a hierarchical system.

4.1 Information Processing

Both *storage* and *transmission* of *data* as well as *processing* of *data* will take place in the system components.

To implement RTI-services one needs a basic instrumentation. This encompasses everything from special measuring, sensoring, registering and signalling units to general storage, transmission and processing elements. Current trends concerning technology as well as demands make it highly probable that such a system will be realized as a distributed system - i.e. as a set of independent but cooperating computers.

Such a system may be looked upon as a mirror image of the physical road network. The system has to be oriented towards generic administration and management of road traffic, and it must include specific services as mentioned in Chapter 2. Logically it may be considered as several functional networks. Regarding hardware, these will to a great extent be realized within a common system.

It will take some time to establish an appropriate infrastructure. Already existing IT-infrastructures must be taken into account, and a trial and error process is needed. An initial basis may be created from some adjacent areas - such as systems engineering and systems programming tools for the novel *broad-band telecommunication services.*

The information processing in such systems may briefly be described as follows:

1. Registration of traffic conditions, environmental data and user needs
2. Filtering and preprocessing of data
3. Continuous build up of knowledge
4. Systematic analysis of knowledge fragments / synthesis of action patterns
5. Selection and triggering of actions
6. Dispatching of control signals / educational information

4.2 Knowledge-Based Components

With regard to the above elements, the output from steps 4 and 5 must be fed back as input to step 3, resulting in an iterative process. This may be implemented with methods and techniques from the combined areas database systems and knowledge-based systems. A key word is *deductive databases* - interpreted in a wide sense as databases that may infer additional data based on existing data and application knowledge.

The very large and dynamic amount of information in such a real-time system makes it impossible to base all the alternative actions at any time on all the available data. One must continuously conduct consequence analyses and include probability evaluations of possible events. The system has to make provisions for the at any time most probable and risky events. Various action patterns must thus be prepared beforehand, and this requires knowledge-based methods and techniques for data representation and data inference.

To simplify slightly: the right information has to be at the right place at the right time. This is a requirement which applies at all times and in all matters. It is not possible to have all information stored at a fixed place and then initiate a transmission of it to the specific place where it is to be processed when this is needed. Actual information must instead float around in the distributed system in an intelligent way - i.e. it must continuously float towards the place where it may be needed.

An important function of knowledge-based components in RTI-systems will be to provide *active decision support* to users. Two aspects of active decision support are particularly relevant. One is to suggest actual solutions to a user's problem. This corresponds to the traditional role of knowledge-based systems, including expert systems. The other is to use application knowledge in order to filter and focus all the potentially relevant information that a user has to deal with. This corresponds to a different role than the former - since the system assists a user in his decision making process instead of concluding on a final result. Both aspects of decision support are relevant to RTI-systems. Here we will specifically emphasize the latter, the knowledge-based information filtering/focusing, and assume that the final decision making is conducted by the user. This information handling support ranges from derivation of non-explicit information at the database level to guiding a user by suggesting expectations from observed and measured data, checking the consistency of the user's input, evaluating the implications of the user's suggestions, and providing 'what-if' predictions at the user's request.

Knowledge-based methods rely on general domain knowledge in terms of various types of relationships among domain concepts. These relationships may include heuristic associations in the form of 'if-then' rules - as well as deeper knowledge such as hierarchical structures, cause-effect relationships, and functional roles. Deeper knowledge models allow the system more flexibility, tolerance and robustness in handling unexpected situations than heuristic association models do. However, they usually require more complex and time-consuming inference and reasoning techniques. Both types of knowledge are difficult to maintain. Hence, another key word is *machine learning* - and specifically the type of learning that addresses the knowledge maintenance problem by continuous learning from experience. A method that particularly emphasizes the reuse of previous experience - and the continuous capturing of new experience for future reuse - is case-based reasoning.

Case-based reasoning integrates problem solving with continuous learning from experience (see for example [1]). It solves problems by adaptive reasoning from previous situations. Here a problem refers to what the system regards as a problem - not necessarily to a real problem in the outside world. For example, given an initial description of a road traffic situation, a problem may be to check whether this situation is normal or abnormal. Case-based reasoning can be viewed as consisting of three major tasks:

1. Analyse the current problem and identify a matching case
2. Retrieve and adapt the solutions from this previous case to solve the new problem
3. Learn through retaining the relevant parts from this new case - and index the additional case for future retrieval

A set of previously solved problem cases are stored in a case base. Each case record has three main parts: a description of the situation in terms of a set of attributes, a set of conclusions for various problems related to this situation, and a justification for each conclusion. A justification may come from the system's testing of a solution, from a user's confirmation of the solution, or from explanations generated on the basis of general domain knowledge. Each new problem is solved by finding a previous case which has a problem description that is sufficiently similar to the new description. Many case-based reasoning techniques take a rather syntax-based approach to these processes. For example, matching two cases can be based on a surface similarity such as the ratio of identical descriptors to the number of total descriptors. This may be a useful approach in domains where little knowledge is available. However, there will be severe search problems when the case base grows larger. Combining such case-based reasoning with reasoning based on more general domain knowledge enables a matching of cases based on relevant descriptors with similar meaning within a given problem context. This results in more focused and tuned processes for case retrieval, case adaption and learning (see for example [6]).

With respect to a concrete realization, the execution of some of the filtering and preprocessing functions and their corresponding actions may take place more or less locally, resulting in a hierarchical system.

It is most important to consider legal and social aspects connected to such a road traffic system. Some key issues are:

- What types of data may be collected, stored and manipulated?
- How may the corresponding information and knowledge be applied?
- Who may have access to such information and knowledge?

5 System Realization

This chapter will describe some key mechanisms for implementation of such an integrated service system, given the requirements and architecture from Chapters 2 and 3. Our approach is inspired by how other advanced systems function, and based on this we have come up with some new notions both suitable and necessary for our system realization. Technologically this represents a combination of techniques from the database systems and knowledge-based systems fields. A generic result is the specification of a new structure for cooperative parallel processing, while a specific contribution is the investigation of new concepts and mechanisms for both *continuous behaviour* and *alert behaviour.* Our approach looks some years ahead, and due to the space limitations our description is far from complete.

Through our refinement of existing concepts for cooperating parallel processes, we need to realize both some novel basic mechanisms and some novel basic services. It is advantageous to start with the human brain - and the way it functions. There are similarities between what we need and what we observe there.

5.1 Human Functioning

The human being is an organic system. Organisms are tolerant - i.e. they adapt to their environments. They are robust - i.e. they continue to function even under unexpected conditions. This adaptability and robustness does not appear to be brought about by decisions made in a central facility - such as the nervous system. Influential theories on human physiology and human cognition suggest that it is rather the result of a very distributed control - with each part of the organism cooperating with the others in a natural manner.

The human nervous system performs a lot of functions similar to those of a process control system. The act of driving a car is a good example. Input from sensor units is turned into output to control units without much ado. However, in parallel with the car-driving there is a very different and continuous process going on. The car-driving individual considers the situation - reflecting more or less consciously on whether it should change its course of action. Here we need to be more precise about two terms: continuous and the situation.

Continuous behavior corresponds to constant awareness as seen from the act of driving the car. The considering process merely listens in on the car-driving - i.e. there is no interaction between the driving and the thinking about where to drive. The monitoring process is not in a client-server relationship with the driving process - in the sense that it is controlling every action. It has rather decided to be alert to the car-driving - ready to take actions to change it if and when appropriate. We are not claiming that this must be so in the human brain. We are merely making a model of what might be going on in order to illustrate the relationship between a low-level activity (driving the car) - and a high-level activity (considering where to drive) that can be said to be continuous with respect to the low-level one. It is continuous precisely because there are no discrete points of interaction between the two processes that are required by either process. If the high-level process were to be modelled as a sequence of discrete events, it would still be continuous in this sense. There is of course interaction between the two processes when the monitoring process decides, for example, to halt the car-driving. Unexpected conditions - such as a pedestrian suddenly blocking the road - are likely to cause such interaction.

Situations correspond to the sum of information that might be relevant to the considering process. Again we are postulating discrete processes performing actions - not as a claim that the human brain must function like this, but as a way of illustrating ideas about such processes. This information is not consistent, and it changes continually. The monitoring process sorts and considers all the information. Several hypotheses about the environment may be updated and evaluated concurrently - being refined and eventually accepted or rejected as more information becomes available.

The considering process represents several complex functions. On the one hand, it chooses the best course of action from moment to moment - for instance with respect to avoiding collisions with other cars. On a slightly more general level, it must choose a route to the fixed destination - and avoid getting lost following the chosen route. On an even more general level, it might be reflecting on whether it really wants to drive there. Each of these functions can be seen as a separate process which is continuous with respect to the processes below. Each process has a knowledge base that it keeps updating - including a model of the present situation. Each analyses the situation for its own purposes, and each makes decisions in its own domain. Each process recognizes given situations and has preprogrammed responses. And each process is monitored by a more general process that decides whether the process' set of responses is sufficient to meet the demands of any known situation. The monitoring process may update the process' set of responses if it manages to generate an appropriate response to an unknown situation. This is otherwise known as learning - or perhaps self-reprogramming. We may consider the system as a super operating system. In known situations, it allocates necessary knowledge as a resource to those processes that can generate correct actions (responses). In unknown situations, it allocates decision support and decision information to those processes that can generate new responses. Hence, the super operating system contains advanced elements from the expert systems field paired with traditional elements from the operating system field.

5.2 Extracted Concepts

Information availability is the most important need of each process in the process hierarchy above. The information must be of the right kind and of the right amount, and it must be accessible sufficiently quickly and reliably. Reason and carefulness are required by any process in selecting and evaluating information - even if we do not postulate any consciousness.

Alert behaviour involves being prepared for surprises. It implies a continuous monitoring of the situation - including analysis of probable future situations. The analysis calls for recognition of relevant state attributes in future situations. It is concurrent with evaluations and actions based on the present situation. Such concurrent monitoring requires that information which represents critical actions is sufficiently accessible. Various information is steadily put into more accessible storage as the situation evolves.

Context changes involve replacing large parts of available reactions - corresponding to immediately accessible information - based on a partial recognition of a given situation. This may be visualized with what happens when a child on the side-walk suddenly runs out in front of a moving car. Fast context changes and fast recovery of old contexts are very essential in many situations. That is likely to influence the information organization.

In such a processing system, each process is monitored by a process at a higher level. The high-level process does not interact with the low-level one except when situations arise that the low-level process is unable to handle. The recognition of such situations is done by the high-level process - i.e. it is not called upon by the low-level one. This makes the monitoring continuous as seen from the low-level process.

The structure of such a processing system is a process hierarchy where the information processed at each level changes continually. Processes at a given level alter their behaviour based on perceived changes in the situation. They monitor and control the processes below themselves, and they are being monitored and controlled by the processes above themselves. The processing system as such adapts on several levels. The high-level processes are reprogramming the low-level processes - thus causing changes in process structure, process definition, information structure and information flow at the level below. The high-level processes may be low-level ones with respect to other monitoring processes - thus being subject to the same kinds of dynamic changes.

6 Conclusion

The system requirements analysed in Chapter 2 place some constraints on what is necessary and what is sufficient in a system architecture. Likewise, the system architecture investigated in Chapter 3 places some constraints on which system applications are possible and on which system applications are desirable. Further, the system applications evaluated in Chapter 4 induce some conditions with respect to how best the system may be realized. Finally, the system realization considered in Chapter 5 induces some conditions with respect to which concepts and mechanisms are pertinent.

Acknowledgments

This work has been partially funded by the Royal Norwegian Council for Scientific and Industrial Research. The authors would hence like to express special gratitude to Olav Grimsbo and Torstein Garnaas for their support of [2], the report that this paper is partially based on.

Electronic Addresses

- Mads.Nygaard@DELAB.SINTEF.No
- Agnar.Aamodt@IFI.UNIT.No

References

[1] **Janet Kolodner:** "An Introduction to Case-Based Reasoning; Artificial Intelligence Review, Vol. 6, No. 1 - 1992.

[2] **Mads Nygård:** "Information Technology within Road Transport - Components and Systems"; SINTEF Technical Report STF40 A92060, Trondheim - 1992. (In Norwegian)

[3] **Mads Nygård:** "Automatic Traffic Control in Norway - Functional and Political Issues"; VNIS '92: 3rd International Conference on Vehicle Navigation & Information Systems, Oslo - 1992.

[4] **Kåre Rumar et. all:** "Evaluative Research of Road-Vehicle Communication Systems, Part I"; OECD Report DSTI/RTR/RVC(92)1, Paris - 1992.

[5] **Kåre Rumar et. all:** "Evaluative Research of Road-Vehicle Communication Systems, Part II"; OECD Report DSTI/RTR/RVC(92)2, Paris - 1992.

[6] **Agnar Aamodt:** "Knowledge Intensive Case-Based Reasoning and Sustained Learning"; ECAI-90: 9th European Conference on Artificial Intelligence, Stockholm - 1990.