Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Situation based Data Distribution in a Distributed Environment Model

Andreas Hermann and Stefan Lutz
Institute for Applied Research
University of Applied Sciences Ingolstadt
85049 Ingolstadt, Germany
{hermanna,lutz}@fh-ingolstadt.de

Abstract— **For Advanced Driver Assistance Systems (ADASs) knowledge of the environment plays a fundamental role. This also includes real-time data distribution and information about the current driving situation.**

A flexible situation model and a uniform situation analysis method are proposed to enable situation dependent information distribution, interpretation and fusion. This is a novel approach in the scope of automotive software. Waning from the proposed situation analysis an event-based programming model and a situation dependent data distribution is introduced.

The introduction of driving situations to optimize data distribution and interpretation enables the ADAS developer to focus on their key algorithms. The proposed framework provides the distribution of the data and the situation analysis.

I. Introduction

Comfort and safety applications rely on the knowledge of the car's environment and therefore environmental sensing plays a fundamental role in this field [1]. For Advanced Driver Assistance Systems (ADASs) not only the detection of the environment but also gathering additional information about the current driving situation of the vehicle is significant. Hence, Situation Awareness is a core part of future ADASs. For example, Adaptive Cruise Control (ACC) systems [2] already take risky situations like cutting-in vehicles into account [3]. In addition driver behaviour [4], collision prevention [5], classification [6] and co-pilot [7] are used to improve next generation of ADASs.

In the AUTOSAR consortium [8] leading automotive manufacturers and suppliers are working together to develop and establish an open industry standard for automotive architectures. Common problems like hardware abstraction and real-time communication environment are addressed by this standard. However, fundamental parts like sensor data distribution and the integration of driving situation awareness are not covered in AUTOSAR. To handle these major parts in ADASs development and deployment we propose the Distributed Environment Model (DEM) as an addition to the AUTOSAR specification. DEM meets ADASs requirements with a distributed framework which is able to support a Situation Model (SM), a Situation Analysis (SA) and hence Situation Awareness. DEM will provide a uniform driving situation for all ADASs and therefore allows the detachment of SA and ADASs resulting in reduced ADAS development complexity. Furthermore, this driving situation is used for the internal data distribution in DEM, which leads to a more efficient resource usage especially for communication capabilities. To improve the performance of data acquisition from a remote host, a proactive data distribution for the most likely next driving situation is proposed.

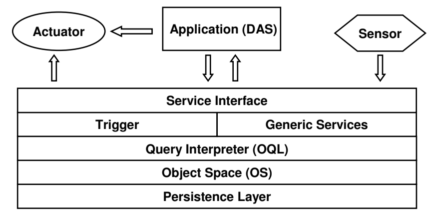


Fig. 1. Layered architecture of DEM

The remainder of this paper is organized as follows. In section II the DEM architecture is presented. The possible manoeuvres are described and ordered into two manoeuvre groups for the description of a driving situation in section III. The core part for the SM and the proposed flexible mapping method follows in section IV. The SA based on this map is described in section V. The event-based programming model is introduced in section VI. Based on the model a situation dependent data distribution is presented in section VII, section VIII shows first experimental results. Summary and conclusion can be found in section IX.

II. Distributed Environment Model (DEM)

Middleware in automotive environments is a weakly researched field. Common embedded object oriented architectures like RT CORBA [9], TAO [10] and ROFES [11] are based on ORB architectures. This is not suitable for automotive environments because of strict hardware limitations in this scope, although ORB architectures are built for remote object calls. In the automotive environment, data transfer can be considered the main requirement for a distributed system because operations on this data are usually processed on local machine to achieve hard real-time requirements. Therefore, we propose Distributed Environment Model (DEM) architecture to provide real-time driving situation recognition and a subsequent adaptive sensor data distribution.

DEM is a distributed embedded framework for sensor data fusion and interpretation in an automotive environment. As shown in Figure 1 DEM is organized as a layered architecture. The Service Interface can be used by Applications (e.g. ADASs) to access the internal functionality. Triggers are event-triggered functions to acquire, store, or modify data in

	MR 1	MR 2	MR 3	MR 4	MR 5	MR 6	MR 7	MR 8	MR 9
MG I	x	x			x	x	x	x	x
MG II		x	x	x					x

TABLE I

ASSOCIATION MATRIX FOR MANOEUVRES AND MANOEUVRE GROUPS

the DEM Object Space. Generic Services provide location independent data access. Furthermore, the communication with sensors and the wrapping of sensor data to DEM objects is implemented by generic services. The Object Space is a system wide container for sensor data. The objects stored in that Object Space are indexed to allow efficient query procession via the Query Interpreter Layer. This Layer is based on a subset of the Object Query Language (OQL) specification. The Persistent Layer is used for error logging which enables subsequent error diagnostics and backtracking.

The DEM core consists of several mechanisms to ensure a consistent representation of the current driving situation and an event-based, driving situation dependent data distribution.

III. Situations

In Tölle, possible actions and interactions for an artificial co-pilot are described [7]. The actions differ from the interactions because they can be accomplished without another traffic participant. Tölle identifies the following 9 distinct manoeuvres (MRs):

MR1 running up	**MR2** follow
MR3 approach	**MR4** pass
MR5 cross	**MR6** lane change
MR7 turning off	**MR8** turning back
MR9 parking	

The two manoeuvre groups (MGs) and the associated actions (MG I) and interactions (MG II) from [7] are listed in Table I. MR2 and MR9 are assigned to both groups.

Interactions (MG II) have the property $0 \leq |S_t^i| \leq |W_t|$ with S_t^i as the set of interactions i (MG II) and W_t as the set of vehicles in the environment at time t. The set of actions a (MG I) at time t has to fulfill $|S_t^a| \leq 1$. Hence MR2 and MR9 could violate the latter equation because of the duality which was made as a simplification in [7] to reduce the number of manoeuvres. To define an accurate set of actions and interactions the definitions of MR2 and MR9 have to be adapted for the situation scope. For MR2 [7] introduces a virtual traffic object which would lead along the lane. Hence, he does not differentiate between a lane and a normal traffic participant. Therefore MR2 has to be broken up again in the manoeuvres **MR21** follow lane and **MR22** follow vehicle. The former can now be categorized as an action and the latter as an interaction. For MR9 the interpretation as an interaction can be avoided by considering that a car can be parked while additionally interacting with other active traffic participants.

Now hybrid manoeuvres are eliminated and two distinct manoeuvre groups can be used for a situation model.

Aside from actions and interactions, the behaviour of the driver depends on the current traffic regulations for inner and outer city and on highways. In order to get a better situational representation, this has to be taken into account. As the vehicle can only be in one (composite) regulation at a given time t, $|S_t^r| = 1$ holds with S_t^r as the regulation S^r active at time t. This leads to an additional criterion. Altogether, there are three different aspects of a situation in an automotive environment from the driver's point of view:

1) Action (MG I)
2) Interactions (MG II)
3) Regulation

Hence for the current situation S_t a tuple of sets $\langle S_t^a, S_t^i, S_t^r \rangle$ is a feasible description, where S_t^i may be composed of more than one interaction (see above).

IV. Situation Model

The SA process relies on the integration of internal and external representations of a situation. Internal representations model the awareness of the process about itself, while external representations specify awareness about the environment [12]. A complete situation model must take into account the following tasks [13]:

1) Situation *perception* composed of Situation Element Acquisition, Common Referencing, Perception Origin Uncertainty Management, and Situation Element Perception Refinement as subtasks.
2) Situation *comprehension* composed of Situation Element Contextual Analysis, Situation Element Interpretation, and Situation Assessment as subtasks.
3) Situation *projection* composed of Situation Element Projection, Impact Assessment, Situation Monitoring, Situation Watch, and Process Refinement as subtasks.

The situation element acquisition implies all the object tracking and sensor data fusion procedures to acquire objects in the environment. This is a complex task and essential for the SA process.

In Table II the association with the DEM representation levels and the general JDL [14] levels 0 to 4 is shown. The first three DEM levels are single source data processing algorithms and can associated with the Source Pre-Processing level of the JDL Model. Object Refinement (level one) is handled in the track fusion and in the classification step. Situation Refinement (level two) is done in the mapping level of DEM. The last two levels of the JDL model are addressed in the strategy level. The DEM representation levels are a specialization of the JDL model. Note that the last level of the DEM representation represents the SA itself and the whole information processing will be situation aware.

One of the most important DEM levels is level 5 (Mapping). Maps can be divided into four classes [15]:

1) grid based [1]
2) feature based
3) topologic [16]
4) sequential monte carlo methods

The map described below fits best into the category *topologic* as it considers the logic links between the different map elements.

Level	Description	JDL	0	1	2	3	4
0	Sensor Data		x				
1	Ego Track		x				
2	Multi Target Tracking		x				
3	Track Fusion			x			
4	Classification			x	x		
5	Mapping				x		
6	Strategy					x	x

TABLE II
DEM REPRESENTATION LEVELS AND ASSOCIATION WITH JDL LEVELS AND ROY [13] SITUATION MODEL DEFINITION

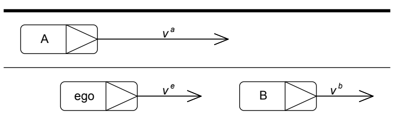


Fig. 2. ego vehicle with two interactions

To support the SA process the map has to represent the elements with logic and a spatial scope. In order to minimize the computational need of the mapping algorithm only the minimal assignment essential for the SA will be computed. For example, this means that only interactions with the ego vehicle as a participant will be considered. In the way the algorithm has a computational complexity of $O(n)$ instead of $O(n^2)$ for n traffic participants to acquire the relevant interactions. Figure 2 shows a simplified overview of a traffic configuration at time t with

- a **road** bordered by the thick horizontal lines,
- two **lanes** separated by the thin horizontal line,
- an **ego vehicle** with velocity v^e,
- and **vehicles *A* and *B*** with velocities v^a and v^b.

So $|W_t| = 3$ with $W_t = \{ego, A, B\}$. The vehicles are arranged in such a way that A is passing the ego vehicle and will pass B within a few seconds which means $v^a > v^e$ and $v^a > v^b$. Since an interaction requires two interacting participants the elements of the interactive situation subset S_t^i are triples $\langle s^i, w^1, w^2 \rangle$ with

- $s^i \in S^i = \{$ *follow vehicle, approach, pass* $\}$
- $w^1, w^2 \in W$
- and $w^1 \neq w^2$.

The triples for the example in Figure 2 are $\langle ''pass'', A, ego\rangle$ and $\langle ''approach'', ego, B\rangle$. Note that a possible third triple like $\langle ''pass'', A, B\rangle$ will not be considered as the *ego* vehicle is not part of this interaction.

For a reduction of complexity there is at most one $s_t^i \in S_t^i$ that has the property $\pi(s_t^i) = w$ for each $w \neq ego \in W$ with $\pi(\cdot)$ as a function that returns the interaction participant $w' \neq ego$. The problem with this statement is that the definition of the possible interaction types in S^i cannot take place exactly. This uncertainty has to be taken into account. Hence, the definition of an interaction has to be extended to represent this additional "information" of uncertainty. An interaction $s_t^i \in S_t^i$ is now a 4-tuple with $\langle s^i, w^1, w^2, p\rangle$ with p as the probability $prob(s_t^i)$. Since there should be only one s_t^i with $\pi(s_t^i) = w$, so

$$\sum_{\{s_t^i | \pi(s_t^i) = w\}} prob(s_t^i) \leq 1$$

holds. If an implicit interaction of type *no interaction* is considered and the interaction set S^i is extended to $\{S^{i\ old}, none\}$ the inequation results in an equation. For the given example, Figure 3 shows the state graph of possible interaction types a single pair of participants can attend, omitting *start* and *end* states for a better overview. Generally, for $n = |W_t| - 1$ as the number of traffic participants without the ego vehicle, there are n state graphs. The exact current state/interaction of a vehicle for such a state graph is uncertain.

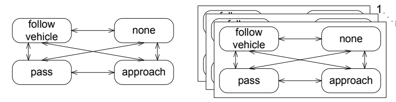


Fig. 3. state graph for interactions

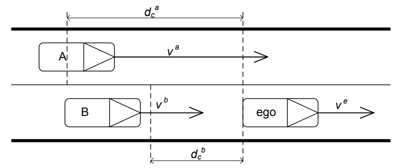


Fig. 4. scheme with critical distances

Before determining the probabilities for the different interaction types, the other participants (SA in section V) are first assigned to the lanes of the road relative to the ego vehicle. The current lane is l_0. Lanes to the left/right get a higher/lower index, respectively. In the configuration of Figure 2 the vehicle set of l_0 is $L_t^0 = \{ego, B\}$ and for l_1 it is $L_t^1 = \{A\}$. If the configuration is not as clear as in Figure 2 and a vehicle is between two lanes the lane sets do not have to be distinct and the association function $a(L_t^x, w)$ for the association value of w to L_t^x can be inside $[0..1]$. These fuzzy lane sets have the additional property $\sum_{L_t^x} a(L_t^x, w) = 1$, so a vehicle w cannot be over-associated.

After completion of the fuzzy lane sets, an additional ordering is made depending on the position in the lane itself. The mapping algorithm proposed here will cut the lanes into three parts around the ego vehicle with the sets ${}^xP_t^b, {}^xP_t^c$ and ${}^xP_t^f$ for the backward, centre, and forward parts of lane x. The elements in the set for the centre section are "too near" vehicles. This property "too near" can be calculated from the vehicle positions and their current derivation(s) in time. A classic attribute is the Time-To-Collision or some recent approach like the Time-To-Break [5].

Figure 4 shows another scenario with three participants driving at different velocities. The sets for the lane parts with more than 0 elements are ${}^1P_t^c = \{A\}$ and ${}^0P_t^b = \{B\}$. Since A is faster than B, the minimum distance for the centre section is bigger than referring to B. Hence participant A is in the set ${}^1P_t^c$ despite its bigger distance to the ego vehicle in comparison to participant B. The association to a section

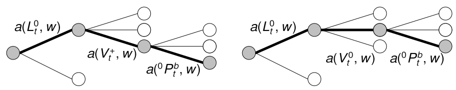


Fig. 5. decision trees for "approach" and "follow"

set is also fuzzy like the association to the lane set with an analogue constraint:

$$\sum_{^kP_t^e} a(^kP_t^e, w) = a(L_t^k, w)$$

with $e \in \{b, c, f\}$ and k as the lane number.

Additionally the speed of the vehicles is taken into account. Like in the description with a natural language the vehicles are ordered in three sets V_t^-, V_t^+ and V_t^0 for slower, faster and equally fast vehicles.

Therefore, the map M_t is a set of lanes L_t^k which consists of sets of parts $^kP_t^e$. The elements (the participants) can belong more or less likely (fuzzy) to one of these sets with the sum of all associations for one element being 1.

V. Situation Analysis

Using the set M_t as the situation model the probabilities of the interactions for each participant can be readily computed. Bayesian Networks [17], [18] could be used to calculate the interaction probabilities but these are computationally not feasible. Therefore, an algorithm with less computational need is proposed. The probability of each interaction s_t^i with $\pi(s_t^i) = w$ obeys a rule for each type contained in S^i. For the three elements of S^i the rules are:

- $follow(w) = a(L_t^0, w) \wedge a(V_t^0, w) \wedge a(^0P_t^b, w)$
- $approach(w) = a(L_t^0, w) \wedge a(V_t^+, w) \wedge a(^0P_t^b, w)$
- $pass(w) = (1 - a(L_t^0, w)) \wedge a(V_t^+, w) \wedge a(^0P_t^c, w)$
- $none(w) = 1 - [follow(w) + approach(w) + pass(w)]$

These rules exclude each other in a way that the constraints

$$follow(w) + approach(w) + pass(w) + none(w) = 1$$

and

$$follow(w) + approach(w) + pass(w) \leq 1$$

hold. The first constraint is obviously always fulfilled because of the definition of $none(w)$. The second constraint can be explained with a decision tree (Figure 5). If the sum of all edges to the child nodes is 1 and the traversal from a parent to a child node means a conjunction of the possibilities the sum of all possible leaves in the tree is 1. If the rules above are interpreted as traversals in such a tree and they do not include each other and are not identical the sum of the possibilities is less than or equal to 1.

It is not feasible for DEM to support ambiguous situations because its triggers can only be activated and deactivated. Thus the most probable interaction will enter the situation set S_t^i except for the artificial "$none$" interaction. As all the observed targets are modelled with physical parameters, the probabilities of the interactions cannot change from one extreme to another. To prevent oscillation of the system a threshold is introduced.

The current action S_t^a can be determined according to similar rules using the attributes velocity, steering angle, and GPS information.

The regulation aspect of the current situation S_t^r can also be determined from GPS data. Knowing the current road type is usually sufficient for this purpose.

VI. Event-based programming model

DEM uses an event-based programming model. Data consumers (e.g. trigger or applications) register their subscriptions $b = \langle d, e \rangle$, with *d* representing a unique data type and *e* as an event raised on this data type, at DEM. Hence for a trigger *g* a set B^g with $b_i^g \in B^g$ describes all *i* subscriptions of this trigger. Data producers (e.g. generic services which act as sensor drivers) register a set *O* of data types, which they are producing at runtime.

During initialization state *g* is inserted into a DEM instance. Afterwards its set B^g is distributed to all connected DEM instances. DEM instances parse the incoming set B^g whether events fulfilling $b_i^g \in B^g$ occur on local host, by reviewing every O_j of the *j* local data producers. Accordance's are stored in a local distribution table and result in a callback of *g* when b_i^g performed at runtime.

The DEM event-based programming model implements the observer pattern [19] in the context of a distributed system. Comparable work was done in Edwards *et al.* [20] by integrating Publisher/Subscriber Services into CORBA Component Model (CCM). Because DEM uses a different architecture as CORBA ORB this work could only be seen as proof of feasibility and initial point for future research.

Due to its programming model, DEM provides location independent data access over the network of ECUs in a vehicle. Hence, DEM supports reuse of ADASs software because of an explicit data distribution interface. Therefore, a location independent trigger deployment is possible. In addition, the event-based programming model is significantly more user friendly than commonly used interrupt-based programming techniques. Furthermore, event-based programming outperforms commonly used polling algorithms in the field of data acquisition by latency and CPU usage.

VII. Situation dependent data distribution

A main intention in automotive software engineering is to reduce the load of the bus system. To address this issue DEM proposes a combination of the event-based programming model (section VI) with the SA (section IV) to provide a situation dependent data distribution. Referring to section III the driving situation detected by SA could be interpreted as a cube with *regulation* as abscissa, *action* as ordinate and *interaction* as depth of the cube. This representation is mapped to a 32bit unsigned integer number to achieve a straightforward processing of the current driving situation in DEM.

To enable situation dependent data distribution, the event-based programming model is extended with driving situation information. So $B^{g\prime}$ with $b_i^{g\prime} \in B^{g\prime}$ and $b' = \langle d, e, c \rangle$ describe all subscriptions of *g* with corresponding driving situation c. At the moment g is inserted into a DEM $B^{g'}$ instead of B^g will be distributed among connected DEM instances. After initialization state every host maintains a distribution table T which contains all subscriptions (local or remote) for local produced data types and the assigned driving situations for these subscriptions.

With this situation dependent data distribution the amount of data which has to be transferred e.g. to a trigger can be reduced. Let h be the amount of data g receives in shorter cycles than driving situations change. In this case a continuous data stream to g is established. Therefore t_p is defined as the period g receives h.

In a system insensitive to driving situations N, g receives h over time span $t_{pN} = t$ with t is the uptime of N. Therefore the amount of transmitted data is defined as

$$\int_0^t h \ dt_{pN} \ \approx \ h \ \cdot \ t.$$

In a driving situation sensitive system D with uptime t the amount of transmitted data is defined as

$$\int h \ dt_{pD}, \ with \ t_{pD} = \sum_{i=0}^{n} t_{p_i D},$$

with $t_{p_i D}$ as the period a driving situation i is active in t and n as the number of active situations of g in t. So the amount of transmitted data to g in t can be approximated with $h \cdot t_{pD}$. From $t_{pN} = t$ and $t_{pD} \leq t$ follows that $t_{pN} \geq t_{pD}$. So in common case the aggregated data amount for g could be reduced by situation dependent data distribution. Due to this, network bandwidth could be saved to meet real-time requirements of automotive applications.

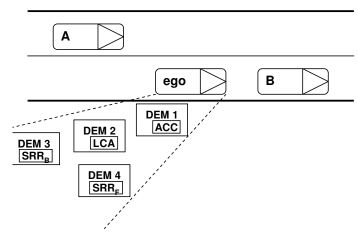
In SA, driving situations were changed based on probabilities. To avoid the system hunt between two situations a threshold for switching to a new driving situation is introduced. Hence, the driving situation change and the next driving situation can be predicted with a certain probability. This information can be utilized by DEM to start a proactive distribution of data, which trigger and applications will be needed for processing in the upcoming driving situation. The predicted driving situation is referred to as shadow situation.

In driving situation c a host H computes its distribution table T_c. If the situation recognition trigger inserts shadow situation c' into DEM, H calculates the proactive distribution table $T'_{c'}$ for predicted driving situation c'. $T'_{c'}$ is defined as

$$T'_{c'} \ := \ \{b_i | b_i \in T_{c'} \wedge b_i \notin T_c\}.$$

After the preparation of $T'_{c'}$, H starts to compute $b_i \in T'_{c'}$ with low priority to not affect real-time requirements of current driving situation tasks. Subscriptions in $T'_{c'}$ are not handled the same way as subscriptions in T_c. If timeliness can be assured DEM will start distributing data which is assigned to these situations, in the cycles before a situation change c to c' occurs. Proactive data distribution does not cause callbacks of trigger or applications, so external tasks are not influenced by this.

Therefore, after a situation change, this data is already available on local host and transmission time can be saved.



regulations	actions	interations
highway	follow street	no interactions
highway	follow street	approach
highway	lane change	pass
highway	follow street	pass
highway	lane change	no interactions
highway	follow street	no interactions
highway	follow street	approach
highway	lane change	approach & pass
highway	follow street	approach & pass
highway	follow street	pass
highway	follow street	no interactions
highway	lane change	no interactions

Fig. 6. experimental setup of situation dependent data distribution

This reduces latency after situation change and saves computing power respectively network usage during the sensitive period of a driving situation change.

VIII. Experimental results

Figure 6 shows the ego vehicle with four DEM instances. These DEMs control the ACC, the Lane Change Assistant (LCA) and convert the measurements of the short-range radars, in the front (SRR$_F$) and back (SRR$_B$) of the ego vehicle, to DEM objects. According to the definition of a driving situation, given in section III, typical driving situations on a highway were simulated and the transferred data volume between the DEM instances was measured. The simulated situations are listed in the table of Figure 6. The chart in Figure 7 shows the amount of data which has to be transferred between the DEMs in this scenario. It is shown that with driving situation dependent data distribution the transferred data volume was decreased significantly. In this scenario to about 47% of the amount without using situation dependent data distribution. Due to this, network bandwidth could be saved which assists to meet real-time requirements of automotive applications.

Fig. 7. Transmitted data in highway scenario of Figure 6

Table III shows measurements belonging to experimental setup in Figure 8, taken on a Intel Pentium 4 D 3,00 GHz with 1 GB RAM, Windows XP SP2 and Ethernet LAN connection. Real-Time environment was approached by assigning corresponding priorities to the threads used for DEM implementation. For example in the scenario below, real-time priority was assigned to trigger g, service u and DEM data distribution service. The proactive data distribution was handled in idle priority.

	scenario	mean time	DEMs	PCs
(1)	without PDD	1.04122 ms	2	1
(2)	with PDD	0.00950 ms	2	1
(3)	without PDD	1.93763 ms	2	2
(4)	with PDD	0.00951 ms	2	2

TABLE III

ACCESS TIME ON HISTORIES OF AN OBJECT USING/NON USING PROACTIVE DATA DISTRIBUTION (PDD)

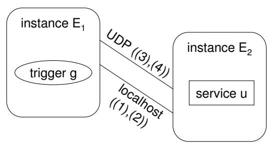


Fig. 8. experimental setup of proactive data distribution measurements

As shown in Figure 8, g was located on DEM instance E_1 and u on DEM instance E_2. g subscribes for tracks o_i and requires histories of o_i in active state. u produces o_i and E_2 take care for histories of this objects. For this benchmark, the time was taken which elapses until g receives history of a given o_i. For (1) and (3) in Table III proactive data distribution was turned off, for (2) and (4) turned on. The measured times represent average access times for the history of o_i. To avoid measurement errors the acquisition of o_i was repeated 1000 times and the determined mean times are shown in Table III. Workload was simulated by inserting other subscriptions and trigger into DEM instances to take care that measurements were taken under real life terms and conditions.

In case of localhost communication proactive distribution speeds up the traditional way of data acquisition by factor 109, in case of LAN communication by factor 203. These results show that proactive data distribution can reduce access times to histories of objects significantly. The results of this benchmark could be applied to automotive environments because

- the ratio $\frac{network\ bandwidth}{computing\ power}$ is similar
- prioritization of threads were made in a similar way
- concepts of DEM are independent from underlying hardware

By changing hardware configuration for this scenario only the factor between non-proactive data distribution and proactive data distribution will diversify. For example, when network bandwidth is increased significant, the factor will decrease. On the other hand, the factor will increase when computer power increases.

IX. Conclusion

In this paper, the problems of situation analysis and situation specific data distribution are addressed. First, the situations are defined and a situation model is developed. Based on this model a suitable situation analysis methodology is introduced. With the results of this analysis, an event based programming model for a situation dependent data distribution is proposed. Furthermore concepts for a shadow distribution which lowers latency and saves computing power and network load during the sensitive period of a driving situation change is presented. First experimental results show the potential of these novel concepts in automotive environment.

Future work will detail situation prediction and improvement of situation specific data distribution. This will include the results of ongoing tests performed on mobile robots with extensive sensor equipment and on a test vehicle in real traffic situations.

X. Acknowledgements

This work was supported by the FuE-Programm "Informations- und Kommunikationstechnik" of the Free State of Bavaria in the *Mobilsoft* project, the AUDI AG and Elektrobit.

References

[1] M. Skutek, D. T. Linzmeier, N. Appenrodt, and G. Wanielik, "A precrash system based on sensor data fusion of laser scanner and short range radars," in *FUSION 2005*, 2005.
[2] A. Gern, U. Franke, and P. Levi, "Robust vehicle tracking fusing radar and vision," in *Proceedings of the MFI*, 2001, pp. 323–328.
[3] I. Dagli, G. Breuel, H. Schittenhelm, and A. Schanz, "Cutting-in vehicle recognition for acc systems- towards feasible situation analysis methodologies," in *Intelligent Vehicles Symposium IEEE*, 2004.
[4] N. Tricot, B. Rajaonah, M.-P. Pacaux, and J.-C. Popieul, "Driver's behaviors and human-machine interactions characterization for the design of an advanced driving assistance system," in *Systems, Man and Cybernetics*, 2004.
[5] J. Hillenbrand, K. Kroschel, and V. Schmid, "Situation assessment algorithm for a collision prevention assistant," in *Proceedings of Intelligent Vehicles Symposium*, 2005, pp. 459–465.
[6] N. Kaempchen and K. C. J. Dietmayer, "IMM vehicle tracking for traffic jam situations on highways," in *Proceedings of the Seventh International Conference on Information Fusion*, 2004, pp. 868–875.
[7] W. Tölle, *Ein Fahrmanöverkonzept für einen maschinellen Kopiloten*. VDI Verlag, 1996.
[8] AUTOSAR GbR, *AUTOSAR Technical Overview V 2.0.1*, June 2006.
[9] Object Management Group, "Real-Time CORBA Specification Version 1.2," Object Management Group, 2005.
[10] D. C. Schmidt, D. L. Levine, and S. Mungee, "The design of the TAO real-time object request broker," *Computer Communication, Elsivier Science*, vol. 21, 1998.
[11] S. Lankes, "Konzeption und Umsetzung einer echtzeitfähigen Verteilungsplattform für eingebettete Systeme," Dissertation, RWTH Aachen, 2003.
[12] A.-L. Jousselme, P. Maupin, and Éloi Bossé, "Uncertainty in a situation analysis perspective," in *FUSION 2003*, 2003.
[13] J. Roy, "From data fusion to situation analysis," in *Proceedings of Fourth International Conference on Information Fusion*, 2001.
[14] D. L. Hall and J. Llinas, "An introduction to multi-sensor data fusion," in *IEEE Proceedings*, 1998, pp. 537–540.
[15] J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *International Journal of Robotics Research*, vol. 21, pp. 311–330, 2002.
[16] A. Belussi, B. Catania, and E. Bertino, "A reference framework for integrating multiple representations of geographical maps," in *GIS '03*, 2003.
[17] C. Howard and M. Stumptner, "Situation assessments unsing object oiented probabilistic relational models," in *FUSION 2005*, 2005, pp. 1489–1496.
[18] O. Bangso, M. J. Flores, and F. V. Jensen, "Plug & play object oriented bayesian networks," Department of Computer Science Aalborg University, Tech. Rep., 2004.
[19] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns*. Addison Wesley, March 1995.
[20] G. T. Edwards, D. C. Schmidt, and A. Gokhale, "Integrating publisher/subscriber services in component middleware for distributed real-time and embedded systems," in *Proceedings of the 42nd annual Southeast regional conference*, 2004.