A Decentralized Approach to Collaborative Driving Coordination

Simon Hallé, Julien Laumonier and Brahim Chaib-draa

Abstract— **Collaborative driving is an important subcomponent of Intelligent Transportation Systems ITS as it strives to create autonomous vehicles that are able to cooperate in order to navigate through urban traffic by using communications. In this paper, we address this problematic using a platoon of cars considered as a multiagent system. To do that, we propose a hierarchical architecture based on three layers (guidance, management, traffic control) which can be used to develop centralized platoons (where a head vehicle-agent coordinates other vehicle-agents by applying coordination rules) and decentralized platoons (where the platoon is considered as a team of vehicle-agents maintaining the platoon together). We propose the model of teamwork used in multiagent systems as a decentralized alternative to previous coordination centralized on the platoon's leader and outline its benefits using collaborative driving simulation scenarios.**

I. Introduction

Transport systems are suffering from traffic flow increasing at an unstoppable rate in every major cities. To address this problem, solutions using Intelligent Transportation Systems (ITS) infrastructure are growing in popularity, as they provide potential capacity improvements as high as 20 percent [1]. The main objectives of ITS include: reduce congestion and environmental impacts, enhance safety and comfort, reduce human stress, etc.

A very promising use of ITS is done by technologies such as Adaptive Cruise Control (ACC) [2], which uses sensors to maintain safe inter-vehicle distances between cars. A recent upgrade to this technology is available through Cooperative Adaptive Cruise Control (CACC), that benefits from a communication system to collaborate between vehicles' ACC. When CACCs are being used as a whole in vehicles driving on a same highway, coordination strategies can emerge between the different neighbor vehicles. These strategies are implemented within a collaborative driving system, which aims the formation of platoons of vehicles using a decentralized control system. Such platoons are coordinated using a leader-follower architecture [3], which centralizes the coordination on the leader, leaving few autonomy to the followers. As a new approach to this centralized coordination system, we aim to incorporate the multiagent vision to the platoon architecture and coordinate the vehicles through Teamwork models [4]. This approach incorporates autonomous agents that make use of the communication system in each vehicle, to coordinate each others in a decentralized platoon model.

In this paper, we address the coordination issue for a platoon of vehicles, by first describing the collaborative driving domain and the simulator used to represent this environment, in section II. Then, section III presents the hierarchical architecture we adopted as the decentralized driving system of automated vehicles. Section IV describes the different coordination strategies we implemented and tested in the previous simulator. Section V reports the preliminary results using the comparison of centralized to decentralized coordination approaches, using teamwork. Finally, section VI presents a discussion, followed by the conclusion.

This work was carried out as part of the Automobile of the 21st Century (Auto21) project supported by the Government of Canada through the Networks of Centres of Excellence (NCE)

Simon Hallé, Julien Laumonier and Brahim Chaib-draa are with the Département d'informatique et génie logiciel, Université Laval, Sainte-Foy, QC, Canada, G1K 7P4 {halle,jlaumoni,chaib}@iad.ift.ulaval.ca

II. Domain of Application

Collaborative driving is a research domain which aims to create automated vehicles that collaborate in order to navigate through traffic. In this sort of driving, one generally form *a platoon* [5], that is a group of vehicles whose actions on the road are coordinated by the means of communication. The first vehicle of a platoon is called the platoon leader and its role is to manage the platoon and guide it on the road. Our work comes as part of the Auto21 project [6] studying the automobile of the 21st century within three levels of system functionality. The first two levels will focus on the longitudinal control (first) and the lateral control (second) of the vehicle, in a platoon lead by a human driver, while the third level will consider every vehicles as fully autonomous, relying on an advanced road and telematic infrastructure. This article will focus on the first level and its usage within the different platoon driving tasks.

A. Collaborative Driving Simulator

The environment in which our vehicle coordination system has been tested is an Automated Highway System (AHS) simulator [7], [8] which recreates the highway environment as well as the vehicles equipped with all of the required technological components. The simulated vehicles' model includes longitudinal and lateral vehicle dynamics, wheel model dynamics, engine dynamics, torque converter model, automatic gear shifting and throttle/brake actuators. The engine and transmission torque converter and differential were translated from a model developed under MATLAB/SIMULINK by our partners at Sherbrooke University [9]. The wheel model and vehicle's lateral and longitudinal dynamics were developed using the theory on wheel slip, tyre side slip angle and friction co-efficients applied to a single-track model, as well as the theory on the chassis' motions models, described in [10]. The simulated sensors were developed using the 3D engine of JAVA 3D^{TM}, and for the current test, each vehicle are equipped with a vehicle-based laser sensor for a low-level, inter-vehicle navigation. This sensor provides information on the front object's (a vehicle) distance and difference of velocity, for distances up to 100 m, using an abstract model of laser. The second type of sensor, used for high-level navigation, is a Global Positioning System (GPS), which gives real-time information on the vehicle's position (latitude, longitude), mapped in a two dimensions system. Finally, we simulated a radio transmitter/receiver onboard each vehicle for two ways point to multipoint communications. This communication model includes adjustable delay, throughput and protocol for different communication devices. We will not go further on a detailed representation of the simulator's components, as it is out of scope for this paper.

B. Simulated Driving Scenarios

Using the precedent simulator, collaborative driving scenarios involving the coordination of a platoon of vehicles have been defined to compare the different coordination approaches. The main problematic, resolved as part of our studies, is the maintenance of the platoon formation, so the two scenarios we will focus on will be the two main disturbance in this formation: a vehicle merging and a vehicle splitting the platoon. Those two scenarios, represented on Figure 1, can be detailed as follows:

A Vehicle merging happens when two non-empty platoons merge together to become one. This manoeuvre requires a platoon formed of only one vehicle, which is *L2* on Figure 1, to communicate to another platoon its intention to join it. Moving from step 1 to 2 (*S1* to *S2*) on Figure 1, the latter platoon will react by creating a safe space while the merging vehicle modifies its velocity to join the meeting point. When the formation has safely reached a stable state *S2*, the merger changes lane to enter the platoon formation. Once the merged vehicle has stabilized its inter-vehicle distance, the platoon can attain its precedent formation plus one vehicle, by diminishing the distances with the new vehicle, thus reaching state *S3*. Although, the steps of the merge task may differ from one coordination approach to another, this represents the general pattern.

A Vehicle splitting is the exact opposite of a merge manoeuvre: a vehicle member of a platoon decides to leave it, thereby forming two non-empty platoons. To execute this manoeuvre, the splitter (*F2*) must communicate its intention of leaving the platoon, so the platoon formation modifies the distances at the front and rear of the splitting vehicle, as shown in *S1*. When this new formation gains stability, the splitting vehicle *F2* can change lane, while the rest of the platoon followers keep the same distances. When the splitting vehicle has safely left the platoon (*S2*), the gap created for its departure can be closed, thus forming back the precedent platoon, minus one vehicle (*S3*).

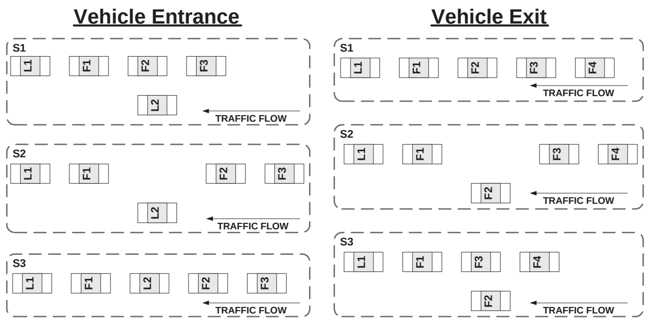

Fig. 1. Three steps of the entrance (merge) and exit (split) of a vehicle.

III. Hierarchical Architecture for Collaborative Driving

The architecture we adopted for our driving system is based on a hierarchical approach [11] that will be used as part of a decentralized control model. This architecture uses a more reactive system as the bottom of the architecture and moves forward to a more deliberative system as it raises to the upper levels. This approach is related to other collaborative driving models, as it was inspired in part by Tsugawa's Advanced Vehicle Control and Safety System (AVCSS) architecture [12] and other concepts coming mainly from the PATH project [13]. The resulting architecture has three major layers: *guidance layer*, *management layer* and *traffic control layer*, as indicated on Figure 2.

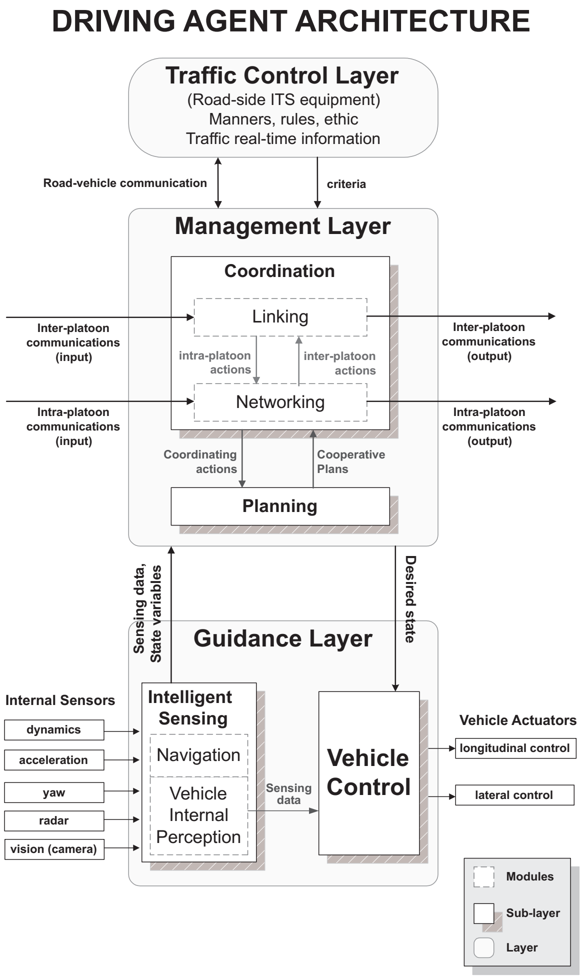

Fig. 2. Hierarchical architecture for collaborative driving.

The *guidance layer* has the function of sensing the conditions and states ahead and around the vehicle and activating the longitudinal or the lateral actuators. For the sensing systems, inputs come from sensors for speed, acceleration, raw rate, machine vision, etc. This layer also outputs sensing data and vehicles state variables to the vehicle guidance layer and then receives steering and vehicle velocity commands from the same guidance layer. These considerations have lead us to divide this layer in *intelligent sensing* and *vehicle control* sub-layers as depicted on Figure 2. The vehicle control will be discussed in section III-A.

The *management layer* determines the movement of each vehicle under the cooperative driving constraints using information provided by (a) the guidance layer, (b) the coordination sub-layer through the inter-vehicle communication, (c) the traffic control layer through the road-vehicle communication. To determine the movement of each vehicle under the cooperative constraints, this layer needs to reason on the place of the vehicle in the platoon when this platoon remains the same (intra-platoon coordination), and its place in a new platoon when this platoon changes (inter-platoons coordination). The first type of coordination is handled by the *networking* module and the second by the *linking* module, together forming the *coordination* sub-layer. Generally, the task of the *linking* module is to communicate with the traffic control layer to receive suggestions on actions to perform. Once the *linking* module has chosen an action to perform, the manoeuvres involved in the action (likely splitting or merging a platoon) will be coordinated through intra-platoon policies. These policies will be maintained using the *networking* module, which is responsible of the intra-platoon coordination. Finally, the management layer should also maintain a platoon formation plan, a task which is devoted to the *planning* sub-layer.

The *traffic control layer* is a road-side system composed of infrastructure equipments like sign boards, traffic signals, road-vehicle communications, as well as a logical part including: social laws, social rules, weather-manners and other ethics (more specific to Canada), etc.

A. Guidance Layer Implementation

To achieve the platoon maneuvers, two guidance controls have been defined : the *longitudinal* control which can use two different gaps, a distance gap based on [3] and a time gap control based on [14] and the *change lane* control. Formally, the time gap control is defined for the i^{th} vehicle of the platoon by :

$$a_i = \delta a + \frac{1}{h}(\delta v + k(\delta x - (gapv_{i-1}))),$$

where a_i is the acceleration of the i^{th} vehicle, δa is the difference of acceleration, δv is the difference of velocity, δx is the inter-vehicle distance and gap the wanted time between vehicles. These controllers will be used and tested in collaboration with the ones of our collaborators at Sherbrooke University [9], which also provide the lower level controller that transforms the desired acceleration in brake and throttle commands.

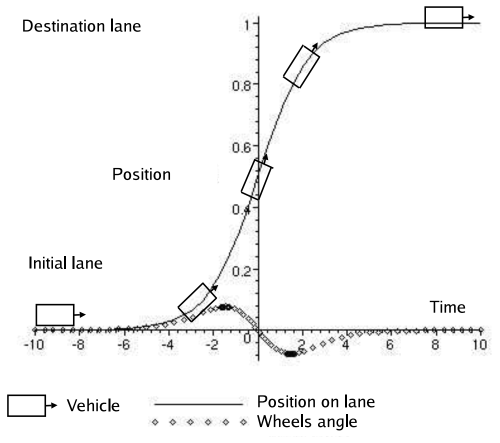

Fig. 3. Change lane control.

For the change lane control, a robust approach has been proposed in [15]. Our change lane control follows the functions drawn in the Figure 3. The vehicle has to follow the path defined by the position function to change lane. This function is a simple sigmoid function $p(t) = \frac{1}{1+e^{(-\alpha t)}}$. The wheel angle is controlled by $\frac{d^2p}{dt}$ and α controls the duration of the maneuver. We assume that the road curative does not change during the lane change maneuver.

IV. Communication and Coordination Methodologies

By using a decentralized guidance and control system [9] for every autonomous vehicle in the platoon, a need for collaboration strategies on driving actions arises. Communication provides more efficiency and safety, and faster response time [2] for Collaborative Driving Systems (CDS), but we must define the most effective way of using it, in order to take full advantage of this technology. The different possible communication methodologies for the platoon of vehicles are implemented in the *coordination* sub-layer of Figure 2. Since those coordinated actions are directly linked to the distributed planning sub-layer, part of each vehicle's driving architecture, we defined a comparison of possible coordination approaches from Durfee's representation of distributed planning [16]. This representation defines planning models for multiagent systems as either: *centralized planning for distributed plans*, *distributed planning for centralized plans*, *distributed planning for distributed plans*. The first, *centralized planning* model can be compared to coordination models used so far for platoons architecture centralized on the leader, as the one of the PATH project [5]. Within this planning model, the distributed plans can include synchronization actions, leaving more flexibility to the plan executors, as it was done in a recent version of PATH's architecture [17]. Furthermore, the fully decentralized *distributed planning for distributed plans* can be implemented using a novel approach to inter-vehicle coordination in CDS: teamwork for driving agents [18]. Four models reaching from *centralized planning for distributed plans* to *distributed planning for distributed plans* are presented on Figure 4, where the inter-vehicle communication involved in each model is highlighted.

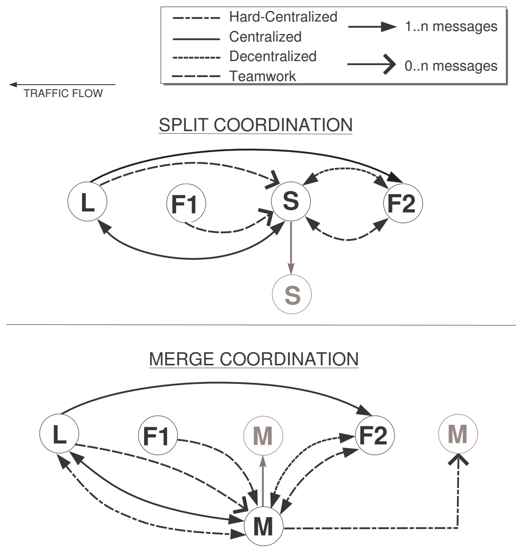

Fig. 4. Four coordination models of the merge and split task.

A. *Centralized Platoons*

A centralized platoon means that the task of communication executed to coordinate the vehicle formation is executed by only one vehicle: the leader. To maintain the platoon formation, the leader (head vehicle) is the only entity that can give orders, in which case the followers only apply requested changes. During a split manoeuvre, three vehicles are involved: the leader, the splitter, the vehicle following the splitter (if it exists), which are vehicles *L1*, *F2*, *F3*, on Figure 1. During a merge, the same configuration of vehicles is involved: the leader, the merger, the vehicle which will follow the merged vehicle (if it exists), that can be defined as vehicles *L1*, *L2*, *F2*. For both of these manoeuvres, the merger or splitter will communicate its need to do a manoeuvre, and the leader will send requests on inter-vehicle distance, change of lane, meeting point or on velocity, to involved vehicles. For the merge task, we have defined two sub-models. The first one simplifies the task and involves only two vehicles, by requesting the merging vehicle to always merge at the end of the platoon. In a second model, the leader will specify the optimal in-platoon merging position, considering the merging vehicle's position (parallel to the platoon). Thus, this model will involve three vehicles, if the merging vehicle's position is in front or farther than the platoon's tail vehicle.

B. *Decentralized Platoons*

In the concept of a decentralized platoon, the leader is still the platoon representative, but this is only for inter-platoon coordination. Thus, every platoon member has a knowledge of the platoon formation and is able to react autonomously, communicating directly with each others. An agent's common knowledge is initialized when it enters the platoon and updated using the broadcasted information about new vehicles' merge or split.

This model represents the simplest decentralization approach and does not rely on any existing framework or complex distributed plan sharing, but tries to lower the communications as much as possible with its simplicity. For the split manoeuvre only two vehicles are involved: the splitter and the vehicle following the splitter (if it exists). For the merge, once the merging vehicle has chosen a platoon, only two vehicles are involved as well: the merger, the vehicle which will follow the merged vehicle (if it exists). For those manoeuvres, we eliminate the intermediate that was the leader because every platoon members have the knowledge of their platoon configuration. Thus, using social laws defined for agents member of the same platoon, the actions of creating a safe gap for the split and merge tasks can be handled by a platoon member without being assigned by the leader. For instance, the intention of creating a safe gap will emerge (through social laws) from the vehicle at the right distance from the merging vehicle, while the other platoon members will determine that it is not their task.

C. *Teamwork for Platoons*

A more organized decentralized concept, gaining in popularity in the field of multiagent, is the one of teamwork for agents. Using Team Oriented Programming (TOP) models, like STEAM [4], the platoon members are assigned roles within a team hierarchy, and team operators relating to those roles are defined, in the same way as an agent's plan assignation. The STEAM architecture also provides domain-independent directives to support responsibilities and commitments for teamwork, which help to maintain safety in the manoeuvres accomplishment.

For the Auto21 project, we have defined three major teams, each requiring specific roles to be filled for their formation and which are defined as follows:

- The platoon formation is a persistent team, using persistent roles, for long-term assignments. For this formation we only require two persistent (long-term assignments) roles: a *Leader* and *Followers*. The first role is filled by the head vehicle and the second, by all of its followers.
- The split task team requires task-specific roles, for shorter-term assignments, defined as: *Splitter*, *Gap Creator*, *Virtual Vehicle*, *Safety Observers*. Figure 5 illustrates this formation, where the leaf nodes represent roles and in this case, the only internal node for the task observers, represents a sub-team. The *Splitter* is the vehicle initializing the team by making a request for his task, and the *Gap Creator* is the vehicle behind the vehicle executing the task (vehicle *F3* on Figure 1). The *Virtual Vehicle* mainly helps the vehicle executing the task to create an inner virtual representation of a front vehicle (vehicle *F1*), by communicating information about its dynamic state, when it has went out of the reach of the task-executing vehicle's sensors. Finally, *Safety Observers* is a role taken by one or more agents that communicate their belief about dangers or unsafe deceleration to others.
- The merge task team is very similar to the previous split team, as it requires the same roles, except that the *Splitter* is replaced by a *Merger*, and the execution of the *Gap Creator* and *Virtual Vehicle* roles will differ.

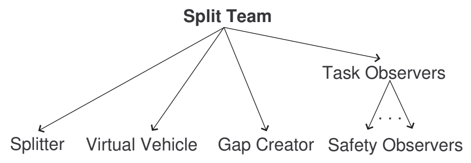

Fig. 5. Split task team's role organization.

Thus, the teamwork strategy results in most vehicles of a platoon to be involved in tasks and communicate if necessary, as shown on the dotted lines of Figure 4, representing "possible" communication.

Different domain level operators have been defined for those three teams' role carriers, and their execution is regulated using STEAM's architecture level operators (domain-independent) as the Coherence Preserving (CP) actions and Selective Communication (SC) actions [4]. A CP action, seen as a communicative act, will be used if the current operator is believed to be unachievable, achieved, or irrelevant. Then, SC actions are being used to synchronize mutual beliefs within the execution of team operators by verifying the likelihood that the information it wants to communicate is already common knowledge, through a process of decision-theoretic communication selectivity. For instance, using SC actions during a task execution, an agent filling the *Safety Observers* role will have a higher probability of communicating deceleration information ahead, if a lane change is being executed by a member of its team (the danger of a high deceleration is high during a lane change).

The choice of communicating information, made by SC actions, is done if the expected utility of communicating this information is higher than not communicating it. Represented as $EU(C) > EU(NC)$, i.e., iff:

$$\rho * \sigma * C_{mt} > (Cc + (1 - \rho) * Cn)$$

Where ρ is the probability that the information it wants to communicate is not known by its teammates. σ is the probability that this information opposes a threat to the execution of the current team operator. C_{mt} is the cost for miscoordination, Cc is the Cost of communication and Cn is the Cost of nuisance. The previous costs are dependant on the task being executed (e.g., critical tasks have high miscoordination cost). Thus, the probabilities and cost are initialized considering domain-specific factors, and will be updated through learning.

V. Evaluation and Results

To develop the previous coordination strategies within the architecture presented on Figure 2, we have used an agent development toolkit called JACK Intelligent Agents^{TM}[19] which supports the Belief Desire Intention (BDI) agent model [20], as well as teamwork related strategies. In the TOP vision relating to the STEAM model, a non-negligible advantage is the reusability and flexibility of the operators [4], since it contains many infrastructure rules that are not directly related to the domain level. Thus, using JACK's vision of agent's capabilities, related to plans and beliefs, we managed to develop collaborative driving teams that follow STEAM and TOP models.

A. Reaction to a Splitting Vehicle

We show as an example on Figure 6, the results we got in the average coordination of a vehicle exit (split manoeuvre), using a centralized coordination (in a thin red line) as opposed to the teamwork model of coordination (in a bold blue line). This graphic highlights the reaction of a platoon to the departure of a vehicle (*F2*), by focusing on its following vehicle (*F3*). In this scenario, *F3* senses the splitting vehicle and has to adjust from its departure by keeping a *safe* gap until the splitter is safely out. The solid lines present the difference between, the front distance between this vehicle and its front vehicle (splitter), and the *safe* front distance. This *safe* distance is defined by a gap in time between vehicles that agents should respect to insure security. In addition, the dotted lines only show the inter-vehicle distance from *F3*'s sensor, without any modification. Around time 14, vehicle *F3* has to create a larger and safer distance with the splitting vehicle, so the solid lines drop, but are readjusted within almost 10 seconds. The second outlined step arrives at time 30, and 27 for the teamwork, when the splitter has went out of range of vehicle *F3*'s laser. At this moment, the sensed distance raises on the dotted line, but there is no gap considering the distance defined as *safe* (solid lines). Before the splitter has stabilized itself on the next lane (time 37 or 34), the gap creating vehicle of the centralized model does not manage to keep the *safe* distance and has a difference of 2 meters with the *safe* distance by the end. On the other hand, the splitting vehicle modelled with the teamwork coordination is using communications from vehicle *F1* through teamwork rules, to maintain his virtual representation and follow it after it has changed lane. By doing so, the splitting vehicle helps its following vehicle (*F3*) to maintain the right *safe* distance. This approach gives much better results, since the difference with the *safe* distance does not go higher than 0.5 m. When the splitter is stable, the distance qualified as *safe* drops to the normal intra-platoon distance. At that moment, the vehicle is at 17 meters (length of the gap created by the vehicle that left) from the *safe* distance, reached by the end. This graphics also shows that using a teamwork model, information is exchanged faster since messages do not have to go through the leader, which results in an overall faster response time of three seconds by the end of the task.

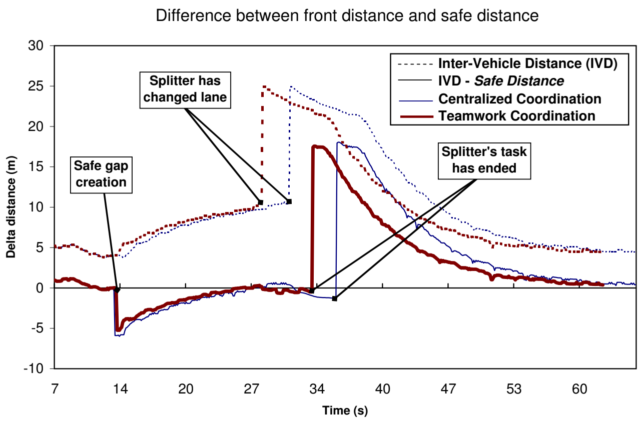

Fig. 6. Inter-vehicle distance with the splitting vehicle.

B. Coordination Models Comparison

Test results received from the simulation of the scenarios presented in section II-B were also collected from the four different coordination models presented in section IV: *centralized agent architecture* using the simplest merge protocol (*hard-centralized*) and using an elaborated centralized protocol (*centralized*), *decentralized agent architecture*, *teamwork for agents architecture*.

Using the preliminary results presented in Table I, we show each of the four models used for both a split and a merge, divided in four rows. We compared on the first pair of column, the average total amount of messages exchanged by vehicles during the manoeuvre. The second pair of columns compares the amount of plans required to develop the *Coordination layer* of the architecture shown on Figure 2. Those plans were defined using JACK [19], and represent contextual rules of applications for communication and actuation. The amount of messages calculated for the merge task considers that the merging vehicle had already chosen his platoon and is ready to merge. Those four models' advantage and inconvenient are summarized as follows:

1. The first centralized model (*Hard-Centralized*) is benefic on the amount of messages it exchanges. But the major disadvantage is the traffic density it creates, as it must reach the platoon's tail by either accelerating or decelerating (considering his relative position), thus creating traffic waves and diminishing the highway's capacity.
2. The second centralized model suffers from the amount of messages it encounters, as the leader redirects all the messages within the platoon. Moreover, in average,

more then three quarters of the messages were sent or received by the leader, creating a bottleneck for this vehicle. But compared to decentralized models, this model does not require the followers to keep a platoon knowledge, which helps lowering communications when using point-to-point communication (not our case).

3. The standard decentralized approach uses less messages but is a lot less safer than the other approaches. This fact will be proven using further simulations, but since only two vehicles (agents) are being used during the tasks executions, and no vehicles ahead in the platoon communicates information on possible dangers, this approach would have to compensate by using more sensors to attain the level of safety of the other models. This model also needs to communicate to initialize and maintain common knowledge within the platoon, but since the "updates" on knowledge are done through a broadcast of messages, it does not require more communications to reach every platoon members.

4. In the teamwork model, we managed to use an amount of messages that is in the average of the three other approaches. It must be mentioned that this number varies more than in the other models, from different contextual simulations, because of the selective communications. Furthermore, the teamwork model was the only one supporting virtual vehicles, which explains a higher number of messages. Then, using a TOP model implemented with the STEAM vision in JACK, even though there are more actors for a safer approach, we do not need much more plans then the standard decentralized approach, which was not developed using generic plans. Moreover, TOP also uses less plans than the centralized model which was developed in a more functional vision, in which more plans are required to handle uncertainty, compared to STEAM's vision.

TABLE I

TOTAL OF MESSAGES AND PLANS USED BY COORDINATION MODEL

	Nb Messages		Jack Plans	
Coordination	Merge	Split	Merge	Split
Hard-Centralized	7	7.5	12	12
Centralized	11.5	8	20	13
Decentralized	8	5.5	12	9
Teamwork	8.75	6.75	14	10

VI. Conclusion and Future Work

Collaborative driving is emerging in the ITS domain and its need for communication is obvious. As this paper presented, building an autonomous driving system on a strong architecture giving a wide latitude to the coordination strategies enables the use different inter-vehicle communication models. From these models, we have presented and tested four different strategies, for which advantage and inconvenient were presented. The coordination model based on teamwork presents great avenues considering the flexibility and safety insurance at low communication cost. In addition, the decision theoretic communication selection allied with STEAM's Coherence Preserving (CP) actions enables a wider use of generic plans, provides code reusability and flexibility and uses more efficient inter-vehicle communications.

The Collaborative Driving System presented in this paper could be used in a fully autonomous system, using vehicles equipped with longitudinal and lateral guidance system. But within the presented scenarios, we did not specify if the lateral control was automated or a simulation of a human driver, thus we are are still opened to both avenues. As future works, the usage of the teamwork model should be analyzed within a vehicle formation different from the platoon, without a particular leader, where vehicles are more autonomous and not necessarily equipped with the same system. Furthermore, these collaborative driving strategies will be used within a simulation of the Canadian climate environment, thus demonstrating the application of CDS to snowy roads, conjointly with the required ITS infrastructure.

VII. Acknowledgments

This work has been carried out as part of the Automobile of the 21st Century (Auto21) project supported by the Government of Canada through the Networks of Centres of Excellence (NCE).

References

[1] R. R. Stough, *Intelligent Transportation Systems: Cases and Policies*. Edward Elgar Pub. LTD, 2001.
[2] Q. Xu, K. Hedrick, R. Sengupta, and J. VanderWerf, "Effects of vehicle-vehicle / roadside-vehicle communication on adaptive cruise controlled highway systems," in *IEEE VTC*, Fall 2002.
[3] S. Tsugawa, S. Kato, K. Tokuda, T. Matsui, and H. Fujii, "A cooperative driving system with automated vehiclues and inter-vehicle communications in demo 2000," in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2001.
[4] M. Tambe and W. Zhang, "Towards flexible teamwork in persistent teams: extended report," *Journal of Autonomous Agents and Multi-agent Systems, special issue on Best of ICMAS 98*, vol. 3, pp. 159–183, 2000.
[5] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on Automatic Control*, vol. 32, 1993.
[6] Auto21, [Online], April 2004, http://www.auto21.ca/, (consulted the 30th of April 2004).
[7] DAMAS-Auto21, [Online], April 2004, http://www.damas.ift.ulaval.ca/projets/auto21/, (consulted the 30th of April 2004).
[8] S. Hallé, B. Chaib-draa, and J. Laumonier, "Car platoons simulated as a multiagent system," in *Proceedings of Agent Based Simulation 4 (ABS4)*, Montpellier, France, Mar. 2003.
[9] X. Huppe, J. de Lafontaine, M. Beauregard, and F. Michaud, "Guidance and control of a platoon of vehicles adapted to changing environment conditions," in *IEEE International Conference on Systems Man and Cybernetics, 2003.*, vol. 4, Oct. 2003, pp. 3091–3096.
[10] U. Kiencke and L. Nielsen, *Automotive Control Systems: For Engine, Driveline and Vehicle*. Springer Verlag, March 2000.
[11] Auto'21, "Architectures for collaborative driving vehicles: From a review to a proposal," Université Laval, Ste-Foy, Québec, Rapport de recherche DIUL-RR-0303, 2003.
[12] S. Tsugawa, S. Kato, T. Matsui, and H. Naganawa, "An architecture for cooperative driving of automated vehicles," in *Procs. IEEE Intelligent Vehicles Symposium 2000*, Dearbsorn, MI, USA, Oct. 2000, pp. 422–427.
[13] J. Lygeros, D. N. Godbole, and S. S. Sastry, "Verified hybrid controllers for automated vehicles," in *IEEE Transactions on Automatic Control*, vol. 43, 1998, pp. 522–539.
[14] P. Daviet and M. Parent, "Longitudinal and lateral servoing of vehicles in a platoon," in *Proceedings IEEE of the Intelligent Vehicles Symposium, 1996*, 1996, pp. 41–46.
[15] C. Hatipoglu, U. Ozguner, and K. A. Redmill, "Automated lane change controller design," *IEEE Transaction on Intelligent Transportation Systems*, vol. 4, no. 1, March 2003.
[16] E. Durfee, "Distributed problem solving and planning," in *Multiagent Systems*, G. Weiss, Ed., Cambridge, MA, 1999, pp. 121–164.
[17] S. Bana, "Coordinating automated vehicles via communication," CS Berkely," PATH Research Report UCB-ITS-PRR-2001-20, 2001.
[18] S. Hallé and B. Chaib-draa, "Collaborative driving system using teamwork for platoon formations," in *Proceedings of AAMAS-04 Workshop on Agents in Traffic and Transportation*, July 2004.
[19] Agent Oriented Software Pty. Ltd., "JACK Intelligent Agents^{TM} 4.1," Software Agents Development Framework, 2004.
[20] A. S. Rao and M. P. Georgeff, "BDI-agents: from theory to practice," in *Proceedings of the First Intl. Conference on Multiagent Systems*, San Francisco, 1995.