Kelsch, J.; Dziennus, M.; Köster, F. (2015): Cooperative Lane Change Assistant: Background, Implementation & Evaluation. Proceedings to AAET 2015, pp. 65-85, 12.-13.02.2015, Braunschweig, Germany, ISBN: 978-3-937655-34-5

Cooperative Lane Change Assistant: Background, Implementation & Evaluation

Johann Kelsch, Marc Dziennus, Frank Köster

Institute of Transportation Systems / German Aerospace Center

Lilienthalplatz 7, 38108 Braunschweig, Germany

johann.kelsch@dlr.de

Abstract

Drivers can cooperate with automation inside the vehicle as well as with traffic participants and intelligent infrastructure outside the vehicle. For that matter cooperation can be a successful concept for designing driver-automation or rather highly automated traffic systems. EU-Project D3CoS was aiming at related knowledge to develop a cross domain framework for design of cooperative systems. For the automotive domain this knowledge was applied to develop a Cooperative Lane Change Assistant (C-LCA) that supports drivers in dense traffic situations while performing lane changes. In this contribution theoretical background, used methods, implementation of the C-LCA, results of its evaluation and lessons learned are described.

Introduction

Modern technical development in the automotive domain push today's traffic systems towards higher levels of automation and distribution. Advanced sensor and actor technologies, e.g. LiDAR and active inceptors, enriched by the proper algorithms, e.g. data fusion and arbitration [9], allow highly automated driving. The driver and the automation inside a highly automated vehicle can share the control [15] or even be team players [2]. Advanced communication technologies, e.g. Car2X technologies, allow moving the high automation from the inside to the outside of the vehicle. This way, a highly automated and highly distributed traffic system is evolving, where several agents, such as drivers, pedestrians, different sorts of automations, traffic management etc., are involved as elements of one system. Though, such development can cause conflicts, which have to be resolved by proper system and interaction design in order to prevent or to avoid unsafe, inefficient and uncomfortable traffic situations.

Imagine the use case (Fig. 1): Two highly automated vehicles (V1, V2) are driving in a dense mixed traffic on a two-lane highway. V1 (120 km/h) is on the right and V2 (80 km/h) on the left lane. Both vehicles are equipped with ACC-automations (adaptive cruise control) aiming at keeping the constant distance as well as adapting the velocity to the related vehicle in front. Both vehicles have the task to move as efficient as possible through the traffic.

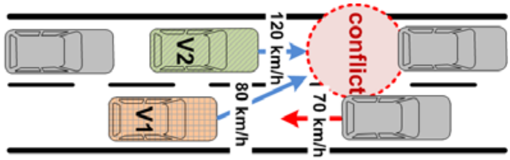


Fig. 1: Possible resource conflict in a highly automated traffic system

If the vehicle in front of V1 slows down (70 km/h), a resource conflict will be induced into the system as a sort of a *problem*. V1 has do brake, since there is no gap on the left lane (resource is in conflict) for changing the lane fluently. This is inefficient in respect to the task of V1. However, this conflict can be resolved by inducing cooperative behaviour into the system as *solution*. In the described use case V2 could slow down just in order to timely open an appropriate gap for V1's lane change. This still roughly formulated design concept of 'cooperation induction' is applicable at manually driven vehicles in the same use case as well.

This contribution is about how such a 'cooperation induction' can be generally designed in the automotive domain and in context of lane change situations in particular. In the next section the method used for the design of the C-LCA is described step by step. We are focussing on how cooperation can be understood from its theory towards being helpful in system design.

Since this contribution is aiming at being more about system and interaction design and less about the theory in behind, this section will not deeply discuss the cooperation phenomenon itself. However, the used definition of cooperation and how it was applied to the C-LCA design will be stated.

It is possible to decompose a cooperation process into appropriate ‘problem-solution pieces’: Cooperation (design) patterns. These, for example, can be separately used for addressing system and interaction design challenges in order to initiate, to maintain and to complete the cooperation process. So, in the subsequent sections cooperation patterns will be shortly introduced, first on the abstract level and then on the level of their concrete application for the C-LCA design. Then the concrete interaction design and implementation of the C-LCA, how it was evaluated and lessons learned will be described.

For the design of C-LCA a six-step approach was used: 1. Use case definition and decomposition; 2. Conflict analysis; 3. Design concept; 4. Interaction design; 5. Implementation; 6. Usability study (Fig. 2).

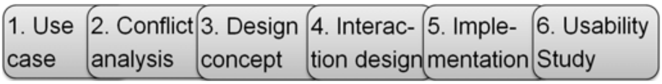


Fig. 2: Used approach for design of C-LCA

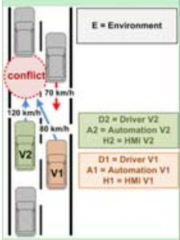
This six-step design approach will be described in detail and the particular design process steps will be used for structuring this contribution.

Use case definition, decomposition and conflict analysis

First of all, a use case was defined that should be properly handled by the C-LCA at the end. This use case is about two vehicles in dense traffic, which is already described (Fig. 1) [9]. To begin with a more generic design of C-LCA we decided both ego vehicles (V1, V2) to be manually driven.

The target use case structure was decomposed into several agents. An agent is “any entity able to act” [1], so a driver, a vehicle automation, the vehicle itself, the environment can be regarded as separate agents within a multi-agent system [9], which can be used as decomposition means. V1 and V2 contain the drivers (D), the automations (A), the vehicles themselves (V) and the HMI agents (H) each. The HMI agents can be further decomposed into a steering wheel and a display agent for each vehicle. The other vehicles in the use case as well as the environment can be aggregated in the environment agent (E) [9].

After the definition and decomposition of the target C-LCA’s use case it was possible to analyse it in order to discover potential system conflicts. On Fig. 3 a short summary of the conflict analysis is depicted.



USE CASE	TASKS		POSSIBLE CONFLICTS: AGENT STATE	AWARE-NESS	ARBITRA-TION	ACTION	SOLUTION
	D1	drive, cooperate, take the gap	distracted	doesn't comprehend that the gap is open	own action opportuninies unknown	uses the gap too early/ too late	**Conflict avoidance:** Resource conflict about a gap is unaviodable
	A1, H1, V1	coordinate, perform lane change	limits, failure	limits, failure	limits, failure	limits, failure	**Conflict solution:** Cooperation between V1 and V2 should be timely initiated, maintained and completed with the aime at opening a gap and guiding the correct drivers' actions. The automation should properly know own limits and failures
	D2	drive, cooperate, open the gap	distracted	doesn't comprehend that he must open the gap	own action opportuninies unknown	fails opening the gap correctly	
	A2, H2, V2	coordinate, produce a gap	limits, failure	limits, failure	limits, failure	limits, failure	
	E	X	resource in conflict	X	X	X	

Fig. 3: Problem analysis for the use case lane change in dense traffic

In the left row (USE CASE) of the corresponding conflict analysis tool, the use case and its decomposition is shown. In the next row (TASKS) general tasks are mentioned, which should be performed by the agents. In the next row (POSSIBLE CONFLICTS) conflicts are described that agents can induce into the system. These conflicts are pre-structured in four categories (AGENT STATE, AWARENESS, ARBITRATION and ACTION) according to the informational processing within the designed system. So, agents can be in certain conflict causing states, e.g. distracted, before informational processing. Then the agents can produce awareness related conflicts, e.g. due to the lack of situation awareness [4]. In the next step agents can fail due to the arbitration during the moderation and decision process [9]. At the end, the agents can fail while performing actions.

The gathered information was helpful for choosing cooperation as design concept and as general solution for all possible conflicts C-LCA should address. It is noticed in the right row (SOLUTION) of the conflict analysis.

Cooperation as design concept

To use cooperation as design concept it is important to understand what cooperation actually is. While analysing scientific literature with respect to the definition of the cooperation phenomenon two main definition approaches can roughly be distinguished. The first one is trying defining cooperation on the ‘agent level’ whereat the second one is doing this on the ‘system level’.

On the ‘agent level’ cooperation is mostly claimed as a result of some agent state preconditions and determined by behavioural rules within the agents. Those state preconditions and rules must be present in the agents to name then evolving system behaviour ‘cooperative’.

For example, the driver knows (is ‘in the state of knowing’) that if he or she behaves cooperatively (follows some rules) by opening a gap for another driver during a lane change manoeuvre, then another driver might open a gap next time in reward (direct and indirect reciprocity) [12]. In those definition approaches one mostly operates with game theoretical methods, such as calculation of agent related cost-benefit functions, relating them to some agent related probabilities and parameters, such as “coefficient of relatedness” or “probability to know someone's reputation”.

Those definition approaches contain a fair number of agent specific eventualities to be considered. Therefore, there can be many cooperation variants being dependant on the possibly huge amount of agent state preconditions, their combinations and the rules for agents' behaviour. Such variants can be, for example, 'symmetric cooperation', when the expected result is better for both agents than each other can achieve by itself, or a 'non-symmetric/symmetric compromise' when one or several agents degrade their benefits from a rational point of view [11] etc.

Such approach has certain advantages, e.g. it can support clear definitions and evaluation of mathematical rules for cooperation design on the agent level. It suggests parameters and metrics to be considered while designing a cooperative system, e.g. in form of agent specific costs, benefits, different agent types and behavioural rules. However, this approach has certain disadvantages as well. Especially when designing a complex highly automated traffic system with its high variability of possible use cases, agents' states and behaviour it can be difficult to choose an applicable set of necessary agent state preconditions and most appropriate behavioural rules.

On the 'system level' cooperation is often defined more loosely. It is argued as a process emerging from the internal complexity and sociality of the agents having personal goals and doing some actions and interactions. The agents are interacting permanently by their goals and therefore actions generating interference on the system level. The negative interference can be considered as conflict or concurrency and the positive one as cooperation between the agents [1]. A fair number of eventualities can be considered here as well, such as distinction between cooperation variants on the system level, such as social commitment, adoption, 'mind reading', as well as cooperation modes [7], such as mutual control, goal and task delegation etc.

The weakness of this definition approach can be the relative abstractness of the discussed concepts. For example, it can be unclear, how to exactly design the social commitment or 'mind reading' in the context of highly automated road transport systems. Though, such definition approach on the 'system level' has a very important advantage: It powerfully supports the holistic picture of the cooperation phenomenon. From the system design point of view the discussed cooperation variants are focussing more on the cooperation itself as a system related process instead of focusing on how to design the particular cooperating agents (e.g. beliefs, desires, intentions etc.)

Definition of cooperation on the system level was the guiding approach for C-LCA's development. For example, it was possible to define a system related metric named 'cooperation tension', which was used for controlling the cooperation process between the drivers and their C-LCAs from the syntax (scheduling) point of view. However, the definitions on the agent level were helpful for developing cooperation patterns used for design of C-LCA's interaction from the semantic (meaning) point of view.

Cooperation patterns for C-LCA's semantics

The EU-Project D3CoS (Designing Dynamic Distributed Cooperative Systems) was aiming at understanding and developing system design and cooperation related knowledge [3]. From the cross-domain perspective of D3CoS, this knowledge was translated into the automotive domain.

Cooperation was understood as positive interference and an emerging process between different agents, such as drivers, automations, HMI agents, intelligent infrastructure, environment etc. Such cooperation process, as any process, can be initiated, maintained and completed. During cooperation tasks and resources (re-)allocation between the agents can take place by changing the cooperation mode [7],[9] and in order to initiate, to maintain and to complete the cooperation. For it, cooperation patterns as well as supporting methods and tools were developed for cross-domain application, i.e. adopted for the automotive domain [16] and for the C-LCA design.

Based on the analysis of cooperation phenomenon and a strict problem-solution oriented approach, cooperation (design) patterns where developed, which can be used for system design. Nine basic cooperation patterns were developed so far. They were structured after their appearance order within the cooperative informational processing: initiation, maintenance and completion of cooperation. On the Fig. 4 all nine patterns are shown.

Information Processing Step	Design Pattern Name
INITIATION	PROBLEM AWARENESS
INITIATION	ROLE ALLOCATION
INITIATION MAINTENANCE	EXPLICIT ADRESSING
INITIATION MAINTENANCE COMPLETION	ALLOCATION TO COOPERATIVE POPULATION
INITIATION MAINTENANCE	TASK AND GOAL ALLOCATION
MAINTENANCE	AWARENESS OF OWN TASK AND GOAL
MAINTENANCE	AWARENESS OF PARTNERS' TASK AND GOAL
MAINTENANCE	CERTAINTY OF TASK AND GOAL
INITIATION COMPLETION	REWARDING

Fig. 4: Nine developed design patterns for cooperation with their informational processing step and name

These patterns are described in detail in [3]; therefore, this contribution is focussing on their particular application for the C-LCA design. For it, five patterns were used: PROBLEM AWARENESS, ALLOCATION TO COOPERATIVE POPULATION, EXPLICIT ADRESSING, TASK AND GOAL ALLOCATION and REWARDING. These cooperation patterns were applied to design the C-LCA. In this section we are describing, how the particular patterns were integrated into definition of C-LCA's semantics.

On Fig. 5 mapping of developed visual elements to their semantics used for C-LCA design is shown. The table is divided into five rows. In the left row the form of the possible appearance of the corresponding visual element is assigned to a particular category. It can be discrete, continuous or mixed. Discrete elements can appear event-based as a whole pictogram; the continuous elements can change their form continuously. In the next two rows, the element variants are shown as pictograms and described as enumerations of variables that can be changed for the particular element. In the right row, the assigned semantic meaning is described.

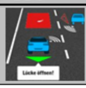
form	element	variants		semantics
discrete	vehicle		colour, size, position on screen	Ego vehicle (blue), lead vehicle (white), special vehicle (red)
discrete	lane marker		colour, size, position on screen, line variant	current state of lane markers around the ego vehicle (white), lane departure warning (red)
discrete	gap		colour, size, position on screen	ocupied and asked gap (red), free gap (green), "ACC-radar on" (dashed)
discrete	intention/ action		colour, size, position on screen, pulsing	intention (red, empty) acion (green, full)
discrete	done		X	Gap is ready!
discrete	message		position on screen, pulsing	message from automation to the driver
continu-ous	communi-cation		position on screen, direction	cooperation is going to be established/ is established in direction
continu-ous	waiting		X	Please wait!
mixed	dyad		position on screen, direction	cooperation dyad established
mixed	9er scheme		nine fields with any simple element	cooperation process guiding

Fig. 5: C-LCA visual elements semantics

This semantic mapping was developed and evaluated in several short pre-studies [10]. However, for the semantic mapping the cooperation patterns as well as cooperation as design concept were used. In the following it is described how the used cooperation patterns were implemented.

PROBLEM AWARENESS: “Lack of recognition and comprehension of problems of other agents, e.g. lack of recognition of situation criticality for other agents, can lead to issues in cooperative behaviour. If a problem of other agent is not obvious, then the agent might not see the necessity to help or to perform tasks cooperatively.” “Communicate problems directly to other agents in sufficient time, for example, the dangerousness of imminent collision with other vehicles.” [3]

This pattern was implemented as a red gap pictogram. It should appear on the screen only when there is a need for a gap, which should be understood by yet not cooperating drivers as a resource conflict. At the same time, there is a shared problem, which has to be solved by cooperative behaviour.

ALLOCATION TO COOPERATIVE POPULATION: “Agents do often not know that and with whom they are cooperating or have to cooperate.” “The issue can be resolved by the allocation of the agents to a particular cooperative population, for example by means of a dyad, triad etc.” [3]

This pattern was implemented through gradual appearance of communication symbols while the cooperation between partners was going to be established. At the end, both communication symbols were showing at each other symbolizing the established cooperation dyad. The allocation to the same ‘cooperative population’ was supported by colourising both cooperation partner vehicles on the display with the same colour (blue).

EXPLICIT ADRESSING: “During a communication within the cooperative system, more than one agent could be affected by the communicated demands of other agents.” “The communication flow must be determined to the particular optimal cooperation partner. Multiple communication flows have to be avoided.” [3]

This pattern was designed trough the explicit addressing of the particular cooperation partner. Instead of using, for example, the turn indicator outside the vehicles, which can affect several uninvolved drivers, only displays inside the vehicles of particular cooperation partners were actively moderating the cooperation until the actual lane change.

TASK AND GOAL ALLOCATION: “Agents might not know which tasks and goals could be adequate in a given cooperative situation.” “Allocate the relevant agents tasks and goals in a clear and understandable form. (For example: Increase distance to lead vehicle to open a gap for enabling lane change from right to left lane).” [3]

Implementation of this pattern were the necessary tasks, such as opening the gap, waiting, using the gap, being explicitly communicated to the drivers on their C-LCA displays by messages as well as by intention/action symbols.

REWARDING: “To behave cooperatively is often based on a voluntary basis, so agents might often refuse to behave cooperatively, especially when there is no direct or obvious benefit from the cooperative action. As a consequence, agents need to be motivated to behave cooperatively.” “If there is no obvious benefit for all involved agents, some sort of rewarding system could be implemented. Especially the agents should be rewarded for their cooperative behaviour after task fulfilment, which have to diverge from their status quo in order to enable other agents’ tasks and goals.” [3]

This pattern was implemented as an extra message to the driver, who opened the gap. The message field contained the phrase “Thank you!”

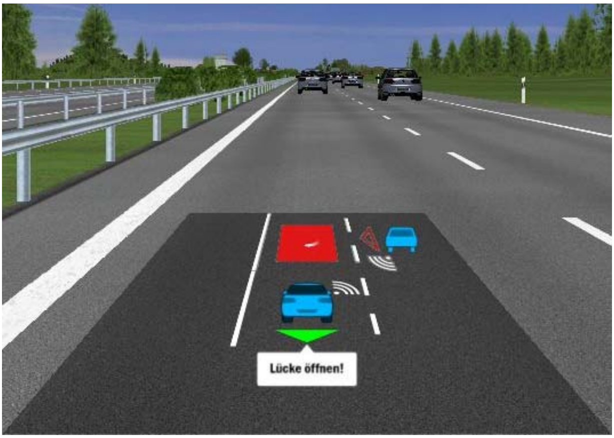


Fig. 6: C-LCA 9er scheme display in vehicle V2 during gap opening for V1

After definition of all visual elements with their mapped semantics, all these ‘simple’ elements were combined into a single ‘complex’ element: “9er scheme” display, which is shown on Fig. 6. Both cooperating vehicles (V1, V2) are equipped with the same display, which can show a slightly different area of the same traffic situation. The particular ego vehicle pictogram is always in the middle of the display. All other vehicles, closed and open gaps, intention/action, communication symbols, dyads etc. are shown at nine fields arranged around the ego vehicle pictogram. This is the C-LCA display. The depicted configuration shows the display inside vehicle V2 (compare Fig. 1) while the driver of V2 is opening a gap for a lane change requested by V1. The display itself is implemented as head-up display.

Besides the definition of visual elements and display semantics it was crucial to define the syntax of the intended interaction. Therefore we used the action tension concept [8] and a metric named 'cooperation tension'. In the next section this metric as well as the interaction syntax are introduced.

Cooperation tension for C-LCA's syntax

A particular action tension is an integrated and operationalized reference value for control of a particular action sequence on system level. It is always directed towards an optimal state [8], which means 'relaxation of a conflict'. Since in the C-LCA's use case (Fig. 1) there is a resource conflict, which can be resolved by cooperation between V1 and V2, we defined an action tension named 'cooperation tension' in order to develop the C-LCA's syntax. The use case, the operationalization of the cooperation tension, the resulting C-LCA syntax as well as the corresponding states on 9er scheme displays in both cooperating vehicles are shown in combination on Fig. 7.

Cooperation tension was determined to have five tension states: Cooperation not necessary, cooperation request V1 and V2, gap production and consumption. These tension states should activate the following actions: Determine cooperation necessity, determine cooperation acceptance for V1 and V2, guide gap production and consumption, complete cooperation. The transition events between these tension states were triggers for activation of the corresponding interaction signals on the 9er scheme display as well as synchronization signals [8] for behaviour of both cooperating drivers.

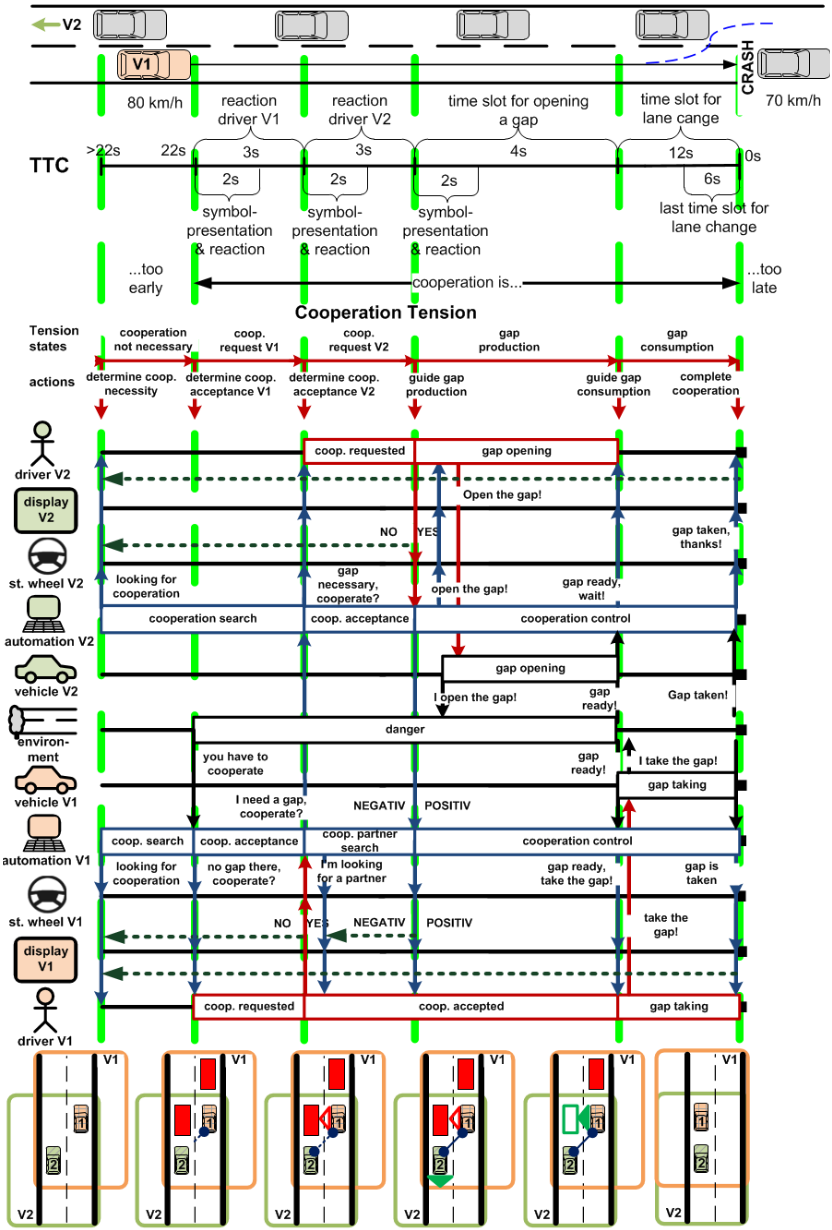


Fig. 7: C-LCA syntax developed by means of cooperation tension

After definition of cooperation tension states and corresponding action sequence the main task was to determine trigger events in-between. It was done by analysing literature dealing with driver behaviour during lane changes on highways and by estimation based on best practice and testing in a driving simulator. As main operationalization means for the cooperation tension the reciprocal value of TTC (time to collision) [6] was used. Driver reaction times and the time slot for gap opening were estimated according to the requirements of the used automation and HMI hardware and software. The time slot for lane change was determined according to [5] stating that most drivers prefer a lane change on a highway at around 12 sec. TTC.

On Fig. 7 (in the middle) a ‘syntactic storyboard’ is shown [9]. It is a sort of a sequence diagram from UML [13]. But instead of using time for life lines the cooperation tension is used as ‘action line’. However, it can be read from left to the right side. Similar to UML this sequence diagram is containing units and messages interchanged in-between. The units are the agents, which were determined during the use case decomposition. The ‘action lines’ are containing the current cooperation tension states mapped to the particular agents’ states. The interchanged messages and actions are depicted by arrows between the agents’ action lines. The former determined trigger events are controlling the overall system behaviour during the initiation, the maintenance, and the completion of the cooperation process.

The syntax of the cooperation process can be developed and documented this way using the former defined semantics (Fig. 5). For example, C-LCA automations of both vehicles can determine the cooperation necessity, arbitrate the cooperation acceptance by the drivers, guide the gap production and consumption, and complete the cooperation process at the end. On the bottom part of Fig. 7 it is schematically shown, how the C-LCA functionality is implemented on both screens of the vehicles V1 and V2.

The developed interaction model of the C-LCA was transferred into state machines and the interaction controller [9]. All together these were implemented as a C-LCA prototype and evaluated in a usability study.

Usability Study

For evaluation of the C-LCA a usability study was conducted in the IDeELab (interaction design & ergonomics laboratory) at DLR in Braunschweig. The IDeELab contains two coupled simulators [14] especially designed for the development and evaluation of interaction designs. In order to evaluate the usability of the C-LCA we focused on the user's subjective evaluation of the complete HMI design with regard to the comprehensibility, handling and acceptance.

For that matter we used a fixed base simulator with ca. 180° of view (Fig. 8) equipped with a customized steering wheel. The buttons of the steering wheel are freely configurable and were used for haptic inputs and visual feedback to the driver. In the present study the second button on the right had a blue colour and was used to accept a cooperation request whereby the second button on the left was red and used to decline cooperation.

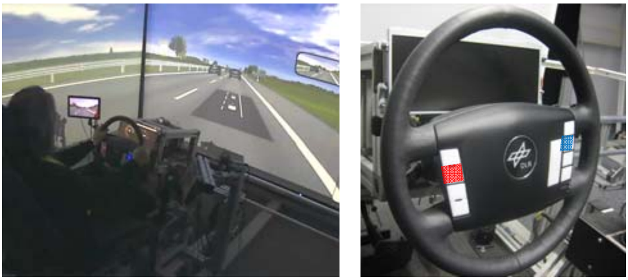


Fig. 8: Fixed base simulator and customized steering wheel

Eight subjects participated in the usability study. All participants (four male and four female) had an average age of 32 years (*SD* = 14.8). All of them possessed a valid driving license and had only very few experience with modern driver assistance systems. The participants were recruited through the participant pool from the German Aerospace Center (DLR). The already introduced use case with the resource conflict on a two-lane highway (Fig. 1) was used as main test scenario. It was mixed with several distractor use cases in order to produce a complete test driving scenario.

Each participant completed the study in approximately one hour on a single day. At first, all participants were introduced to the IDeELab and filled out a consent form and a demographic questionnaire. The participants were instructed that the purpose of the study was to test a new assistance system, which should help the driver to perform a lane change. After that all participants drove a training run of approximately eight minutes to familiarize themselves with the driving simulator. The training consisted of different driving tasks, such as accelerating, braking and multiple lane changes. After the training run, the participants started with the naive run. The participants were not informed how the C-LCA works, the only instruction was that their vehicle was equipped with a new assistance system, which should help the driver to perform a lane change. Furthermore, they were told the C-LCA helps the driver if another vehicle wants to change onto the driver's lane.

In the naive run the driver started on the right lane of a two-lane highway (V1 perspective). The driving task was to change to the left lane as soon as the vehicle in front begins to reduce its velocity. To perform a lane change the participant had to use the new assistance system (C-LCA). Approximately two minutes after the first run the participants experienced the HMI from the other perspective (V2). This time the participant drove on the left lane and got a cooperation request form another driver on the right lane. After the naive run a short interview was conducted and the participants were asked to fill out questionnaires to rate important variables, such as the usefulness and confirmability of the C-LCA. Afterwards, the participants were informed about the functionality of the C-LCA and had the chance to train the handling of the C-LCA. After three training trials a 'hot run' with trained participants was started. At the end, the participants were asked to fill out final questionnaires and received their payment.

Focus of this study was the C-LCA's usability. We analysed the subjective data collected in questionnaires and interviews. We compared the ratings after naive runs with ratings after introduction of the system (hot runs).

Both runs (naive and hot) were rated very similarly (Fig. 9), so that no significant differences could be found. Participants perceived the C-LCA as very helpful, understandable and useful. Working with the C-LCA was rated as pleasant, easy to learn and efficient. Participants felt excellently supported by the system regarding own and cooperation partner's lane changes and were willing to trust the system. Participants felt a noticeable increase of comfort and safety while using the C-LCA.

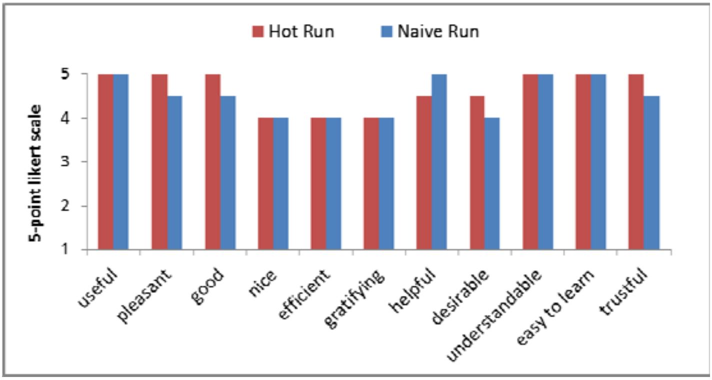


Fig. 9: Comparison of participants rating in naive and hot run

Discussion and lessons learned

C-LCA achieved a very good rating by the participants and proved its usability. The naive run results illustrate the very good comprehensibility of the design even for naive users without any knowledge of the system. Participants appreciated the constant feedback of the gap size and felt safer and more comfortable when using C-LCA. It was very easy to learn and after few training trials all participants felt well supported by the C-LCA. The participants rated the C-LCA as a very good and trustable system. As the biggest advantages of the C-LCA participants mentioned that they don't need to closely observe the traffic around them in order to know when there is a good point of time for a lane change. Furthermore, participants liked that C-LCA told them what to do (braking, steering) and when to start with the corresponding action.

Nevertheless, participants had some ideas for improving the C-LCA. Three of eight participants mentioned that it would be helpful to have a clear identification of the vehicle they are cooperating with. Further, it would be important to have an assistance system, which is able to support a lane change from left to the right lane as well. Furthermore, through observation we discovered at least one possible Human Factors related issue of the current C-LCA implementation. Two of eight drivers were performing the steering action for a lane change right after their commitment for cooperation in the perspective V1. Although there was no accident, it was very dangerous, since the gap was not ready yet. This particular point should be very closely investigated in the next C-LCA implementation.

During the C-LCA's interaction design, especially during the design of its semantics, we learned that it was important to focus on the "gap" as the main design element. That influenced the whole design towards the particular interaction signal statements, such as "open the gap, gap is ready, take the gap" etc. Further experience we made is that the whole design process, which is here claimed as linear, was actually iterative. For example, we had to change several times from syntax to semantic design and back. We had to tune on the conflict analysis while implementing etc.

However, our general conclusion is that the C-LCA can provide a great contribution to a safer traffic environment and decrease accident rates.

Summary and outlook

In this contribution we mostly focussed on the design aspects of a Cooperative Lane Change Assistant (C-LCA). It was expected as useful assistance function in dense traffic on highways, which can timely help producing and consuming gaps. Cooperation was stated as an overall system design concept for initiation, maintenance and completion of cooperative driver behaviour. It was decomposed into design patterns, which were used for definition of C-LCA's interaction semantics. The concept of cooperation tension was used for definition of interaction syntax.

The implemented C-LCA prototype was rated as useful, comprehensible and pleasant. Even though the current C-LCA version is designed for a head-up display, its structure makes it possible to use it also in a head down display. However, important challenges, such as distraction and gaze behaviour in a head down version are still unanswered. Furthermore, an adaptation to urban traffic is imaginable. Further research is needed to meet requirements of the more complex urban environment.

References

[1] Castelfranchi C. (1998): *Modeling Social Action for AI Agents.* Journal Artificial Intelligence - Special issue: artificial intelligence 40 years later archive Volume 103 Issue 1-2, Aug. 1998

[2] Christoffersen, K.; Woods, D. D. (2002): *How to make automated systems team players.* In: Salas, E. (Hrsg.): Advances in Human Performance and Cognitive Engineering. Research Bd. 2. Elsevier Science Ltd., 2002, p. 1-12

[3] D3CoS Community (2012): *Reference Designs and Design Patterns for Cooperation & DCoS State Inference and Adaptation.* D3-03 Deliverable for EU-D3CoS

[4] Endsley, M. R. (1995): *Toward a theory of situation awareness in dynamic systems.* Human Factors 37, p. 32-64

[5] Fastenmeier, W.; Hinderer, J.; Lehnig, U.; Gstalter, H. (2001): *Analyse von Spurwechselvorgängen im Verkehr.* Zeitschrift für Arbeitswissenschaft 55, 1

[6] Hayward, J. Ch. (1972): *Near miss determination through use of a scale of danger*. Report no. TTSC 7115, Pennsylvania State University, Pennsylvania

[7] Hoc, J. M. (2001): *Towards a cognitive approach to human-machine cooperation in dynamic situations.* International Journal of Human-Computer Studies, 54, p. 509-540

[8] Kelsch, J.; Heesen, M.; Hesse, T.; Baumann, M. (2012): *Using human-compatible reference values in design of cooperative dynamic human-machine systems.* EAM 2012, 11-12.09.2012, Braunschweig, Germany

[9] Kelsch, J.; Temme, G.; Schindler, J. (2013): *Arbitration based framework for design of holistic multimodal human-machine interaction.* AAET 2013, 6.-7. Feb. 2013, Braunschweig, Germany

[10] Klemm, R. (2013): *Willst Du mit mir kooperieren? Ja, Nein, Vielleicht!* Masterarbeit, HBK Braunschweig.

[11] Martin, J. (2002): *Organizational culture: Mapping the terrain.* Thousand Oaks, CA: SageMartin (2002)

[12] Nowak M.A. (2002): *Five rules for the evolution of cooperation.* Science (New York, N.y.) 2006; 314(5805):1560-1563. doi:10.1126/science.1133755

[13] OMG (Object Management Group) (2011): *Unified Modeling Language (OMG UML),* Superstructure, Version 2.4.1, http://www.omg.org /spec/UML/2.4.1/Superstructure/PDF/, Last access 25.04.2013

[14] Schieben, A.; Heesen, M.; Schindler, J.; Kelsch, J.; Flemisch, F. (2009): *The theater-system technique: agile designing and testing of system behavior and interaction, applied to highly automated vehicles.* Proceedings: 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications. pp. 43-46, ACM New York, NY, USA

[15] Sheridan, T. B.; Verplank, W. (1978): *Human and Computer Control of Undersea Teleoperators.* Cambridge, MA: Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT

[16] Temme, G.; Kelsch, J.; Köster, F. (2014): *Domänenübergreifende Gestaltung von Kooperativen Systemen.* Deutscher Luft- und Raumfahrtkongress 2014; 09/2014