Project no.:	**IST-FP6-FET-16276-2**
Project full title:	**Learning to emulate perception action cycles in a driving school scenario**
Project Acronym:	**DRIVSCO**
Deliverable no:	**D7.1**
Title of the deliverable:	**Specification of Learning Requirements**

Date of Delivery:	10.9. 2006
Organization name of lead contractor for this deliverable:	BCCN
Author(s):	F. Wörgötter, BCCN, M. Tamosiunaite, VMU
Participant(s):	BCCN
Work package contributing to the deliverable:	WP7
Nature:	D
Version:	1.0
Total number of pages:	10
Start date of project:	1 Feb. 2006 **Duration:** 42 **months**

Project Co-funded by the European Commission		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Specification of the Requirements for Structured Visual Events (SVEs) and Structured Action Events (SAEs) in the context of Learning

BCCN, Göttingen
Vytautas Magnus University, Lithuania

Bernstein Center Göttingen
Bunsenstrasse 10
37073 Göttingen
18. Dec. 2006,

Table of Contents

1. Introduction

1.2. Short Project Overview

The main goal of the DRIVSCO project is to arrive at a largely autonomously car driving system by combining adaptive learning mechanisms with conventional control. A major role is assigned to predictive mechanisms that will lead to an anticipatory, and therefore safe, driving style.
To acquire such a system, several learning techniques will be employed, e.g. correlation learning and statistical methods (s. also D 4.2).
The core idea is to learn a model that maps sensory input to output. This in- and output is clustered into units called Structured Visual Events, SVEs, and Structured Action Events, SAEs.

1.3. Importance of Hierarchy

In the following we will give an overview of which features SVEs and SAEs ought exhibit, in order to be used with any learning method. One important issue will be, that we can subdivide SAEs and SVEs into different levels of a hierarchy.

2. SVEs, SAEs and Learning Agents

Driving a car on a street has the advantage of taking place in a highly structured environment. Due to this, successful driving can be seen as result of solving 3 tasks: First, the vehicle must stay on the street while following it at an appropriate speed, second it should not crash into obstacles, fixed or moving, and third it must obey street laws, such as left yields right.
Each of these tasks can be solved individually and applied when there is need for it. E.g. a car should always follow the street unless it must avoid an obstacle. The interplay of these indidvidual tasks can then be regulated by a higher level control unit, which could be seen as a decision unit. Following the street would be on the lowest level and traffic rule obedience and obstacle handling would be above it.
For these reasons we assume that tasks can be treated individually, e.g. each being learned by a different agent, where their interplay can be learned separately.

3. Mapping SVEs to SAEs

The paragraph above motivated the division of the driving problem into 3 subtasks to solve. However, it left open how these subtasks can be addressed, which will be discussed now.
Each of the 3 tasks can again be described into different levels of complexity, or rather of a semantic hierarchy. Street following, for example, can be segmented into being on a straight stretch of the street, or a right or left curve, etc.. The same applies to the sensor information.
On the lowest level there is steering data as an angle alpha dependent on time and speed information in our system in form of a numeric value between 128 and -128 also dependent on time. But this data can also be clustered into discrete entities on different levels. One major issue here is, that a task is discrete on a high level and a continuous function of time on the lowest level. And this holds for both datatypes, i.e. input and output data, or sensory information and action commands, or

SVEs and Sues. Therefore, to apply any learning algorithm to these tasks their different appropriate levels must be identified. “Appropriate” here, refers to their relevance for learning. It naturally follows that correlations between input and output data are are best established between entities on the same level of the hierarchy, as illustrated in Fig. 1.

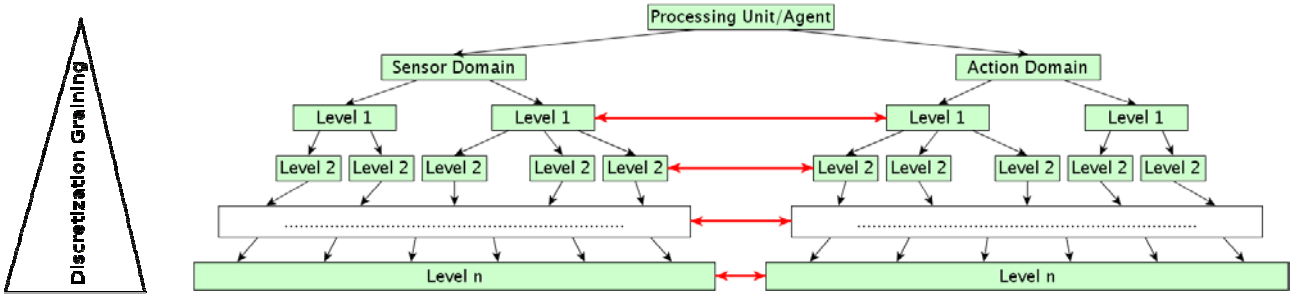

Figure 1: Abstract view of an agent. The horizontal arrows (in red) indicate where linkage between sensor and action domain can take place.

In this sense, SVEs and SAEs can be identified as entities (on the lowest level we would not have entities anymore, but continuous functions of time) in the hierarchy of each task. Identification of SVEs and SAEs, thus, is the same as determining the appropriate levels. Another important issue is to find out, on which level SVEs and SAEs shall best be correlated. Here, an additional complication occurs: The various levels of such a hierarchy are not independent from each other. Situations might arise in which the accordant action is also influenced by higher level or lower level information. So information flux must also be guaranteed in a vertical way as shown in Fig. 2. The proposed structures stay in the domain of layered architectures normally used for organizing autonomous robot behaviours as well as for driving robots (Gat, 1998, Thrun et al, 2006).

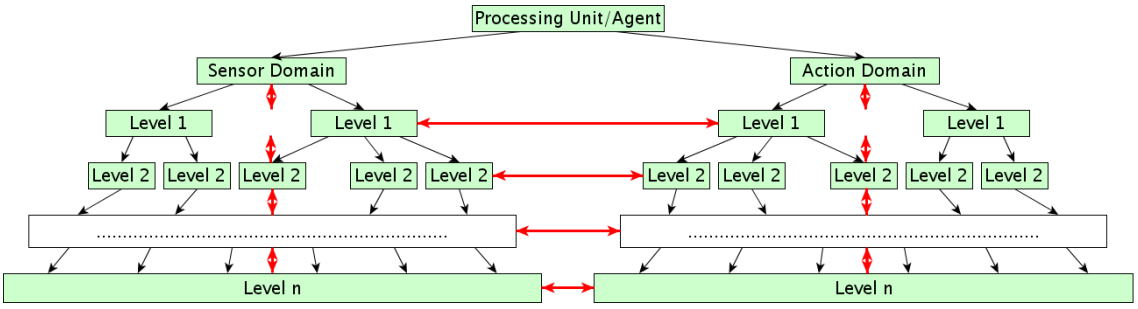

Figure 2: The vertical arrows indicate that sensor and action events must be seen in a certain context, which can be given by higher, or lower level information.

4. Specification of Requirements for SVEs and SAEs

In order to map SVEs to SAEs via a learning algorithm they must not only lie on the same level in the according hierarchy, but they must also be represented in a way, that they can be correlated. That is, every SVE must be described by parameters that can specify the related action. So one requirement for the specification of SVEs is not only the identification of the appropriate levels in the hierarchy as explained in the paragraph above, but also the identification of the needed

parameters. Here it makes sense to first identify the possible actions and then to trace back to what caused them.

4.2. SAEs

In our study action events are based on steering s(t), velocity v(t), and braking b(t) time series. Indicator signals, that contain information about the intentions of the driver, such as eye movements, are available from the test car setup (true car indicators also exitst), but in the first approach we will not be analyzing intentional driving, so actions where indicators are used will be excluded from analysis. In the simplified laboratory setup only s(t) and v(t) series will be considered. In real road experiments attention will also be investigated, and attention events can be attributed to action events. Attention will be investigated through the usage of an eye tracking system. Gaze position on the camera image series p_x(t), p_y(t) will be collected during real-road driving, but attention issues will not be analyzed in this deliverable.

Structured action events will be described at several levels of complexity, as mentioned above. At the lowest level we distinguish actions: (1) acceleration, (2) deceleration, (3) braking, (4) steering left, (5) steering right. The specificity of this level is that each event intakes only a single control variable: steering, velocity control and braking respectively.

At a higher level, bigger scale action events are defined, which we will be calling "driving conditions" for example: (1) straight driving, (2) curve taking, (3) tailgating, (4) stopping, (5) obstacle handling, (6) lane changing, (7) overtaking, (8) corner taking (at the intersection). These higher level actions are specific in the sense that they include more than one control variable (e.g. steering, velocity control and, possibly, braking during taking a turn). The first four driving conditions we classify as *automatic driving behaviours*, and the last 4 (5)-(8) as *intentional driving behaviours*. In the first stage only automatic driving behaviours (1)-(4) will be analyzed using real road experiments. In addition obstacle handling will be addressed in laboratory experiments. Straight driving formally might be analyzed as a special case of curve taking with zero curvature. Similarly overtaking and tailgating might be interpreted as special cases of obstacle handling. But to remain closer to everyday understanding of driving sequences we will be treating those cases separately.

To explain variability in behaviours, combination of events should be analyzed (e.g. taking a turn with tailgating (a car in front), and taking a turn without tailgating). Combination of events will form again higher level of action description. We are choosing the country road scenario with most prevailing situations for real road driving as mainly composed of the four automatic driving conditions as mentioned above. Obstacle handling will be investigated under laboratory conditions first, due to difficulties in collecting enough real world data.
Specifications of the 4 automatic driving conditions are provided in table 1:

Condition	Visual inputs	Non-visual inputs	Control parameters
Straight driving	- Lane marking signals - Heading vs. lane marking signals - Position within the lane - Uphill/downhill condition – Possibly optic flow (flow rate)	- Current speed of a car - Current steering angle	- Throttle position - Steering angle
Curve taking	- Lane marking signals - Heading vs. curvature - Position in the lane - Curvature estimation vs. distance - Optic flows (curved flow vectors) - Tilt of the road surface towards the center of the curve - Uphill/downhill condition	- Current speed of a car - Current steering angle - Current break force of a car Possibly: - Previous speed of a car - Previous steering angle	- Throttle position - Steering angle - Brake force
Stopping (or slowing down to given speed limitation)	- Obstacle description (size, distance, position on the road) - Intersection/traffic sign/traffic light - tail lights flashing - depth map of the scene - Uphill/downhill condition	- Current speed of a car - Current brake force	- Throttle position - Brake force

Tailgating	- Fact of a car in front - Relative speed of a car in front - Distance to a car in front - Time to contact of a car in front - Back lights of a car in front - Position on the road of a car in front - Orientation of a car in front on the road	- Current speed of a car - Current steering angle of a car - Current break force of the car - possibly distance and relative speed of the car if measured other than vision	- Throttle position - Steering angle - Brake force
Obstacle handling[1]	- Obstacle type - Obstacle size - Position of the obstacle on the road - Position of a car on the road	- Current speed of a car - Current steering angle of the car - Current break force of a car Possibly: - Previous speed of the car - Previous steering angle of the car	- Throttle position - Steering angle - Break force

Table 1: A listing of driving conditions and their required input, visual and non-visual.

The SAEs specified here are provided at an intermediate level of the described hierarchical structure. Higher actions would be composed by combining several actions of this level (e.g. curve taking with tailgating). Lowest level actions would correspond to simple controls, like staying on the road or keeping distance to a car in front. Although specific lowest level action would be dependent on the branch of the hierarchy through which we are descending towards them (e.g. possibly different steering algorithms would be used to stay on the road during straight driving or curve taking).

4.3. SVEs

With the results of section 4.1 it is now possible to specify the requirements for the desired SVEs. Mainly column 3 (“Inputs”) of table 1 already lists the necessary information that is to be extracted from the visual sensors. Straight driving and curve taking can be derived from an SVE that includes the following information: Curvature of right and/or left street boundary, which corresponds to “lane marking signals”, disparity between point of infinity of the cameras and of the street, which can give the “heading vs. lane marking signal” and also “position within the lane”. “Uphill downhill condition” will be ignored at this early stage.

Thus, we can create a visual event, let' s call it “trajectory”, which is formed by the curvature of each pixel on the right and/or left street lane border, e.g. c(lane_right, L). Where, c is a function that returns curvature, of the right lane (lane_right) and the pixel p(x,y) which is given by the parameter L, which is the arc length of the lane border. (Notice that L includes temporal information, since a small value refers to the near future and a larger L value to something further ahead in space and thus time.)
The corresponding event on the action side would be steering, which, again on the lowest level, can be described by a function s(t), which is the steering angle at a time t. Furthermore, street following should

1 Obstacle handling experiments will be done in laboratory with the simplifying assumption that there is only one obstacle.

happen at an appropriate speed, so the current velocity v(t) of the car, which is also sensory information must be taken into account. Thus we can derive the following specification of the structured visual event trajectory:
SVE_TRAJECTORY = (CURVATURE = c(x, t), VELOCITY = v(t) , CONTEXT). The variable "context" can be seen as a placeholder for other information that influences the corresponding action. This can be higher level information, as mentioned paragraph 3 ("Mapping SVEs to SAEs").

For the task of obstacle avoidance we need to distinguish between different kinds or classes of obstacles depending on their influence on the driving style. For example an obstacle that is on the street and not moving is calling for another action than an obstacle that is moving with the traffic flow, like another car ahead. A moving obstacle can further be subdivided into, as said before, one that is moving with the traffic flow, or in a different way, for instance a pedestrian crossing the street. Depending on the kind of obstacle, different parameters are needed to specify the appropriate action.

- SVE_OBSTACLE_FIX(DISTANCE, LOCATION, TIME_TO_CONTACT, CONTEXT)
- SVE_LEADER_CAR(DISTANCE, LOCATION IN THE LANE, TIME_TO_CONTACT, HEADING_DIRECTION, STOP LAMP STATE, CONTEXT).
- SVE_MOVING_ELSE(DISTANCE, LOCATION, TIME_TO_CONTACT, HEADING_DIRECTION, CONTEXT).

Finally an SVE called "traffic_law", according to driving condition "stopping, slowing down" shall be specified. For this traffic sign (or traffic light) detection and recognition is required.

- SVE_TRAFFIC_LAW(SIGN | LIGHT, SEMANTICS, DISTANCE, CONTEXT).

These SVEs, although still rough, are sufficiently well defined to derive which information must be extracted from the visual sensors.

1. Curvature and length of the street lanes are fundamental information that require detection and further processing of the street borders in the camera images. Furthermore it might turn out, that the effects of perspective projection must be erased.
2. Depth information must be acquired, for calculation of distances and time-to-contact.
3. Street sign detection and recognition is required.

5. Learning

To cover the requirements of learning raised by the hierarchical structures presented above a set of learning methods should be employed:

- Simple correlation/regression-based learning;
- Model-based learning;
- Learning to combine agents (experts);

We are also planning to check suitability to driving tasks of several biologically-inspired learning algorithms with elements of unsupervised learning.

In simple correlation/regression type learning we will investigate (1) how much speed correlates with visual flow rate; (2) how much speed correlates with time to contact signal; (3) how much steering correlates with curved flow vectors on a curve, etc. If the correlations will be substantial we will include these parameters in speed and steering control rules. Here only more scientifically interesting

correlation trials are outlined. More of these sorts of correlations would be needed to develop feedback control rules for velocity and steering in the set of situations defined by structured action events.

Model based learning will be needed for action events having predefined time course, e.g. for stopping or obstacle handling. We know that stopping requires dropping speed from the current value to zero at the predefined point. This requires choosing a model structure (some class of functions), and learning parameters of those functions depending on current speed and distance to the stopping point. For obstacles we know that we need change the position on the road (as a function of a distance to an obstacle, position on the road of an obstacle and speed of a car), then drive by an obstacle, and change back to the previous lane. Here again function parameters have to be tuned. Predictive aspects here are included in model structure itself, where the whole action is described as a unity and contains an action plan. Obviously feedback control rules mentioned above are needed on top to adhere to the plan.

To perform driving action, simpler controls have to be combined to perform bigger action sequences. E.g. actions proposed by straight driving should be combined with obstacle handling if there is an obstacle on the straight road segment, or actions proposed by tailgating should be combined with feedback control to stay on the road on the curve. Here several approaches would be used. One needs to distinguish which actions are additive, and combine those by simple summation. If simple summation is not sufficient then more complex methods shall be used. Specifically, Committee Machines are considered as an approach to combine control from several agents. Here returning to the tailgating on the curve example, a decision unit of a committee could be trained to distinguish in which proportion to include control suggested by tailgating, and in which proportion algorithms of taking the curve autonomously should be used.

For some control subtasks, specifically following the curvature of the road, specific biologically-inspired control methods using receptive field representations will be analyzed. Here in the procedure of human driving receptive fields will be created representing vision information. After the receptive fields have formed, this information will be used to control curve taking. This is an alternative approach, more black-box type as compared to explicit SVEs-SAEs definitions we were doing before.

Predictiveness of the approach is in interpretation of the visual scene as a unity, where lower fragments represent nearer future, while upper fragments represent more distant future. Though the approach is different from the previously defined SVE-SAE ideology, but for some situations these approaches could be used, and combined with other types of action experts using e.g. earlier mentioned Committee Machines.

Other variety of biologically-inspired method, proposed by participants of a project as a possible way to learn actions includes self-organizing network, where prototypical 2D visual scenes are associated directly with actions. This is close to learning by an example which could be a feasible method in learning to drive where it is not easy to otherwise systematize variety of possible visual inputs and relation of those to actions.

6. Decision Making issues

Decision making is a substantial component of driving, and is specifically required for the intentional driving conditions described above: obstacle avoidance (if several obstacles are possible at the same time), lane changing, overtaking, corner taking. The first three conditions usually require decision in situations where uncertainty is introduced by failing to evaluate presence of other vehicles (obstacles)

e.g. in overtaking with limited visibility, or failing to evaluate other drivers' behavior (e.g. in lane changing). The corner taking requires decision at the level of the travel plan of the driver. A usual way to handle uncertainty concerning actions of other traffic participants is by developing several parallel scenarios and evaluating measures of utility and measures of dangerousness (evaluated by time to contact or similar measures), and acting not exceeding dangerousness threshold (Holzman et al, 2005). Decisions about the driving plan (corner taking) can be dealt inside route choosing scenario. Although, in this project we are going to concentrate on automatic driving conditions, where the higher level knowledge required for making decisions can be neglected.

7. References

Gat, E. (1998). On Three-Layer Architectures. In Artificial Intelligence and Mobile Robots, Kortenkamp D., Bonnasso R. &Murphy R. (eds), MIT Press, Cambridge, p.195-210,

Thrun, S., Montemerio, M., Dahlkamp, H. et al (2006). Stanley: The Robot that Won the DARPA Grand Challenge, Journal of Field Robotics 23(9), p. 661-692.

Holzmann, F., Kolski, S., Sulzmann, A., Spiegelberg, G., Siegwart, R., and Bubb, H (2005). Improvement of the driving safety using a virtual driver. Intelligent Transportations Systems. Proceedings IEEE, p. 546- 551