Advanced driver assistance functions need anticipatory map data and foresights to work efficiently. The Electronic Horizon (EHR) component provides these required information by integrating topographical and digital map data with the vehicle position. Being a central component deployed at various instances over the vehicle Boardnet, the Electronic Horizon serves a multitude of different client/customer functions. As each of these customers has its specific functional and API-related requirements, deployment of the Electronic Horizon involves a complex variant handling. With topics such as Autonomous Driving and Connected Driver Assistance Systems ahead, this complexity is likely to grow even stronger and implies that EHR will need to process a bigger amount of data and offer a flexible and generic interface.

To address the difficulties mentioned previously and to prepare the future of the EHR component we propose to adopt a data oriented approach in designing EHR using ontologies. This new approach will provide a uniform model to represent, store and extract map data in an abstract and uniform format. It offers to the EHR component the possibility to be equipped with a generic and flexible interface via a query language in an independent way from its customer functions or the provider of the information. Additionally, a reasoning system can be attached to this model and offer to EHR the possibility to infer and extract new knowledge from the existing map data. Finally, the ontology approach will allow us to update the internal architecture and design of EHR, add new data and make them available to our customer functions without affecting them.

So knowing all these advantages, the project should cover the following:

- Build the map data ontology-based Knowledge Base (KB) system needed for EHR using the Basic Formal Ontology (BFO) process.
- Evaluate and choose a reasoning system (HermiT, Pellet, etc.) connected to KB to validate the ontology model.
- Design a generic and flexible interface for EHR based on an appropriate query language (ΣQL, SPARQL, etc.) and test it with some customer functions like SLI and GWW.
- Implement a C++ prototype of the model with its interface and use it with some real and synthetic map data.

Good implementations skills in Java and/or C++ and knowledge of Semantic Web technologies (OWL, SPARQL, . . . ) are required.

Contact

Birte Glimm
Tel.: +49 (731) 50 24 125
Birte.Glimm@uni-ulm.de

Institute of Artificial Intelligence
Building O27
Room 448