Towards Simulation of Ambient Intelligence in Autonomous Vehicles using Car Racing Games

Markus Schatten, Bogdan Okreša Đurić, Igor Tomičić
Artificial Intelligence Laboratory
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia
{markus.schatten, dokresa, igor.tomicic}@foi.hr

Abstract. *An initial implementation of an ambient intelligence (AmI) system for autonomous vehicles (AVs) inside a car racing game (namely TORCS) is presented and analyzed. The proposed system is based on a preferential routing and re-routing algorithm and a knowledge-base implemented in $\mathcal{F}$LORA-2 . The routing system is implemented as a module to B.A.R.I.C.A., an open artificial intelligence system, while the client based on the TORCS racing game is an application in B.A.R.I.C.A. terms. The current version of the system supports routing and rerouting the AV agent based on its understanding of the map and set preferences.*

Keywords. autonomous vehicles, computer games, ambient intelligence, artificial agent, smart city

1 Introduction

Everyday environments are made intelligent and sensitive to us (Aarts and Wichert, 2009; Cook et al., 2009) using ambient intelligence (AmI). On the other hand, autonomous vehicles (AVs) are a disruptive technology that is slowly entering one important aspect of our lives especially with regard to transport (Fagnant and Kockelman, 2015; Gerla et al., 2014). The aim of this paper is to make a first step towards implementing AmI in AVs – i.e. making AVs sensitive to its passengers and its surroundings. Whilst there are various possible systems of AmI in AVs, herein we will focus on preferential routing and re-routing based on the passengers preferences (Schatten et al., 2011).

In order to test and simulate a possible AmI system for AVs we will use an open source racing game called TORCS[1]. We argue that car racing games and especially open source games present the ideal environment (apart from building your own 3D simulator) for such tasks. Herein we present a work-in-progress system called Barely an ARtificial Intelligence CAr (B.A.R.I.C.A.) for which we have built a proof-of-concept application connecting a preferential routing knowledge base to a machine learning based agent, which allows simulating an AV driving in TORCS.

The rest of this paper is organized as follows: firstly in 2 we provide an overview of related work. Then in 3 we showcase our initial implementation of the system and discuss possible future steps to be taken in the implementation of the system. Finally, in 4 we give a conclusion and provide guidelines for future research.

[1] Available at http://torcs.sourceforge.net/

2 Related Work

In the early nineties, Weiser stated that "*the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*", (Weiser, 1991; Weiser, 1993), setting the background for Ubiquitous Computing - the idea of pervasive, but non-intrusive computing.

A group of authors describe ambient intelligent environments as "*unobtrusive, interconnected, adaptable, dynamic, embedded and intelligent.*" (Sadri, 2011), presenting a comprehensive survey of ambient intelligence with the overview of application domains such as home, health care, assisted living, shopping, recommender systems, business, museums, tourism, institutions, etc.

The principle idea of AmI stems from the concept of artificial intelligence (AI) applied to and embedded in various elements of an environment. Such environments, named intelligent virtual environments (IVEs), comprise autonomous intelligent artificial agents both as actors physically situated in the said environment, either in their free-to-roam form, or stationary, and as actors that are not physically situated in the given environment (Okreša Đurić et al., 2018).

An IVE (Barella et al., 2012) is therefore a specific variety of a multiagent system (MAS), in the context of information communication technology (ICT), wherein agents are considered not only as software or virtual entities, but as gadgets that help digitize human agents and foster their interaction with the rest of the intelligent environment, i.e. agents that virtualize human agents for the sake of interaction with the IVE they are situated in.

Since AmI is a particular application domain of MAS, it can be modeled using the models applicable to MASs and large-scale multiagent systems (LSMASs), such as those recently published and described in (Mayr et al., 2016; Okreša Đurić, 2017).

In the context of AVs, the majority of conducted research is steered towards improvement of individual car agents and their interaction with their environment using various sensors and actuators paired with advanced inference engines (for example, (Reina et al., 2016). This paper provides arguments towards the rather opposite, yet unmistakably compatible, perspective promoting the importance of AmI in combination with AVs, whose combination is known as the concept of intelligent traffic systems. AmI is the driver of research towards AVs actively communicating with their environment, especially under the presumption that those environments develop into intelligent environments comprising both human and artificial agents.

Due to the expected low latency in the message exchange process between agents located in an IVE, and quick and easy data transfer in such an environment in general, backed up by unambiguous meaning of the transferred data, communication modeling is one of the key features to be observed. Some recently published research provides interesting insight towards modeling communication between human and artificial agents using conceptual and semantic models (Okreša Đurić et al., 2018; Okreša Đurić and Maleković, 2018). Such an approach, using semantic modeling and utilizing ontologies, for the purposes of AmI paired with AVs is what this paper is a step towards.

3 Implementation

The proof-of-concept system is implemented as part of the B.A.R.I.C.A. autonomous vehicles subsystem[2] which is shown in Fig. 1. It consists of a back-end knowledge-base module that uses $\mathcal{F}$LORA-2 to provide AmI routing as well as a front-end client application based on Snake Oil, a Python library for interfacing with TORCS[3] or more precisely the MADRaS[4] simulator's implementation of the library (Kaushik et al., 2018).

The B.A.R.I.C.A. AI platform is an attempt to provide a cloud based interface between AI developers (general purpose modules and knowledge bases) and AI users (domain-specific applications). It is currently a work in progress, but already consists of a cloud platform (cro. BARICA-u-oblacima), as well as four identified application domains: (1) virtual assistants (cro. BARICA-asistentica), (2) Internet of Things (IoT) (cro. BARICA-sobarica), (3) autonomous vehicles (cro. BARICA-vozačica), and (4) computer games (cro. BARICA-gejmerica).

[2] Available at https://github.com/AILab-FOI/B.A.R.I.C.A.
[3] Available at http://scr.geccocompetitions.com/
[4] Available at https://github.com/madras-simulator/MADRaS

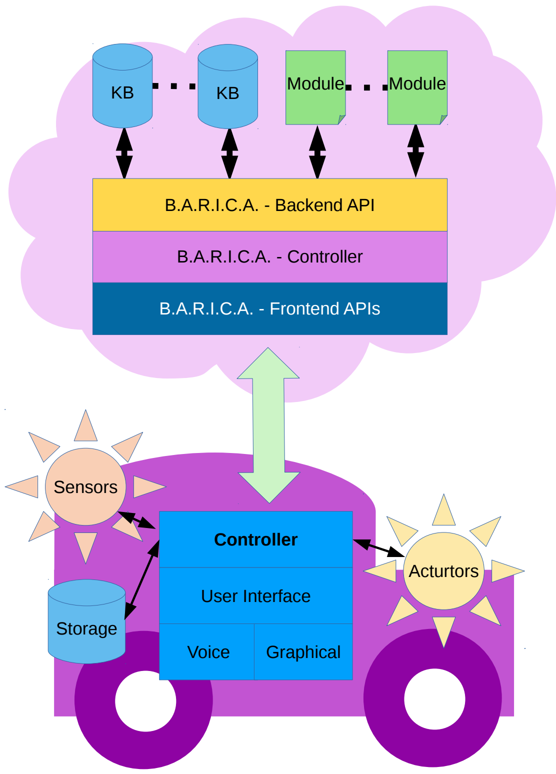

Figure 1: Architecture of the B.A.R.I.C.A. autonomous vehicles subsystem - a work in progress

Figure 2: Initialization screen of the TORCS racing game based simulator

The cloud platform features a back-end module implementation application programming interface (API) that allows AI modules to be implemented in any programming technology given that they can be run in a shell, accept command line arguments in a specified order and provide output in a specified JavaScript Object Notation (JSON) or streaming format. Modules can be defined as accepting a stdin stream of data, providing a stdout stream of data (i.e. if they are both they behave as pipeline filters over a netcat based network), or work in an asynchronous fashion.

For the sake of this paper we have implemented a module named TORCS-map that is based on the preferential routing and re-routing algorithms presented in (Schatten et al., 2011) and implemented in $\mathcal{F}$LORA-2 and Python. The module accepts three arguments two of which are the starting and destination nodes of a map and the third is a list of filters or ambient preferences of the user.

Besides the back-end API the cloud platform also provides a number of front-end APIs. At the time of writing this paper, the APIs includes a representational state transfer (REST) interface, a Web Socket interface as well as a Slack chatbot interface. These APIs allow for the implementation of specific applications which use various modules provided by the platform.

Figure 3: Autonomous vehicle driving in the TORCS racing game based simulator

To this end, we have started developing an application based on TORCS that uses the REST front-end API. In the current version, the application is able to steer the virtual car in TORCS as well as to query B.A.R.I.C.A. cloud platform for ambient and routing information.

In this way the AV agent can query the preferential re-routing module based on user-preferences to generate a map route that is in accordance with passengers' preferences. For example, passengers might prefer to use a road with specific attributes (i.e. forest rich roads, roads that are low in traffic or roads that do not require tolls). An example query might look follows Schatten et al., 2011:

```
flora2 ?- newmodule{ pref },
preference_path( a, g,
 [type->highway, landscape->farmland],
 ?p ),
erasemodule{ pref }.
?p = [a, b, e, g]
?p = [a, d, f, h, g]
```

In this example, firstly a new dynamic module is created to minimize computation requirements, then a preference path is sought (see Schatten et al., 2011 for a detailed explanation of how the `preference_path/4` predicate is implemented) where `a` and `g` are starting and destination nodes on the map respectively and `type` and `landscape` are road attributes of passengers' preference, and in the end the module is erased to minimize memory consumption. Variable `?p` returns two possible routes that obide to the given preferences.

Our next steps will be to map these requirements in two ways: (1) to annotate the racing game's map semantically with various (potentially dynamic) attributes that can be queried by the re-routing knowledge base, and (2) to make the AV agent sensitive to the needs and preferences of the passengers. The first part is the more complex and time consuming since it will include a definition of attributes and manual annotation of each racing map from TORCS. The second part can be simulated easily by defining the passengers by the start of the simulation or by querying the preferences on demand.

4 Conclusion

In this paper we have presented and analysed an initial implementation of an AmI system for AV inside a car racing game. We have built a proof-of-concept application which is able to connect a preferential routing knowledge base to a machine learning based agent that allows simulations of driving an autonomous vehicle. In the current version, the application is able to steer the virtual car as well as to query a cloud platform for ambient and routing information.

The implementation of the proposed system is based on a preferential routing and re-routing algorithm, and a knowledge-base implemented in $\mathcal{F}$LORA-2 within the system's back-end. The routing system is implemented as a module to B.A.R.I.C.A., a custom open artificial intelligence system that we have implemented as well. The front-end client application is based on the Python library called Snake Oil, which provides an interface with MADRaS, a Multi-Agent Autonomous Driving Simulator built on top of TORCS, which is an open-source 3D car racing simulator game.

Future research might include more complex interaction between the AV and its environment in the context of AmI and heterogeneous preferential choices. For example, the map might contain various semantic meanings integrated into objects that would interact with person's registered preferences, such as favorite POIs (points of interests), quality and nature of roads, passing sightseeing possibilities, etc.

References

Aarts, E. & Wichert, R. (2009). Ambient intelligence. In *Technology guide* (pp. 244–249). Springer.

Barella, A., Ricci, A., Boissier, O., & Carrascosa, C. (2012). MAM5: multi-agent model for intelligent virtual environments. In *10th european workshop on multi-agent systems (eumas 2012)* (pp. 16–30).

Cook, D. J., Augusto, J. C., & Jakkula, V. R. (2009). Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, *5*(4), 277–298.

Fagnant, D. J. & Kockelman, K. (2015). Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, *77*, 167–181.

Gerla, M., Lee, E.-K., Pau, G., & Lee, U. (2014). Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In *2014 ieee world forum on internet of things (wf-iot)* (pp. 241–246). IEEE.

Kaushik, M., Prasad, V., Krishna, K. M., & Ravindran, B. (2018). Overtaking maneuvers in simulated highway driving using deep reinforcement learning. In *2018 ieee intelligent vehicles symposium (iv)* (pp. 1885–1890). IEEE.

Mayr, H. C., Al Machot, F., Michael, J., Morak, G., Ranasinghe, S., Shekhovtsov, V., & Steinberger, C. (2016). HCM-L: Domain-Specific Modeling for Active and Assisted Living. In D. Karagiannis, H. C. Mayr, & J. Mylopoulos (Eds.), *Domain-Specific Conceptual Modeling* (pp. 527–552). Cham: Springer International Publishing. doi:10.1007/978-3-319-39417-6_24

Okreša Ðurić, B. (2017). Organisational Metamodel for Large-Scale Multi-Agent Systems: First Steps Towards Modelling Organisation Dynamics. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, *6*(3), 17. doi:10.14201/ADCAIJ2017631727

Okreša Ðurić, B. & Maleković, M. (2018, September 6). How to Manage Knowledge With Domain Specific and General Conceptual Modelling Examples. In E. Bolisani, E. Di Maria, & E. Scarso (Eds.), *Proceedings of the 19th European Conference on Knowledge Management* (Vol. 2, pp. 615–622). European Conference on Knowledge Management. Reading, UK: Academic Conferences and Publishing International Limited.

Okreša Ðurić, B., Rincon, J., Carrascosa, C., Schatten, M., & Julian, V. (2018). Mambo5: A new ontology approach for modelling and managing intelligent virtual environments based on multiagent systems. *Journal of Ambient Intelligence and Humanized Computing*.

Reina, G., Milella, A., Rouveure, R., Nielsen, M., Worst, R., & Blas, M. R. (2016). Ambient awareness for agricultural robotic vehicles. *biosystems engineering*, *146*, 114–132.

Sadri, F. (2011). Ambient intelligence: A survey. *ACM Computing Surveys (CSUR)*, *43*(4), 36.

Schatten, M., Bača, M., & Koruga, P. (2011). The transportation paths routing semantic description. In *Transactions on transport systems telematics & safety*. Silesian University of Technology.

Weiser, M. (1991). The computer for the twenty-first century (pp. 94-100). *Scientific American, September Issue*.

Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM*, *36*(7), 75–84.