Towards autonomous embedded systems

Sagar Behere, Martin Törngren, Jad El-khoury, DeJiu Chen
KTH, Stockholm, Sweden
{behere,martint,jad,chendj}@kth.se

***Abstract*—Machines incorporating embedded systems display a trend towards increasing autonomy. In this position paper, we outline an approach for introducing autonomy in embedded system architectures. The approach involves the creation of an artificial consciousness within the machine. We propose that the artificial consciousness may be represented in the form of domain specific reference architectures. We illustrate the approach with the aid of a validated reference architecture for cooperative driving.**

***Keywords*-Autonomy, artificial consciousness, embedded systems, reference architecture**

I. Introduction

Two trends are apparent in the design and construction of many machines that surround us:

1) There is an increasing usage of electronics, computers and software within the machines (a.k.a embedded systems)
2) There is an increasing desire to make the machines autonomous, where autonomy is defined as 'operation without direct human intervention'.

Therefore, a question that will gain increasing importance is:

What should the hardware and software architecture of a machine look like, so that autonomous operation is easy to achieve?

This work proposes an approach for answering the above question. The approach is valid under a specific set of preconditions and constraints, which are also described in this work.

The proposed approach is still a work in progress. Nevertheless, an application of the approach in the form of an automotive reference architecture for cooperative driving[3, 4] and two instantiations thereof has already been made. This text describes the approach, the reference architecture and its instantiations as well a non-exhaustive list of questions whose solutions need to be found, to develop the approach further.

II. Proposed approach

A. Autonomy, intelligence and machine consciousness

One of the principal aids to achieving machine autonomy is machine intelligence. Intelligence can be defined as the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal[1]. There may be several ways to construct a machine so that it *displays* intelligence. The display of intelligence can be evaluated solely by observing external behavior, without concerning oneself with the mechanisms within the machine that generate the displayed intelligent behavior (If it looks, walks and quacks like a duck...). If the external behavior of the machine is indistinguishable from that which would be generated by an intelligent entity, then the machine may be considered as intelligent[7] (a.k.a the 'Turing test'). The external display of intelligence must, however, be differentiated from the internal mechanisms in the machine that give rise to that behavior.

For the purpose of this text, we constrain[1] the term 'consciousness' as the quality or state of being aware of something within oneself. This leads to the question, "Who, or What, is it that is being aware?" This question must be answered. We will answer it in section II-C.

It must be pointed out that consciousness does not imply intelligence and intelligence does not imply autonomy. The converse is also true: intelligence does not imply consciousness, nor does autonomy imply intelligence.

We claim that **the combination of consciousness and intelligence is a useful engineering method to achieve machine autonomy, when the machine is a system composed out of multiple sub-systems**. In the context of such a machine, our notion of consciousness can be reified by a distinct sub-system that

1) is aware of the overall purpose(s) of the machine, can understand the short term goals of the machine's user and the behavior expected for the fulfillment of those goals
2) recognizes the presence of the other sub-systems of the machine, and knows how they should interact to generate expected machine behavior
3) understands the environment that the machine operates in, and the expected interactions of the machine with its environment
4) is capable of interpreting the commands of the machine's user and orchestrating behavior of other sub-systems in order to execute those commands

Such a consciousness subsystem may contain algorithms for machine learning and intelligence, which could be combined with elements of intelligence present in other sub-systems. With such a construction, the machine may be deemed to be equipped with mechanisms for understanding the commands of its user, and for executing those commands. This is nothing but the essence of machine autonomy.

[1] The terms 'conscious' and 'consciousness' have been expounded with significantly greater meaning elsewhere. See for example Van Gulick [8]

B. Pre-conditions and constraints

Our approach to machine autonomy assumes the existence of the following pre-conditions and constraints

- Embedded computer systems and software are the primary means to generate and control machine behavior
- The embedded systems are the sole focus area for the achievement of machine autonomy. All efforts to realize machine autonomy will be concentrated on the embedded hardware and software only.
- The embedded systems within a machine are organized into sub-systems. Each sub-system consists of one (or a small group of) computers.
- There is constant communication between the embedded sub-systems. This communication may be used for exchanging data and/or altering the specific software functions being executed by a sub-system.
- There exist legacies of proven sub-system designs, together with strong reasons for minimizing changes to these sub-system designs.

C. The Self and progressive autonomy

Our approach to embedded systems autonomy consists of introducing into the system architecture a sub-system that reifies the notion of machine consciousness. We denote such a sub-system by the term 'Self'. It is this Self that lends an identity to the machine. Users interacting with the machine are in fact interacting with the Self. Earlier in section II-A we asked, "Who, or What is it that is aware within the machine?" The answer is: the Self. In this way, our approach endeavors to partially mimic the notion of consciousness and self-awareness in human beings.

From the architectural perspective, some interesting questions are:

- What should be the structure and interfaces of the Self?
- What are the patterns of interaction between the Self and the other sub-systems?
- How should the sub-systems be designed so that they can interact more easily with the Self?
- What particular sequence of design iterations should be followed to evolve existing architectures towards those that incorporate and utilize the Self?
- How are cross-cutting extra-functional properties like system safety, reliability, error management etc. affected by the Self? Can the presence of the Self be exploited to favorably affect these properties?

Introduction of the Self into the architecture needs to be complemented by examination of aspects related to formal representations of system construction and desired behavior. Formal representations provide the Self with the knowledge necessary for appropriate reasoning and control of the system. The representations in turn will be affected by the algorithms used by the Self for reasoning and decision making. Given the dependency of these aspects on the domain, task and implementation specific details, it might not be possible to specify a sufficiently general solution that works for all types of embedded systems. General solutions however, could be in the form of *domain specific reference architectures* that are instantiable for specific use cases. Reference architectures[5] are essentially proven solution templates and patterns that are useful for solving a specific category of problems. An example of a reference architecture based on our approach is given in section III.

One particular salient benefit of our approach must be highlighted. The benefit is that the approach enables progressive autonomy. Progressive autonomy means that the autonomy of the system is gradually increased over successive design iterations. This implies that the degree of human intervention needed to operate the machine decreases over successive product versions. Progressive autonomy is important for two reasons

1) It enables cautious increase in capabilities of a function that is inherently susceptible to uncertainty and errors that have safety consequences.
2) Existing and legacy systems can be the starting point. These can be gradually evolved towards autonomy, making large design changes unnecessary. This is appreciated by commercial companies whose products are already in the market (example: automotive manufacturers).

The reason why progressive autonomy is enabled by our approach is that the capabilities of the Self (together with the architectural changes necessary to take advantage of those capabilities) can be developed in an incremental fashion. At its simplest, the Self need be no more than a passive component that is fed some status data by the rest of the subsystems. It need take no active role in determining and affecting the system behavior, but could be used, for example, to provide diagnostic information and/or warnings. Next, the Self may be allowed to interpret user inputs as intentions to achieve specific system behaviors, while still permitting the Self no control over the other sub-systems. The inputs and internal reasoning performed by the Self could be logged over time to validate the correctness of the related algorithms, and only then could the Self be granted executive powers that affect the functioning of the system.

III. Application example: A reference architecture for cooperative driving

This section gives an example of adding a Self as an additional sub-system to a set of existing sub-systems. The example also demonstrates a recursive characteristic of the approach: the Self is implemented as an additional sub-system which in turn consists of sub-subsystems. One of the sub-subsystems is a Self (sub-Self?)!

A. A Self for cooperative driving

The electrical/electronic (E/E) sub-systems of a modern automobile meet the constraints listed in section II-B. For the purpose of introducing autonomy, an automobile can be considered as a set of interconnected, embedded computer (sub-)systems, each of which has a specific purpose. Our approach to autonomy suggests the addition of another sub-system (the Self) that can function as the system's consciousness. To illustrate this approach, we considered the specific problem of creating autonomous motion under cooperative driving conditions. Cooperative driving conditions are those where continuous wireless communication exists between a vehicle and its surroundings, which consist of the local road infrastructure as well as the other vehicles in the vicinity. The Self of the autonomous system should then understand that

1) the purpose of the system is autonomous driving (under cooperative driving conditions) and the short term goals of the system are to navigate the vehicle in a specific environment
2) there are other sub-systems in the vehicle (like the engine, the brakes and the transmission) which have defined roles and that the correct interaction of these sub-systems will generate the desired behavior
3) the environment of the vehicle consists of objects that include other vehicles and road infrastructure (traffic lights, speed limit signs etc.) and how the vehicle should react to the presence/absence of these objects

When given the appropriate commands by the user, the cooperative driving Self should be able to correctly interpret the commands and orchestrate the other sub-systems to realize safe, cooperative driving. This problem is domain specific and sufficiently detailed to generate a reference architecture for the Self, as mentioned in section II-C.

Accordingly, a reference architecture for cooperative driving was created, which is described in [4]. This reference architecture was instantiated[6, 2] on two separate occasions, one of which was the Grand Cooperative Driving Challenge (GCDC) 2011. The GCDC consisted of vehicle platooning on public roads. An instantiation of the reference architecture was installed on a Scania R730 commercial truck, and the modified truck successfully completed the driving challenge. A second instantiation of the reference architecture was installed on a Scania R480 commercial truck, which was then utilized during further cooperative driving demonstrations. The two instantiations differed in capabilities and had very little in common.

Thus, on two separate occasions, the concept of adding a Self i.e. consciousness sub-system (for the purpose of autonomous cooperative driving) was demonstrated to produce desired system behavior.

B. A Self within a Self

The introduction of a Self happens in the form of an extra sub-system in the system architecture (see Figure 1). If we denote this extra sub-system as 'Self0', then it is entirely possible that Self0 itself comprises of multiple sub-subsystems and that one of these sub-subsystems is a Self (denoted 'Self1' in Figure 1) and so on. Thus, the approach demonstrates recursive characteristics.

In the case of our particular reference architecture for cooperative driving, one of the key architectural elements is a Self, present in the form of a Supervisor component. Specifically, *"..It is the supervisor that encodes an understanding of the various architectural elements, their capabilities and limitations. Thus, it is the supervisor that is aware of the presence of the world model, the control and other elements*[of the reference architecture] *and how they must function in order to generate specific behaviors of the cooperative driving system. The supervisor "knows" what behavior is expected of the cooperative driving system in a given context and uses them to achieve the expected behavior. The elements in turn pass on all unknown inputs, locally unresolvable errors and requests to the supervisor and expect instructions on how they should proceed."*[4].

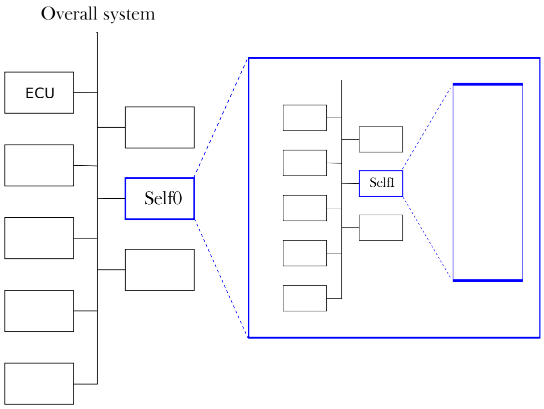

Figure 1. Recursive Selves

Thus the reference architecture provides a blueprint for a Self (Self0, as per Figure 1) that generates autonomous behavior in the vehicle. Within the reference architecture, there is another Self (Self1, as per Figure 1) that generates the desired behavior of the reference architecture. This "second-level Self" illustrates precisely the same principles of creating autonomous embedded systems as outlined by our proposed approach.

IV. Conclusions and future work

We have outlined an approach towards embedded systems architecture to achieve machine autonomy. Parts of the approach have been validated in the context of cooperative driving for which a reference architecture was developed.

Further work is required to elaborate the approach and to encompass other autonomy settings. Introducing autonomy will also require specific efforts for addressing safety and reliability aspects. In particular there is a dichotomy between the determinism required by safety practices vs. the dynamic behavior which is inherent in autonomy. Safety standards, legislation and supporting technology all need further work

References

[1] J.S. Albus. Outline for a theory of intelligence. *IEEE Transactions on Systems, Man, and Cybernetics*, 21 (3):473–509, 1991. ISSN 00189472. doi: 10.1109/21.97471. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=97471.

[2] Sagar Behere. Scoop Technical Report: Year 2011. Technical report, KTH Royal Institute of Technology, Stockholm, 2011. URL http://kth.diva-portal.org/smash/get/diva2:567028/FULLTEXT01.

[3] Sagar Behere. *Architecting Autonomous Automotive Systems*. Licentiate thesis, KTH, Stockholm, 2013.

URL http://kth.diva-portal.org/smash/record.jsf?searchId=6&pid=diva2:615888.

[4] Sagar Behere, Martin Törngren, and Dejiu Chen. A reference architecture for cooperative driving. *Journal of Systems Architecture*, 2013.

[5] Philippe Kruchten. *The Rational Unified Process*. Rational Software White Paper. Addison-Wesley, 2003. ISBN 0321197704.

[6] Jonas Må rtensson, Assad Alam, and Sagar Behere. The Development of a Cooperative Heavy-Duty Vehicle for the GCDC 2011: Team Scoop. *IEEE Transactions on Intelligent Transportation Systems*, 13 (3):1033–1049, September 2012. ISSN 1524-9050. doi: 10.1109/TITS.2012.2204876. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6236179http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6236179.

[7] A. M. Turing. Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL http://mind.oxfordjournals.org/cgi/doi/10.1093/mind/LIX.236.433.

[8] Robert Van Gulick. Consciousness. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 2011. URL http://plato.stanford.edu/archives/sum2011/entries/consciousness/.