Context Aware Software Stacks for Mobility

Composive.ai Overview

Naci M. Dai
Eteration
Istanbul, Turkey
naci.dai@eteration.com

Deniz Memis
Eteration
Istanbul, Turkey
deniz.memis@eteration.com

Burak Saglam
Eteration
Istanbul, Turkey
burak.saglam@eteration.com

***Abstract*—Humans are very good at adapting to and driving in very different conditions and environments. Complex AI based software such as Autonomous vehicles (AV) are biased by the training data and they do not do well in different contexts. Environment perception and context awareness can be used to adapt and deploy complex networks of AI systems that are expected to perform safely on many different environments and edge cases. Typical Autonomous Drive (AD) software stacks provide localization, object detection and tracking using an array of sensor data. Additional sources of information such as V2X connectivity, passenger behavior, person-to-device mapping, urban context, degradation of traffic etc. complements the sensor data that can be used to describe a context. This context is used to predict and adapt the system of software and applications that are deployed to the vehicle.**

Context Aware Software Stack for Mobility (CASSM) describes a model based approach for definition of a software stack for complex systems such as those found in autonomous vehicles. Machine Learning is used to detect a context and adapt the software stack, which can then be used to identify and deploy services to autonomous vehicles. CASSM will be an open source in-vehicle delivery of context-driven services and applications. This paper describes an architecture to define context-aware personalized & dynamically provisioned capabilities, applications & services to autonomous vehicles.

***Index Terms*—Autonomous Vehicles, Context Awareness, Adaptive Software Stacks, Machine Learning**

I. Introduction

Experienced human drivers are good at adapting to different driving conditions and environments. They can instinctively transfer their earlier learning to the current driving context. The advent of big data and availability of computing allowed statistical deep learning methods to power innovation in many applications including AD. AD software are complex systems of various interconnected AI and deterministic modules. These software do very well in the context that they are designed and trained for, but perform poorly in other environments. Starting with Darpa challenges, statistical learning methods have been used in implementing very successful AD software. The quality of these solutions depend on their statistical models and the big data that is used to train them. These systems are very good at learning, perceiving and adapting to the environment. These cars can successfully, and safely drive on highways and in the cities, interacting with the environment, the traffic, other cars and pedestrians.

There are still many challenges to this learning process as huge learning data leads to many difficult to explain biases and edge cases where the algorithms come up with unexpected results. Deep neural networks learn from the data to cleanly separate the dataspace for the intrinsic dimensionality (nicely demonstrated by Colah's Neural Networks, Manifolds and Topology [1]), and require careful engineering and modelling to do this efficiently. These systems have very little or no contextual capability, or ability to abstract knowledge or reason about the drive [2]. DARPA recently announced the Next AI challenge, and is focusing its investments on a third wave of AI that brings forth machines that understand and reason in context [3] (See Fig.1.).

The Contextual Adaptation means the systems construct explanatory models for classes of real world phenomena and the explanations and the context can be used to adapt both the learning models and in our case the overall AD software stack.

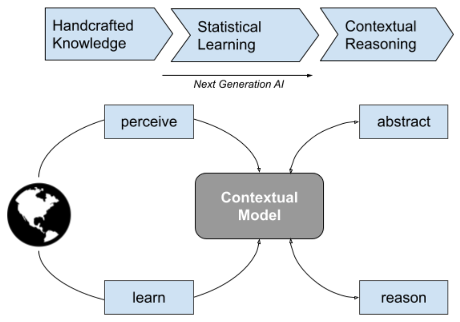

Fig. 1. DARPA - The Next Wave of AI (Figure Adapted from John Launchbury)

Humans are very successful at understanding the context such as parking, urban driving, or driving fast on a highway, abstracting experiences and transferring ideas and reacting appropriately. However, for AD software the design, capabilities, sensors and actions taken for these contexts can differ significantly. Even urban driving in a well organized city versus a city with narrow streets (i.e. Istanbul) will require training with completely different datasets, probably using different models.

We can define the context as the set of all the inputs from the sensors, and additional inputs from the environment, people's behavior, even if they are not used in the AD stack. The action of driving takes place and all these inputs interact with the system and change its behavior. It involves the driver, the vehicle physics, its parts, brakes, steering, mechanical components, the weather condition, the destination etc. Context-awareness means that a service, although given the same request parameters, is perceived differently with respect to a given context [4]. In autonomous driving context-awareness means, (1) Semantics and modelling the context for which AD software is trained and designed for and (2) Using the information about the context to adapt how AD perceives, predicts, and acts.

II. Contextual AI and Context Awareness for Autonomous Vehicles

A. Contextual AI

Contextual AI is focused on providing a service or information to the user that is relevant to the acquired (explicit or implicit) context information [5]. This approach allows contextually-aware systems to provide tailor-made services and solutions under varying circumstances. There are mainly four pillars to Contextual AI, based on these the system must be [5]:

- Intelligible: Ability to explain what it knows, how it knows and what it is doing.
- Adaptive: Ability to meet user's needs and expectations under varying environmental circumstances.
- Customizable: Ability to be controlled fully by the user.
- Context-aware: Ability to perceive at a human level.

Even for the most successful AD software, its capabilities are still bounded by its compatibility to the current context and the constraints it was trained for, making such systems heavily dependent on human intervention when it is faced with individual cases where the inferences and plan fail.

B. Explainable AI

Like all other safety and sensitive applications of AI, Explainability of AD software is strongly related to context-awareness. AD software travelled successfully for billions of kilometers but when they fail, the reasons are not easily interpretable by a human. Interpreting and explaining is important to gain trust into the prediction, and to identify potential data selection biases or artefacts. We need to know when an AD system is operating within the limits of its training. These limits identify contexts that the AD is designed to safely and reliably operate, and prevent dangerous inferences and outputs. For example, if the perception system cannot classify the intent of a cyclist, being able to explain why the input data led to an incorrect calculation will help in the adaptation of the underlying learning models. The explanations could be related to the structure or models of the system, of single decisions or cases, or generalize to the complete overall behavior. Samek et. al [6]. provide an extensive review of methods and applications of some of the most recent advances in AI explanations.

C. Context Awareness

Numerous context-aware systems have been presented in the literature, they albeit differ in the way context related data is acquired and utilized; the main commonalities of these systems' structure could be listed as below:

- Acquiring information.
- Manipulating the information (filtering, compressing, transforming) to contextually represent the system.
- Interpreting the context.
- Distributing information based on the contextual interpretation.

In the AV domain, these systems have been utilized in a variety of applications such as "route planning and learning [7]–[9], real time perception of traffic congestion [10], collision avoidance [11] and finding vacant car parks [12]". Considering these use cases that provide a service to the user and above mentioned system characteristics there are two key aspects relevant to context-awareness: First is the knowledge acquired about the context of the service to be provided and the second is to adjust the service to be contextually aware considering the information within the context.

The first absolute benefit of context-awareness for AV, as is true for all autonomous systems, is adaptation [13]. Adaptation implies modifying a service that will, given the same parameters, yield a relevant distinguished output depending on the context and the objective. This may include different realizations such as refining or tracking other information, invoking or inhibiting other services. The other and relatively more sophisticated benefit compared to other autonomous systems is proactivity, currently provided reactive services work upon the invocation of the user (a pull type service), on the other hand proactive services could deliver the information without any explicit action required by the user. Furthermore, these proactive services could keep delivering based on the predicted outcome of future context [13].

III. The Model Driven Adaptive Software Stack

Simon described means-end analysis with the notion of adaptation, which is finding the difference between an existing state and a desired state and then finding the correlating process that will erase the difference [14].

Our work focuses on software systems that are amenable to model-driven methods. It is possible to dynamically adapt these models to better address a given context. A large percentage of AD software is based on the Robot Operating System (ROS). ROS has the concept of computational nodes that can be connected with the data flowing between them. Many AD implementations involve sensor data, and higher level objects that are produced by sensor-fusion, prediction, planning and control tasks that are implemented with ROS. This is typically done with a code-first, development heavy practice.

RobMoSys attempts to address this difficulty by providing standards for composable models and Software for Robotics Systems. We are exploring RobMoSys model-driven approach to define models for the proven ROS based open-source AD frameworks such as Autoware [15], Baidu Apollo [16]. Refactoring existing systems as model-based software will provide fast and effective feedback for the functionality and also will prove the platform independence. The metamodel-driven approach, that composes the building blocks (or a combination of them) into various systems in a purposeful way, the context detection models and algorithms (ROS nodes) will be in use. According to context, these nodes will be used to create a context-level metamodel and will enable other nodes to be deployed and to be used.

The contextual adaptation of an AD software stack would change the active stack to better address the constraints and trade-offs of the context. Each stack model is mapped to a context that reflects the conditions of its training data. It is conceivable that one can define an “uber” model that would satisfy most contexts and there will be no need for adaptation, but this is probably not achievable with today’s understanding. Instead, we propose a pragmatic context-based adaptation approach that will identify a context based on the inputs of the system and dynamically change the software stack (a model-based computation graph) to satisfy the requirements in the new context (See Fig.2. & Fig.3.).

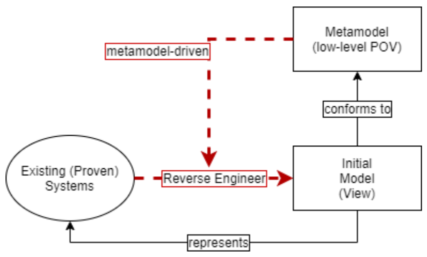

Fig. 2. Metamodel extraction from existing systems.

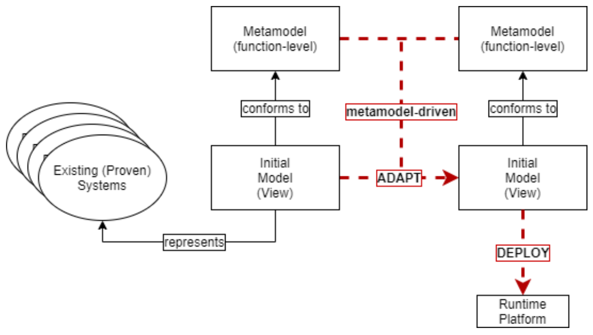

Fig. 3. Adaptation using metamodel-driven feedback.

Similar approaches produced promising results for other problems. Odena et.al [17] use reinforcement learning to change model behavior at test-time. They use a composer model, which consists of a set of modules and a controller network. The composer network is a graph of deterministic and stochastic nodes made of neural networks that are organized into ‘metalayers’. The composer network is very similar to the typical software stack like the ones used by Autoware (See Fig.4.).

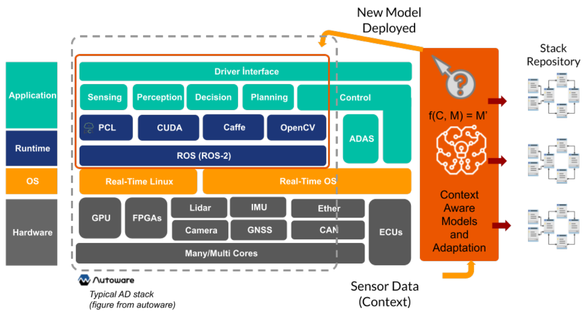

Fig. 4. Autoware AD Software Stack.

In their work, they improve inferences by using the notions of adapting the composer model and policy preferences. Alternatively, MROS [18] leverages the RobMoSys [19] model-based approach at runtime, to provide a solution for ROS2 systems, based on architectural self-adaptation driven by ontology reasoning on the architecture models.

Of course this approach is not limited to AD. The applications in the mobility space exist in many different domains. Connected cars and smart cities make use of cloud based services and innovative human computer interactions. The same context-based adaptation techniques can be applied to model-based software applications in these domains. For example, the service orchestrations based on BPMN-like flows or component-based user interfaces that are amenable to view-flow like structures would also benefit from this solution. Composive.ai, will deliver our components and tools for building adaptive software stacks as open-source software (See Fig.5.).

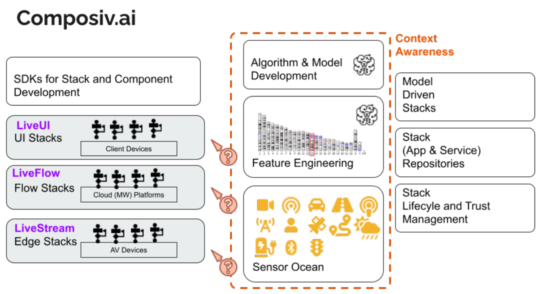

Fig. 5. Composiv.ai Adaptive Software Stacks.

Conclusion

State of the art AD software makes extensive use of statistical learning methods with proven results. These applications are typically limited by the context, size and quality of the datasets that were used to train them. Contextual adaptivity and reasoning is the natural next step for AI methods to get closer to human like capabilities. In our project, Composive -

Context Aware Software Stacks for Mobility, is a solution for adapting model-driven complex autonomous systems (such as AD). Composive will solve the adaptation problem by being able to define a context and detecting the variations within the defined context from a repository of pre-existing models using a metamodel-driven approach, subsequently it will adapt and deploy other context-driven services that are compatible with the future contextual model.

IV. Acknowledgments

This work is being partially funded by The Scientific and Technological Research Council of Turkey, TEYDEB 1509 grant.

References

[1] http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

[2] Launcbury, John, A DARPA Perspective on Artificial Intelligence, 2017 https://youtu.be/-O01G3tSYpU

[3] AI Next Campaign, https://www.darpa.mil/work-with-us/ai-next-campaign

[4] Pichler, Mario et al. ”Context-awareness and artificial intelligence”. ÖGAI Journal 23. (2004): 4-.

[5] Contextual AI: The Next Frontier of Artificial Intelligence,https://blog.adobe.com/en/2019/04/09/contextual-ai-the-next-frontier-of-artificial-intelligence.htmlgs.ytw2bv

[6] W Samek, G Montavon, S Lapuschkin, C Anders, KR Müller, Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications Proceedings of the IEEE, 109(3):247-278, 2021

[7] M. Younes, A. Boukerche, and G. Rom'an-Alonso, "An intelligent path recommendation protocol (ICOD) for VANETs," Comput. Netw., vol. 64, pp. 225–242, May 2014.

[8] M. B. Younes and A. Boukerche, "A performance evaluation of a contextaware path recommendation protocol for vehicular ad-hoc networks," in Proc. IEEE Global Telecommun. Conf. (GLOBECOM), Dec. 2013,

[9] M. Masikos, K. Demestichas, E. Adamopoulou, and M. Theologou, "Energy-efficient routing based on vehicular consumption predictions of a mesoscopic learning model," Appl. Soft Comput., vol. 28, pp. 114–124, Mar. 2015.

[10] J. Liu et al., "High-efficiency urban-traffic management in context-aware computing and 5G communication," IEEE Commun. Mag., vol. 55, no. 1, pp. 34–40, Jan. 2017.

[11] B. Kihei, J. A. Copeland, and Y. Chang, "Automotive Doppler sensing: The Doppler profile with machine learning in vehicle-to-vehicle networks for road safety," in Proc. IEEE Workshop Signal Process. Adv. Wireless Commun. (SPAWC), Jul. 2017, pp. 1–5.

[12] A. Alhammad, F. Siewe, and A. Al-Bayatti, "An infostation-based context-aware on-street parking system," in Proc. Int. Conf. Comput. Syst. Ind. Inform. (ICCSII), Dec. 2012, pp. 1–6.

[13] Pichler, Mario Bodenhofer, Ulrich Schwinger, W.. (2014). Context-awareness and artificial intelligence. ÖGAI Journal. 23. 4-.

[14] Sciences of the Artificial, Herbert Simon, 1996, MIT Press

[15] Autoware Foundation - https://www.autoware.org/

[16] Apollo Auto - https://apollo.auto/

[17] Augustus Odena, Dieterich Lawson, Christopher Olah - Changing Model Behavior at Test-Time using Reinforcement Learning, Workshop track - ICLR 2017 https://arxiv.org/pdf/1702.07780.pdf

[18] https://robmosys.eu/mros/

[19] https://robmosys.eu Composable Models and Software for Robotics Systems