MASTER THESSES 2019

We are looking for highly motivated students with background in computer science or similar. Our Master Theses projects require a good knowledge in embedded systems design and/or eclipse-based development for some additional knowledge in model based design would also be beneficial. We prefer Master Theses projects handled by two students.

Register your interest at http://career.arccore.com/. Include CV and your area of interest.

Students are welcome with their own ideas but here is suggestions from our team:

**MODELING**

1) **LOSS-LESS MODEL TRANSFORMATIONS**
   Investigate possible improved storage format for AUTOSAR models, capable of being version controlled.

2) **LAZY LOADING CACHE FOR REFERENCE HEAVY MODELS**
   Develop a way to efficiently perform lazy partial loading of massive automotive models.

3) **AUTOSAR MODEL ABSTRACTION**
   Develop a high level “language” for modeling automotive systems.

**TESTING**

4) **STANDARDIZED PERFORMANCE METRICS**
   Develop a standardized test-suite to analyze and compare adaptive AUTOSAR implementations.

**EMBEDDED SYSTEMS**

5) **SIGNAL AND SERVICE BASED, SYSTEM INTERACTIONS**
   Develop smart ways of interacting between old signal based automotive systems and new service-based ones.

6) **FAULT ANALYSIS OF COMPLEX SYSTEMS**
   Investigate ways to analyze and look for faults in complex service based automotive systems.

**APPLICATION**

7) **EVALUATE THE POSSIBILITY TO CREATE AN APPSTORE FOR AUTOSAR**
DETAILED DESCRIPTION

The following is some additional information, background and suggestions about the topics presented above. In no way is this the only way these topics can go, so if you have interests, suggestions or completely different ideas just contact us and we can discuss them.

MODELING

LOSS-LESS MODEL TRANSFORMATION

AUTOSAR models are usually saved in an XML variant called ARXML. This model format is not friendly to traditional version control programs such as git or subversion, and this results in major complications when a model is continuously changed over time.

It would be a major benefit to the Automotive industry if a loss-less conversion could be developed that allows the saving of ARXML in traditional version control systems, while being readable by traditional automotive industry programs.

LAZY LOADING CACHE FOR REFERENCE HEAVY MODELS

AUTOSAR models tend to grow larger and larger as the automotive software gets more and more complicated. This causes longer loading times and higher RAM memory usage from the modeling tools, and it's starting to reach the point where lazy loading of pieces of models is a must. Complications however arise in that AUTOSAR models are rich in cross-references, both single and bi-directional and that users' behavior and changes could break or invalidate those links.

The problem is to develop a lazy-loading cache that will only load parts of a full model, allow changes, and still not display any out-of-date or misrepresenting data to a user of the system. Ideally the cache should also be able to warn when operations would break the data-links without loading the entire model into memory.

AUTOSAR MODEL ABSTRACTION

AUTOSAR models are getting larger and larger as the car systems they are describing are getting more and more complex. It is now reaching the point where it is difficult or even impossible for humans to have any kind of overview of how the entire system works. This problem is made worse by the fact that AUTOSAR demands very detailed models, and thus larger models.

Could an abstraction to AUTOSAR be found by analyzing how general car ECUs are modeled and what parts of AUTOSAR are actually used? The goal being to describe the system on a higher level from which the detailed AUTOSAR format can then be automatically generated, or compiled?

TESTING

STANDARDIZED PERFORMANCE METRICS

Adaptive AUTOSAR is a very new standard, and for now it is very difficult to know what kind of minimum requirements that it will place on hardware. It also constitutes a large change in how automotive ECUs work and it is not properly known what kind of requirements and performance can be expected of an Adaptive AUTOSAR system.

It would therefore be good to study similar architectures from other domains and check how they work to measure and improve the performance of their core systems. This could then be developed into an Adaptive AUTOSAR performance test. Which could be used to compare different implementations, and to ensure consistent improvement of ARCCOREs own products. In other words, to establish a benchmark test with metrics for Adaptive AUTOSAR. Such a benchmark would likely involve stressing the system, error injection and AUTOSAR coverage.

EMBEDDED SYSTEMS

SIGNAL AND SERVICE BASED, SYSTEM INTERACTIONS

In the automotive systems world of yesterday, almost all communication was made with "signals", that is fixed communication links with known content between known parties. With Adaptive AUTOSAR and the industry in general moving more into a "service" based world, where one "provider" of content can be "subscribed" to by any other component in the car.

For at least the foreseeable future these two design methodologies will have to coexist, both for financial reasons (it’s expensive to remake everything) and for safety reasons (safety critical functionality must be deterministic). Therefore, there is a need to investigate
different methodologies to interact between those two designs. Should any signal be directly mapped to a service? Should there be a central lookup of signals for the service world? What are the pros and cons of each approach? What is most efficient? Most safe? Fastest?

**FAULT ANALYSIS OF COMPLEX SYSTEMS**

As the automotive systems gets increasingly more complex it is getting more and more difficult to create fault detecting systems that are meaningful. A traditional approach is to set up "report" points where the different parts of the system have to report what they have done and since the entire system is known this can be used to ensure the right things happen in the right order.

However, in the newer dynamic system, no one can know from the start exactly what should happen in the system, or exactly when. In such a system, it might be difficult, or maybe even meaningless to use the traditional approaches. Instead there is a need to investigate how other computer architectures handle fault detection, for example how does widely distributed data servers check for faults? Possible approaches might be analyzing data flow using metrics/AI, having fixed behavior subsystems or maybe something completely different.

People interested in this area will likely need experience of fault detection strategies from at least one service oriented domain.

**EVALUATE THE POSSIBILITY TO CREATE AN APPSTORE FOR AUTOSAR**

In the new upcoming standard Adaptive AUTOSAR it will be possible to download new application software to the car without reprogramming the complete unit and without going to the workshop. This creates new possibilities for development of new car functions but also deploying fixes for existing problems. On the other side this possibility also creates vulnerabilities for the car i.e. there is a risk for viruses, security leaks and other types of malfunctions.

In this master thesis we would like to evaluate the needs and requirements that should be put on the infrastructure that distributes the software to the car. I.e. what protocols to use, authentication mechanisms, recovery strategies and so on. The master thesis should result in a theoretical study of available mechanisms used in other IT systems and a prototype implementation of such a system for the automotive industry.

Nothing interesting? Got your own ideas? Come talk to us and let's make it happen!

**Contact information**

Annie Fridén  
email: annie.friden@arccore.com  
mobile: +46 724 00 83 83  
office: +46 31 301 28 30  
www.arccore.com