HW/SW Architecture and Design

TG, 10th October 2016

General definition:
Architecture is the breakdown process of a system in such a number parts that makes it is possible to establish an overview of the system.
When the parts are defined the interface between the parts must be defined.

At first: establish a deployment diagram to make a rough definition of the hardware and software blocks, which can be derived from the requirement specification. The software blocks are based on nouns from the text in the requirement specification. These blocks are typically domain classes.
The deployment diagram is a transition from the requirement specification to the architecture document.

HW architecture and design, based on SysML

HW architecture

1. Draw BDD's (Block Definition Diagrams) to an overview level.
2. Draw IBD's (Internal Block Diagrams) to an overview level.
3. Fill in signals from IBD's in tables to describe each interface between the blocks.

HW design:

If necessary (if you are going to develop a new HW block): Calcalate the values of each component, make a part list, draw a schematic and if necessary: make a PCB layout.
When the "HW architecture and design" is finished you can build the HW of your system!

SW architecture and design, based on UML and N+1 view model

N is a number of necessary views for your SW architecture, e.g. Logical View, Process View, Data View, Deployment View, Security View, Implementation View …)
The "+1" view is SW-architecture, based on use cases from the requirement specification (Use Case View)

1. Make SW architecture based on *Logical View* to define a static class diagram for each CPU (node) and for each use case at this CPU. Find candidate classes based on domain classes (from domain diagram), control classes (based on the activity in the use case) and boundary classes (based on the interfaces to the surrounding world: user, ports …).
Analyze the behavior for the use case by an UML Sequence Diagram to find the interaction between the classes belonging to the use case. By this process you can find the important methods for each class and the important parameters for these methods.
Draw a Static Class Diagram based on the analyzed domain/controller/boundary classes for the use case and insert the methods/parameters from the Sequence Diagram.
By doing this process you have an Application Model for the Use case.
By analyzing all use cases for each CPU you have a static class diagram for each CPU. (An Application Model for the CPU)

2. Make SW architecture based on *Process View* to describe the critical interaction between active classes (threads). Your use of concurrency tools as Semaphores, Mutex'es, Mailboxes, Pipes etc. can be described here.

3. Make SW architecture based on *Data View* to describe the layout of the important data in your system (database layout, table layout).

4. Make SW architecture based on *Deployment View* to describe all the necessary communication protocols between the CPU’s in your system. Describe the low level part of the communication (cable, RS232, USB, Bluetooth, WiFi, Mobile Broadband: 2G/3G/4G). Describe the transport layer level TCP (reliable) or UDP (best effort). Describe the messages in your system which are sent between the CPU’s (using boundary classes) e.g. Temperature, Humidity, Position etc.

5. Make SW architecture based on *Security View* to describe security aspects in you system (login, authentication encryption etc.)

6. Make SW architecture based on *Implementation View* to describe the files in your system, how you can build (compile) your system and further: mathematical expressions, register setups etc. – these things are difficult to describe as comments in your source code.

When the “SW architecture and design” is finished you can write the source code for your system!