KPI based design methodology

Francesco Regazzoni, USI

Horizon 2020
European Union funding
for Research & Innovation

What are KPIs?

- **KPI (Key Performance Indicators)**
- CERBERO DEFINITION: **A KPI is a quantifiable parameter associated with a metric. A KPI evaluates one critical parameter of a CPS and evaluate the discrepancies from its long term goals**
- **This implies:**
  - Quantifiable parameter: it should be quantifiable (measurable or at least “rankable”)
  - Associated with a metric: the way to measure of compare is part of the KPI definition
  - Evaluates: constantly evaluates, should be measureable at the design and on the field
  - Discrepancies: to eventually correct them, needs adaptation
  - Long term goals: follows the CPS from the initial design stages to the phasing out of the device

Properties of KPIs

- **KPIs are always defined with a metric**
  - The metric depends on the system (the use of generic KPIs could be meaningless)
- **KPIs are measuring a specific CPS**
  - The set of KPIs used to measure a system are specific to that system (the use of generic KPIs is suboptimal)
- **Each KPI belongs to a family**
  - KPIs is tailored to the CPS, but it will belong to a family. Properties of family of KPIs are the reusable element of the library of KPIs
- **KPIs are always defined with structured/common formalism**
  - Formally defined KPIs automatically inherit all the properties of a family
- **KPIs drives the evaluation of the system during the whole live cycle**
  - KPIs are used at design time, but they also be used to drive the adaptation

KPI based design methodology

1. State the problem
2. Select the KPIs
3. Model the physical process and the KPIs
4. Characterize the problem and the KPIs
5. Derive the control algorithm
6. Define the toolchain
7. Synthesize the low level requirements
8. System architecture
9. Define the hardware
10. Synthesize the monitors
11. Synthesize the software
12. Assemble and simulate
13. Construct
14. Verify, validate, and test

KPI based design methodology

1. State the problem
2. Select the KPIs
3. Model the physical process and the KPIs
4. Characterize the problem and the KPIs
5. Derive the control algorithm
6. Define the toolchain
7. Synthesize the low level requirements
8. System architecture
9. Define the hardware
10. Synthesize the monitors
11. Synthesize the software
12. Assemble and simulate
13. Construct
14. Verify, validate, and test

KPI based design methodology highlights 1/2

- **Select the KPIs**
  - Selection of the KPIs that should be optimized
  - At this stage, definition in simple language
- **Model the physical process and the KPIs**
  - Provide a simplified representation of the KPI and a metric to measure it
  - Process of defining a model is iterative (till the model is sufficient to monitor the needed quantity)
- **Characterize the problem and the KPIs**
  - Each KPI is characterized and assigned to a family
  - This step dictates the model of computation (no need to select the model of computation in a separate step)

KPI based design methodology highlights 2/2

- **Define the toolchain**
  - KPIs are following the CPS since its lifespan
  - Selected toolchains should support the defined KPIs and their monitors (use of existing toolchains is a constraint)
- **Synthesize the low level requirements**
  - Low level requirements for the toolchains should be inferred directly from the KPIs
  - This step express the KPIs by means of tools constraints
- **Synthesize the monitors**
  - Monitors for adaptivity should be derived directly from KPIs
  - Monitors for specific KPIs could not be available, then should be designed or inferred from other existing monitors

Focus of KPIs in the tool

- **Two types of support:**
  - being able to directly use, at design time or at run time, one or more KPI during the generation of the output. Implies the capability of calculate the given KPI.
  - producing the infrastructure needed for exposing one or more KPI at runtime or at design time (if needed in simulation) or producing the infrastructure for modifying, at run time, the behavior of a system in a way that affect the value of one or more KPIs.

Example of KPIs in the tool (type one)

- **PREESM / SPIDER (similar behavior regarding KPIs)**
  - Inputs: model of the architecture, model of the application (as dataflow), set of constraints
  - Output: optimized mapping, optimized scheduling, optimized memory allocation for sharing
  - Which KPIs does the tool support? How do you express them?
    - Throughput (using the model of application, ex. specific sampling rate of a sensor)
    - Latency: application and architecture depended, passed via constraints
    - Resource utilization: input (passed as constraints) and output
  - Extensions
    - Energy consumption: passed as annotation on the architectural model and application and as constraints
    - Reconfiguration time: suitable place to pass to the tool is annotate in the model of architecture

Example of KPIs in the tool (type two)

- **ARTICO³**
  - Which KPIs does the tool support? How do you express them?
    - Latency / throughput: the tool generates the infrastructure to react at run time to measures coming from dedicated monitors
    - Energy consumption: the tool generates the infrastructure to react at run time to measures coming from dedicated monitors
    - Reliability: the tool generates the infrastructure with the desired redundancy to react to measures coming from dedicated monitors

Thanks for your attention!