Resource Contracts for deploying multiple applications on MPSoC systems

Zubair Wadood Bhatti, Davy Preuveneers, Roel Wyuts, Yolande Berbers

{Zubairwadood.Bhatti, Davy.Preuveneers, Roel.Wyuts, Yolande.berbers}@cs.kuleuven.be

DistriNet

Why do we need resource contracts?

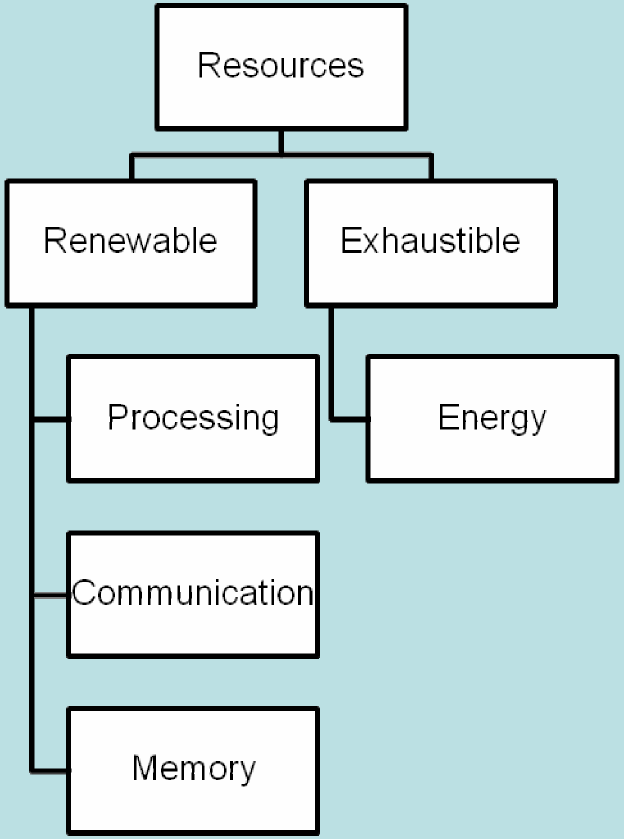
- Unbounded set of multiple application = unexplorable dynamism
- Different types of application
  - Data dominated, control, hard, soft.
- Different mapping methodologies
  - Design space exploration: memory aware, scenario aware
  - pure design/run-time.
- Resource contracts

Context: MP-SoC

Objective:
Minimize resource usage for exhaustible resources

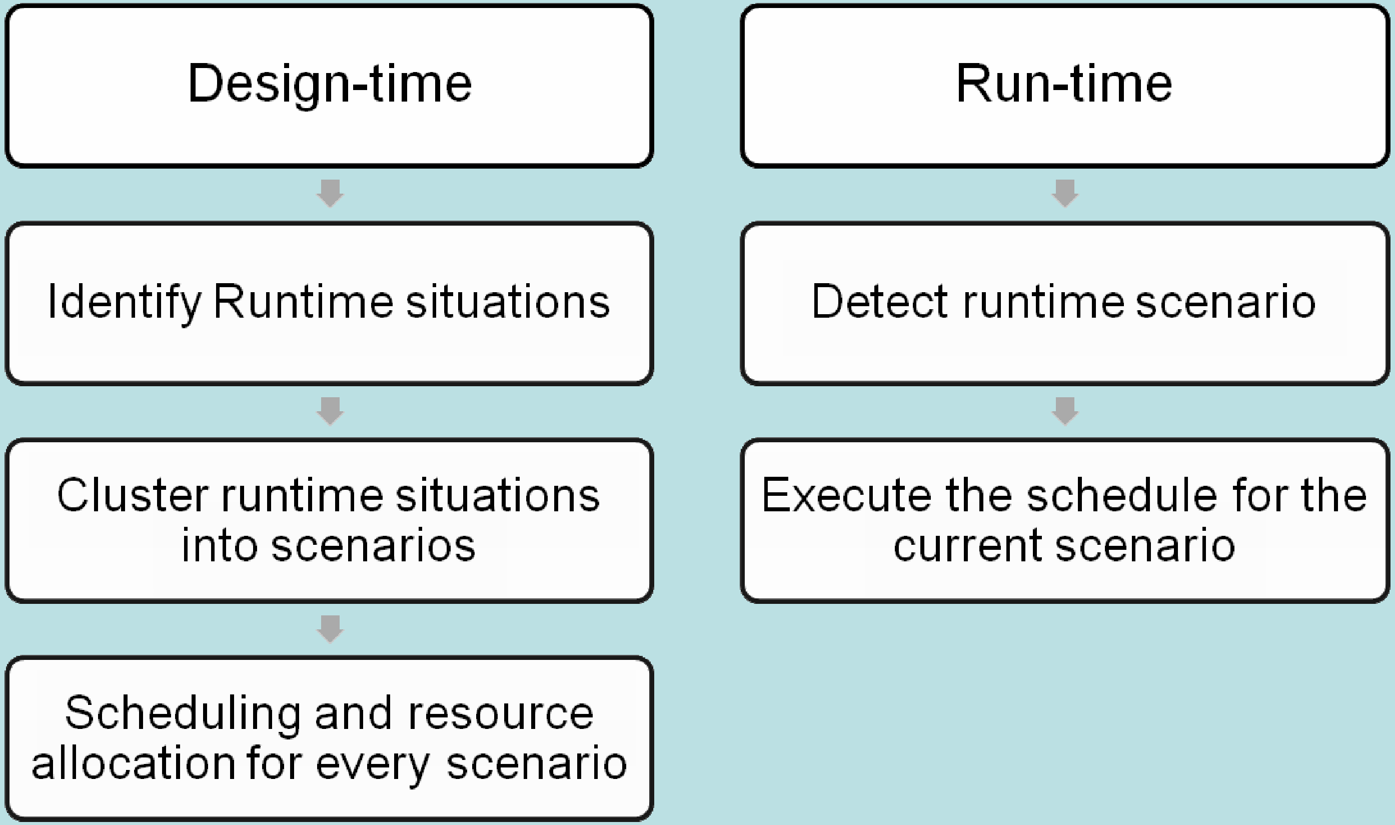
Constraints:

1. Renewable resources
2. QoS requirements



System-scenario-based design

Explorable Dynamism



Design-time

Identify Runtime situations

Cluster runtime situations into scenarios

Scheduling and resource allocation for every scenario

Run-time

Detect runtime scenario

Execute the schedule for the current scenario

[SV Gheorghita et al.]

Application 1

Execution time / us	Energy / uJ
345431	129562
297921	137909
232187	153552
177263	270560

Table: 1 mapping tradeoffs for 3D application in a frame with 22 objects being rendered

Execution time / us	Energy / uJ
274528	73905
149104	115509
56714	304492
15420	377924

Table: 2 mapping tradeoffs for 3D application in a frame with 13 objects being rendered

Application 2

Execution time/ us	Energy / uJ
17954	4307
9295	5812
7260	13604
1921	47778

Table 3: Mapping tradeoffs for static application

QML for resource contracts

```
type Energy = contract {
energy: decreasing numeric uJ;
};
3D_Application_Contract = Energy contract ( Max_Energy: function; Scenario: int;
      Available_Processor_Time: int;)
{
Maximum = Max_Energy(Scenario, Available_Processor_Time);
};
```

The function of Max_Energy is derived from Table 1 and Table 2.

```
Static_app_Contract = Energy contract ( Max_Energy: function;
      Available_Processor_Time: int;)
{
Maximum = Max_Energy(Available_Processor_Time);
};
```

Where the Max_Energy function comes from table 3.

Conclusion

- Mapping techniques should have a way to communicate with each other or standardized contracts
- This can’t be done unless there is interest from the community
- Requirements analysis