TECHNISCHE UNIVERSITÄT DRESDEN

ResUbic Qualitune COOL Software

Faculty of Computer Science, Institute of Software- and Multimedia-Engineering, Software Technology Group

Runtime Variability Management for Energy-Efficient Software by Contract Negotiation

Sebastian Götz, **Claas Wilke**, Sebastian Cech, Uwe Aßmann

Models@runtime 2011, Wellington, NZ
October, 17th 2011

Europa fördert Sachsen. ESF Europäischer Sozialfonds SACHSEN
Bundesministerium für Bildung und Forschung
DFG Deutsche Forschungsgemeinschaft
DRESDEN concept Exzellenz aus Wissenschaft und Kultur

Context

- **Target:** Run **software** w.r.t. **non-functional requirements** in an **optimal** way.

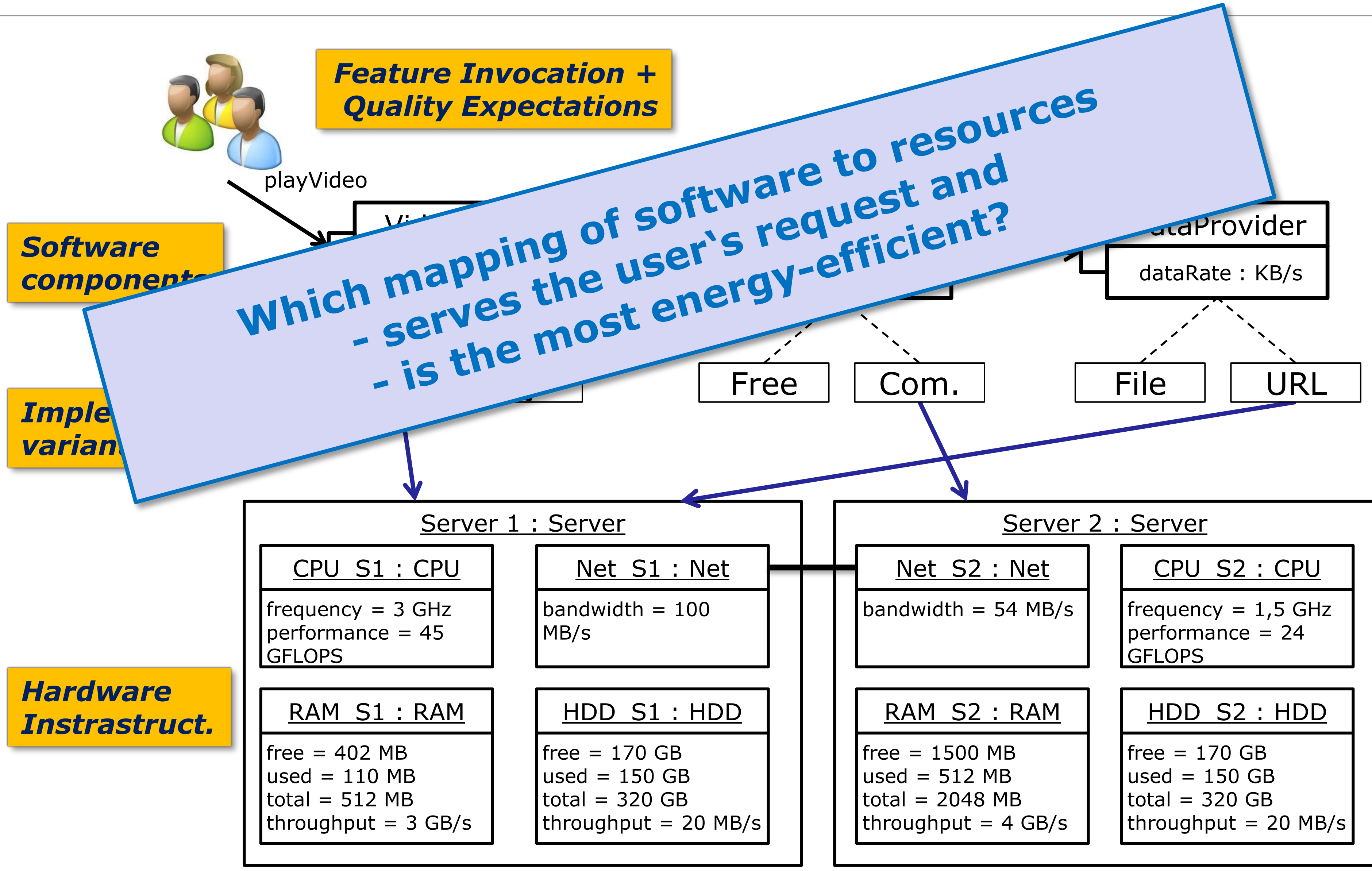
→ **Energy Auto-Tuning (EAT) [1,2]**

- **System knows**
  1. Its **hardware** (resources)
  2. Its **software** (components)
  3. Their **energy behavior**
- **System adapts**
  1. The **deployment of software** components
  2. On available **hardware**
  3. W.r.t. their **energy consumption**
  4. And **non-functional properties**

[1] S. Götz, C. Wilke, M. Schmidt, S. Cech, and U. Aßmann. Towards energy auto tuning. In Proceedings of First Annual International Conference on Green Information Technology (GREEN IT), pages 122–129. GSTF, 2010.

[2] S. Götz, C. Wilke, S. Cech, and U. Aßmann. Architecture and Mechanisms of Energy Auto-Tuning. To appear in: Sustainable Green Computing: Practices, Methodologies and Technologies, IGI Global, 2011

Example Application



Quality Contracts for Components

```
contract VLC implements VideoPlayer {

  mode highQuality {
    requires component Decoder {
      min dataRate: 50 KB/s
    }

    requires resource CPU {
      max cpuLoad: 50 percent
      min frequency: 2 GHz
    }
    requires resource Net {
      min bandwidth: 10 MBit/s
    }

    provides min frameRate: 25 FPS
    provides min imageWidth: 1024 Pixel
    provides min imageHeight: 768 Pixel
  }

  mode lowQuality {
    /* More requirements and provisions here ... */
  }
}
```

Quality Modes

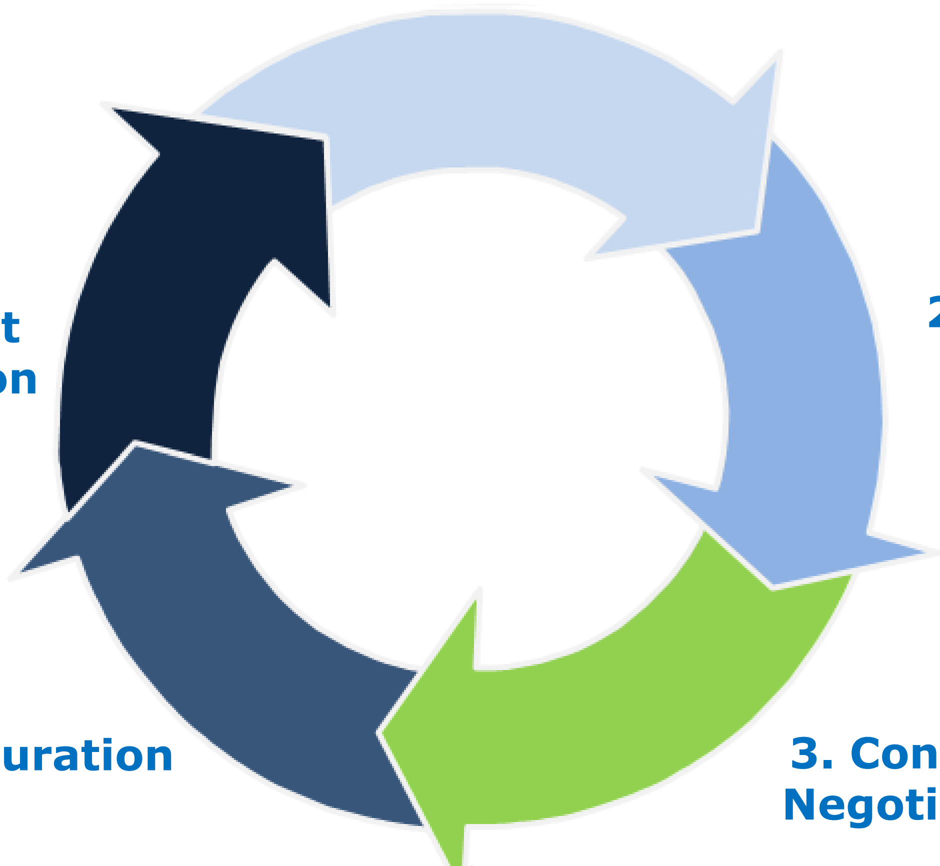
Software Dependencies

Resource Dependencies

Quality Provisions

Quality Modes

Energy Auto-Tuning Process [2]



[2] S. Götz, C. Wilke, S. Cech, and U. Aßmann. Architecture and Mechanisms of Energy Auto-Tuning. To appear in: Sustainable Green Computing: Practices, Methodologies and Technologies, IGI Global, 2011

Solution at a Glance

- HW/SW Modeling using c**omponents** and **contracts**
- Application of **Contract Negotiation** to compute the **optimal system configuration** at runtime.

Definition: „System Configuration“

A selection of components and their mapping onto resources.

- **Transformation into an MILP**

> **Definition: MILP (Mixed Integer Linear Program)**
>
> Constraint System with **objective function** comprised of **linear constraints**, where some **variables** have to be **integers** (must not be reals).

- **MILP consists of**
  - Variables
  - Constraints
  - Objective Function

MILP for Contract Checking

- **Four kinds of variables:**

1. **Base load**
(e.g., baseload#Server1)

2. **Resource usage**
(e.g., usage#Server1#RAM_[s1]#size)

3. **Implementation Mapping (Boolean variables, flags)**
(e.g., b#FreeDecoder#fast#Server2)

4. **NFPs**
(e.g., bitrate, throughput, framerate, …)

MILP for Contract Checking

- **Objective Function**:

$$\min \left(\begin{array}{l} \sum \left(\text{weight}_{XYZ} \cdot \text{usage\#container}_X \#\text{resource}_Y \#\text{property}_Z \right) \\ + \sum \textit{baseload\#server}_i \end{array} \right)$$

- **Selection criteria/mappings** → **Boolean** variables

```
    b#FileReader#file#Server2
  + b#FileReader#file#Server1
  + b#URLReader#url#Server2
  + b#URLReader#url#Server1
  = 1.0;
```

Exactly one DataProvider on exactly one server

  - 1 such constraint per software component type

MILP for Contract Checking

- **Three constraints per usage variable of HW component**
  - Upper bound (according to variant model)
    - `usage#Server1#RAM_[s1]#size <= 512.0;`
  - Lower bound (>= 0 or according to variant model)
    - `usage#Server1#RAM_[s1]#size >= 0.0;`
  - Requirements (from contracts)
    - **usage#Server1#RAM_[s1]#size =**

```
  512.0 * b#FreeDecoder#fast#Server1   0 or 1
+ 256.0 * b#FreeDecoder#slow#Server1
+ 128.0 * b#CommercialDecoder#slow#Server1
+ 512.0 * b#CommercialDecoder#fast#Server1
+ 1536.0 * b#CommercialDecoder#ultrafast#Server1;
```

- **Baseload constraints**
  - baseload#server$_i$ = b#impl$_x$#mode$_y$#server$_i$

MILP for Contract Checking

- **Constraints for SW component NFPs**
  - **throughput =**

```
  5.0 * b#URLReader#url#Server1
+ 20.0 * b#FileReader#file#Server2
+ 5.0 * b#URLReader#url#Server2
+ 20.0 * b#FileReader#file#Server1;
```

- **User Request** reflected by constraint, too
  - `framerate >= 20.0;`

TECHNISCHE UNIVERSITÄT DRESDEN

MILP for Contract Checking

Variables	MILP ...
	2433
b#CommercialDecoder#fast#Server1	0
b#CommercialDecoder#fast#Server2	0
b#CommercialDecoder#slow#Server1	0
b#CommercialDecoder#slow#Server2	1
b#CommercialDecoder#ultrafast#Server1	0
b#CommercialDecoder#ultrafast#Server2	0
b#FileReader#file#Server1	0
b#FileReader#file#Server2	0
b#FreeDecoder#fast#Server1	0
b#FreeDecoder#fast#Server2	0
b#FreeDecoder#slow#Server1	0
b#FreeDecoder#slow#Server2	0
b#URLReader#url#Server1	1
b#URLReader#url#Server2	0
b#VLC#highQuality#Server1	1
b#VLC#highQuality#Server2	0
b#VLC#lowQuality#Server1	0
b#VLC#lowQuality#Server2	0
bitrate	5
framerate	20
throughput	5
usage#Server1#CPU_[s1]#frequency	1500
usage#Server1#HDD_[s1]#size	0
usage#Server1#HDD_[s1]#throughput	0
usage#Server1#Net_[s1]#bandwidth	5
usage#Server1#RAM_[s1]#size	0
usage#Server2#CPU#frequency	800
usage#Server2#HDD#size	0
usage#Server2#HDD#throughput	0
usage#Server2#Network#bandwidth	0
usage#Server2#RAM#size	128

Optimal Configuration

Concrete NFPs

Concrete Resource Usage

Future Work

1) Resource usage

- Improve resource utilization / energy relation
- Introduce time-dependant resource utilization models

2) Case studies

- VideoServer, StockTracking and further scientific case studies
- Industrial case study with a green enterprise application planning (EAP) system

Summary

- Energy Auto-Tuning **requires contract negotiation**
- Goal: select the **optimal variant** w.r.t. **user demands** **and** **energy consumption**
- Solution: **MILPs** can be used **for contract negotiation**
  - Are generated from models@runtime
  - Result includes additional information
    - Concrete resource usage of variant → resource control (e.g., setting the CPU frequency)
    - Concrete provided qualities

Thank You!

Contact

http://st.inf.tu-dresden.de/

claas.wilke@tu-dresden.de

Software Technology Group

ResUbic
www.resubic.org

Qualitune
www.qualitune.org

COOL Software
www.cool-software.org

Europa fördert Sachsen. ESF Europäischer Sozialfonds SACHSEN
Bundesministerium für Bildung und Forschung
DFG Deutsche Forschungsgemeinschaft

TECHNISCHE UNIVERSITÄT DRESDEN

BACKUP

Related Work

Dynamic Variability in Complex, Adaptive Systems (DiVA) [3, 4]

- Management of dynamic adaptive systems
- Special focus on exponential growth of potential configurations
- Automated adaptation at runtime with
  - Goal-based optimization of NFPs
  - Rule-based system reconfiguration

Major difference to our approach

- Level of abstraction for non-functional property values
- DIVA symbolizes non-functional properties (e.g., LOW, MEDIUM, HIGH memory)
- We consider subsymbolic information (e.g., the actual value of free size of memory in MB)

→ allows deriving **finer-grained configurations**

[3] F. Fleurey and A. Solberg. A domain specific modeling language supporting specification, simulation and execution of dynamic adaptive systems. In: Proceedings of MODELS ’09, pp. 606–621, Springer, 2009.

[4] B. Morin, O. Barais, G. Nain, and J.-M. Jézéquel. Taming dynamically adaptive systems using models and aspects. In: Proceedings of the 31st ICSE ’09, pp. 122–132, IEEE, 2009.

MADAM/MUSIC [5]

- Management of **mobile** dynamic adaptive systems
- MUSIC aims to maximize **user utility** using manually written utility functions.
  - Goal is to maximize **user satisfaction**
  - Dependencies between qualities are not considered

Major difference to our approach

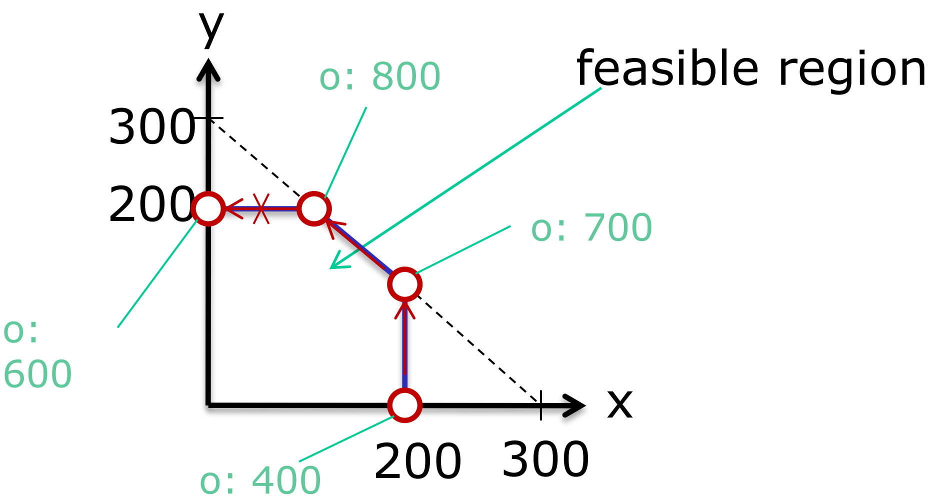
- We aim to maximize user utility **while minimizing resource usage** based on **quality contracts** between user, soft- and hardware
  - We focus on **efficient** user satisfaction
  - Dependencies between qualities are considered
    - Tradeoffs are negotiated

[5] http://ist-music.berlios.de/ - MUSIC project homepage.

Mixed Integer Linear Programming (MILP)

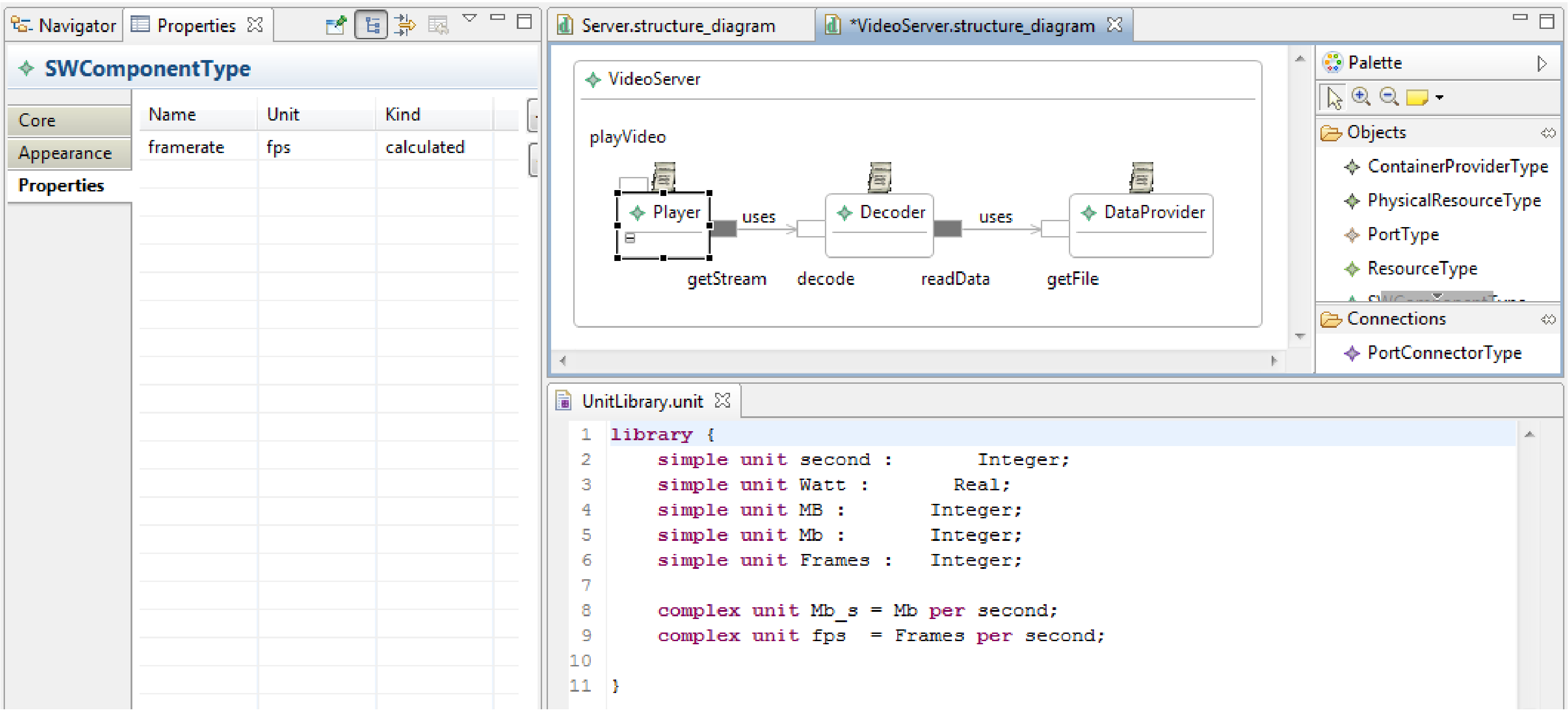
- **Objective Function** (maximize profit, minimize cost, …)
- **Set of Constraints**
- **Example:** Production of 2 types of items. Item x profit: 2€. Item y profit: 3€. Max production capacity: 300, demand for x and y each: 200. Objective function: maximize profit.

max: 2x + 3y
x, y >=0 x + y <= 300 x,y <= 200



- **Simplex-Alg.**
- Integer:
  dom(x) = dom(y) = N

Example: VideoPlayer (SW types)

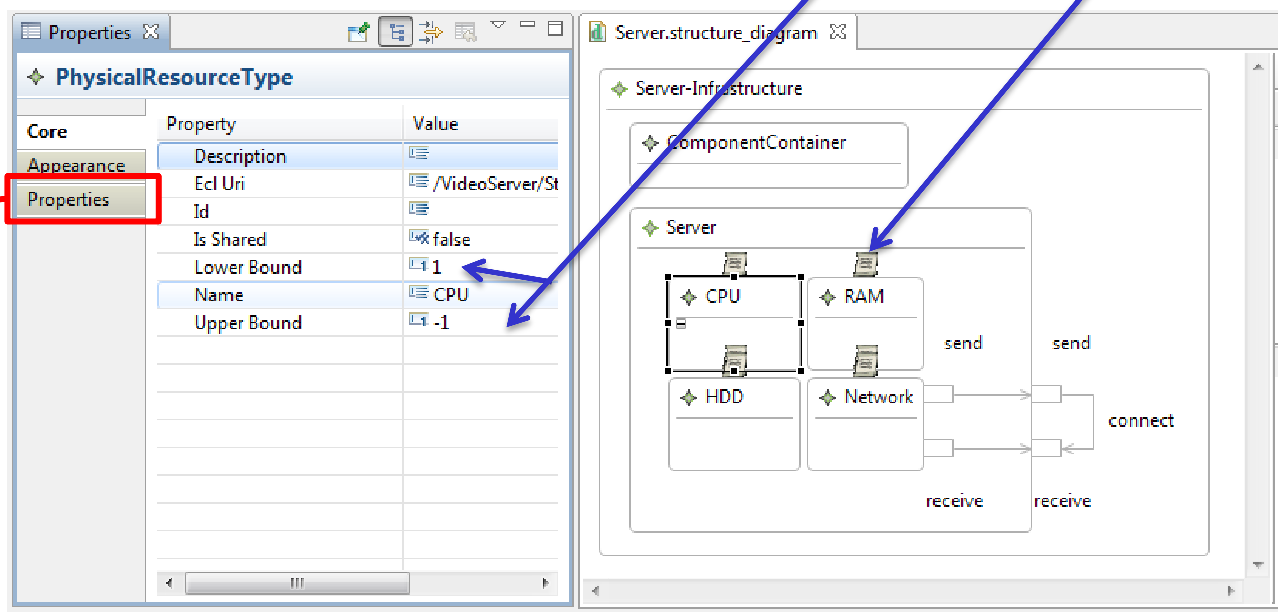


Structural Model of SW Component Types + NFPs

Example: VideoPlayer (HW types)

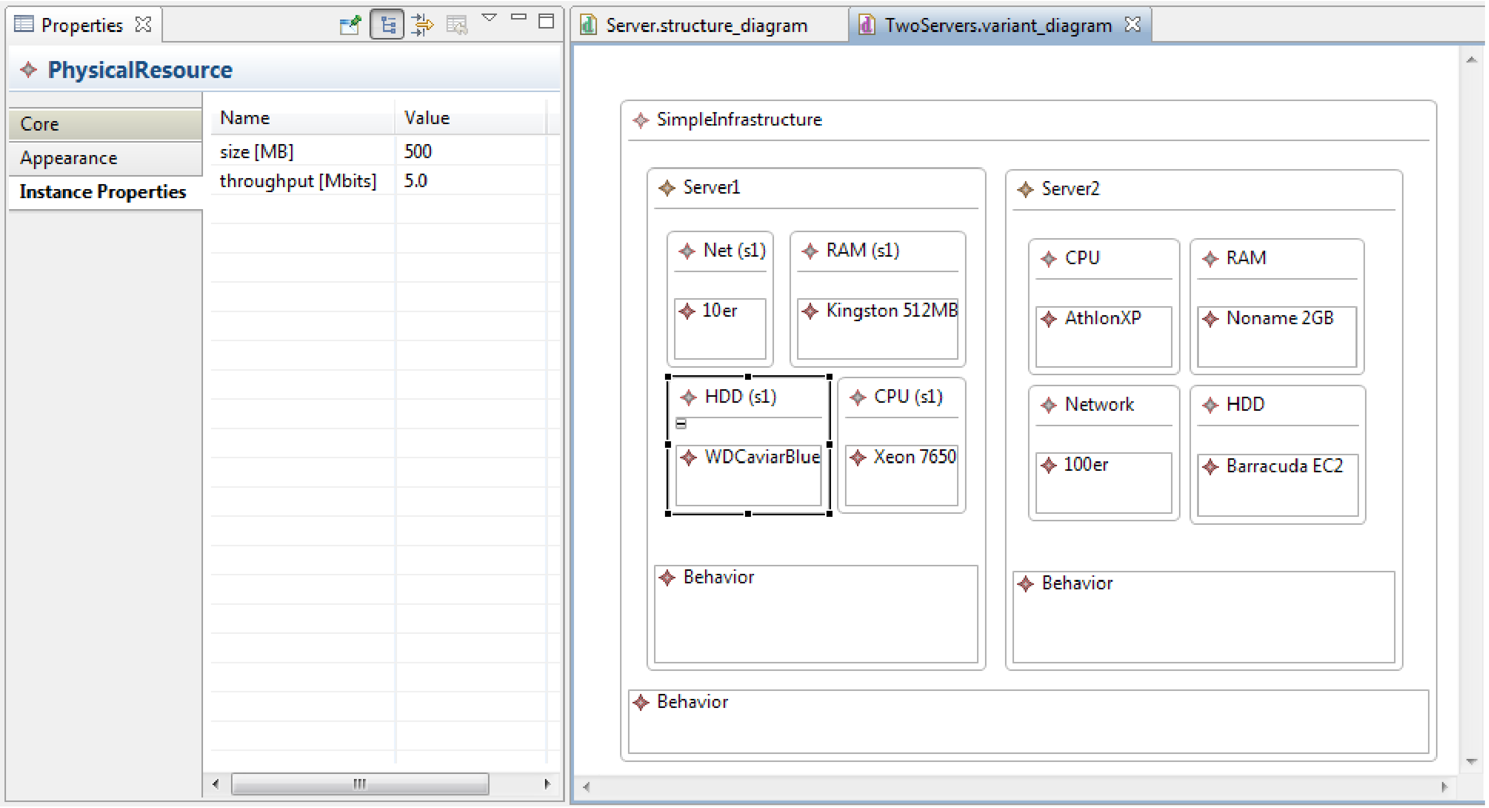
- Which ResourceTypes exist
- How they are connected
- Multiplicities
- Contracts

NFPs



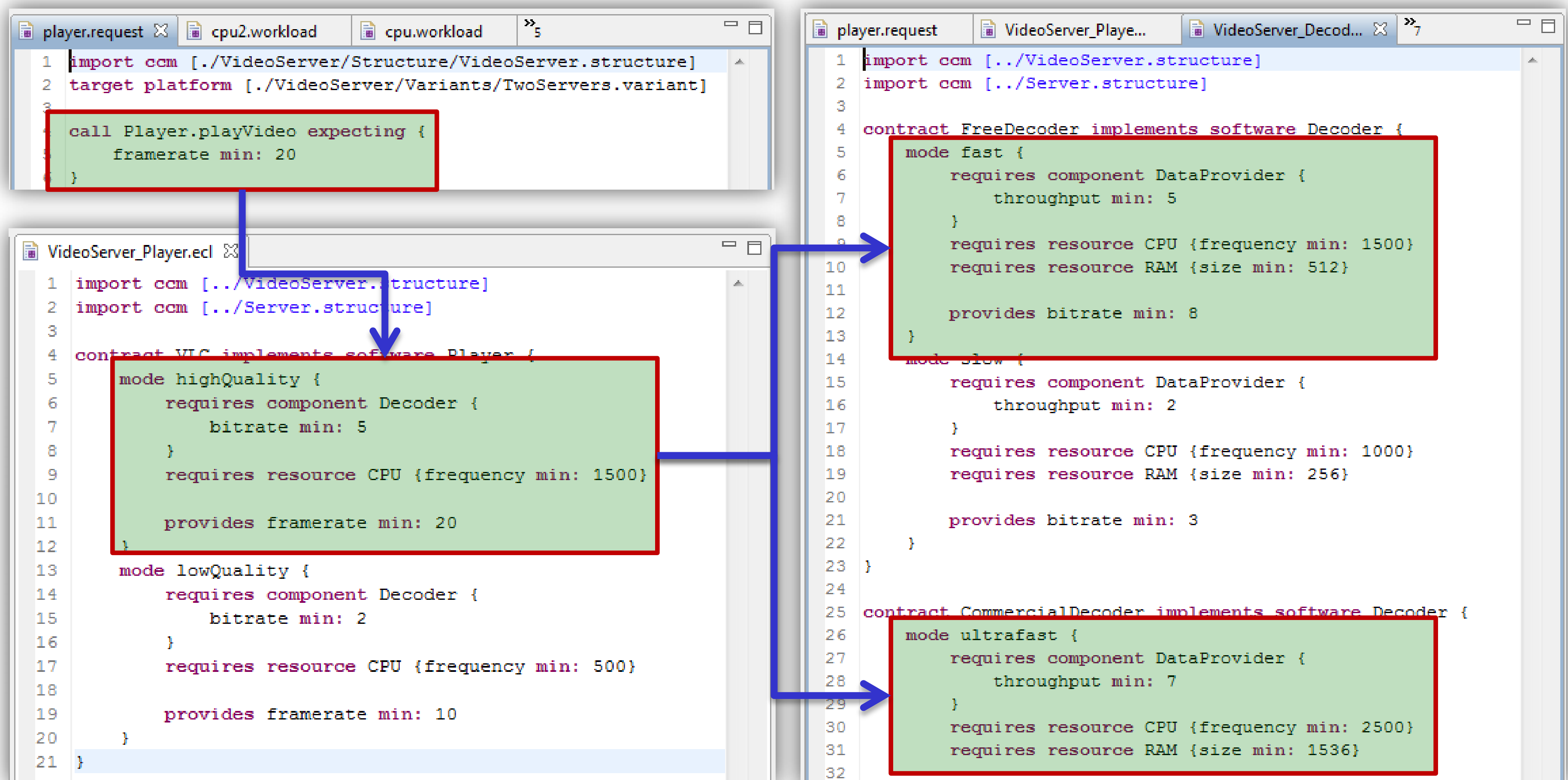
Structural Model of HW Resources

Example: VideoPlayer (HW variants)



Variant Model of Resources + NFP values

Example: VideoPlayer (User + Contracts)



LP vs. ILP

Variables	LP O...	MILP ...	result
	2433	2433	2433
b#CommercialDecoder#fast#Server1	0	0	0
b#CommercialDecoder#fast#Server2	0	0	0
b#CommercialDecoder#slow#Server1	1	0	0
b#CommercialDecoder#slow#Server2	0	1	1
b#CommercialDecoder#ultrafast#Server1	0	0	0
b#CommercialDecoder#ultrafast#Server2	0	0	0
b#FileReader#file#Server1	0	0	0
b#FileReader#file#Server2	0	0	0
b#FreeDecoder#fast#Server1	0	0	0
b#FreeDecoder#fast#Server2	0	0	0
b#FreeDecoder#slow#Server1	0	0	0
b#FreeDecoder#slow#Server2	0	0	0
b#URLReader#url#Server1	1	1	1
b#URLReader#url#Server2	0	0	0
b#VLC#highQuality#Server1	0,333...	1	1
b#VLC#highQuality#Server2	0,666...	0	0
b#VLC#lowQuality#Server1	0	0	0
b#VLC#lowQuality#Server2	0	0	0
bitrate	5	5	5
framerate	20	20	20
throughput	5	5	5
usage#Server1#CPU_[s1]#frequency	1300	1500	1500
usage#Server1#HDD_[s1]#size	0	0	0
usage#Server1#HDD_[s1]#throughput	0	0	0
usage#Server1#Net_[s1]#bandwidth	5	5	5
usage#Server1#RAM_[s1]#size	128	0	0
usage#Server2#CPU#frequency	1000	800	800
usage#Server2#HDD#size	0	0	0
usage#Server2#HDD#throughput	0	0	0
usage#Server2#Network#bandwidth	0	0	0
usage#Server2#RAM#size	0	128	128

LP vs. ILP

Literature

[1] S. Götz, C. Wilke, M. Schmidt, S. Cech, and U. Aßmann. **Towards energy auto tuning**. In: Proceedings of First Annual International Conference on Green Information Technology (GREEN IT), pp. 122–129. GSTF, 2010.

[2] S. Götz, C. Wilke, S. Cech, and U. Aßmann. **Architecture and Mechanisms of Energy Auto-Tuning**. To appear in: Sustainable Green Computing: Practices, Methodologies and Technologies, IGI Global, 2011

[3] F. Fleurey and A. Solberg. **A domain specific modeling language supporting specification, simulation and execution of dynamic adaptive systems**. In: Proceedings of MODELS '09, pp. 606–621, Springer, 2009.

[4] B. Morin, O. Barais, G. Nain, and J.-M. Jézéquel. **Taming dynamically adaptive systems using models and aspects**. In: Proceedings of the 31st ICSE '09, pp. 122–132, IEEE, 2009.

[5] http://ist-music.berlios.de/ - **MUSIC project homepage**.