universität wien

Faculty of Computer Science

Austrian-Italian Workshop on "Future Internet Challenges"

An Optimization Framework for Service Selection and Service Composition in Distributed, Heterogeneous Environments

Erich Schikuta

RG Workflow Systems and Technology

University of Vienna

Pavia, May 2011

Agenda

Future Internet Challenges
- Personal Vision of IT of the Future
- Layered research area and research group

Service Selection and Composition Optimization Framework
- Problem and Goals
- Use Cases
- QoS Data Model and Ontology
- Heuristic Optimization Approaches
  - Blackboard method
  - Gentic algorithm
- Performance Analysis and Lessons Learned

Future Work

Future Internet Challenges

Future IT – An old Computing Vision in SF Literature

Isaac Asimov, 1956, "The Last Question"

> "…
>
> *Man said, "Can entropy not be reversed? Let us ask the Cosmic AC (Analog Computer)."*
>
> ***The Cosmic AC surrounded them but not in space.*** *Not a fragment of it was in space. It was in hyperspace and made of something that was neither matter nor energy.* ***The question of its size and Nature no longer had meaning to any terms that Man could comprehend.***
>
> …
>
> *He stared somberly at* ***his small AC-contact****. It was only two inches cubed and nothing in itself, but it was connected through hyperspace with the great Galactic AC that served all mankind.*
>
> …"

The first **Computing Utility** in literature?

Why not …

No more buying of resources

- Processing power, Storage space, Software
- …

“Just sharing what is needed (and paying for the usage)”

- Vision
  - “Writing a letter” can be as simple as using a telephone:
    - forget of buying software and hardware;
    - all we need is a simple interface to the services on the underlying enabling infrastructure, both the wordprocessor functionality (downloadable code) and the necessary physical resources (processor cycles and storage space);
    - and everything is paid transparently via our telephone bill.

IT resources as **Utility**

Term **Utility Computing** suggested by Prof. John McCarthy in 1961

Today working solutions: Grid Computing and **Cloud Computing**

Cooperation is key – We are the Cloud

universität wien

"I want to see a Soccer play in HD on my iPhone"

- I lack on processing power and bandwidth

Why not using collaboratively some/all the iPhones in this room?

- My iPhone and the other iPhones are building a **Virtual Organization**
  - Transparently for me and the other iPhone owners
  - Bundle processing power and bandwidth
- Deliver the Soccer play in HD to me
- I will be charged 2 cts via my "telephone" bill
- The contributing iPhones share 1 ct
  - Will be credited via their telephone bill
- 1 ct will be earned by the infrastructure provider

Business in action, transparently!

So the Subject turns to Resource

Research Vision

Organisations and Businesses in **future service-oriented infrastructures** (**XaaS** – Everything as a Service) need to act in a more agile fashion than ever before.

The **ICT environment has to adapt automatically** to changing needs.

We need an adaptive service-oriented utility infrastructure for applications supporting **Virtual Organisations**

Virtual organizations are temporary or permanent alliances of enterprises or organizations that come together to share resources, skills, core competencies in order to better respond to business opportunities or large-scale application processing requirements, and whose cooperation is supported by computer networks.

We aim for the development of methods, techniques, and architectures to allow for an infrastructure for the automated and autonomous building of processes within and between virtual organisations.

Areas of research

Advanced Data Management

Technologies for management of data in focus

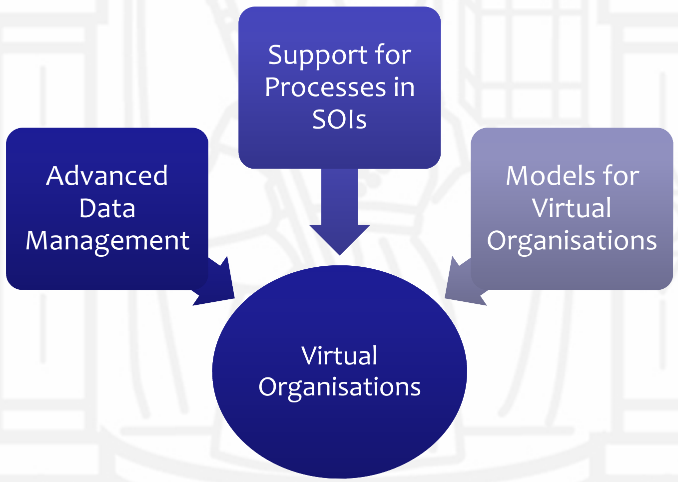
Support for Processes in Service Oriented Infrastructures

Establishment and management of global (business) workflows

(Business) Models for Virtual Organisations

Semantic component for knowledge based decision making in autonomous resources to allow for automated business processes

- Areas of research complement each other
- Integration of results delivers a comprehensive methodological and technological basis for our vision



Big Picture –Research Group

universität wien

Advanced Data Management

- **Distributed and Parallel Database Operations in Heterogenous Infrastructures (*Werner Mach*)**
- **Query Optimization in Service Oriented Database Architecture (*Peter Beran*)**
- **Distributed Databases for Large-Scale Scientific Data Processing (*Elisabeth Vinek*)**

Support for Processes in SOAs

- SLA Management (*Irfan Ul Haq*)
- RESTfull Self-repairing SOA infrastructure for workflows (*Jürgen Mangler*)
- Optimization of Workflows (*Helmut Wanek*)

Models for Virtual Organisations

- Refernce Architecture for Virtual Organisation (*Wajeeha Khalil*)
- Architectures for Biobanks (*Konrad Stark*)
- N2Cloud (*Altaf Ahmad Huqqani*)

Service Selection and Composition Optimization Framework

Presentation today is based on fruitful cooperation with

- **Elisabeth Vinek**

  Genetic algorithm

- **Peter Beran**

  Blackboard method

- **Werner Mach**

  Mathematical Modelling of Distributed DB Operators in Heterogeneous Environments

Introduction

Selection and **Composition** of **Services** in SOI/SOA is **key**

Builds basis for Virtual Organisations

Specific interesting for **data management issues on a world wide scale**

Heterogeneous environments have **no particular importance** in actual research until now

Parallel databases and their operations are well studied for homogeneous environments

Stimulating **new application domains**, huge data sets on a worldwide basis

e.g. **high energy physics,** bioinformatics, genomics

Different available infrastructures: **Web, Grids, (federated) Clouds, "Sky"**

Show typically **heterogeneous characteristics**

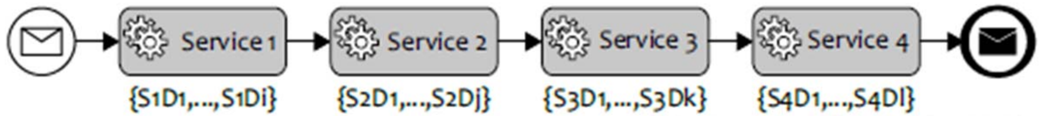
The demand in the industry for using heterogeneous environments is growing

Problem: workflow optimization and execution planning is **NP-hard**, thus we need **good heuristics**!

Problem Statement

Given: Workflow with **abstract services**

Simple Abstract Workflow (Sequence of Services)

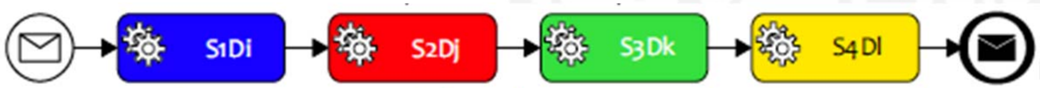


Several deployments of specific services exist

QoS (non-functional) properties allow to distinguish between deployments with identical functionality

e.g. better-worse reliability, expensive-cheap, ...

Goal: Select a **concrete deployment** for each abstract service AND **maximize a utility function** under satisfied constraints



Simple Concrete Workflow (Sequence of Deployments)

Mathematically a multi-dimensional multi-choice knapsack problem

Known NP-hard

Use Cases
Characteristics and Problem Cube

TAG Meta Data

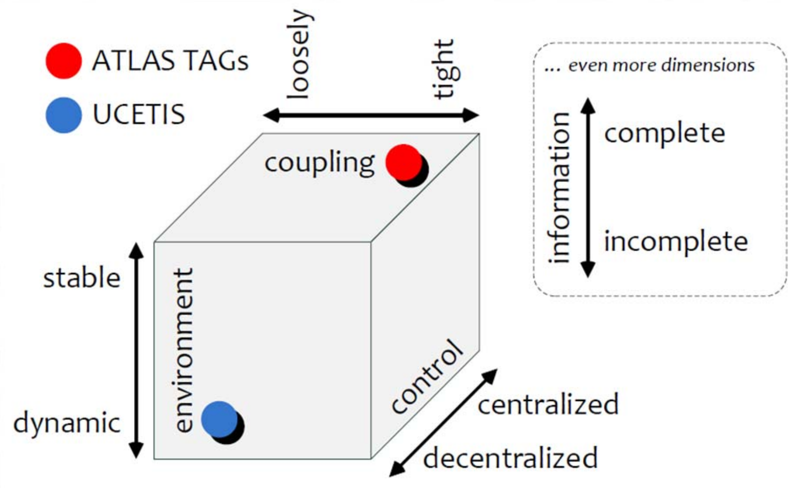
Metadata based pre-selection of high energy physics events of the ATLAS experiment at the CERN LHC

- tightly coupled components
- centralized control
- stable environment

University Information System UCETIS

Data access to a multitude of data scouces and applications of a University Information System

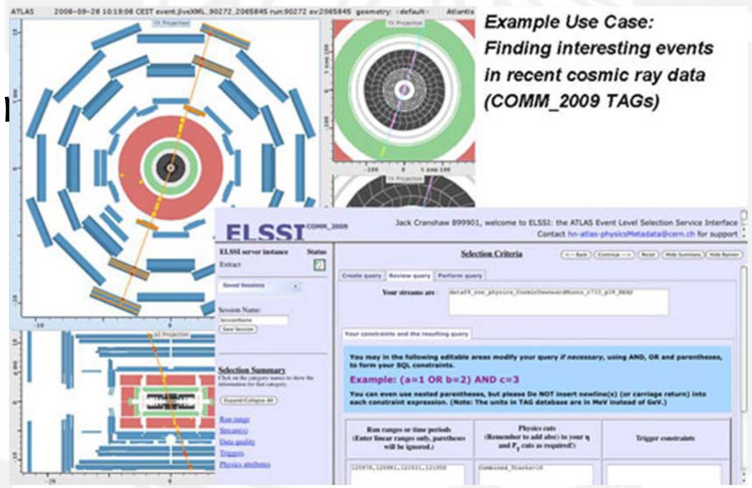
- loosely coupled components
- decentralized control
- dynamic environment



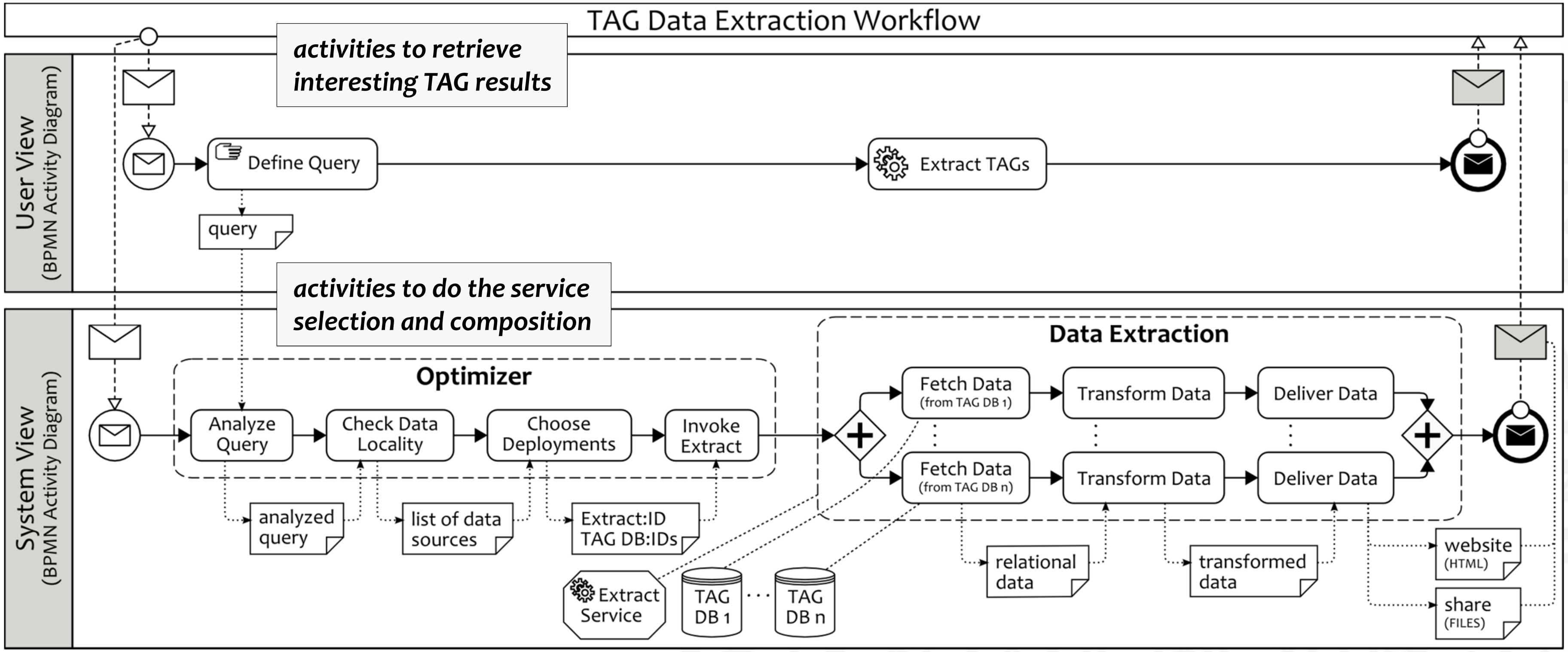
Motivational Application: TAG database ATLAS experiment LHC

universität wien

- Project “**Smart Selection of Distributed Database Services for Metadata Queries in the ATLAS Experiment**”
  - ATLAS is one of the detectors of the Large Hadron Collider (LHC) hosted at CERN, the European Organization for Nuclear Research
- Goal: to allow for a fast selection of physics events via TAGs
  - TAG is a small, high-level summary of an event with key physics quantities and information for back-navigation to the underlying Event
- Stored in relational databases
  - Mainly web-based services are provided for accessing data, making queries, transformi output formats, and shipping outputs
- **Databases and services are distributed/ replicated around the world**
- Design of a **service selection optimizer**
  - Relies on up-to-date system statistics



Motivating Example
The TAG Use-Case



TAG = event-based metadata for observed proton-proton collisions

- TAGs are distributed among ATLAS sites, are extracted by a:
  - **latency-sensitive** query: low data amounts have to be transferred
  - **bandwidth-sensitive** query: high data amounts have to be transferred

→ results in different optimization objectives (utilization vs. throughput vs. time)

Smart Selection of Distributed Database Services for Metadata Queries in the ATLAS Experiment

- massive amount of event-level metadata (TAGs)
  - stored in relational databases
  - accessible via Web services

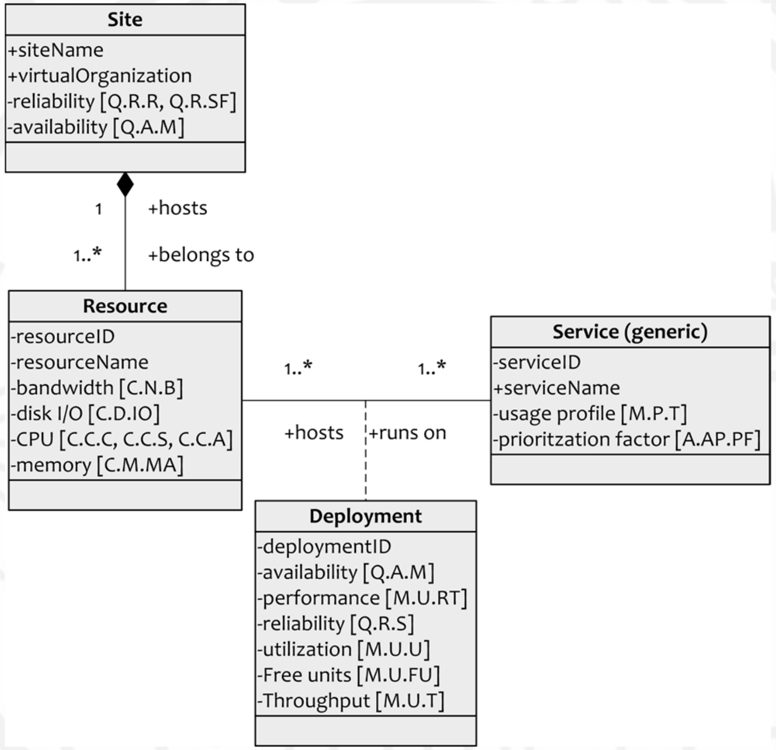
Problem

- **user perspective:**
  - make all TAG databases look like one
  - make service deployments transparent to the user
- **system perspective:**
  - ensure efficient use of available resources
  - treat all requests equally – or conforming to a defined policy
  - enable load balancing and fail over mechanisms

Solution: TASK (→)

- **metadata registry:** central control and information instance
- **service selection optimizer:** relies on up-to-date system statistics, which can be gathered from logs and/or by active/passive monitoring

- tightly coupled components
- centralized control
- stable environment

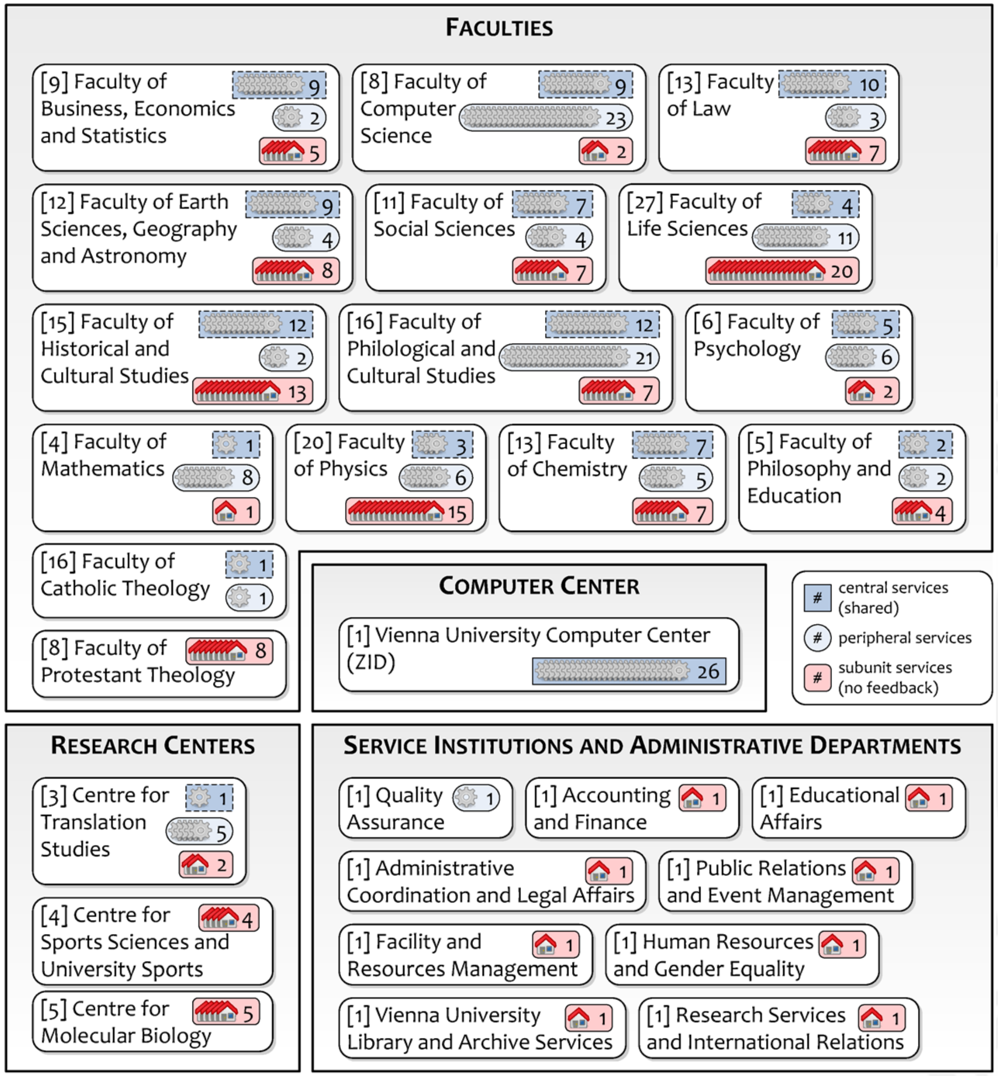


TASK
(TAG Application Service Knowledgebase)

Motivational Example

University Information System Use Case

universität wien



(University Cross European Transfer of Information System)
UCETIS survey in 2008

University of Vienna

- **organization:** 28 main units (departments) divided into 240 subunits (institutes)
- **software:** 107+ applications
  - 25 central
  - 82+ peripheral hosted by the institutes
- **hardware:** a lot of **workstations** but also some **Grid** and **Cluster** systems (Computers Science, Physics, Chemistry)

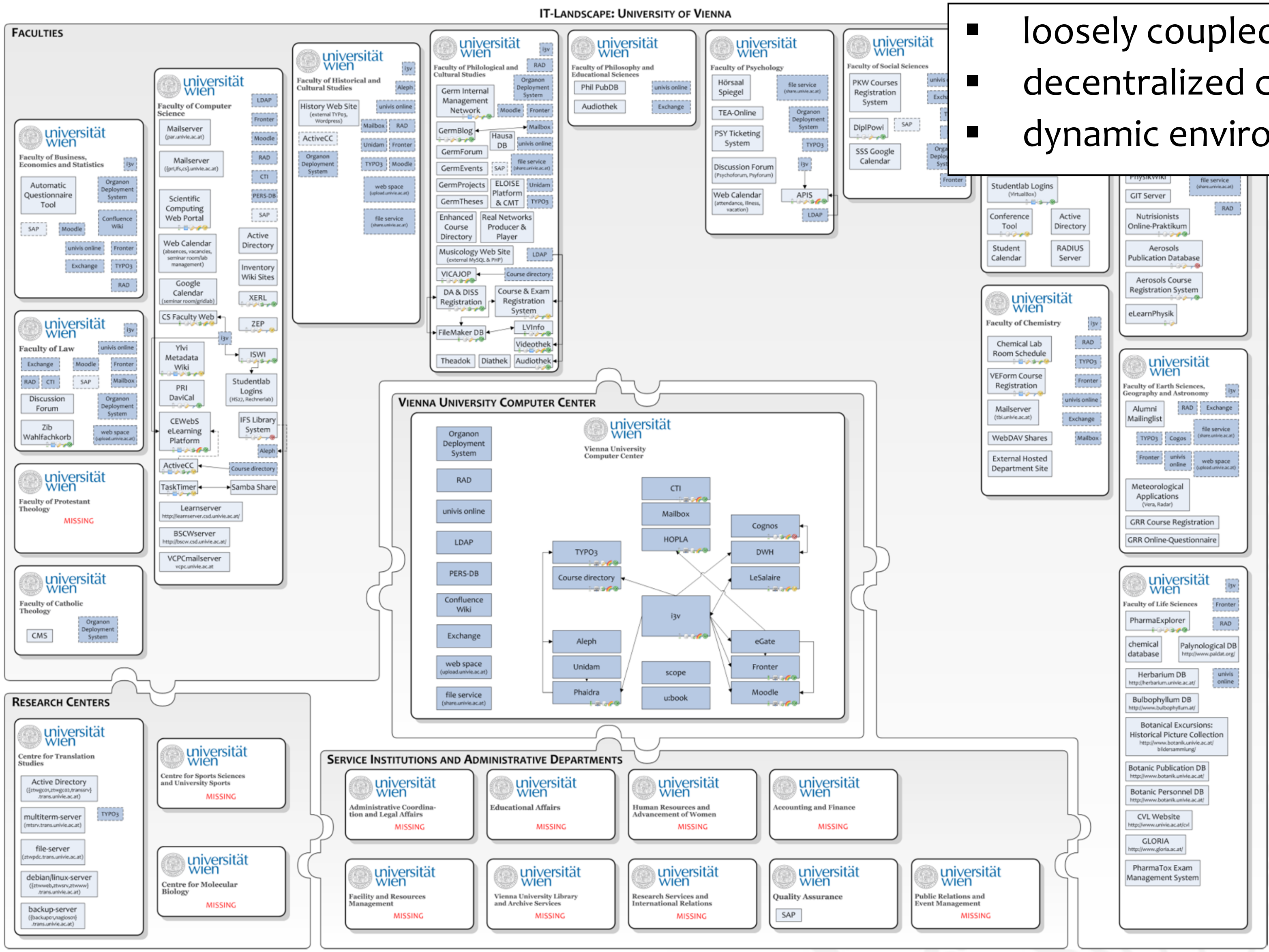
Problems

- various exchange formats/protocols
- multiple similar/overlapping apps
- weak connectivity between apps
- many database-related apps

Question: What have to be taken into account to choose the best app(s) for a given user request?

Model Verification and Applications

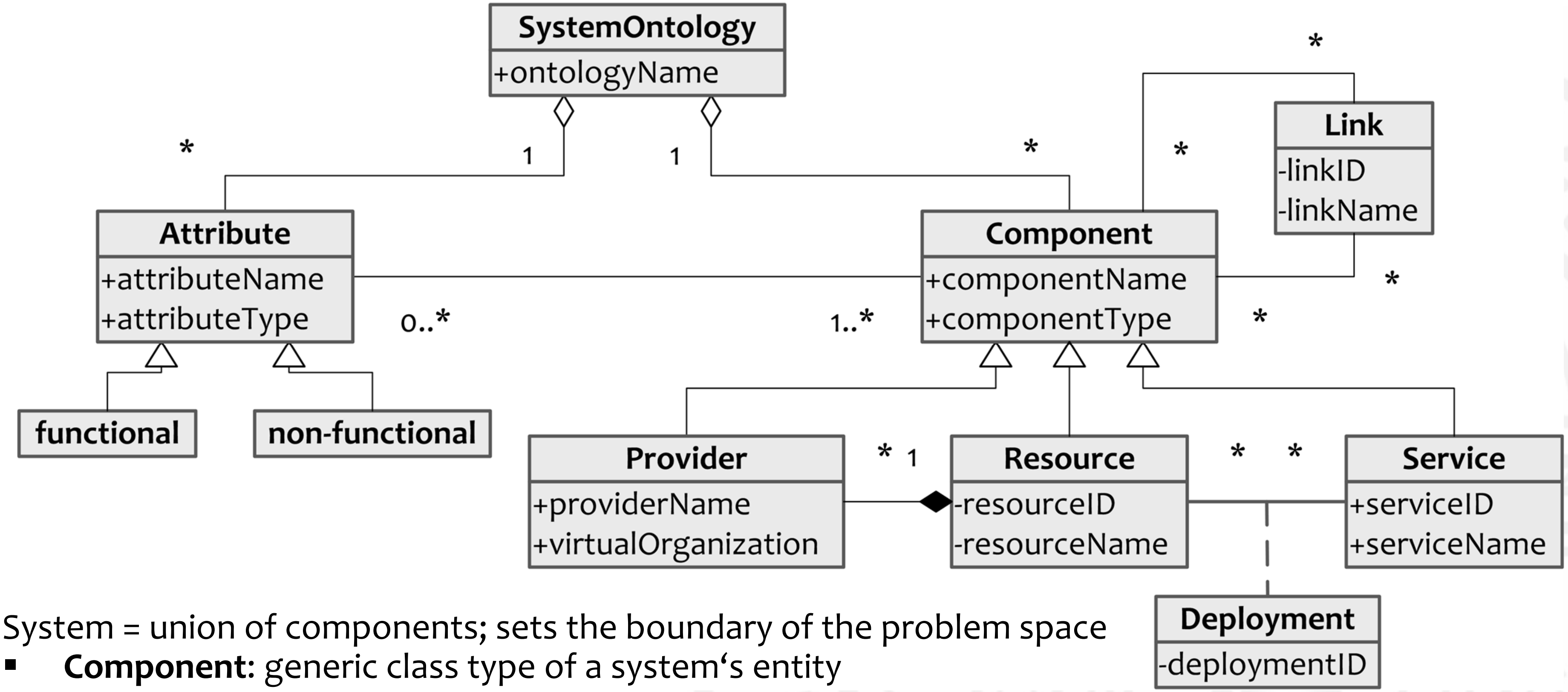
Distributed Database Management System (2)



- loosely coupled components
- decentralized control
- dynamic environment

QoS Data Model and Ontology

System Ontology



System = union of components; sets the boundary of the problem space

- **Component**: generic class type of a system‘s entity
- **Link**: allows to connect multiply *Components* (usually 1:1 connections)
- **Resource**: physical, logical or virtual entity (e.g. DB server, web server, virtual machine)
- **Service**: entity that captures a generic functionality (e.g. Oracle DB, Apache WebServer)
- **Deployment**: concrete instance of a *Service* running on a *Resource*
- **Provider**: owner or hoster of a *Component*
- **Attribute**: describe a *Component* regarding Functional (what?), Non-Functional Properties (how?), is defined more precisely in the QoS Attribute Ontology

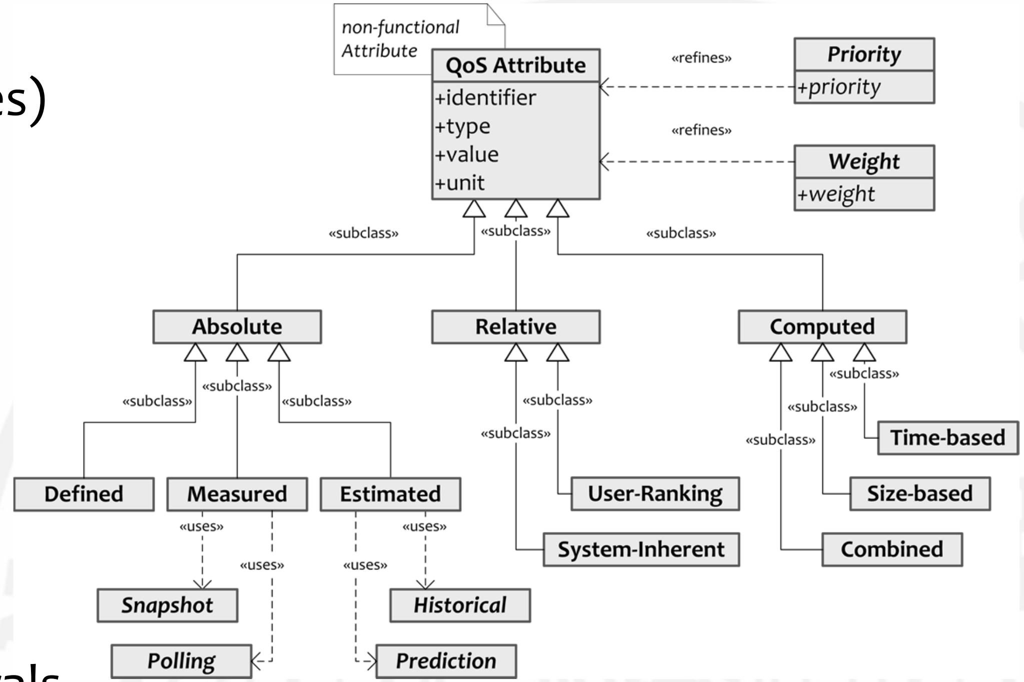
→ describe the service environment in a consequent and adequate manner

Attribute Categories

(reflecting the different attribute types)

- **Absolute:** basic, simple value of numeric, bool, char, date, list type
  - **Defined:** CPU count, RAM amount
  - **Measured:** CPU load, free RAM
  - **Estimated:** CPU time, used RAM
- **Relative:** value defined in scales
  - **User-Rating:** user satisfaction
  - **System-Inherent:** privacy, security
- **Computed:** value by several numerals
  - **Time-based:** availability
  - **Size-based:** reliability
  - **Combined:** performance

Are not mutually exclusive / can overlap!



Significance of Attributes

- **Priority:** ranks attributes according to their optimization importance
- **Weight:** ranks attributes according to their resource intensity

E. Vinek, P. P. Beran, and E. Schikuta, "Mapping distributed heterogeneous Systems to a Common Language by applying Ontologies," in *Tenth IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2011)*, Innsbruck, Austria, 2011

Classification and Description

Composition Patterns & Aggregation Functions

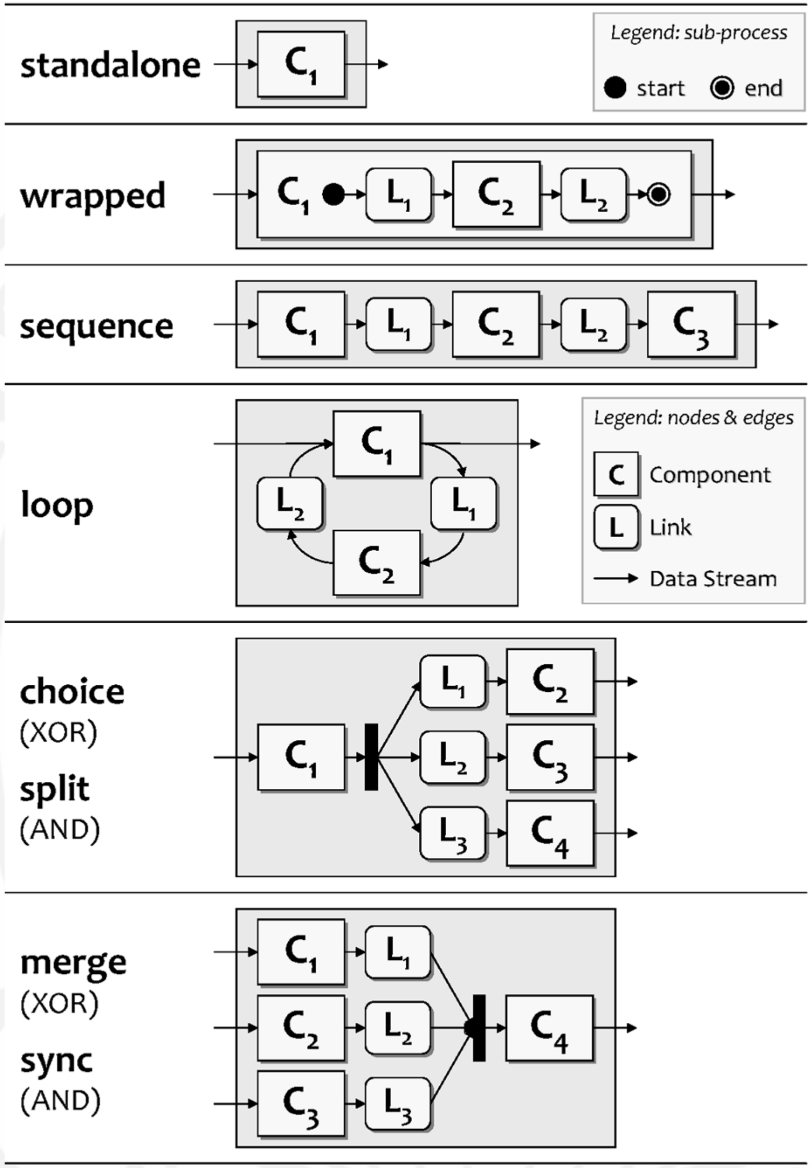
Composition Patterns

- **standalone:** a single component
- **wrapped:** embedded component(s)
- **sequence:** multiple components in a row
- **loop:** recurring components
- **choice:** xor-fork, only one subsequent component receives data
- **split:** and-fork, all subsequent components receive data
- **merge:** xor-join, only one component delivers data to the subsequent component
- **sync:** and-join, all components deliver data to the subsequent component

Aggregation Functions

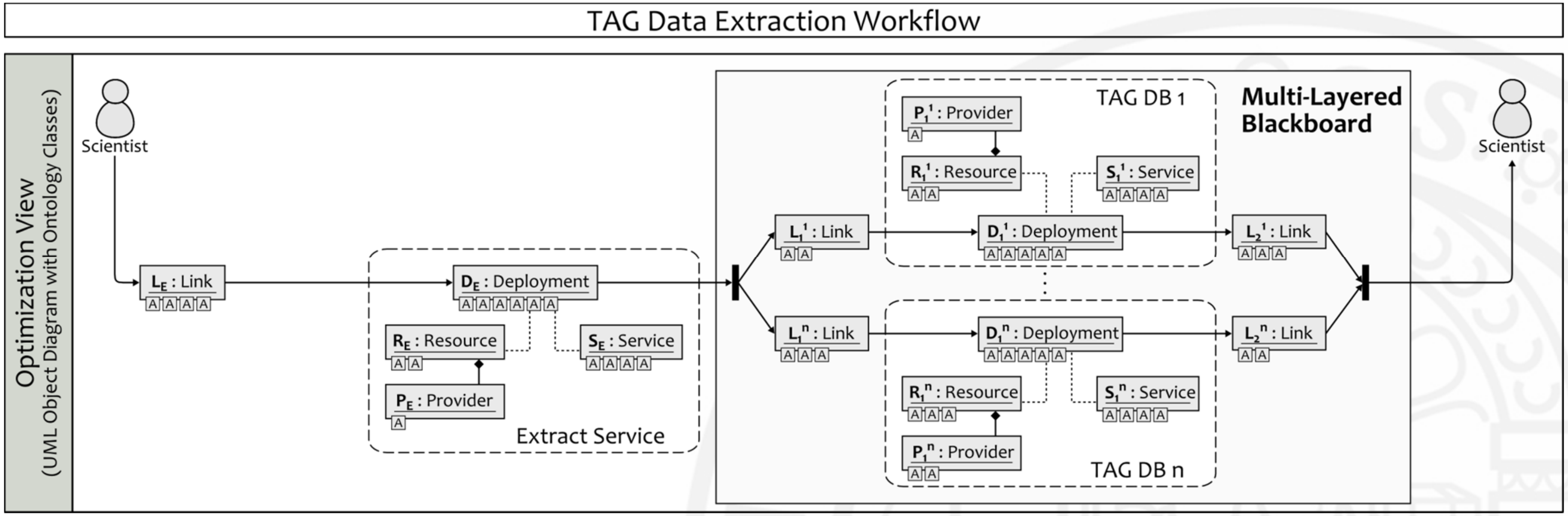
- average, sum, min, max, product, count, mode, mean, median, range, standard deviation, bestNofM

E. Vinek, P. P. Beran, and E. Schikuta, "Classification and Composition of QoS Attributes in Distributed, Heterogeneous Systems," in *11th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2011), Newport Beach, CA, USA, 2011*.



Motivating Example

TAG Workflow Model: Optimization View



Components: L_E, [D_E, R_E, P_E, S_E], L_1^1, [D_1^1, R_1^1, P_1^1, S_1^1], L_2^1, … L_1^n, [D_1^n, R_1^n, P_1^n, S_1^n], L_2^n

Composition Patterns:

- **2x sequence:** start to split (L_E, D_E), split to sync (L_1^x, D_1^x, L_2^x), whereas x = 1… n
- **1x split** (1 in n branches)
- **1x sync** (n to 1 branch)

Goal Function: **minimize Execution Time** (ET)

- λ = user satisfaction metric
- n = number of parallel branches
- k = number of components of a sequence

$$ET_{total} = ET(D_E) + ET(L_E) \max_{i=1\ldots n} \left(\sum_{j=1}^{k} \lambda_{D_j^i} ET(D_j^i) + \sum_{l=1}^{k+1} \lambda_{L_l^i} ET(L_l^i)\right)$$

Motivating Example
XML Binding for System/Attribute Ontologies

Necessary to describe the abilities of a system according to the ontologies

XML Schema Parts

- **Declaration (→):** describes **classes** (*SysOnto*) and **attributes**, covering their name, type (*AttrOnto*) and aggregation functions for composition patterns
- **Definition:** describes concrete instances of system components regarding the:
  - **Infrastructure (→):** lists all concrete components and links; defines their attributes and concrete values
  - **Connections:** define inter-connections between components and links to build-up the workflow graph

```
<declaration>
  <classes>
    <class>COMPONENT</class>
    <class>LINK</class>
  ...
  </classes>
  <attributes>
    <attr id="C.C.ET" unit="TIME">
      <name>Execution Time</name>
      <typeOf>ESTIMATED</typeOf>
      <typeOf>TIME-BASED</typeOf>
      <composition type="SEQUENCE">
        sum(i=1..n of fct(C[i])) + sum(j=1..n-1 of fct(L[i]))
      </composition>
      <composition type="SPLIT">
        fct(C[1]) + max(i=2..n of fct(L[i-1]) + fct(C[i]))
      </composition>
      <composition type="SYNC">
        max(i=1..n-1 of fct(C[i]) + fct(L[i])) + fct(C[n])
      </composition>
      ...
    </attr>
    ...
  </attributes>
</declaration>
```

```
<component id="d1">
  <typeOf>DEPLOYMENT</typeOf>
  <name>TAG DB (Tier 0, CERN) & Oracle 11g</name>
  <attributes>
    <attr id="C.C.ET" unit="msec">125</attr>
    <attr id="C.N.B" unit="MBit/s">1000</attr>
    ...
  </attributes>
</component>
```

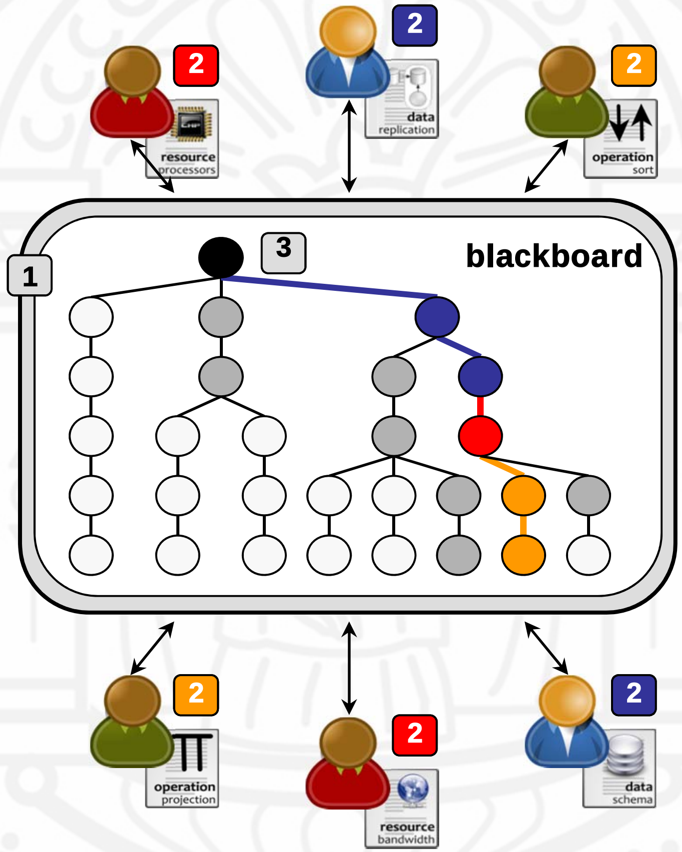
Blackboard Method

Heuristic approach, borrowed from AI

- Can cope with uncertain and incomplete knowledge

Idea (simplified)

- **Experts** gather around a blackboard
- Experts have **specific competence** (disjoint) in a specific area
- The **problem** is put on the **blackboard**
- **Each expert improves** the solution to the problem by his specific knowledge
- Step is repeated until **solution** is produced



Can reduce the search space dramatically

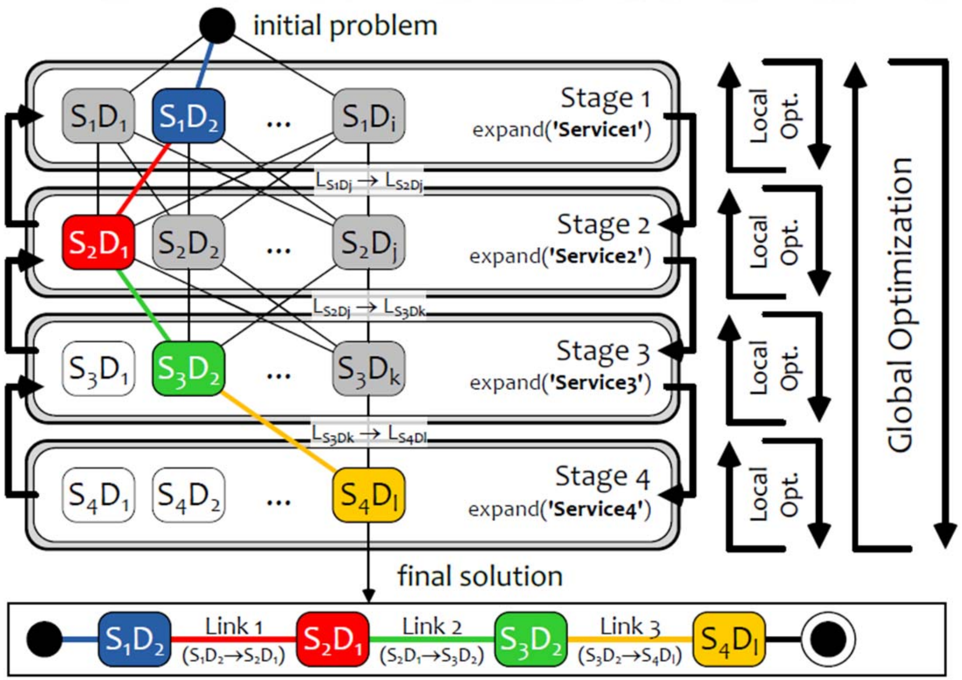
Blackboard Architecture consists of …

1. **global blackboard:** shared space
2. **knowledge base:** expert regions
3. **control component:** phases and activities

Blackboard Query Optimization

Architecture & Components

- Realization by A/A* algorithm
- Idea, try most promising solution first
  - Goal function: minimize cost(v) = v.c + h(v) h(v) estimation of costs
  - Each expert rule generates possible sub-solutions (expansions)
  - Sub-solutions are evaluated by cost function and sorted accordingly
  - „best" sub-solution is expanded next
- Allows to prune search space
  - avoids exhaustive search
    - e.g., white rectangles never visited
      - only grey are candidates
- If algorithm underestimates real costs, it finds always the best solution



P. P. Beran, W. Mach, E. Schikuta, and R. Vigne, "A Multi-Staged Blackboard Query Optimization Framework for World-Spanning Distributed Database Resources," in *International Conference on Computational Science (ICCS 2011)*, Singapore, 2011.

Problem solution approach similar to biological evolution

- Produce solution proposal repetively by mutation, crossover and selection

Start by encoding problem with a suitable genome

Selection

- Survival probability of chromosome reflected by its fittness (copied n times), random selection

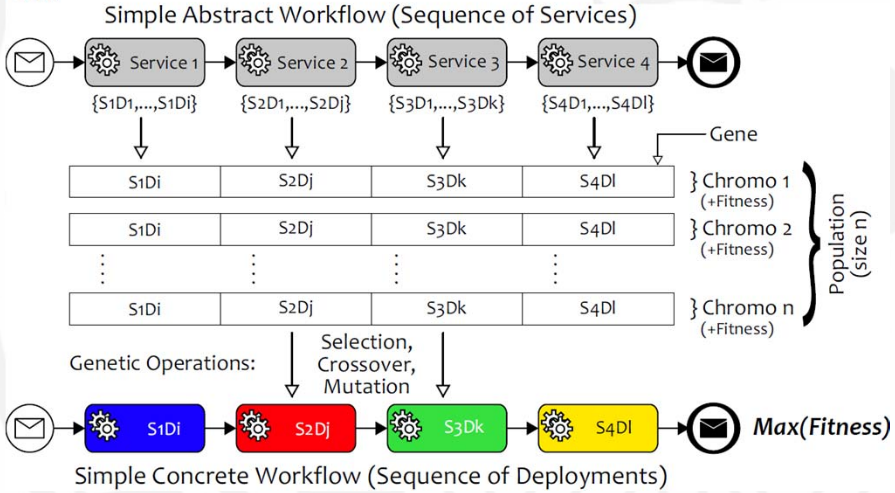
Crossover

- Two chromosomes (parents) breed two children, i.e. head from one parent tail from the other (p=0.65)

Mutation

- Spontaneous random changes, i.e. single gene replaced by another one (p=0.035)

$t = 0$ and initialize population $P(0)$ randomly;
for $t = 0$ *to generations* **do**
  evaluate($P(t)$, $M(t)$);
  $P'(t)$ = rouletteWheelSelection($P'(t)$);
  crossover($P'(t)$, p_c);
  mutate($P'(t)$, p_m);
  $t = t + 1$;
end



Solution are Chromosomes, i.e. ordered list of deployment names or refs

E. Vinek, P. P. Beran, and E. Schikuta, “A Multi-Staged Blackboard Query Optimization Framework for World-Spanning Distributed Database Resources,” in *International Conference on Computational Science (ICCS 2011)*, Singapore, 2011

Genetic Algorithm coping with dynamics

Challenge: capture system dynamics such as

- added or removed services and resources
- changes in system behavior (logging information available)
- changes in optimization objectives, contraints, or prioritization

Idea: Dynamic Genetic Algorithm

- Using explicit memory techniques(2)
- Re-use of previously computed solutions based on system state similarity
- Depending on the similarity degree, a fraction of the „old“ solutions is feeded into the initial population

```
t := 0 and initialize population P(0) randomly;
t_M := rand(5, 10) and initialize memory M(0) randomly;
v_0 := 0 (default profile vector);
repeat
    evaluate(P(t),M(t));
    if v! = v_0 then
        find the closest profile vector V_M(t);
        denote the best memory point ⟨B_M(t), V_M(t)⟩;
        I(t) := create α × (n − m) individuals from V_M(t);
        P'(t) := replace the worst individuals in P(t) by ones in I(t);
    else
        P'(t) := P(t);
    end
    if t = t_M then
        t_M := t+rand(5, 10);
        denote the best individual in P'(t) by B_P(t);
        V_P(t) := profile vector in P'(t);
        if still any random point in M(t) then
            replace a random memory point by ⟨B_P(t), V_P(t)⟩;
        else
            ⟨S^c_M(t), V^c_M(t)⟩:= the memory point closest to ⟨B_P(t), V_P(t)⟩;
            if f(B_P(t)) ≥ f(S^c_M(t)) then
                ⟨S^c_M(t), V^c_M(t)⟩ := ⟨B_p(t), V_p(t)⟩;
            end
        end
    end
    P'(t) := selectForReproduction(P'(t));
    crossover(P'(t), p_c);
    mutate(P'(t), p_m);
until termination condition true;
```

Performance Analysis

Three algorithms implemented

- Random Search (for setting the fitness baseline)
- Blackboard Method
- Genetic Algorithm

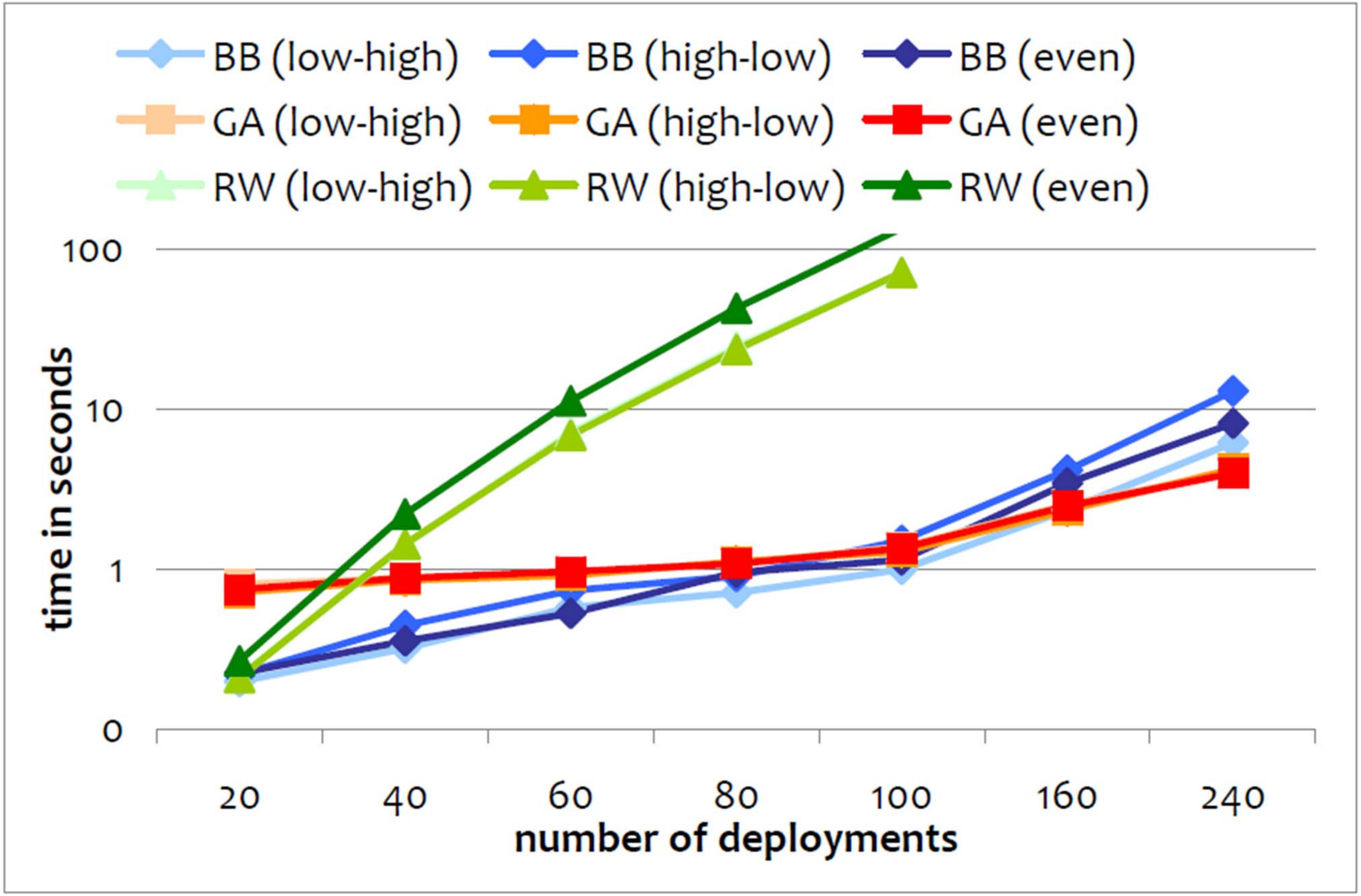
Seven test cases for 20, 40, 60, 80, 100, 160 and 240 deployments

For each test case three deployment distribution scenarios

- Equal (e.g. 5-5-5-5)
- High-low (e.g. 8-6-4-2)
- Low-high (e.g. 2-4-6-8)

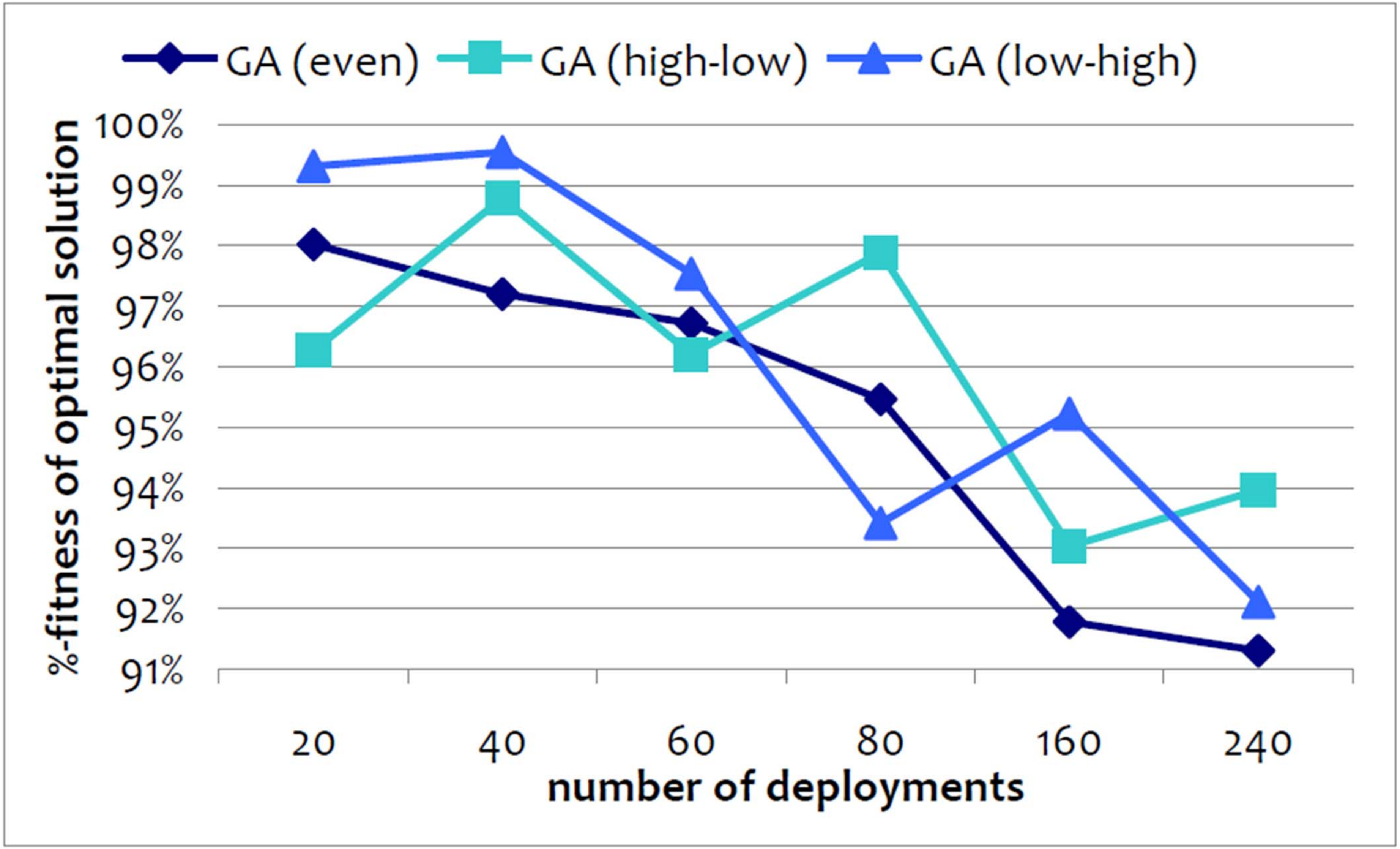
Basis for test data TAG db

Performance (in Time)



Performance of GA (in Quality/Fitness)

universität wien



Lessons Learned

All three approaches feasible

- Random search grows exponential

Performance

- BB outperforms GA up to 100 deployments, after this GA shows better performance than BB
- Regarding deployment distribution no performance implication for GA and RW
- BA bad for (high-low) distribution

Fitness

- BA always returned optimal solution
- GA not less than 91% from optimum

Future Work

Towards a fully autonomous and automatic adaptive system service selection and composition optimization suite

- Adapting to different goal functions reflecting different strategies/policies
- Adapting to new user needs (physists in TAG use case) by identifying user profiles and shaping optimization process
- Evaluation of new optimization approaches, e.g. neural networks, etc.
- Extension to database query optimization
- Parallelization of optimization process

Extension to cope with dynamic behaviour of environment

Hierarchical Blackboard Approach to support federated Clouds

- To cope with visibility, security and information hiding issues

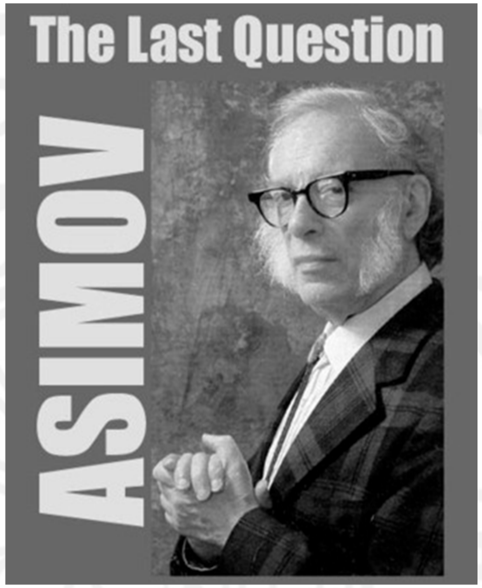
Finally: The Circle Closes

Remember my IT vision from the beginning:

Subject becomes resource!

Isaac Asimov, „The Last Question“:

http://www.multivax.com/last_question.html



“ … And it came to pass that AC learned how to reverse the direction of entropy.

But there was now no man to whom AC might give the answer of the last question. No matter. The answer – by demonstration -- would take care of that, too.

For another timeless interval, AC thought how best to do this. Carefully, AC organized the program.

The consciousness of AC encompassed all of what had once been a Universe and brooded over what was now Chaos. Step by step, it must be done.

And AC said, "LET THERE BE LIGHT!" And there was light---- “

The resource turned into subject! So be careful!

universität wien

Faculty of Computer Science

Questions?

Erich Schikuta

erich.schikuta@univie.ac.at

http://www.cs.univie.ac.at/erich.schikuta

Elisabeth Vinek

elisabeth.vinek@cern.ch

Peter Paul Beran

peter.beran@univie.ac.at

Werner Mach

werner.mach@chello.at