An XML-based Framework for Loosely Coupled Applications on Grid Environments

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

An XML-based Framework for Loosely Coupled Applications on Grid Environments

Fraunhofer Resource Grid

Andreas Hoheisel
(andreas.hoheisel@first.fraunhofer.de)

Uwe Der
(uwe.der@first.fraunhofer.de)

FhRG
Fraunhofer Resource Grid

Fraunhofer FIRST Institut
Rechnerarchitektur
und Softwaretechnik

Outline

Fraunhofer Resource Grid

Grid Application Definition Language

Why Petri Nets?

Grid Job Handler

Application: Environmental Risk Analysis and Management System (ERAMAS)

Conclusions and Future Work

Fraunhofer Resource Grid

Fraunhofer Resource Grid (FhRG)

Challenge

Development and implementation of a stable and robust grid infrastructure

Software layer on top of Globus to enable fast realizations of distributed applications

Integration of available resources

Institutes of the Fraunhofer Resource Grid

ITWM

Institut
Techno- und
Wirtschaftsmathematik

FIRST

Institut
Rechnerarchitektur
und Softwaretechnik

IAO

Institut
Arbeitswirtschaft und
Organisation

IGD

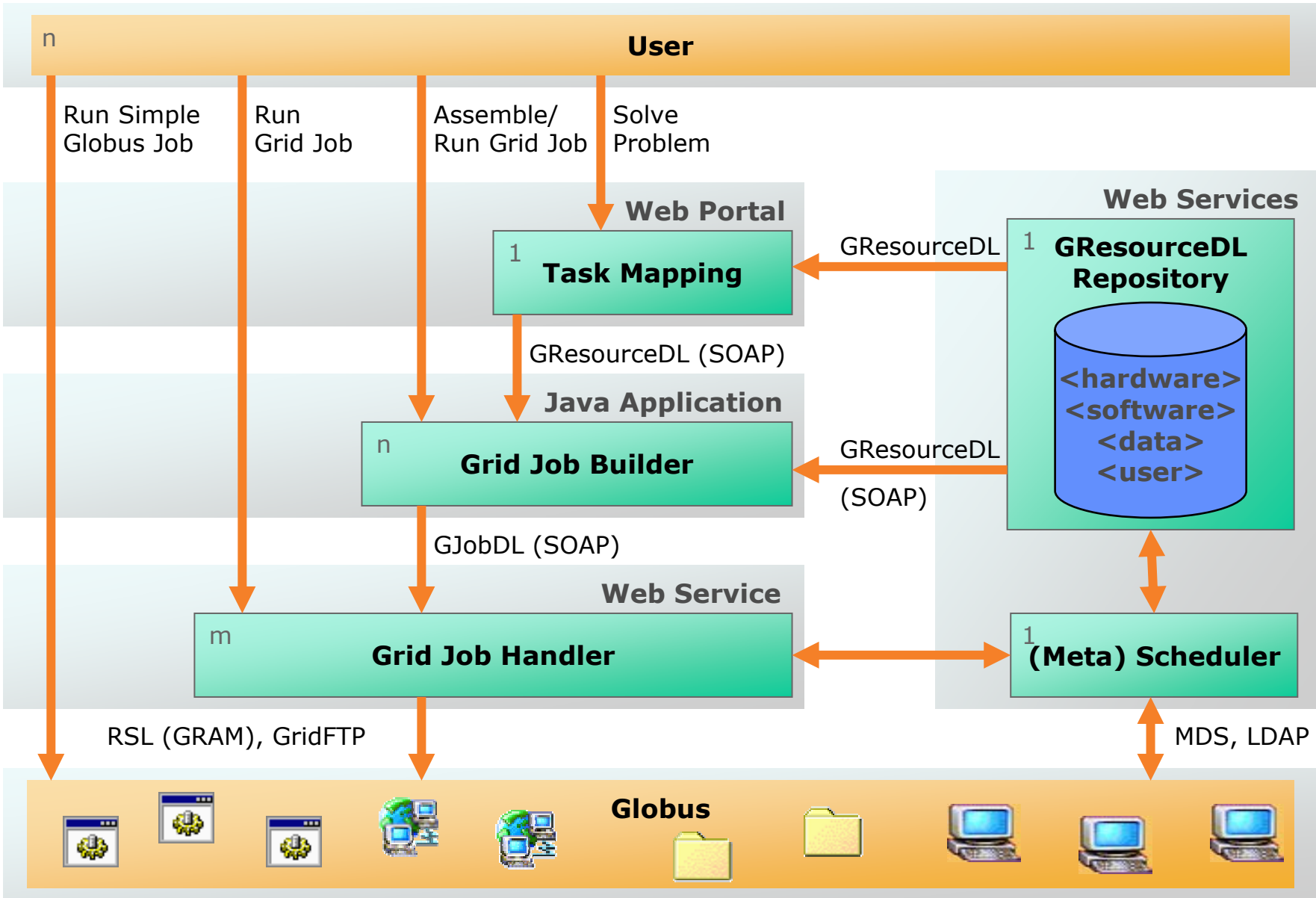
Institut
Graphische
Datenverarbeitung

SIT

Institut
Sichere Telekooperation

FhRG Fraunhofer Resource Grid

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik


n
User
Run Simple Globus Job
Run Grid Job
Assemble/ Run Grid Job
Solve Problem
Web Portal
1
Task Mapping
GResourceDL
Web Services
1
GResourceDL Repository
<hardware>
<software>
<data>
<user>
GResourceDL (SOAP)
Java Application
n
Grid Job Builder
GResourceDL
(SOAP)
GJobDL (SOAP)
Web Service
m
Grid Job Handler
1
(Meta) Scheduler
RSL (GRAM), GridFTP
MDS, LDAP
Globus

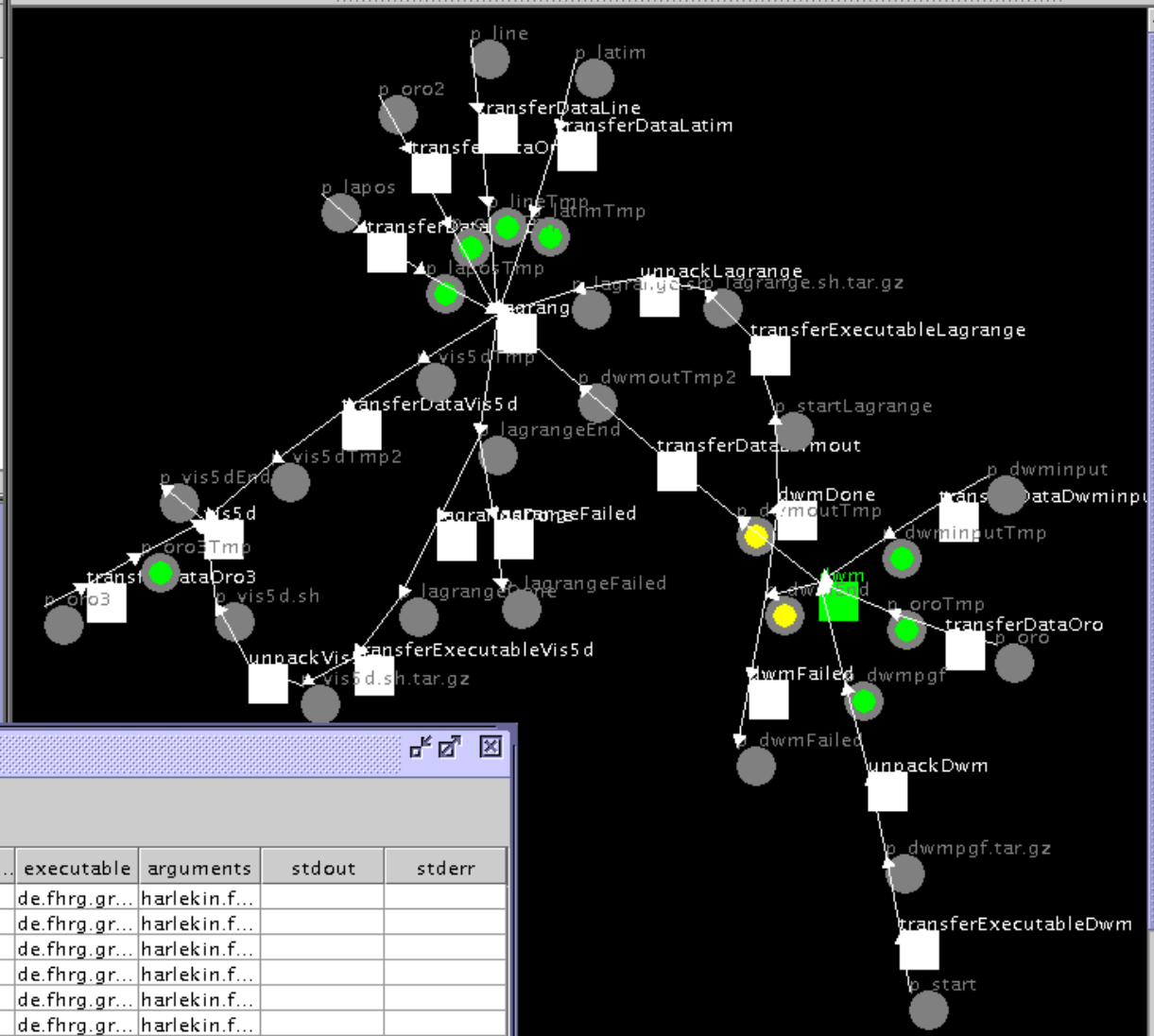
File Grid Job Security Help

GJobDL: /export/demeter/hoheisel/I-Lab/executo

Create Grid Job

```
        <transitionRef id = "transferDataOro3"/>
    </arc>
    <arc id = "arc16b" type = "T2P">
        <transitionRef id = "transferDataOro3"/>
        <placeRef id = "p_oro3Tmp"/>
    </arc>
    <arc id = "arc16c" type = "P2T">
        <placeRef id = "p_oro3Tmp"/>
        <transitionRef id = "vis5d">
            <inputRef id = "oro" type = "file"/>
        </transitionRef>
```

Petri Net -- Grid Job #1



Grid Job #1

Run | Show Workflow | Stop

status	start time	end time	ID	resourceM...	executable	arguments	stdout	stderr
DONE	Feb 05 11...	Feb 05 11...	1	local	de.fhrg.gr...	harlekin.f...		
DONE	Feb 05 11...	Feb 05 11...	2	local	de.fhrg.gr...	harlekin.f...		
DONE	Feb 05 11...	Feb 05 11...	3	local	de.fhrg.gr...	harlekin.f...		
DONE	Feb 05 11...	Feb 05 11...	4	local	de.fhrg.gr...	harlekin.f...		
DONE	Feb 05 11...	Feb 05 11...	5	local	de.fhrg.gr...	harlekin.f...		
DONE	Feb 05 11...	Feb 05 11...	6	local	de.fhrg.gr...	harlekin.f...		
DONE	Feb 05 11...	Feb 05 11...	7	local	de.fhrg.gr...	harlekin.f...		
DONE	Feb 05 11...	Feb 05 11...	8	local	de.fhrg.gr...	harlekin.f...		
DONE	Feb 05 11...	Feb 05 11...	9	p1.itwm.f...	/bin/tar	-xzf tmp...	tmp1044...	tmp1044...
ACTIVE	Feb 05 11...	N/A	10	p1.itwm.f...	dwmpgf	-output d...	dwmpgf.s...	dwmpgf.s...

Grid Application Definition Language (GADL)

Grid Application Definition Language (GADL)

GADL	Set of XML-based description languages needed to define and to execute grid applications

The GADL consists of:

GResourceDL	Description of resources
GJobDL	Description of grid jobs → Set of resources + workflow
GInterfaceDL	Interface definition of software components
GDataDL	Description of data

GADL: Grid Resource Definition Language (GResourceDL)

GResourceDL

Description language for categorization and description of resources

Used to select suitable resources to solve a given problem (task mapping)

Definition of dependencies between resources

Extension of resource descriptions with inheritance allows formulation of recursive descriptions

Everything is a resource!

Software components

Hardware resources

Measuring devices

Data

GResourceDL example that depends on other resources

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE fhrgResources (View Source for full doctype...)>
<fhrgResources>
 <resource id="sleep" type="software">
 ...
 <dependencies type="depends">
  <resourceRef id="linux" type="softwareClass" />
  <resourceRef id="glibc6" type="softwareClass" />
  <resourceRef id="x86" type="hardwareClass" />
 </dependencies>
 ...
</fhrgResources>
```

depends, conflicts, provides, suggests

GResourceDL example that provides other resources

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE fhrgResources (View Source for full doctype...)>
<fhrgResources>
 ...
 <resource id="de-fhrg-first_harlekin" type="hardware">
  <dependencies type="provides">
   <resourceRef id="x86" type="hardwareClass" />
   <resourceRef id="network-ethernet-100" type="hardwareClass" />
   <resourceRef id="linux-kernel-2-4-18" type="softwareClass" />
   <resourceRef id="glibc6" type="softwareClass" />
  </dependencies>
  <authorization>
   <userGroup id="all" read="true" write="false" />
   <userGroup id="first" read="true" write="true" execute="true" />
  </authorization>
 </resource>
</fhrgResources>
```

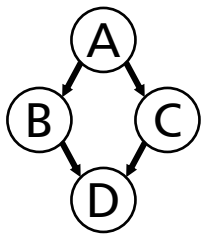
Why Petri Nets?

GADL: Grid Job Definition Language (GJobDL)

Problem

Description of complex workflows of grid jobs

DAG

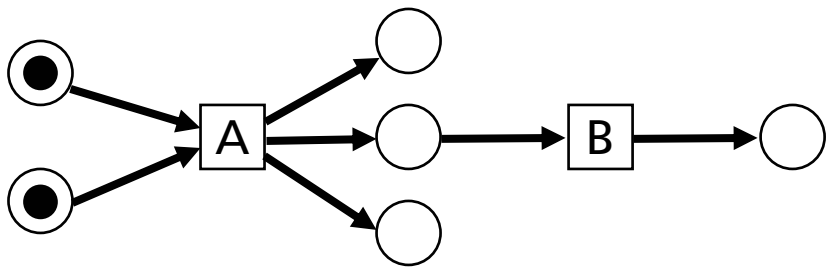


Directed Acyclic Graph (see Condor, Cactus, UNICORE)
no bidirectional coupling (interaction)
no loops

```
PARENT A CHILD B C
PARENT B C CHILD D
```

Petri Nets

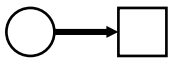
Graphical flow control of discrete systems



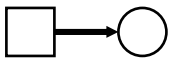
Petri Nets

Places Files, buffers, control places

Transitions Software components, control transitions



Arcs from places to transitions (Place is input place of transition)



Arcs from transitions to places (Place is output place of transition)

Tokens Data, State (active, done)

Rules A transition is activated if all input places are filled with tokens and all output places have not reached their maximum capacity of tokens

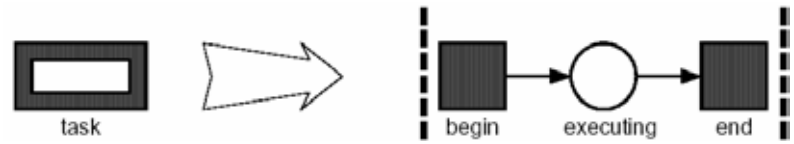
Refinement A transition can be replaced by a Petri Net

Description of state A Petri Net describes workflow and state of a system

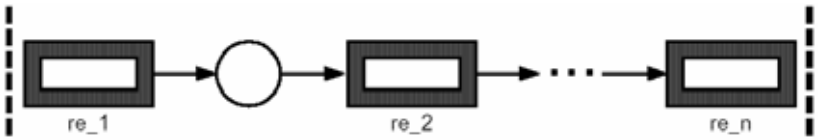
Petri Nets

(from *van der Aalst und Kumar, 2000*)

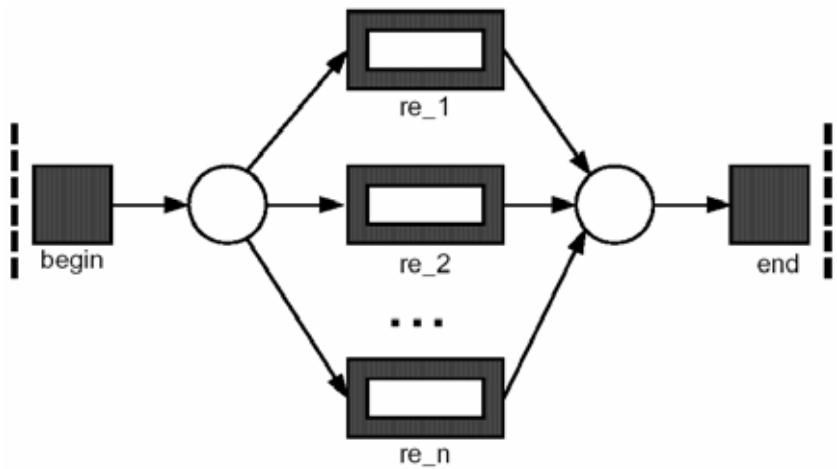
Task



Sequence

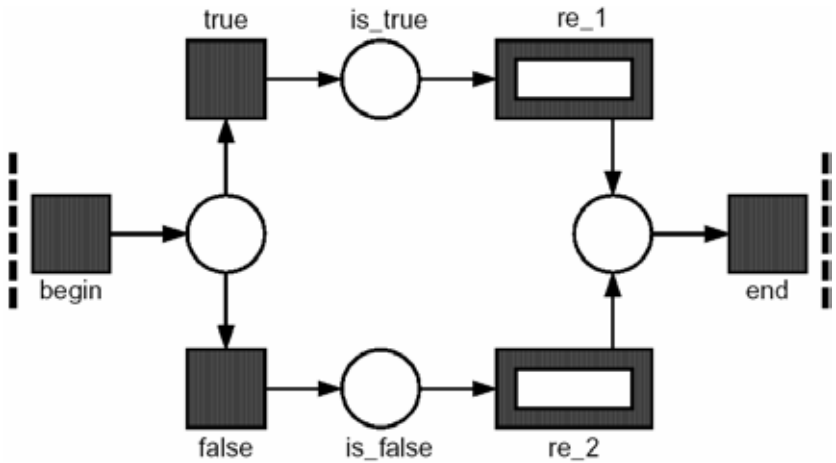


Choice



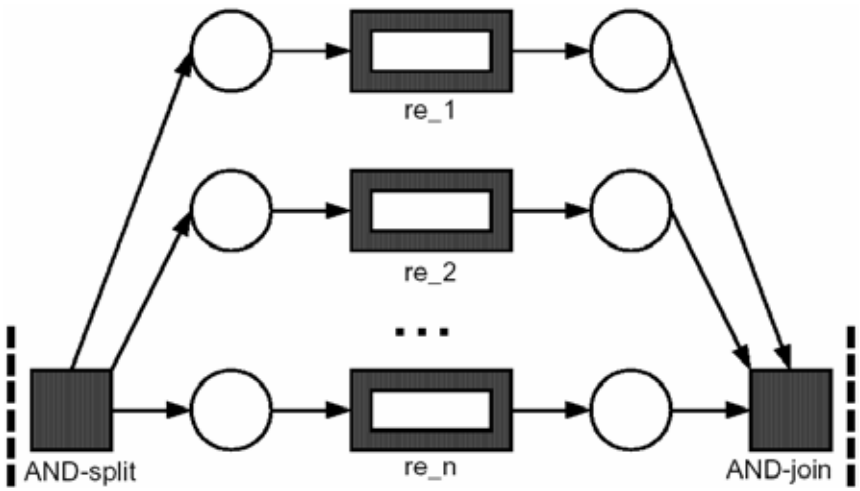
Petri Nets

Condition



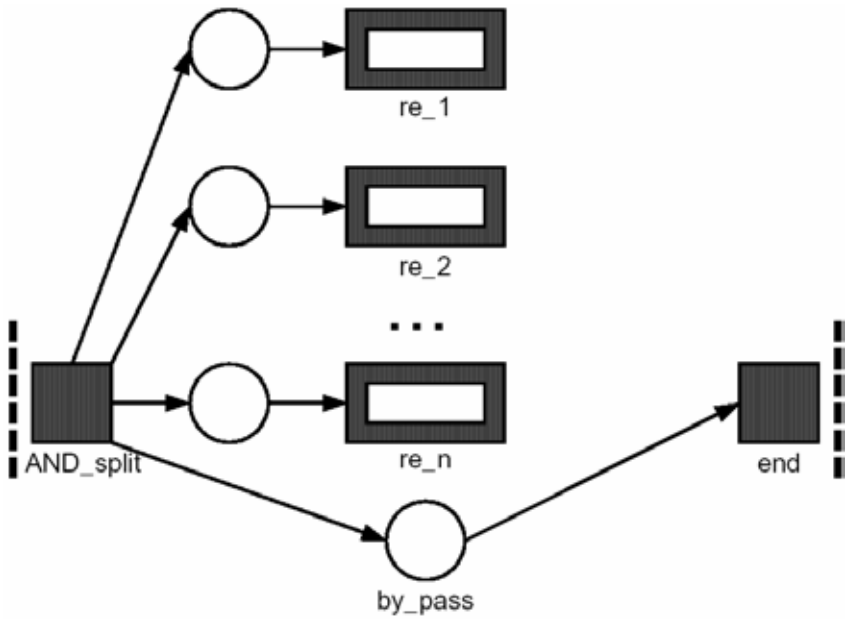
Petri Nets

Parallel execution with synchronization



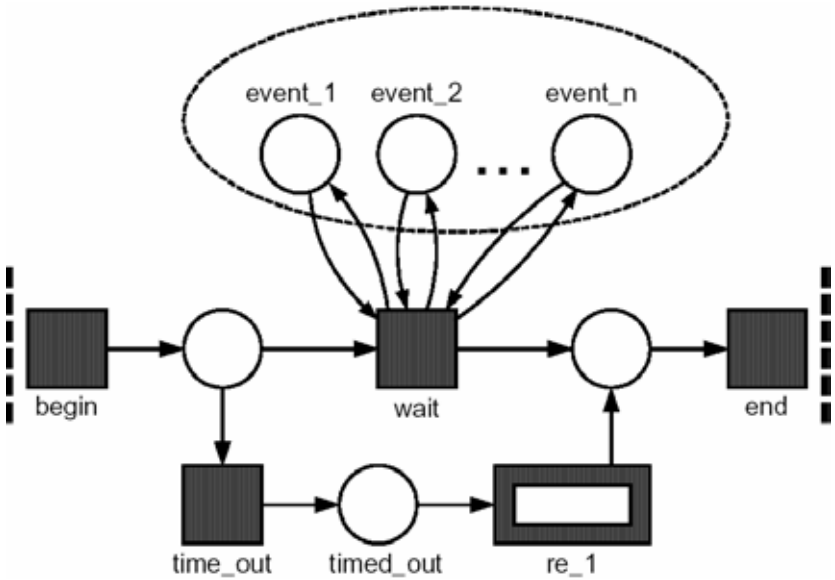
Petri Nets

Parallel execution without synchronization



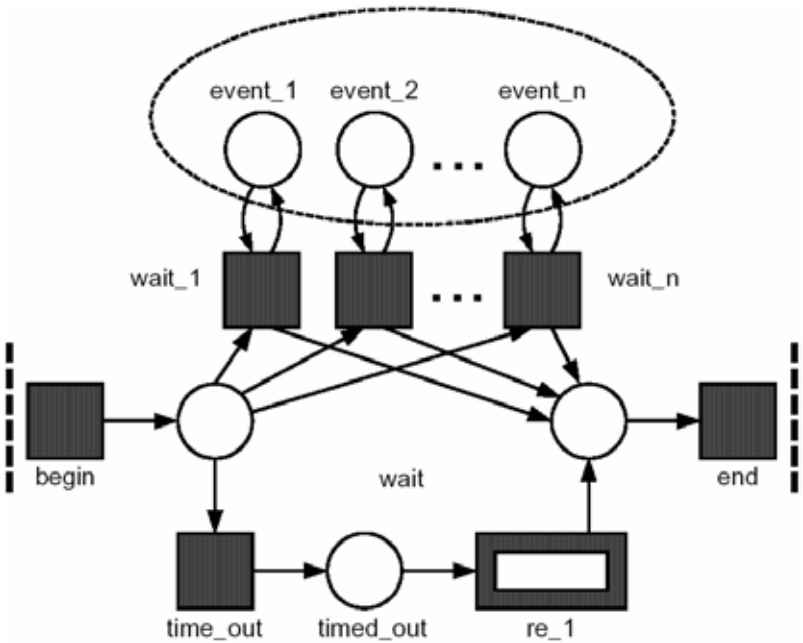
Petri Nets

Wait all with time out



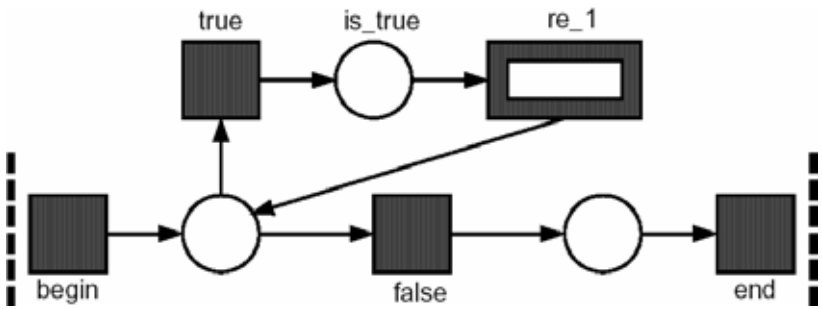
Petri Nets

Wait any with time out



Petri Nets

While … do



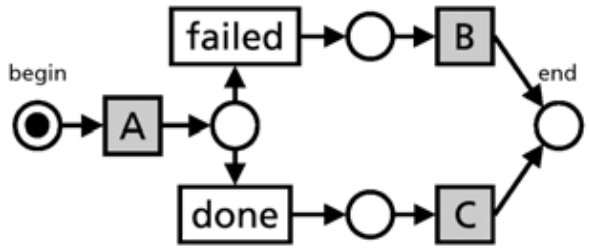
Transitions of type “MethodCall“

```
<transition id = "unpack_data">
  <methodCall>unpack()</methodCall>
</transition>
```

- unpack()
- transferData()
- transferExecutable(String softwareTransitionId)
- transferData(String softwareTransitionId)

Transitions of type “Condition“

```
<transition id = "A_done">
  <condition>isDone()</condition>
</transition>
```



- isDone()
- isFailed()
- countLT(int maxCount)
- negate(String transitionID)

Description of Petri Nets in XML

PNML — Petri Net Markup Language
(modified from *Jüngel, Kindler, Weber; HU-Berlin*)

```
<place id="start">
  <initialMarking>
    <value type="boolean" op="eq">true</value>
  </initialMarking>
</place>
<place id="output"/>
<transition id="program"/>
<arc type="P2T" id="arc1">
  <placeRef id="start" />
  <transitionRef id="program" />
</arc>
<arc type="T2P" id="arc2">
  <transitionRef id="program"/>
  <placeRef id="output" />
</arc>
```

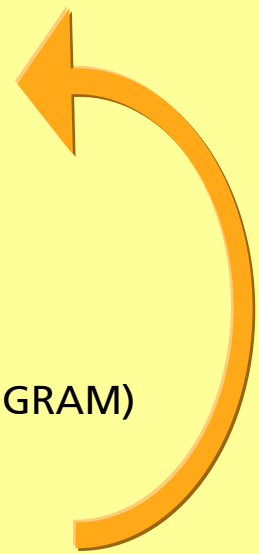
Grid Job Handler

Grid Job Handler

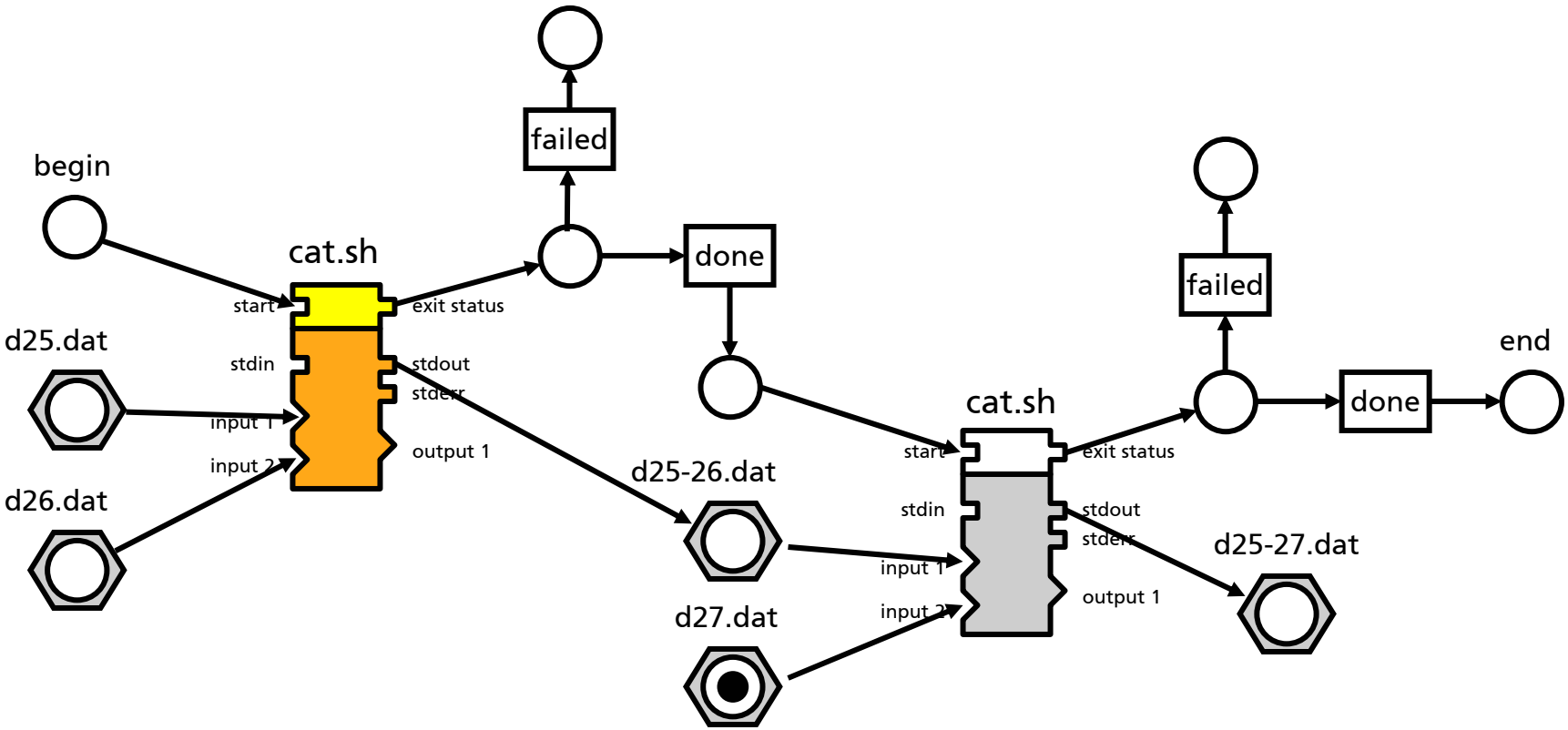
Grid Job Handler	Software for executing and controlling coupled grid applications
	Workflow modeling
	Mapping of Grid Jobs onto appropriate distributed resources
	Execution of jobs using grid middleware (e.g. Globus → Java Commodity Grid Kit)
Component environment	Coupling of components using file input/output
Monitoring	Job status (pending, active, failed, done)
Integration into the FhRG	Communication with other FhRG services using SOAP (→ WebService)

Application flow of Grid Job Handler

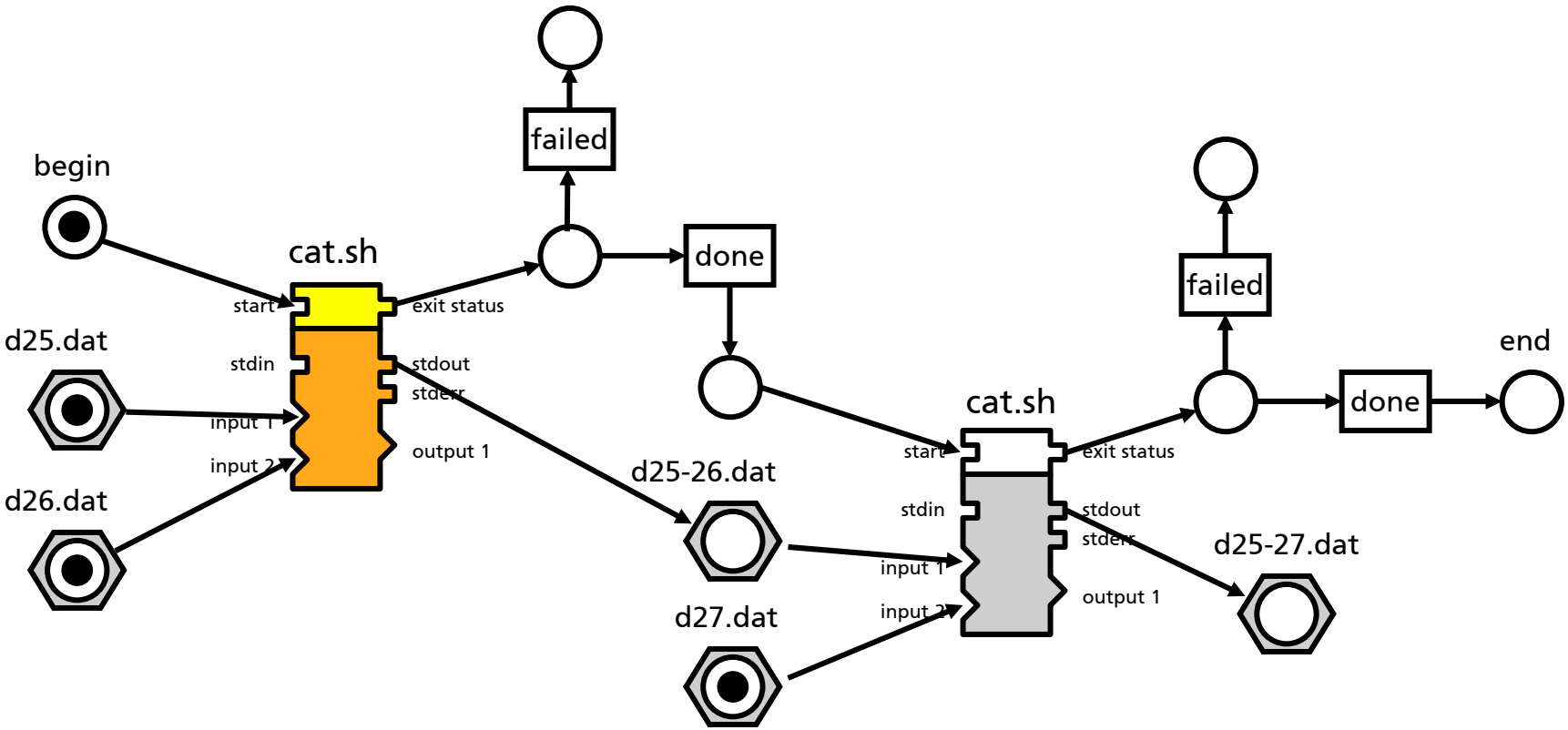
- ✓ Read the GJobDL document
- ✓ Create Petri Net from this job description
- • Verify the Petri Net (well-formedness, liveliness, deadlocks, pits, ...)
- ✓ Start the Grid Job (own thread)
- ✓ Collect all enabled transitions
- ✓ Evaluate conditions
- ✓ Lock input and output places
- ✓ Invoke resource mapping → repository, (meta-)scheduler
- ✓ Refine the Petri Net → insert GridFTPs if necessary
- ✓ Create and submit atomic jobs using grid middleware (e.g. GRAM)
- ✓ The transition fires, if atomic job is “done“ or has “failed“.
  Unlock input and output places



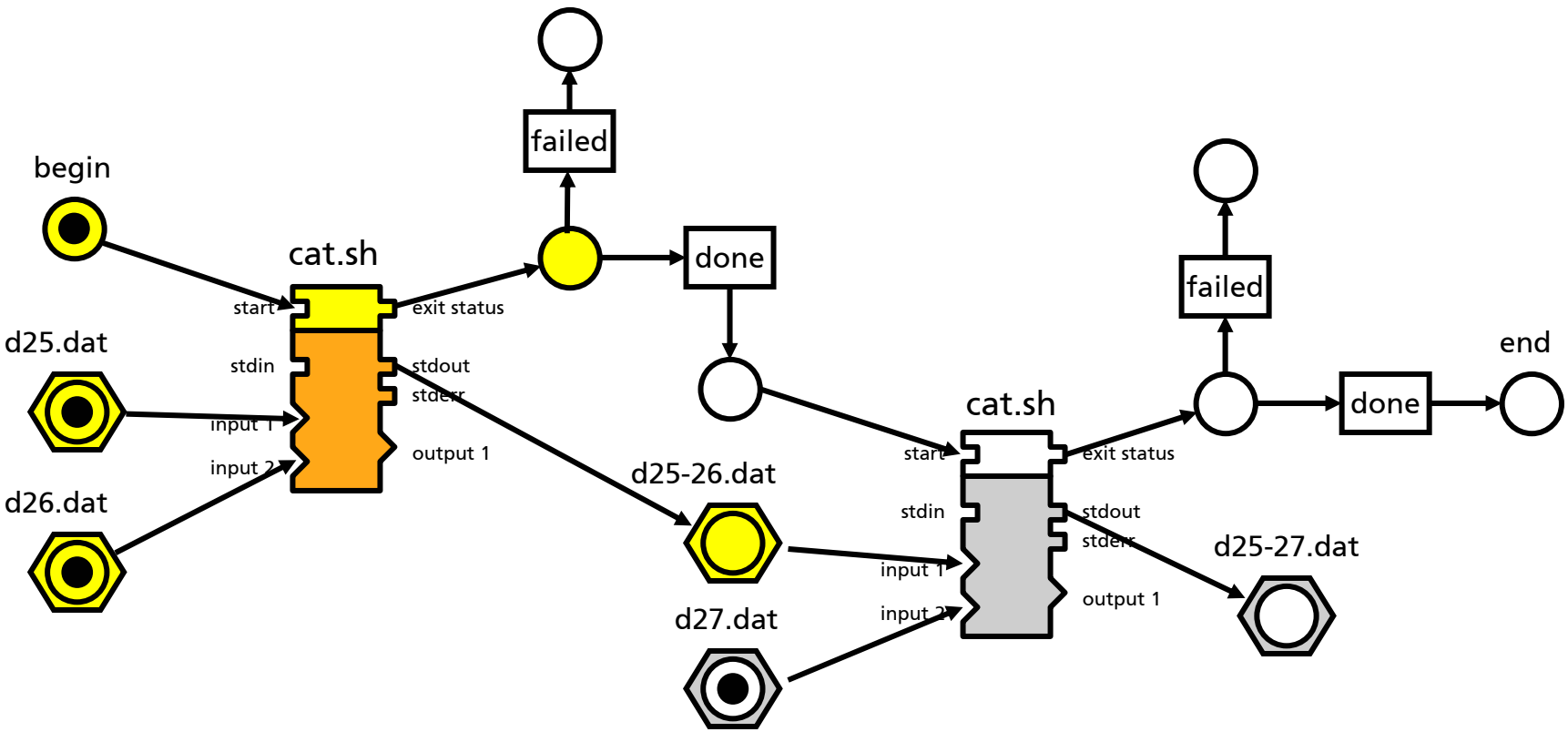
Grid Job Handler



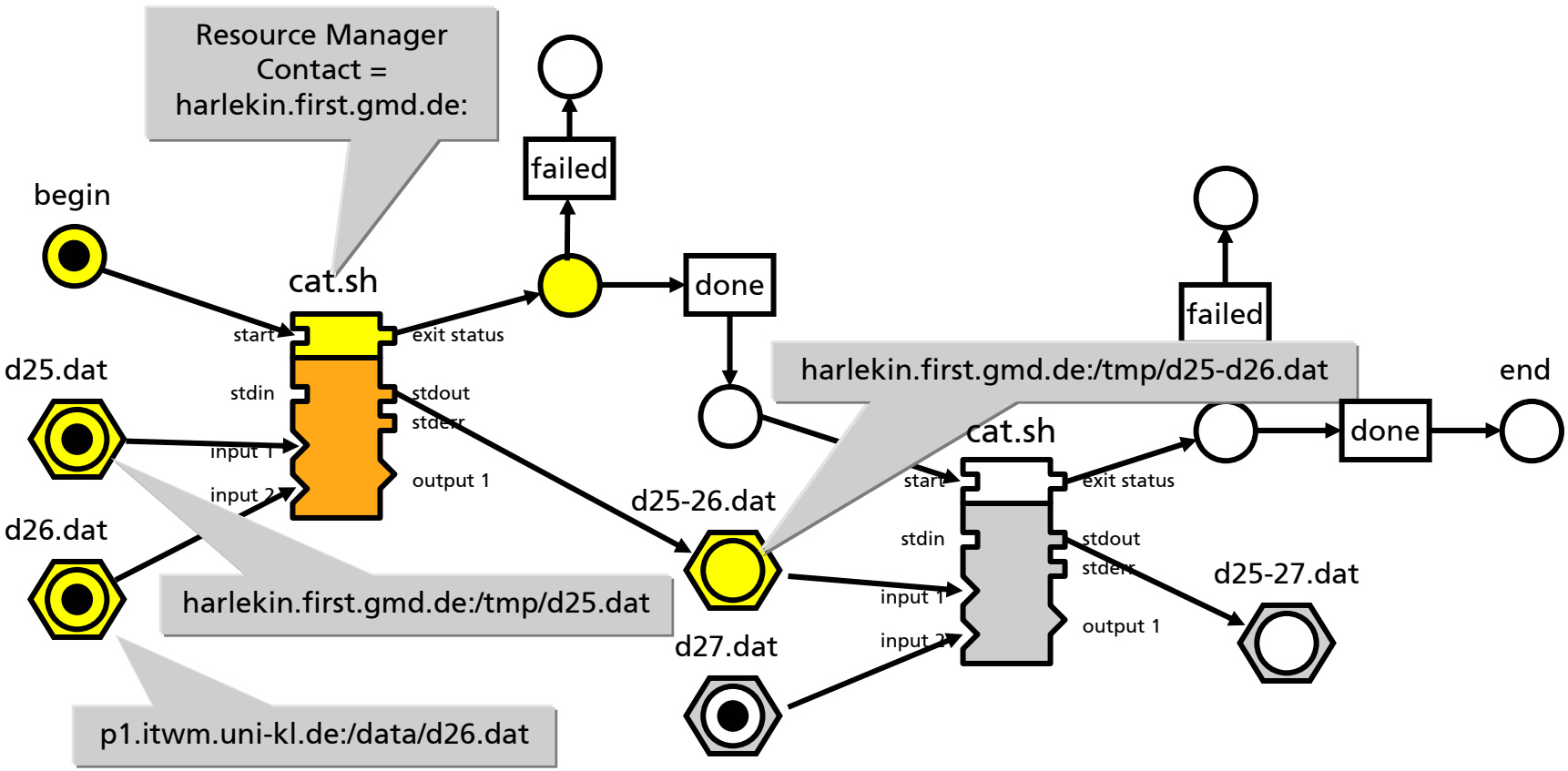
Collect all enabled transitions …



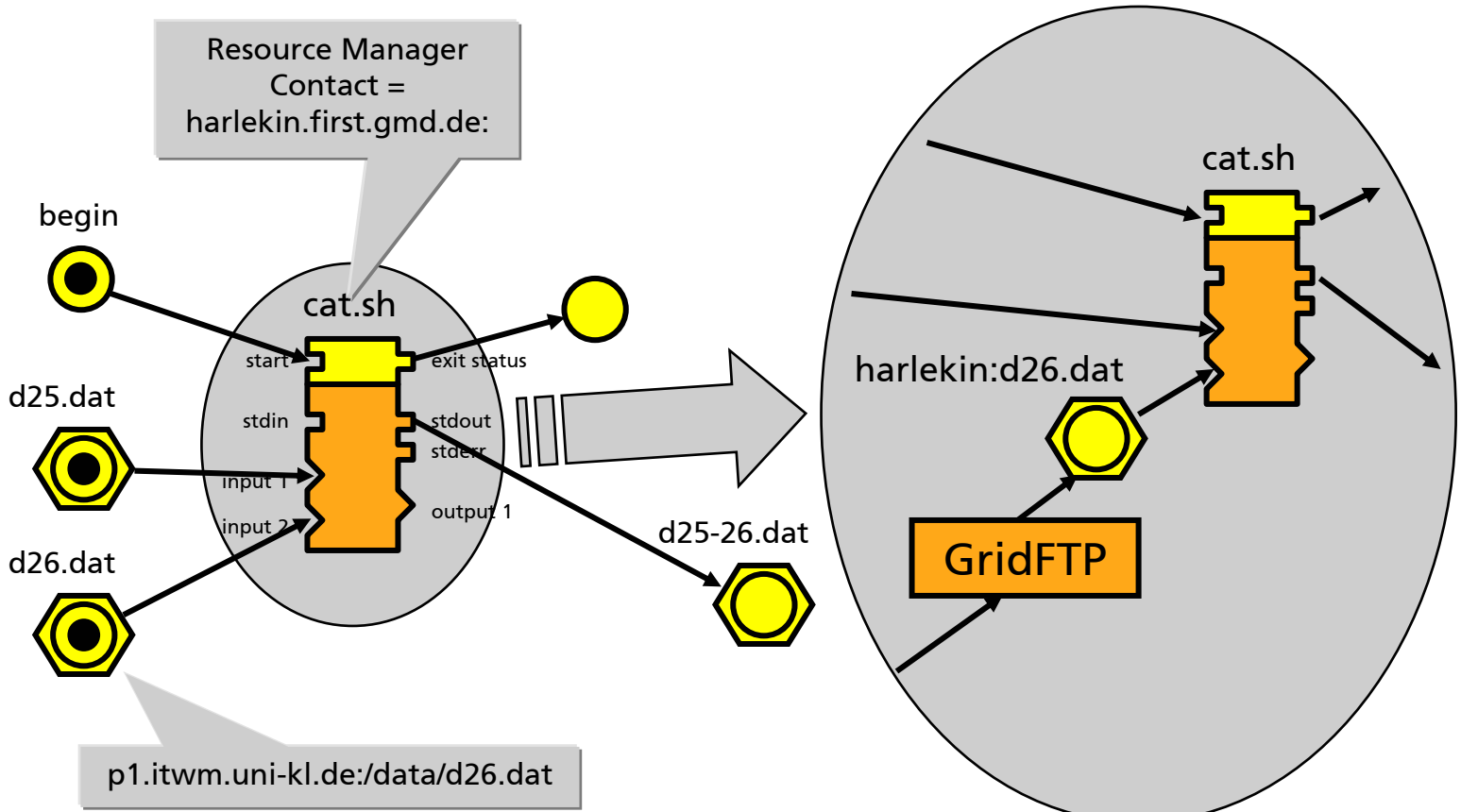
Lock input and output places



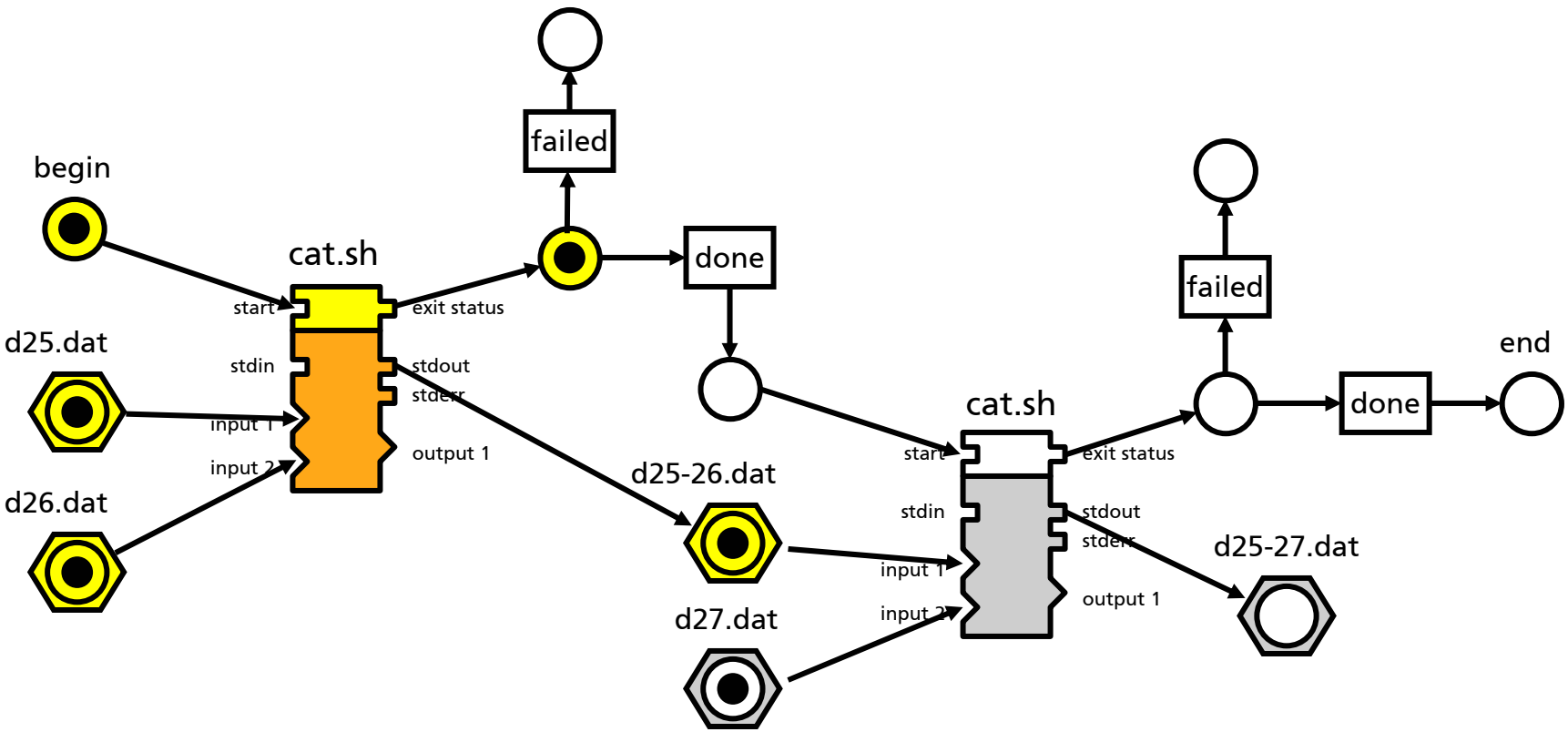
Resource mapping



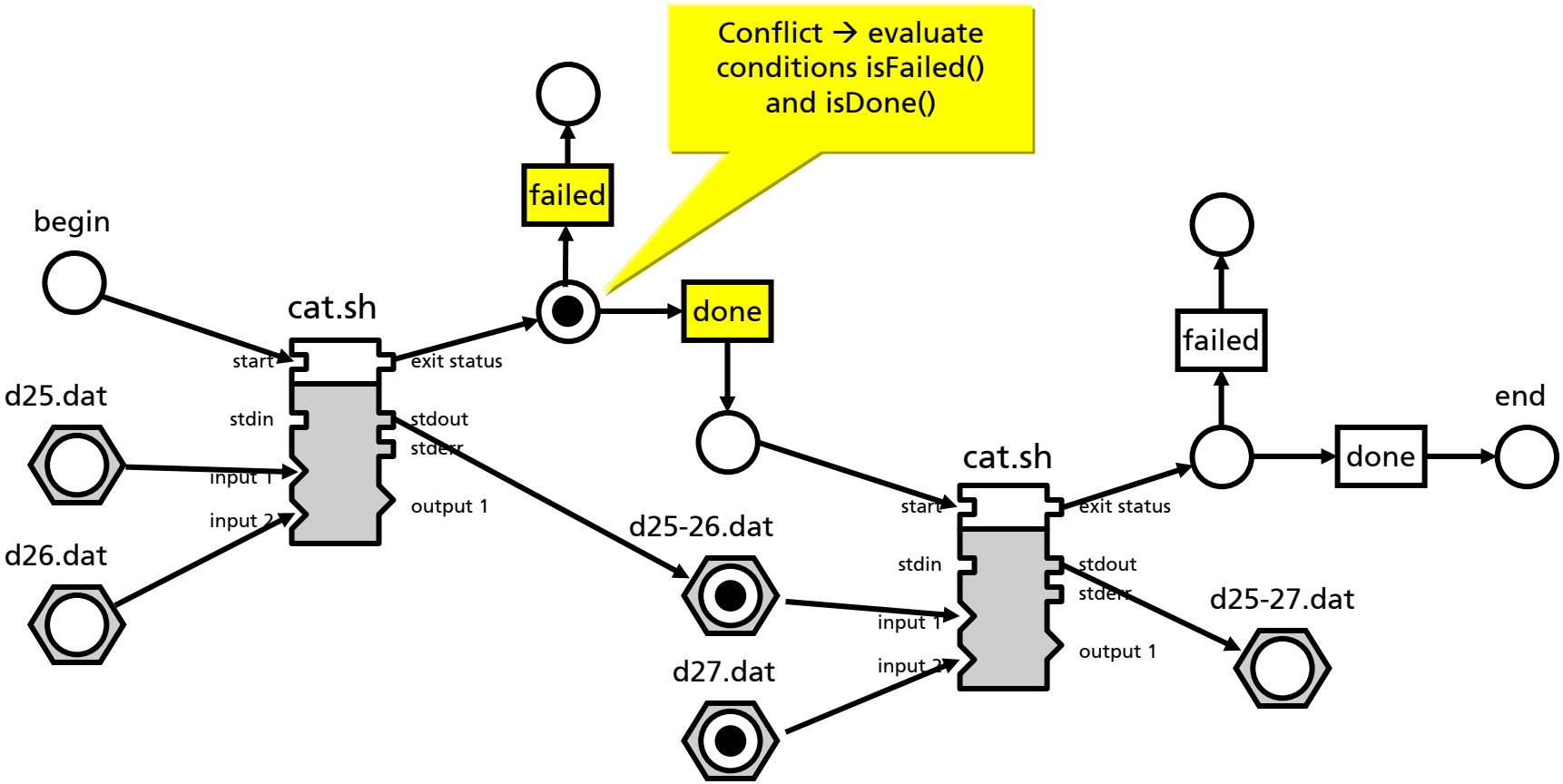
Refine the Petri Net → insert GridFTP



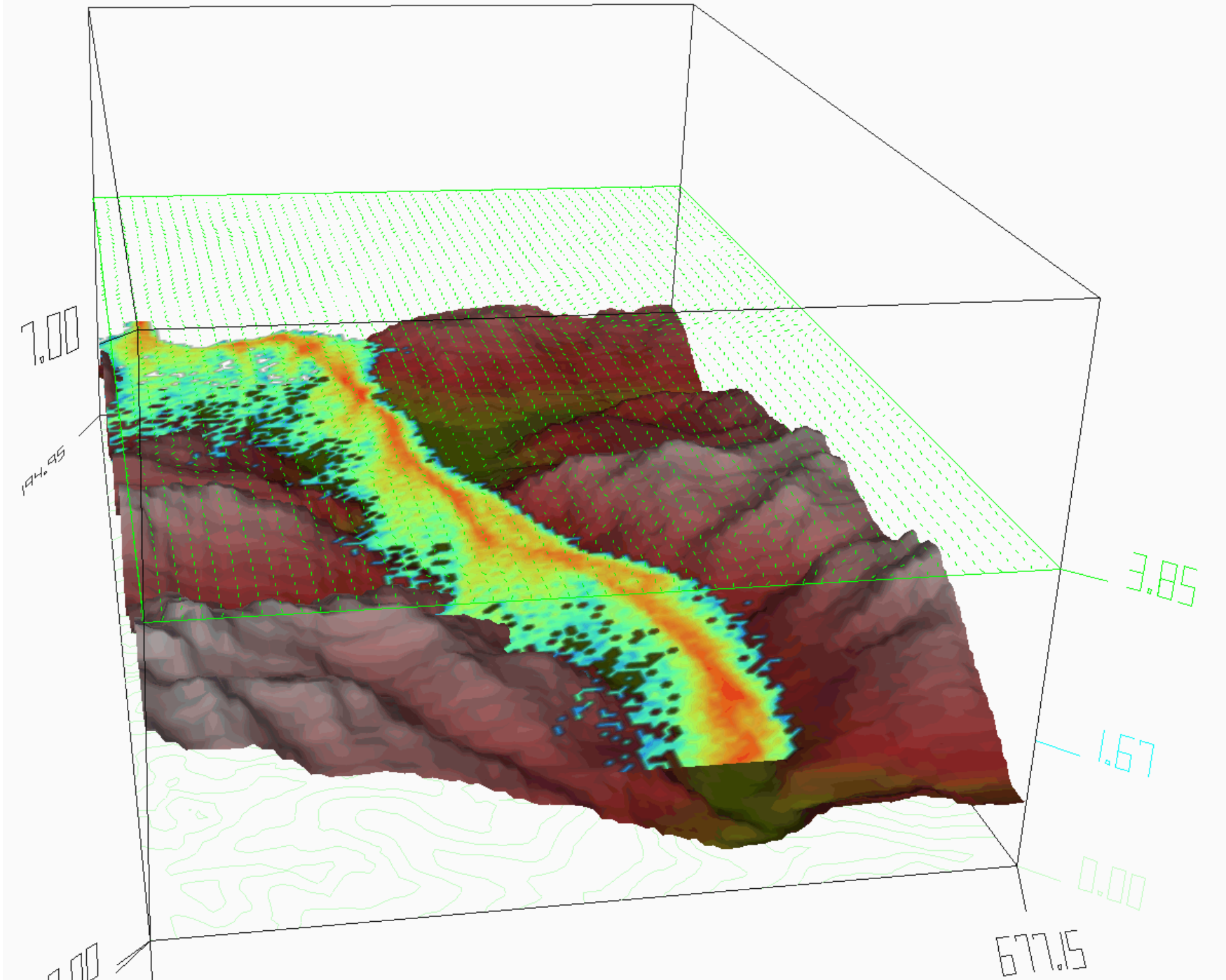
... The transition fires

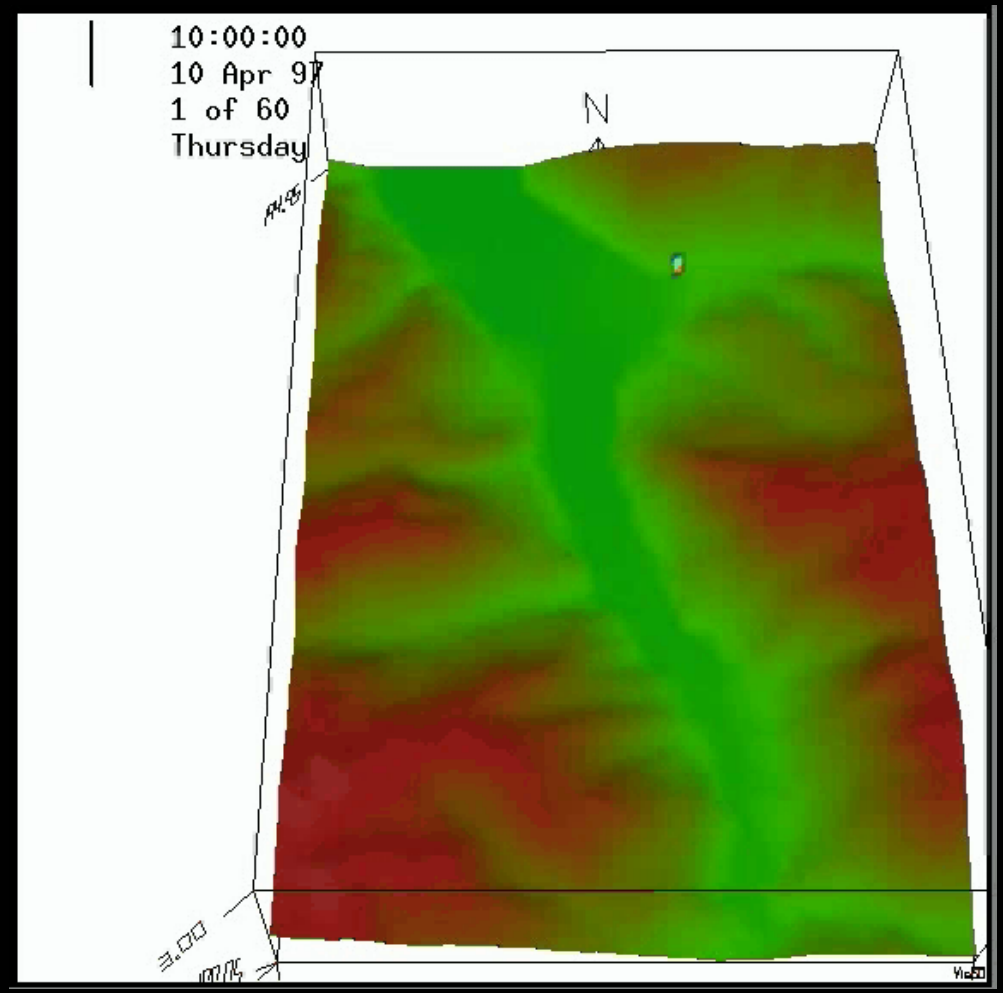


Collect all enabled transitions …



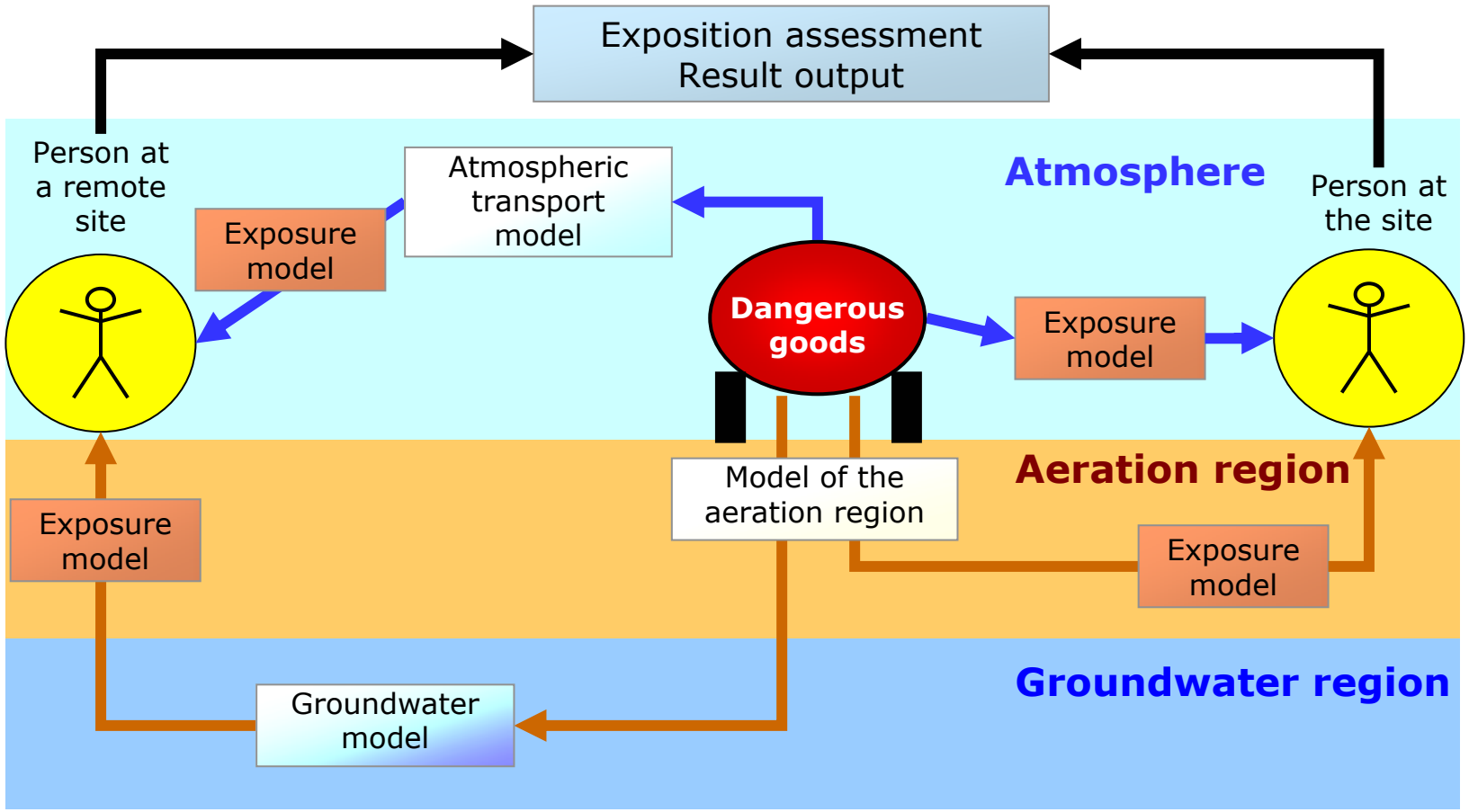
Application: Environmental Risk Analysis and Management System (ERAMAS)


7.00
194.95
3.85
1.67
0.00
677.15


10:00:00
10 Apr 97
1 of 60
Thursday
N

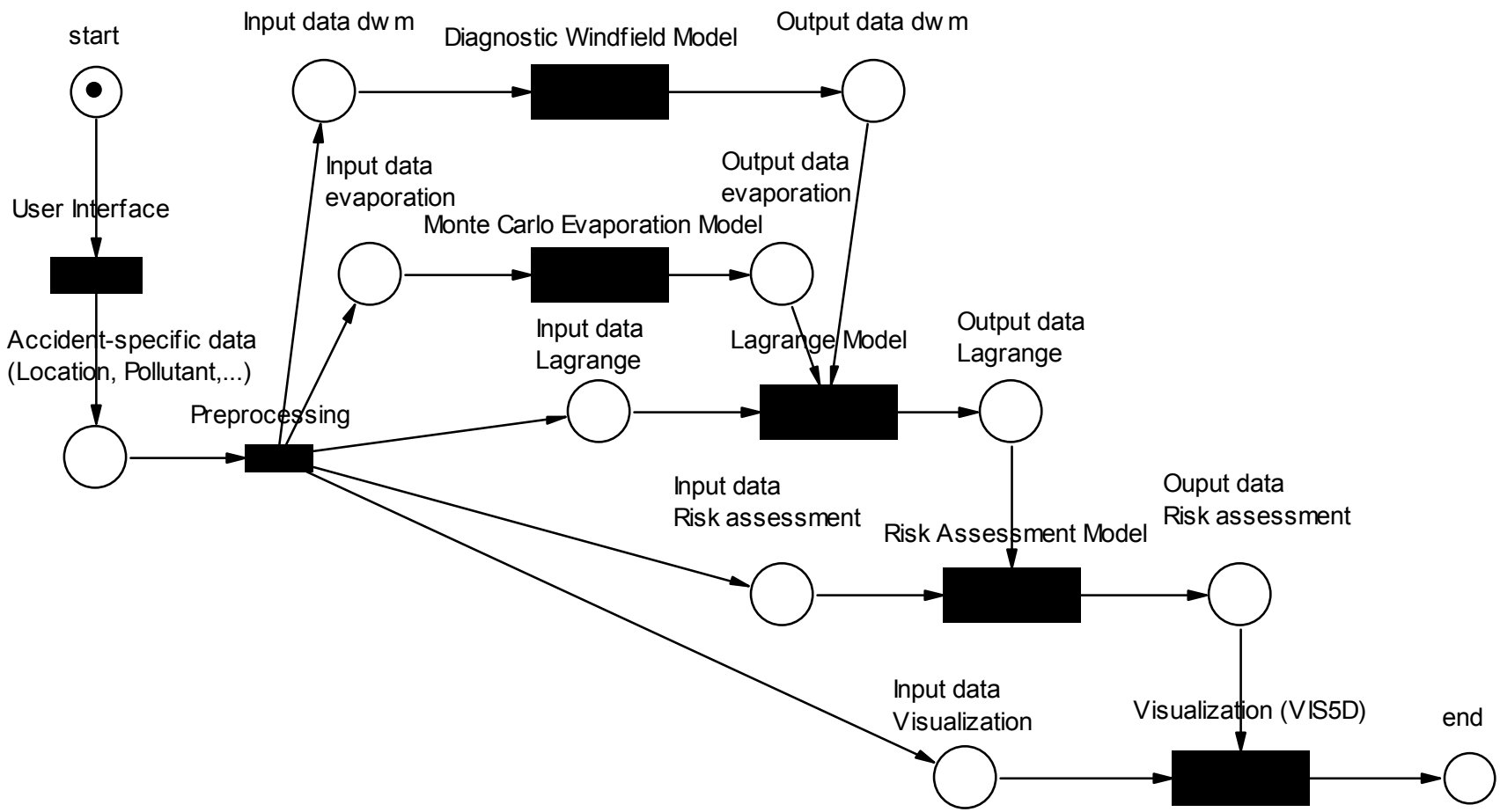
ERAMAS

ERAMAS	Environmental Risk Analysis and Management System
	Simulation-based analysis and management system for environmental risks caused by dangerous substances
Problem	Release of carcinogenic and chemically toxic substances
Scenario	Accidents in industrial installations
	Transport of dangerous goods
	Terrorist attacks



Pollutant Transport in the Atmosphere:

Accident → Source → Atmospheric Transport → Exposure



Fraunhofer Resource Grid

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

Conclusions and Future Work

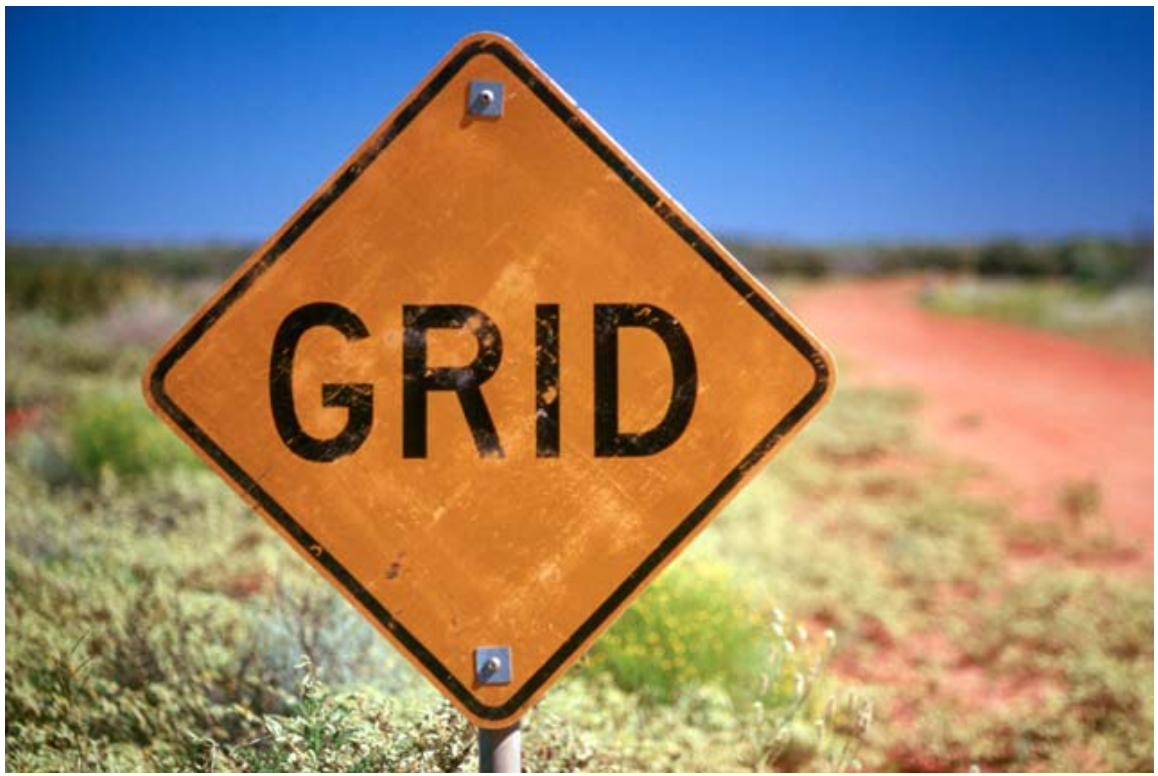
Conclusions

Fraunhofer Resource Grid	Integrates existing resources into a single platform and makes them available for common usage
GResourceDL	Everything is a resource! Design principles: instances vs. classes, extension, inheritance
GJobDL	Petri Nets instead of directed acyclic graphs
Petri Nets	Easy aggregation of complex workflows, including conditions and loops

Future Work

Petri Nets and OGSA?	How does workflow management with Petri Nets adapt to OGSA?
Tight coupling scheme?	Now: one transition → one software component Future: one transition → one method call ➔ Component architecture like CORBA, WebService
Simulation of Petri Nets	Prediction for advanced reservation of resources (→ scheduler) based on software und hardware benchmarks

More Information: http://www.fhrg.fhg.de/
http://www.andreas-hoheisel.de/
mailto:andreas.hoheisel@first.fhg.de



FhRG Fraunhofer Resource Grid

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik