



Der Grid Job Handler des Fraunhofer Resource Grid

Fraunhofer FIRST, Berlin

Version: \$Name: ver20040308\$
\$Id\$

1 Einleitung

Dies ist eine Kurzanleitung des Programmpakets de-fhrg-jobhandler Version 20040308 zur Ausführung von Grid-Jobs auf dem Fraunhofer Resource Grid (FhRG). Neben dem Ausführen von einfachen, ungekoppelten Anwendungen, die mittels der Globus-eigenen Beschreibungssprache XML-RSL definiert wurden, ermöglicht diese Version des Grid Job Handlers auch die Ausführung gekoppelter Grid-Anwendungen, basierend auf der Beschreibungssprache GJobDL Version 1.1. Der Kontroll- und Datenfluss wird hierbei durch ein Petrinetz beschrieben. Der Grid Job Handler verwendet den Globus-Dienst GRAM, um die einzelnen Komponenten der gekoppelten Grid-Anwendung auf gewöhnlichen Globus-Grid-Knoten auszuführen. Mit Hilfe des Dienstes GSI-FTP können vor der Ausführung die benötigten Daten und ausführbaren Dateien auf den Zielrechner übertragen werden. Globus muss hierfür auf dem lokalen Rechner, auf dem der Grid Job Handler gestartet wird, nicht installiert sein.

2 Installation

2.1 Systemvoraussetzungen

Zum Ausführen der Programme ist eine Java Virtual Machine (JVM) Version 1.4 oder höher erforderlich. Für das Ausführen der Unix-Shellskripte in dem Verzeichnis `de-fhrg-jobhandler-20040308/bin/` muss die Umgebungsvariable `FHRG_HOME` auf den absoluten Pfad des Verzeichnisses `de-fhrg-jobhandler-20040308` gesetzt werden. Des Weiteren wird davon ausgegangen, dass unter `/bin/bash` eine zur bash kompatible Shell installiert ist. Zum Kompilieren der Sourcen unter Verwendung der `build.xml`-Datei wird `ant` Version 1.3 oder höher benötigt. Die Programme wurden unter Linux (Debian GNU/Linux, Kernel 2.4.19) sowie unter MS-Windows XP mit Java Version 1.4.1 getestet.

Globus muss auf dem lokalen Rechner nicht installiert sein. Der Zugriff auf Globus erfolgt mit Hilfe des Java CoG-Kits CVS-Version vom 23.7.2003 [5], welches in diesem Programmpaket in einer leicht modifizierten Form mit enthalten ist. Auf den Grid-Rechnern die zur Ausführung der Grid-Anwendungen verwendet werden, muss Globus Version 2.4 sowie GSI-FTP auf den üblichen Ports installiert sein. Falls die Rechner durch Firewalls geschützt sind, müssen die entsprechenden Ports für den eigenen sowie für alle anderen Grid-Rechner freigeschaltet werden. Eine Liste der üblichen Globus-Dienste und den dazugehörigen Port-Bereichen findet sich unter <http://www.globus.org/security/v1.1/firewalls.html>. [6]

Des Weiteren benötigt der Benutzer ein gültiges Benutzerzertifikat mit dem dazugehörigen privaten und öffentlichen Schlüssel, die von einer im Grid akzeptierten Certificate Authority (CA) zertifiziert wurden. Zudem werden die öffentlichen Zertifikate der CAs benötigt, denen der Benutzer vertraut.

2.2 Konfiguration

Zunächst muss das Benutzerzertifikat (X509) für Globus und der dazugehörige private Schlüssel auf dem lokalen Rechner vorhanden sein. Es ist zu beachten, dass der private Schlüssel ausschließlich von seinem Besitzer eingesehen werden darf und die Leserechte entsprechend gesetzt sein müssen. Der private Schlüssel darf nie über unsichere Übertragungswege, wie zum Beispiel FTP oder `rcp`, auf andere Rechner übertragen werden. Zudem wird das Zertifikat der CA (Certificate Authority) benötigt, welche das Benutzerzertifikat signiert hat. Standardmäßig befinden sich die Benutzerzertifikate in dem Verzeichnis

`~/ .globus` (Linux),

`C:\Windows\ .globus` (MS-Windows98) oder

`C:\Dokumente und Einstellungen\\ .globus`

(MS-Windows2000, MS-WindowsXP). Weitere Informationen zu Globus-Zertifikaten siehe <http://www-fp.globus.org/security/v1.1/certs.html>. Mitarbeiter der Fraunhofer Gesellschaft können Zertifikate von der Fraunhofer-CA unter

<http://pki.fraunhofer.de/> beantragen. Zusätzlich wurde eine eigene CA für das FhRG eingerichtet. Anfragen bitte per E-Mail an <ca@fhrg.fhg.de> richten.

Die Konfiguration des CoG-Kits erfolgt durch Klicken auf den Menüeintrag **File - Configure** in der grafischen Benutzeroberfläche. Die Einstellungen werden in der Datei `~/ .globus/cog.properties` gespeichert und sind erst nach einem Neustart des Programms wirksam. Die Einstellungen in der Datei `cog.properties` können auch manuell vorgenommen werden. Dabei sind folgende Einträge möglich:

usercert: Speicherort des Globus Benutzerzertifikats (notwendig)

userkey: Speicherort des zum Benutzerzertifikat passenden privaten Schlüssels. Ausschließlich der Besitzer des privaten Schlüssels darf Leserechte für diese Datei besitzen (notwendig)

proxy: Speicherort des Benutzer-Security-Proxys. Der Name des Proxys wird üblicherweise zusammengesetzt aus der Zeichenkette „x509up_u“ und der Systemabhängigen Benutzerkennung. Ausschließlich der Ersteller des Proxys darf Lese- und Schreibrechte für diese Datei besitzen (notwendig)

cacert: Enthält eine durch Kommas getrennte Liste der CA-Zertifikate denen der Benutzer vertraut (notwendig)

org.globus.dev.random: **true** oder **false**, je nachdem, ob das Java CoG Kit zur Zufallszahlengenerierung nach Unix-Art /dev/urandom verwenden soll (optional)

random.provider: Gibt einen auf dem System installierten Java-Provider für Zufallszahlen an, der standardmäßig verwendet werden soll (optional)

random.algorithm: Gibt den Algorithmus an, der zur Zufallszahlengenerierung durch den Zufallszahlen-Provider verwendet werden soll (optional)

proxy.strength: Standardmäßige Größe des Grid-Proxys in Bits (optional)

proxy.lifetime: Standardmäßige Gültigkeitsdauer des Grid-Proxys in Stunden (optional)

ip: Die IP-Nummer, unter der der Rechner von außen erreichbar ist. Wird zum Beispiel benötigt, falls der Rechner zusätzlich in einem lokalen Subnetz hängt und daher mehrere IP-Nummern besitzt (optional)

http.proxyHost: Die IP-Adresse des HTTP-Proxys über den die HTTP-Kommunikation nach außen erfolgen soll, falls der Rechner zum Beispiel durch eine Firewall geschützt ist (optional)

http.proxyPort: Der Port des oben genannten HTTP-Proxys (optional)

ftp.proxyHost: Die IP-Adresse des FTP-Proxys über den die ftp-Kommunikation nach außen erfolgen soll, falls der Rechner zum Beispiel durch eine Firewall geschützt ist (optional)

ftp.proxyPort: Der Port des oben genannten FTP-Proxys (optional)

socksProxyHost: Die IP-Adresse des Socket-Proxys über den die Socket-Kommunikation nach außen erfolgen soll, falls der Rechner zum Beispiel durch eine Firewall geschützt ist (optional)

socksProxyPort: Der Port des oben genannten Socket-Proxys (optional)

tcp.port.range: Der freie Port-Bereich in der Form **<minport>**, **<maxport>** auf den die lokalen Netzdienste eingeschränkt werden sollen (optional)

usermapping: Falls der Grid Job Handler als Web-Service betrieben wird, findet im Allgemeinen eine Abbildung von Benutzerkennungen auf Pseudonymzertifikate statt. Mit dem Parameter usermapping kann der Dateiname angegeben werden, in dem die Abbildung definiert wird. Die usermapping-Datei enthält pro Zeile einen Eintrag in der Form **<Benutzer-ID>=<certDir>**, wobei **<certDir>** das Verzeichnis angibt, in dem das jeweilige Zertifikat (usercert.pem) und der unverschlüsselte private Schlüssel (userkey.pem) abgelegt sind (optional).

anonymous.certdir: Das Zertifikatsverzeichnis für Benutzerkennungen, die in der usermapping-Datei nicht aufgeführt sind.

Beispiel von
Einstellungen in der
Datei cog.properties
unter Linux oder
Unix

```
#Java CoG Kit Configuration File
#Tue Feb 25 22:30:30 CST 2003
usercert=/home/albert/.globus/usercert.pem
userkey=/home/albert/.globus/userkey.pem
proxy=/tmp/x509up_u1000
cacert=/usr/local/globus/share/certificates/42864e48.0
ip=140.221.56.12
tcp.port.range=6000,6060
```

Beispiel von
Einstellungen in der
Datei cog.properties
unter MS-Windows

```
#Java CoG Kit Configuration File
#Tue Jul 15 10:00:33 CEST 2003
usercert=C:\\Dokumente und Einstellungen\\albert\\.globus\\usercert.pem
userkey=C:\\Dokumente und Einstellungen\\albert\\.globus\\userkey.pem
proxy=C:\\Dokumente und Einstellungen\\albert\\.globus\\x509up_u_albert
cacert=C:\\Dokumente und Einstellungen\\albert\\.globus\\42864e48.0
ip=140.221.56.12
tcp.port.range=6000,6060
```

Damit eine Grid-Anwendung ausgeführt werden kann, muss zunächst ein neuer Security Proxy erzeugt werden. Dies geschieht mittels des Menüeintrags **Security - Grid Proxy - Create**.

2.3 Inhalt der Verzeichnisse

/bin: Skripte zum Starten der Programme unter Linux / MS-Windows

/doc: Dokumentation zum Grid Job Handler

/examples: Beispiele für die RSL-XML sowie GJobDL-Beschreibung von Grid-Anwendungen. Zur Ausführung dieser Grid-Anwendungen werden Globus-Zugriffsrechte auf den entsprechenden Grid-Knoten benötigt

/jar: Die verwendeten Java-Klassen und anderen Dateien in Form von jar-Paketen

/javadoc: Die HTML-Dokumentation zu den Java-Paketen de.fhrg.** und zum Java CoG-Kit (Pakete org.globus.**)

/log: Protokolldateien

/resources: Dateien, die unabhängig vom Sourcecode zur Ausführung der Programme benötigt werden. Diese Dateien sind zusätzlich in den jar-Paketen enthalten.

/sources: Die Sourcen der Java-Pakete de.fhrg.** sowie des verwendeten und leicht modifizierten Java CoG-Kits. Eine Beschreibung der einzelnen Java-Pakete ist in der HTML-Dokumentation unter /javadoc enthalten.

3 Eigenschaften

3.1 Bedienung

Der Grid Job Handler kann in mehreren verschiedenen Modi gestartet werden: Während mit der Klasse **de.fhrg.jobhandler.JobHandler** der Grid Job Handler über Kommandozeilenparameter bedient wird, stellt die Klasse **de.fhrg.jobhandler.desktop.JobHandlerDesktop** eine grafische Benutzeroberfläche zur Verfügung. Zudem kann der Grid Job Handler als Web Service bereitgestellt werden.

3.1.1 Grafische Benutzeroberfläche

Zum Starten der grafischen Benutzeroberfläche können die Skripte **de-fhrg-jobhandler-desktop.sh** (Linux) bzw. **de-fhrg-jobhandler-desktop.bat** (MS-Windows) im Verzeichnis `de-fhrg-jobhandler-20040308/bin` verwendet werden. Wahlweise kann auch direkt in der Kommandozeile im Verzeichnis `de-fhrg-jobhandler-20040308/jar` der Befehl

```
java -cp de-fhrg-jobhandler-20040308.jar
de.fhrg.jobhandler.desktop.JobHandlerDesktop
```

zum Starten der grafischen Benutzeroberfläche eingegeben werden.

3.1.2 Kommandozeilentool

Als Kommandozeilentool kann der Grid Job Handler unter Linux mittels des Skriptes **de-fhrg-jobhandler.sh** oder wahlweise auch direkt durch

```
java -cp de-fhrg-jobhandler-20040308.jar
de.fhrg.jobhandler.JobHandler
```

gestartet werden. Dabei gelten folgende Kommandozeilenparameter:

```
de.fhrg.jobhandler.JobHandler
  -filename <XML_FILENAME>
  [-jobType <JOB_TYPE>]
  [-v <VERBOSE_LEVEL>]
  [-home <FHRG_HOME>]
  [-schedulerType <SCHEDULER_TYPE>]
```

<XML_FILENAME>: Dateiname der XML-Datei, welche die Beschreibung der Grid-Anwendungen enthält.

<JOB_TYPE>: Entweder "GJobDL" oder "RSL". Standardwert ist "GJobDL" (optional)

<VERBOSE_LEVEL>: Menge der Protokollausgabe (optional): 0: nur Fehlermeldungen bis 5: alle Debug-Meldungen

<FHRG_HOME>: Das Verzeichnis, in dem der Grid Job Handler installiert ist (zum Beispiel `/home/albert/de-fhrg-jobhandler-20040308`). FHRG_HOME kann auch als Umgebungsvariable gesetzt werden.

<SCHEDULER_TYPE>: Art des zu verwendenden Schedulers (optional). Entweder „test“ oder „MDS“. Der Test-Scheduler wählt immer die Hardwareressource aus einer Liste möglicher Ressourcen aus, die am längsten nicht ausgewählt wurde. Der MDS-Scheduler hingegen prüft den LDAP-Server des FhRG und wählt den zurzeit schnellsten Rechner unter Berücksichtigung der aktuellen Auslastung aus. Der MDS-Scheduler befindet sich noch in der Testphase und funktioniert noch nicht einwandfrei.

3.1.3 Grid Job Handler Web Service

Auf den Grid Job Handler kann auch per Web Service zugegriffen werden. Im Paket `de.fhrg.jobhandler.webservice` finden sich die dazu notwendigen Klassen. Eine Beispielanleitung für die Bereitstellung des Grid Job Handlers per axis und tomcat findet sich unter `doc/jobHandlerWebServiceDeployment.txt`. Die Zuordnung der Benutzer auf Pseudonymzertifikaten erfolgt zurzeit auf Basis einer Benutzerzuordnungsdatei. Näheres hierzu siehe Kapitel 2.2.

Auf den Grid Job Handler Web Service kann auf verschiedene Arten zugegriffen werden. Ein Beispiel hierfür ist der Kommandozeilen-Client im Paket `de.fhrg.jobhandler.webservice.client`. Um diesen Kommandozeilen-Client zu starten, kann das Skript `de-fhrg-jobhandler-webservice-client.sh` verwendet werden. Die Kommandozeilenparameter sind analog zu denen in Kapitel 3.1.2 beschrieben. Auch die Applets im Paket `de.fhrg.jobhandler.applet` greifen auf den Grid Job Handler Web Service zu.

3.2 Beschreibung von Grid-Anwendungen mit der GJobDL

Das Konzept und die Spezifikation der GJobDL werden ausführlich in [1], [2] und [4] beschrieben. Hier soll nur auf die darüber hinausgehenden implementierungsspezifischen Eigenschaften eingegangen werden. Einige hilfreiche Beispiele für GJobDL-Beschreibungen von Grid-Anwendungen können dem Verzeichnis `<FHRG_HOME>/examples` entnommen werden. Diese Beschreibungen sind zurzeit noch etwas umfangreich, da die aktuelle Version des Grid Job Handlers nur für Teilgebiete automatische Verfeinerung der Petrinetze unterstützt. Die GJobDL-Dokumente können entweder per Hand mit einem geeigneten XML-Editor oder mit Hilfe des **Grid Job Builder** erstellt werden, der zurzeit vom Fraunhofer IGD entwickelt wird.

3.2.1 Methodenaufrufe in Transitionen

Transitionen in GJobDL-Dokumenten können Methodenaufrufe oder Bedingungen enthalten. Hier ein Beispiel für die Definition eines Methodenaufrufs:

```
<transition id="singleTransfer">  
  <methodCall>transferData () </methodCall>  
</transition>
```

Der Grid Job Handler unterstützt in der aktuellen Version folgende Methodenaufrufe (siehe auch das Interface `de.fhrg.jobhandler.MethodCalls`):

transferExecutable(String softwareTransitionId): Überträgt die ausführbare Datei oder das Archiv einer Softwarekomponente zum Ort der Ausführung. Falls der Ort der Ausführung noch nicht feststeht, wird dieser durch Resource-Mapping und Scheduling festgelegt.

unpack (): Entpacken aller tar.gz-Dateien, die mit den Eingabestellen dieser Transition verknüpft sind.

transferData(String softwareTransitionId): Überträgt die Dateien, die mit den Eingabestellen dieser Transition verknüpft sind, zum Ort der Ausführung der Softwarekomponente der genannten Software-Transition. Falls der Ort der Ausführung noch nicht feststeht, wird dieser durch Resource-Mapping und Scheduling festgelegt.

transferData (): Überträgt die Datei, die mit der Eingabestelle dieser Transition verknüpft ist, zu allen Orten, die über die Ausgabestellen definiert sind.

clear (): Räumt auf den benutzten Rechenknoten auf, indem alle bisher im Rahmen des Grid-Jobs angelegten temporären Verzeichnisse gelöscht werden.

3.2.2 Bedingte Transitionen

Es können auch bedingte Transitionen definiert werden, z.B.:

```
<transition id="count.lt.7">  
  <condition>countLT (7) </condition>  
</transition>
```

Der Grid Job Handler unterstützt zurzeit folgende Bedingungen (siehe auch das Interface `de.fhrg.jobhandler.Conditions`):

isDone (): Diese Bedingung gilt als erfüllt, wenn ob alle Eingabestellen einen Boolean mit dem Wert "true" enthalten.

isFailed (): Diese Bedingung gilt als erfüllt, wenn mindestens eine Eingabestelle einen Boolean mit dem Wert "false" enthält.

countLT(int maxCount): Zählt die Anzahl, wie oft eine Transition bisher geschaltet hat. Falls die Anzahl kleiner als maxCount ist, gilt die Bedingung als erfüllt und die Transition schaltet.

negate(String transitionID): Hiermit kann die Bedingung einer anderen Transition negiert werden (entspricht ELSE bei IF THEN), z.B. negate(count.lt.7) schaltet nur dann, wenn die Bedingung der Transition "count.lt.7" nicht erfüllt ist.

timeGE(long msSince1970): Vergleicht die aktuelle Systemzeit mit dem gegebenen Zeitstempel. Die Transition schaltet nur, falls die aktuelle Systemzeit größer oder gleich der angegebenen Zeit ist. Das Zeitformat entspricht dem Java-Zeitformat, das in der Date-Klasse verwendet wird (Millisekunden seit 1.1.1970 00:00 UTC).

idleTimeGE(long milliseconds): Vergleicht die Zeitdauer nach dem letzten Schalten der Transition mit der angegebenen Zeitdauer. Die Transition schaltet nur, falls nach dem letzten Schalten die angegebene Zeit in Millisekunden verstrichen ist.

3.3 Tastenbelegung

<Strg>+<q>	Beenden des Programms
<Strg>+<t>	Verwende Scheduler „Test“
<Strg>+<m>	Verwende Scheduler „MDS“
<Strg>+<l>	GJobDL-Dokument aus lokaler Datei laden
<Strg>+<r>	GJobDL-Dokument aus Repository laden
<Strg>+<s>	aktuelles GJobDL-Dokument in lokaler Datei speichern

3.4 Änderungen

Version 20040308 – Datum: 08.03.2004

Änderungen gegenüber Version 20031127

- Java VM 1.4.0 oder höher notwendig, da das JavaCoG dieses verlangt.
- Neuer Methodenaufruf „clear()“ löscht alle bisher angelegten temporären Verzeichnisse eines Grid-Jobs.
- Maus-Cursor als Sanduhr, falls Desktop-Handler beschäftigt
- Automatischer Zeilenumbruch bei der Darstellung von Fehlermeldungen
- Anbindung an PBS und andere Globus Jobmanager. Hierzu kann nun in der Software-Ressourcenbeschreibung mittels <parameter name="jobType"> der Typ angegeben werden (single, multiple, mpi, condor)
- Angabe der geografischen Koordinaten von Hardware Ressourcen im Event Channel nach ISO6709 zur Darstellung auf Karten.
- Neue Version des Schedulers. Zur Auswahl stehen jetzt ein Standard- (ehemals „Test“), ein interner MDS- sowie ein externer Client/Server MDS-Scheduler.
- Automatische Verfeinerung des Petrinetzes für Installation von Software (Software Deployment)
- Petrinetze werden nun reproduzierbar angeordnet

- Neues Farbschema für Darstellung der Petrinetze

Version 20031127 – Datum: 27.11.2003

Änderungen gegenüber Version 20030708

- Dokumentation jetzt in /doc.
- Grid Job Handler Client greift per Web Service auf Grid Job Handler zu.
- Grid Job Handler jetzt auch als Web Service.
- Grid Job Handler unterstützt Pseudonymzertifikate und erzeugt entsprechend X509-Proxies mit beschränkter Lebensdauer.
- Neue Axis-Bibliotheken (Version 1.1).
- Zusätzliche Events für dynamische Verfeinerung des Petrinetzes: addTransition, addPlace, addArc, removeArc
- Update CoG1.1a auf CVS-Version vom 23.7.2003
- Neue Bedingungen für Transitionen: timeGE und idleTimeGE
- Einbindung von Touchgraph zur Darstellung der Petrinetze. Dadurch entfällt die Anordnung der Graphen durch ein externes C-Programm.
- Fortschrittsanzeige beim Laden von Ressourcenbeschreibungen und Erstellung von Grid Jobs.
- Neue Schnittstelle zum Scheduler.
- Neu: Security → Grid Proxy → Information: Gibt den Namen und die verbleibende Gültigkeitsdauer des aktuell verwendeten Security-Proxys aus.
- Status von Grid-Jobs: undefined, defining, pending, active, suspended, failed, done. Der Status wird über den Java-Beans propertyChange-Mechanismus weitergeleitet.

Version 20030708 – Datum: 08.07.2003

Änderungen gegenüber Version 20030424:

- Update CoG-0.9.13 → CoG1.1a mit neuer Verschlüsselungsbibliothek (Bouncycastle) mit besseren Lizenzbestimmungen.
- Funktioniert jetzt auch mit Globus 2.4.
- Verwendet GADL1.1: Schema statt DTD, Ressourcen-Identifizier entspricht der URL, unter der die Ressourcenbeschreibung verfügbar ist.
- Der Grid Job Handler greift über das Internet auf die verteilten Ressourcenbeschreibungen zu, falls sie nicht im lokalen GJobDL-Dokument definiert werden.
- Standardverzeichnis ist jetzt FHRG_HOME+ "/examples".
- Shellskripte verwenden jetzt bash statt tcsh.
- Alphaversion einer Deutschlandkarte mit geographischer Repräsentation des Grid-Jobs.
- Die Konfiguration der lokalen IP-Nummer und des Internetproxy über die grafische Benutzerschnittstelle ist vorübergehend nicht verfügbar und wird in einer späteren Version wieder eingebaut.
- Viele kleinere Bugfixes.

Version 20030424 - Datum: 24.04.2003

Änderungen gegenüber Version 20021113:

- Automatische Verfeinerung von Petrinetzen für Datentransfer.
- Erste Version des MDS-Schedulers vom ITWM integriert.
- Im "Configuration Wizard" kann jetzt die lokale IP-Nummer angegeben werden, die zur Kommunikation nach außen verwendet werden soll.
- Nötig für Rechner mit zwei IP-Nummern (internes/externes Netz). Erste Anbindung an das Repository mit SOAP.
- Bugfix wegen Probleme mit <executionLocation> in der GJobDL.
- GEM zur automatischen Anordnung der Petrinetze jetzt auch für MS-Windows.
- Der JobHandlerDesktop kann nun auch zum Verändern und Abspeichern der GJobDL-Beschreibungen verwendet werden.
- PetriNetFactory und GridResourceFactory zur Erstellung von Instanzen zur Repräsentation von Petrinetzen und Grid-Ressourcen.

Version 20021113 - Datum: 13.11.2002

Änderungen gegenüber Version 20021002:

- Neuer Kommandozeilenparameter: -home
- Hinzufügen von MethodCalls und Conditions. erlaubte MethodCalls: transferExecutable(String), transferData(String), transferData(), unpack() erlaubte Conditions: isDone(), isFailed(), countLT(int), negate(String).
- Grafische Darstellung des Prozessablaufs in einem Petrinetz (Monitoring).
- Transitionen und Stellen haben jetzt einen Status (IDLE, ENABLED, FIRING, ...).
- Beim Submitten von Softwarekomponenten wird jetzt das Verzeichnis gesetzt: RSL: (directory = \$(HOME) # "/...")
- Verbesserte Benutzerführung im JobHandler-Desktop
- Einige neue Beispiele von GJobDL-Beschreibungen hinzugefügt.
- Bugfix: Transition feuert jetzt erst, wenn ALLE dazugehörigen atomicJobs fertig (=DONE oder FAILED) sind.
- Bugfix: Temporäre Orte von Dateien werden jetzt auch als solche behandelt.
- X/Y-Koordinaten von Stellen und Transitionen werden nun berücksichtigt.
- Globus-Proxy wird jetzt jedem AtomicJob getrennt mitgegeben.

Version 20021002 - Datum: 02.10.2002

Änderungen gegenüber Version 20020214:

- GJobDLExecutor umbenannt in JobHandler
- Paketstruktur geändert (de.fhrg.first.** -> de.fhrg.jobhandler.*, de.fhrg.util.*, ...)
- Kapselung von GRAM-Jobs durch die abstrakte Klasse AtomicJob.
- AtomicGramJob ist jetzt eine Erweiterung der Klasse AtomicJob.
- Neue Art von AtomicJob: AtomicGridFtpJob. Verwendet das
- Paket de.fhrg.gridftp zum Übertragen von Dateien mittels GSI-FTP.
- Neue Kommandozeilenparameter: -jobType, -home, -v, -filename
- GADL 0.9 wird in den Grundzügen unterstützt
- Einlesen des GJobDL-Dokuments per validierenden DOM-Parser
- Petrinetze mit Transitions, Arcs und Places.
- Resource Mapping: Löst Abhängigkeiten zwischen Grid-Ressourcen auf.

- Scheduler: Einfaches Scheduling ohne Verwendung zusätzlicher Informationen über Grid-Ressourcen.
- Update cog-0.9.12 -> cog-0.9.13
- neue Version von iaik_*.jar

Version 20020214 - Datum: 14.02.2002

Änderungen gegenüber Version 20020129:

- Monitoring von Grid-Jobs
- Umstellung des XML-Parsers auf Xerces 2.0.0
- Unterstützung von Proxies beim Zugriff auf das Internet (wird insbesondere bei Rechnern hinter einer Firewall benötigt). Konfiguration der Proxies mit dem Konfigurations-Assistenten. Starten des Assistenten durch Klicken auf den Menüeintrag File->Configure in der grafischen Benutzeroberfläche
- Verbessertes Fehlermanagement
- Diese Datei (LIESMICH.txt) jetzt auch auf Englisch (README.txt)

Version 20020129 - Datum: 29.01.2002

Erste Version zum Ausführen von Grid-Jobs, die mit der XML-RSL-Beschreibung definiert wurden. Verwendet CoG-Kit Version 0.9.12. Diese Version beinhaltet keine Funktionalitäten, die über die normale Globus-Implementierung hinausgehen. Wesentliche Funktionen dieser Version:

- Konfiguration der Umgebung (Schlüssel, Zertifikate, ...) mit Hilfe eines Konfigurations-Assistenten
- Erstellen und Prüfen eines Grid Proxies
- Laden von Grid-Job-Beschreibungen im Format XML-RSL
- Ausführen von Grid-Jobs mithilfe des GRAM-Dienstes von Globus
- Anzeigen dieser Hilfedatei

3.5 Bugs, Vorschläge, Beiträge

Bitte per E-Mail an Andreas Hoheisel <andreas.hoheisel@first.fraunhofer.de>

Bekannte Bugs:

- Exit-Status der Anwendungen wird von GRAM nicht korrekt an den JobHandler weitergegeben. Der JobHandler erhält den Status "done" auch wenn der Exit-Status des ausgeführten Programms ungleich 0 ist. Dies ist eigentlich kein Bug des JobHandlers, sondern ein Bug/Feature von Globus.
- Nicht alle der durch die GJobDL gegebenen Möglichkeiten sind im Grid Job Handler implementiert, z.B. der Vergleich mit Hilfe von Operatoren und die Authentifizierung.
- Zum Teil ist die URL des Grid Job Handler Web Service noch im Code festgeschrieben. Sollte eigentlich über das Konfigurationstool vorgegeben werden.

4 Ausblick

Folgende Erweiterungen sind für die nächsten Versionen des Programmpakets de-fhrg-jobhandler geplant:

- Unterstützung zusätzlicher "Conditions" und "MethodCalls" von Transitionen in GJobDL-Anwendungen.
- Zusätzliche Autorisierung per Access-Control-Komponente (SIT).
- Vollständige Anbindung an das FhRG-Repository (IAO).

5 Weitere Informationen

5.1 Internetseiten

- Fraunhofer Resource Grid: <http://www.fhrg.fraunhofer.de>
- Fraunhofer FIRST: <http://www.first.fraunhofer.de>
- Fraunhofer Gesellschaft: <http://www.fraunhofer.de>
- Fraunhofer CA: <http://pki.fraunhofer.de>
- OpenSource eXeGrid: <http://www.exegrid.net>
- Java CoG Kit: <http://www.globus.org/cog/java>
- RSL: http://www-fp.globus.org/gram/rsl_spec1.html
- Touchgraph: <http://www.touchgraph.com>
- Andreas Hoheisel: <http://www.andreas-hoheisel.de>

5.2 Literatur

- [1] Hoheisel, Andreas, Uwe Der. An XML-Based Framework for Loosely Coupled Applications on Grid Environments. In P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 245-254, 2003. <http://www.andreas-hoheisel.de>
- [2] Hoheisel, Andreas, Uwe Der. Dynamic Workflows for Grid Applications. In Proceedings of the Cracow Grid Workshop '03, 2003. <http://www.andreas-hoheisel.de>
- [3] Hoheisel, Andreas. Ein Komponentenmodell für Softwarekomponenten des Fraunhofer Resource Grid, Interner Report, Fraunhofer FIRST, 2002. <http://www.andreas-hoheisel.de>
- [4] Hoheisel, Andreas. Grid Application Definition Language - GADL 0.2, Interner Report, Fraunhofer FIRST, 2002. <http://www.andreas-hoheisel.de>
- [5] von Laszewski, Gregor, Ian Foster, Jarek Gawor, Peter Lane. A Java Commodity Grid Kit, Concurrency and Computation: Practice and Experience, pages 643-662, Volume 13, Issue 8-9, 2001. <http://www.globus.org/cog/java/>
- [6] von Laszewski, Gregor, et al. The Java CoG Kit User Manual, Draft Version 1.1a, MCS Technical Memorandum, ANL/MCS-TM-259, March 14, 2003. <http://www.globus.org/cog/manual-user.pdf>

5.3 Kontakte

- Andreas Hoheisel <andreas.hoheisel@first.fraunhofer.de>
- Uwe Der <uwe.der@first.fraunhofer.de>
- Dr. F.-J. Pfreundt (Projektleiter) <pfreundt@itwm.fraunhofer.de>
- CA (Certificate Authority) des FhRG <ca@fhrg.fhg.de>

6 Rechtliches

6.1 Lizenzbestimmungen

Es ist geplant, den Grid Job Handler als Open-Source-Projekt unter der GPL freizugeben. Bis dahin bleibt alles bei Copyright © 2003 Fraunhofer FIRST. Unveränderte Kopien der Software inklusive dieser Lizenzbestimmung dürfen für nichtkommerzielle Zwecke bis auf Widerruf kostenlos weitergegeben werden.

6.2 Marken und Urheberrecht

Die verwendeten Namen und Produktbezeichnungen sind möglicherweise eingetragene Marken oder Marken der betreffenden Firmen.

This product includes software developed by and/or derived from the Globus Project (<http://www.globus.org/>) to which the U.S. Government retains certain rights.

This product includes software developed by TouchGraph LLC (<http://www.touchgraph.com/>)

6.3 Haftungsausschluss

Fraunhofer FIRST haftet nicht für Schäden einschließlich Folgeschäden, insbesondere für keine Schäden an Hardware und Software des Benutzers, die dem Benutzer der von ihr vertriebenen Software entstehen, gleich, ob dieser Schaden durch von ihr produzierte, lizenzierte oder in sonstiger Weise benutzte Software verursacht wurde.