DGrid CP
Contract no.:
Start: 01 September 2005
Dauer: 36 Monate

D-Grid

InGrid

Innovative Grid Entwicklungen für ingenieurwissenschaftliche Anwendungen

AP 2.1: Interaktive Visualisierung und Monitoring

D 2.1.1
Voruntersuchung zur Integration der AA Infrastruktur und des Schedulers in COVISE

Projekt finanziert durch das BMBF unter Vertrags Nr.		
D-Grid intern		
IN	Internal	

Autoren **Organisation**
Florian Niebling HLRS

Inhaltsverzeichnis

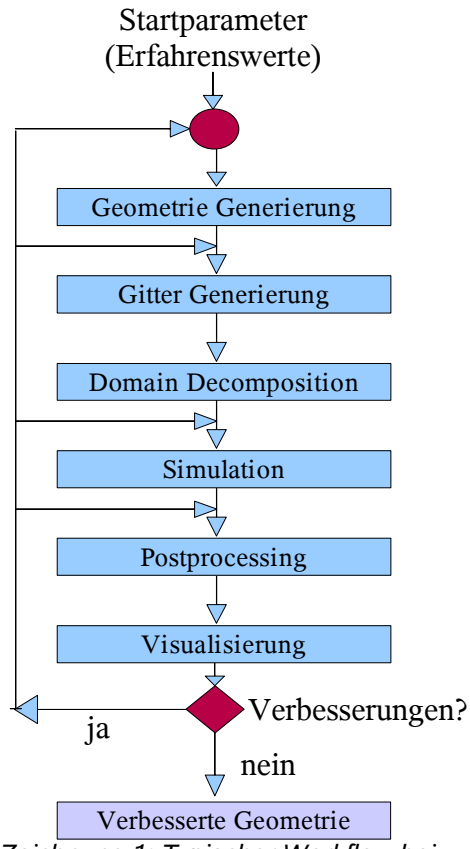
1. Einleitung

Virtuelle Prototypen werden erstellt um Entwicklungsprozesse zu verbessern und zu beschleunigen. Hauptmerkmale sind dabei die enge Integration aller notwendiger Vorverarbeitungsschritte in der Arbeitsumgebung des Benutzers, sowie die Verbesserung der Kommunikation innerhalb und zwischen Entwicklungsteams.

Die typischen Workflows bei der Erstellung virtueller Prototypen sind gekennzeichnet durch den mehrmaligen, iterativen Ablauf verschiedener Arbeitsschritte (siehe Zeichnung 1). Abhängig von den Ergebnissen am Ende eines Iterationsschrittes ist möglicherweise der Eingriff eines Experten in den Workflow notwendig. Da sich die einzelnen Teile des Workflows relativ einfach verteilen lassen, kann der Ablauf durch Service-orientierte Architekturen (*SOA*), wie man sie in heutigen GRID-Implementierungen vorfindet, stark verbessert werden.

Im Vergleich zum statischen Ablauf des Workflows, z.B. in einem Batch-System, wird es dem Benutzer durch die Integration von GRID-Ressourcen in den Entwicklungs-prozess ermöglicht, interaktive Änderungen an der Geometrie sowie weiteren Parametern zur Laufzeit der Simulation vorzunehmen.

Dieses Teilprojekt beschäftigt sich mit der Integration von GRID-Ressourcen in das Simulations- und Visualisierungs-system COVISE.



Zeichnung 1: Typischer Workflow bei der Erstellung virtueller Prototypen

Eingegangen wird dabei auf die Anforderungen an die Entwicklungsumgebung im Hinblick auf

1. Sicherheitsaspekte (Authentifizierung / Autorisierung)
2. Job-Submission und Scheduling

1.1 Authentifizierung / Autorisierung

Authentisierung / Authentifizierung beschäftigt sich mit den Fragen

1. Wer ist der Benutzer?
2. Ist er wirklich der Benutzer der er vorgibt zu sein?

Im Gegensatz dazu befasst sich Autorisierung mit der Festlegung bzw. dem Ermitteln von Zugriffsrechten eines bestimmten Benutzers auf Ressourcen im System.

In diesem Teilarbeitspaket wird hierbei vor allem die Authentisierung des Benutzers gegenüber dem System bearbeitet. Zur Authentisierung werden in D-Grid X.509 Zertifikate verwendet (siehe [1]). Hierbei stellt eine Zertifizierungsstelle (*Certificate Authority, kurz CA*) dem Benutzer ein von ihr signiertes Zertifikat aus. Der Benutzer erhält ein öffentliches Zertifikat und einen privaten Schlüssel, mit dem er sich gegenüber der GRID-Middleware authentisieren kann. Das Zertifikat enthält dabei keinen Eintrag über die Zugehörigkeit des Benutzers zu einer bestimmten Organisation oder eines bestimmten Projektes.

Zur Autorisierung eines Benutzers werden daher Virtuelle Organisationen (VO) verwendet. Dabei kann ein Benutzer einer oder mehreren VOs zugewiesen werden. Die Zugehörigkeit zu einer VO entscheidet dann über die Zugriffsrechte des Benutzers auf eine Ressource. VOs können über sogenannte VO Management Systeme (VOMS) administriert werden. Zu VOMS im D-Grid siehe auch [2].

1.2 Scheduling

Ein Hauptaspekt der Nutzung von GRID-Ressourcen in diesem Teilprojekt ist die Ausführung von Simulations-Jobs auf im GRID beteiligten Rechnern. Hierzu wird, je nach verwendeter GRID-Middleware, Unterstützung für verschiedene Scheduler bereitgestellt. Die für InGrid relevanten GRID-Middlewares wie GLOBUS Toolkit 4, Unicore und LCG/gLite, liefern in der Regel keine eigenen Scheduler mit, sondern können konfiguriert werden lokal vorhandene Scheduling-Systeme zu benutzen. Abseits von reiner Job-Submission bieten *SOA* GRID-Middleware Systeme die Möglichkeit, Webservice Methoden der installierten Services direkt über Programmierschnittstellen aufzurufen. Die Möglichkeiten der hierdurch hinzugewonnenen Flexibilität sollen im Rahmen dieses Teilprojektes untersucht werden.

2. Anforderungsanalyse

Die Entwicklungsumgebung muss es dem Benutzer ermöglichen, vorhandene GRID-Ressourcen zu nutzen. Dem Benutzer muss insbesondere Funktionalität bereitgestellt werden damit er:

1. sich möglichst einfach gegenüber verschiedener GRID-Middleware authentisieren kann
2. Simulationen auf GRID-Ressourcen ausführen kann
3. auf die Ergebnisse seiner Simulationen zugreifen kann

Diese Funktionalität sollte möglichst unabhängig von der verwendeten GRID-Middleware durch den Benutzer nutzbar sein.

Die Sicherheit seiner Daten und gegebenenfalls seiner verwendeten Simulationsprogramme stehen dabei für den Benutzer im Vordergrund. Die sichere Kommunikation sämtlicher im System beteiligter Komponenten ist daher sicherzustellen. Dem Benutzer soll eine einfache Benutzeroberfläche geboten werden, über die er seine Zertifikate verwalten und *Single-Sign-On* gegenüber der GRID-Middleware realisieren kann.

Zusätzlich zu den bisher in COVISE möglichen Methoden zur Job-Submission soll dem Benutzer die Nutzung der von der GRID-Middleware verwendeten Methodik zur Ausführung seiner Simulationen ermöglicht werden. Die Interaktive Visualisierung von Simulationsergebnissen ist hierbei auf eine möglichst zeitnahe Ausführung der Simulation angewiesen.

Die Endergebnisse der Simulationen sollen dem Benutzer möglichst auch aus dem Visualisierungssystem zugänglich gemacht werden.

3. Entwurf

3.1 Authentifizierung

Wie in [1] beschrieben erfolgt die Authentisierung im D-Grid über X.509 Zertifikate. Einer der Vorteile dabei ist die erwünschte Unabhängigkeit von der verwendeten GRID Middleware, da X.509 Zertifikate sowohl von GT4 als auch von UNICORE unterstützt werden.

Um zu verhindern dass der Benutzer bei jeder Nutzung einer GRID-Ressource erneut das Passwort des verschlüsselten *private keys* seines Zertifikates angeben muss, soll *Single-Sign-On* implementiert werden. Dazu wird ein neues Zertifikat (*grid-proxy* Zertifikat) mit beschränkter zeitlicher Gültigkeit erstellt und mit dem Zertifikat des Benutzers signiert. Im Gegensatz zum Zertifikat des Benutzers bei dem der *private key* verschlüsselt abgespeichert wird, kann der *private-key* des *grid-proxy* Zertifikats wegen seiner vergleichsweise geringen Gültigkeitsdauer unverschlüsselt und nur mit Standard-Sicherheitsmechanismen, wie z.B. den Zugriffsrechten der Datei im Dateisystem, abgespeichert werden. Der Benutzer kann nun für die Benutzung von Grid-Ressourcen das neu generierte Zertifikat benutzen, die Notwendigkeit einer mehrmaligen Eingabe seines Passwortes entfällt dadurch.

Zusätzlich zur Verwendung lokaler Zertifikate soll der Benutzer Proxy-Zertifikate aus einem *MyProxy* Webservice (siehe dazu auch [4]) verwenden können.

Im COVISE Map-Editor soll ein Menü eingefügt werden in dem der Benutzer *grid-proxy* Zertifikate aus lokalen Zertifikaten oder aus einem MyProxy Zertifikat erstellen kann. Im selben Menü soll der Benutzer nach Beendigung seiner Simulationen sein erstelltes *grid-proxy* Zertifikat wieder löschen können. Die Benutzerschnittstelle soll dem Benutzer Auskunft geben über aktuell erstellte *grid-proxy* Zertifikate.

3.2 Scheduling

COVISE soll dem Benutzer die Möglichkeit bieten, Simulationen auf GRID-Ressourcen zu starten.

Dazu soll zuerst Job-Submission auf GT4 basierte Systeme implementiert werden. Der Auswahldialog zum Startverfahren von Simulationen in COVISE muss dazu erweitert, sowie Funktionalität bereitgestellt werden die die Simulation an den Execution Manager des GRID übermittelt. In einem weiteren Schritt soll die Möglichkeit hinzugefügt werden, aus einem COVISE Dialog heraus Webservice-Methoden aufzurufen, um damit Simulationen die als GRID-Service implementiert sind zu starten. Diese Funktionalität muss anhand von Simulationen der InGrid Projektpartner getestet werden.

Der Simulations-spezifische Teil der COVISE Simulations-Bibliothek (*coSimClient*) soll ebenfalls um Funktionalität erweitert werden der den Aufruf von Webservice-Methoden erlaubt. Dadurch können Simulationen in die die coSimClient Bibliothek eingebunden wird direkt auf GRID-Ressourcen zugreifen. Hierfür soll die Weitergabe von Berechtigungen (*credential-delegation)* vom Benutzer an die von ihm gestarteten Simulationen implementiert werden.

Die Simulationsergebnisse werden zur Laufzeit der Simulation zur Visualisierungsumgebung COVISE übertragen. Dies geschieht über die in der coSimLib-Bibliothek implementierte Socket-Schnittstelle. Um die Integration in die GRID-Middleware zu verbessern, soll untersucht werden, wie diese Kommunikation über GRID-Schnittstellen zu realisieren wäre. Hierbei müssen vor allem Datenübertragungsraten und Latenzzeit überprüft werden, die, gegenüber zur bisherigen Implementierung, durch die Verwendung von GRID-Schnittstellen beeinträchtigt werden könnten.

Eine Übertragung der Simulationsergebnisse über GRID-Methoden wie z.B. bestimmte Protokolle wie GridFTP oder GRID-Portale soll untersucht und gegebenenfalls implementiert werden.

Die von der GRID-Middleware unterstützten Scheduler müssen auf ihre Eignung für interaktive Simulationen untersucht werden. Hierbei ist zu überprüfen ob eine zeitnahe Ausführung der Simulationen, z.B. über Queues hoher Priorität, gewährleistet werden kann.

4. Stand der Implementierung

4.1 Authentisierung

Es wurde eine graphische Benutzeroberfläche zur Zertifikatsverwaltung in den COVISE MapEditor integriert (siehe Abbildung 1). Diese fügt sich in ein Menü in die Toolbar ein.

Zum Erstellen, Anzeigen und Zerstören von Proxy Zertifikaten wurden GLOBUS GSI Bibliotheken [3] verwendet. Die Anfragen an MyProxy Server geschehen über Webservice-Aufrufe über C-Schnittstellen, die aus der MyProxy Service-Beschreibung generiert werden. Um diese Funktionalität nutzen zu können muss COVISE gegen GLOBUS Bibliotheken gelinkt werden, welche dann auch zur Laufzeit zur Verfügung stehen müssen. Dies betrifft lediglich GLOBUS Client Funktionalität, eine vollständige GLOBUS Installation auf dem Visualisierungsrechner ist daher nicht erforderlich. Die in der graphischen Benutzeroberfläche implementierte Funktionalität ist vollständig kompatibel zu den vom GLOBUS Toolkit zur Verfügung gestellten Programmen zur Erstellung von Proxy-Zertifikaten.

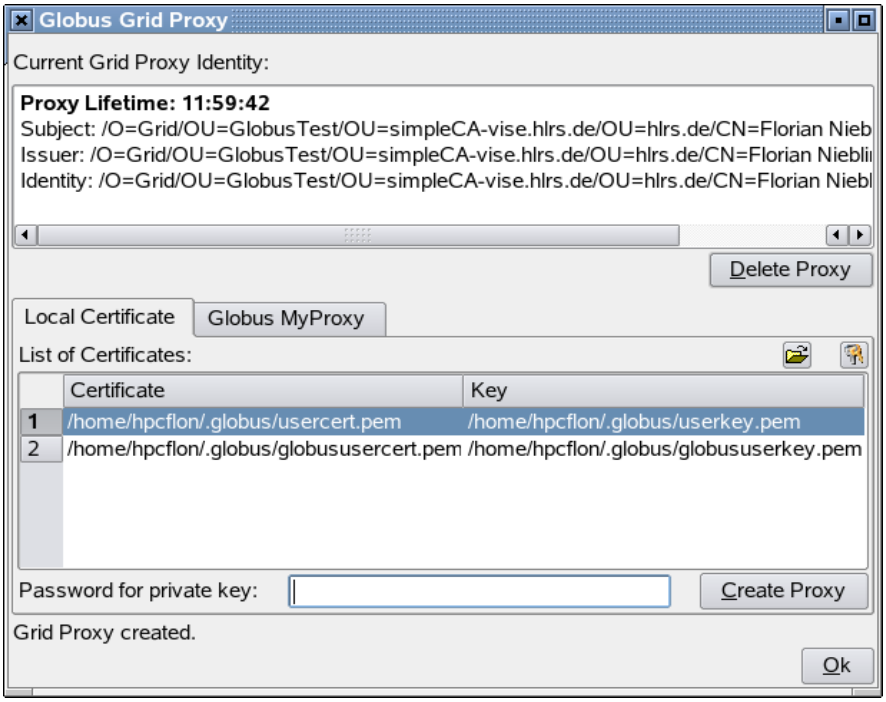


Abbildung 1: COVISE Benutzeroberfläche zur Zertifikatsverwaltung

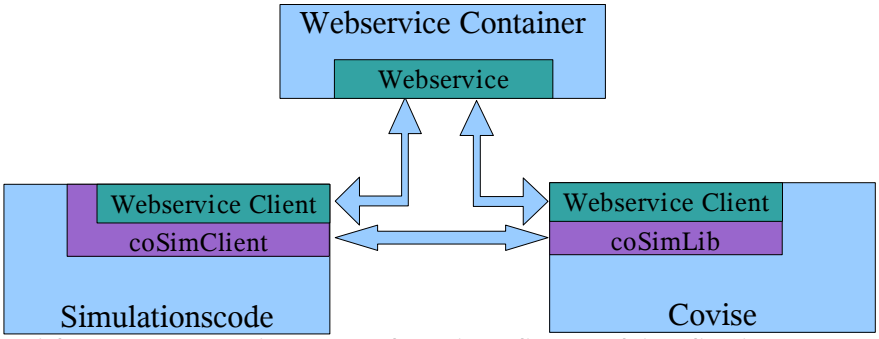
4.2 Scheduling

Job-Submission auf GLOBUS basierte GRIDs wurde in COVISE integriert. Zusätzlich zu den vorhandenen Methoden wie z.B. ssh, kann der Benutzer nun Job Submission durch Nutzung eines GLOBUS GRAM Webservices auswählen. Hierzu werden bis jetzt aus COVISE heraus die vom GLOBUS Toolkit zur Verfügung gestellten Programme aufgerufen. Die direkte Verwendung von Webservice-Schnittstellen zum Zugriff auf den GRAM Webservice wird noch überprüft.

Die Möglichkeit des Aufrufs von Webservice-Methoden aus der Simulation heraus wurde implementiert durch die Integration von Webservice-Client Methoden in den Simulations-spezifischen Teil der COVISE Simulationsbibliothek (*coSimClient*, siehe Zeichnung 2). Die Authentisierung der Simulation geschieht hierbei über die Weitergabe der Credentials des Benutzers (*credential-delegation*) an die Simulation.

Der Aufruf von Simulationen die als GRID-Service implementiert sind ist aus mangelnder Verfügbarkeit einer derartigen Simulation zur Zeit nicht getestet. Da Webservice Aufrufe, z.B. zum MyProxy Webservice, aber generell möglich sind, ist die Grundfunktionalität hierfür bereits vorhanden. Diese Funktionalität muss noch im Zusammenspiel mit einer als GRID-Service verfügbaren Simulation der InGrid-Partner getestet werden.

Verschiedene Scheduler konnten bis zum jetzigen Zeitpunkt noch nicht evaluiert werden. Mit der Bereitstellung eines auf dem GLOBUS-Toolkit basierenden Testclusters am HLRS soll dieser Punkt jedoch untersucht werden.



Zeichnung 2: Integration von Webservice-Client Funktionalität in COVISE

4.3 Übertragung von Simulationsdaten

Simulationsdaten können über die in der coSimLib zur Verfügung gestellte Socket-Schnittstelle zur Visualisierungsumgebung übertragen werden. Diese Schnittstelle ist in anderen Szenarien bereits erprobt und es existieren Anbindungen zu verschiedenen Simulationen. Mit der Evaluation von Methoden zur Übertragung von Simulationsdaten zur Online-Visualisierung über GRID-Schnittstellen wurde bereits begonnen. Jedoch zeigte sich bereits frühzeitig dass die standardmäßig von GLOBUS bereitgestellten Serialisierungs-/Deserialisierungsmethoden für große Binärdaten, wie sie bei der Online-Visualisierung anfallen, aus Gründen der hohen Latenz sowie geringen Datenübertragungsrate bei gleichzeitig hoher Prozessorlast, nicht geeignet sind. In diesem Bereich sind deshalb noch weitere Untersuchungen nötig.

Desweiteren muss noch untersucht werden, inwieweit das GridFTP Protokoll, sowohl zur Online-Visualisierung, als auch zur Übertragung der Endergebnisse der Simulationen, verwendet werden kann. Hier ist gegebenenfalls eine Implementierung in COVISE sinnvoll. Außerhalb von COVISE ist die Verwendung von GRID-Portalen, vor allem auch in Hinblick auf die Implementierung von Simulationen als GRID-Services, noch zu untersuchen.

Literatur

[1] : *Authentifizierung im D-Grid*,

http://www.d-grid.de/fileadmin/dgrid_document/Dokumente/vorschlagspapier-authz_v2.pdf

[2]: *Thesenpapier zum VO Management in D-Grid*,

http://www.d-grid.de/fileadmin/dgrid_document/Dokumente/VOMS-Thesenpapier.pdf

[3] : *Grid Security Infrastructure (GSI)*,

http://www.globus.org/toolkit/docs/4.0/security/

[4] : *MyProxy Credential Management System*,

http://myproxy.ncsa.uiuc.edu/