

# RCE – Remote Component Environment

O. Krämer-Fuhrmann, J. Klein

Fraunhofer Institut für Algorithmen und Wissenschaftliches Rechnen SCAI

Der Einsatz von Simulationswerkzeugen sowie die schnelle und sichere Datenübertragung von CAD- und Berechnungsdaten gehört heute in vielen industriellen Einsatzszenarien zum unverzichtbaren Mittel für die Produktentwicklung. Die verschiedenen Softwareplattformen, die während der Produktentwicklung dabei eingesetzt werden, wie Entwurfssysteme, Simulationssoftware und Datenübertragungssoftware, ja bereits das Aushandeln von Übertragungsmöglichkeiten für verteiltes Arbeiten komplizieren die Verfahren und kosten den Ingenieur unter Umständen viel Zeit.

Das „Remote Component Environment“ RCE [1] ist eine Service-basierte Integrationsplattform für verteilte Ingenieursanwendungen. Es verbirgt die Komplexität von heterogenen und verteilten IT-Systemen mittels wohldefinierter Benutzerschnittstellen und verbessert so die Sicherheit beim Zugriff auf Daten und bereitgestellte Dienste. Als vollständig in Java™ implementierte Anwendung ist RCE plattformunabhängig und kann in verschiedensten Hardware-Umgebungen eingesetzt werden. Unternehmen aus der Schiffbauindustrie entwickelten das „Schiffentwurfs- und Simulationssystem“ SESIS [2], welches RCE als Laufzeitumgebung benutzt.

RCE wird in Zusammenarbeit des Fraunhofer-Institutes für Algorithmen und Wissenschaftliches Rechnen SCAI in Sankt Augustin [3] mit der Einrichtung Simulations- und Softwaretechnik SISTEC des Deutschen Zentrums für Luft- und Raumfahrt DLR in Köln [4] entwickelt.

## 1 Einleitung

RCE ist eine Integrationsplattform, die verschiedene Dienste zur Verfügung stellt, um kollaboratives Arbeiten in einer verteilten Umgebung zu ermöglichen. Anwendungsentwickler, die RCE als Basissystem nutzen, werden weitgehend von Aspekten der Verteiltheit und deren Komplexität entlastet. Übertragung und sicherer Zugriff auf verteilte Daten werden für den Entwickler transparent durch das RCE-Basissystem weitgehend automatisiert bereitgestellt. Dadurch wird das Programmieren von verteilten ingenieurstechnischen Anwendungen auf Basis von RCE nahezu so einfach wie das für lokale Applikationen. Zentrale Eigenschaften von RCE sind

- Plattformunabhängigkeit,
- Sicherheit und
- Erweiterbarkeit.

## 2 Die Laufzeitumgebung

RCE basiert auf „Eclipse™ Equinox“ [5], einer Implementierung des „Dynamic Module System for Java™“ OSGi [6], dem weit verbreiteten Standard für modulare und dynamische Java™-Applikationen. OSGi spezifiziert ein komponentenbasiertes Programmiermodell, das in verschiedenen Implementierungen in unterschiedlichen Anwendungsszenarien eingesetzt wird. Ursprünglich für den Einsatz in „Residential Internet Gateways“, also die mittlerweile immer noch voranschreitende Heimvernetzung (Set-Top-Boxen, DSL-Modems, usw.) entworfen, wird OSGi mittlerweile immer mehr auch in anderen Bereichen eingesetzt, so zum Beispiel in Desktop- und Server-seitigen Unternehmensanwendungen, in Smartphones und auch im Automobilbereich, wo es u.a. die Grundlage für Telematik- und Navigationssysteme ist.

Neben Aspekten der Modularisierung von Software-Komponenten in so genannte Bundles, welche die Erweiterbarkeit und Wartbarkeit von Software wesentlich verbessern kann, sind Konzepte, wie das Management von Zuständen, die ein solches Bundle annehmen kann, Teil der OSGi Service Plattform. Ergänzt wird dies durch sicherheitsrelevante Aspekte wie die Möglichkeit, Bundles zu signieren und die Ausführung einzelner Bundles

einzuschränken. Als wesentlicher Bestandteil von OSGi ist die Service-Schicht zu betrachten. Diese spezifiziert ein allgemeines Dienste-Modell, in dem Dienste über eine „Service Registry“ verfügbar gemacht werden können. OSGi als Service-Framework ist in logische Schichten unterteilt, die diese Konzepte kapseln.

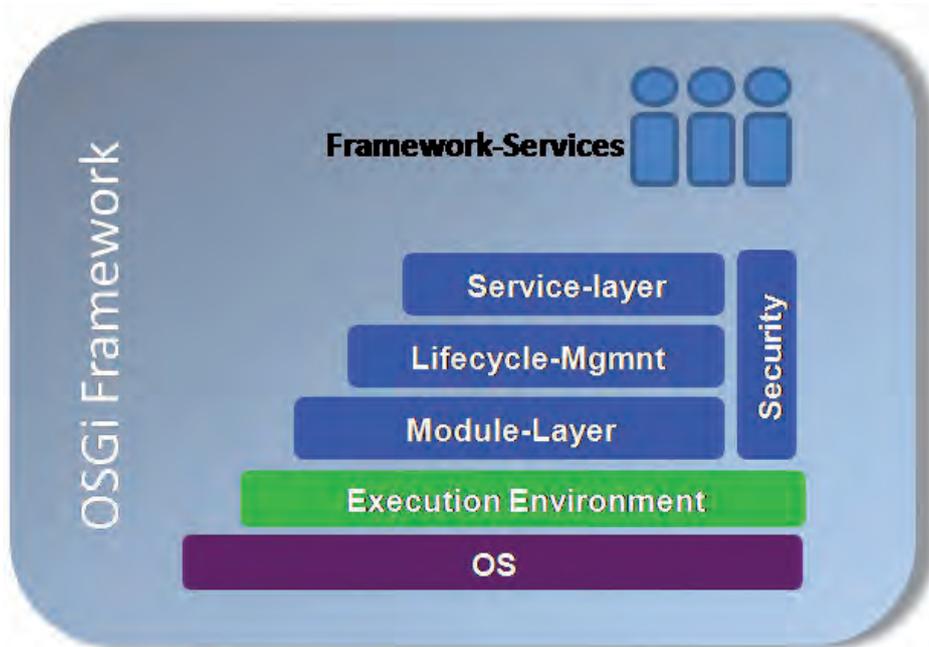


Abb. 1 - OSGi-Schichtenmodell

Software-Komponenten, die als Bundle in OSGi registriert sind, können über das Lifecycle-Management während der Laufzeit des Frameworks (de-)installiert, gestartet und gestoppt werden. Das Lifecycle-Management sieht verschiedene Zustände für ein Bundle vor:

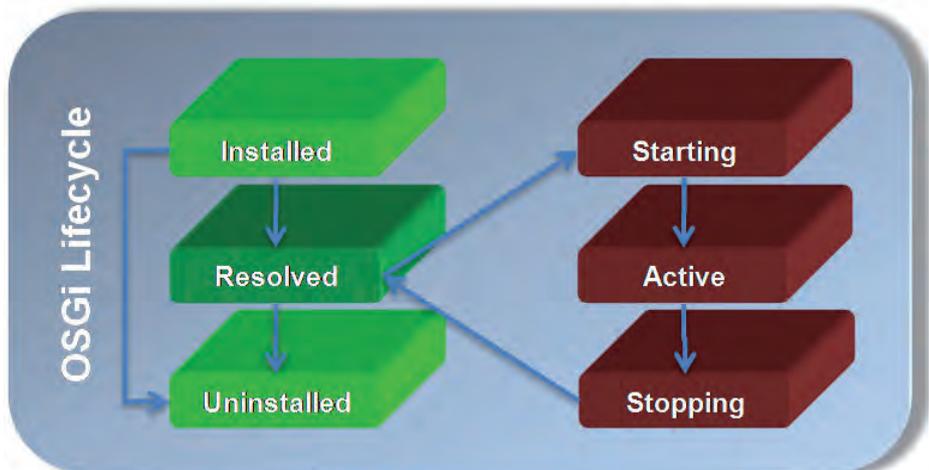


Abb. 2 - OSGi-Lifecycle

In verteilten Umgebungen, in denen Software-Komponenten „remote“ benutzt werden sollen, ist es wichtig, Überblick über die im Netzwerk verfügbaren Dienste zu haben. Daher ist es auch notwendig, die Software-Komponenten so zu implementieren, dass sie auf Nichtverfügbarkeit von entfernten Diensten richtig reagieren können. Während beispielsweise das zeitweilige Fehlen eines Logging-Services keine wesentliche Systembeeinträchtigung erzeugt, muss die Verfügbarkeit anderer Dienste, wie z.B. der Kommunikation, sichergestellt sein, um die korrekte Ausführung der verteilten Anwendung zu garantieren. OSGi bietet hierfür die Möglichkeit, Komponenten als Services bereitzustellen. Dabei werden die Dienste in einer so genannten Service Registry veröffentlicht, über die sie von der Anwendung gefunden werden können.

### 3 RCE-Architektur

Die Software-Architektur von RCE basiert auf diesen Konzepten, indem all ihre Komponenten als Bundles realisiert sind. Diese Bundles werden innerhalb der OSGi-Implementierung Equinox als separate Komponenten deployed. Dies vereinfacht in großem Maße die Entwicklung von anwendungsspezifischen Komponenten. RCE stellt zu den von Eclipse™ Equinox bereitgestellten Standard-Services, wie beispielsweise Logging, Preferences, usw. eigene Dienste zur Verfügung. Diese bieten dem Anwendungsentwickler Funktionalitäten, die das Zusammenspiel verteilter Komponenten organisieren. Ein Service Broker ermöglicht das Auffinden von Diensten auf einer entfernten RCE Installation.

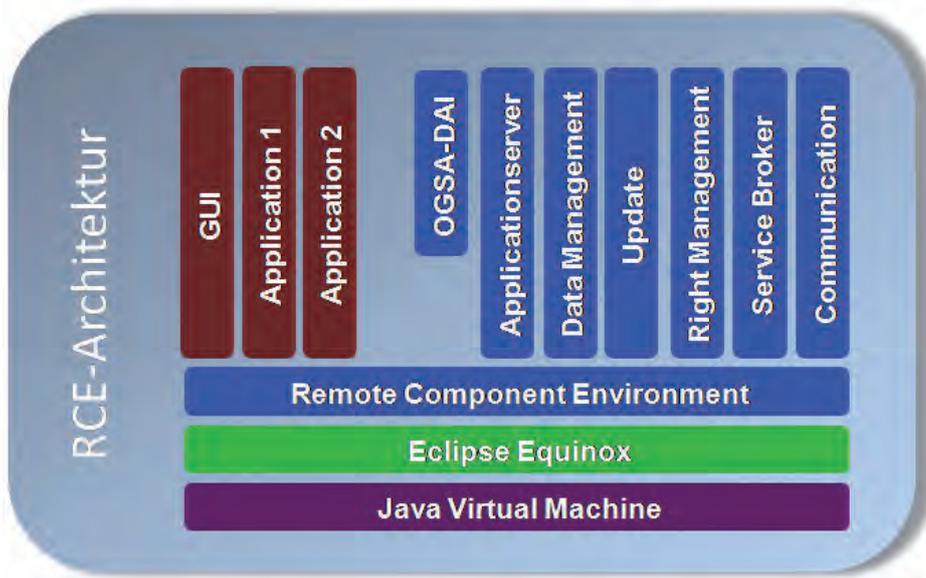


Abb. 3 - RCE-Architektur

### 4 Kommunikation

Die Kommunikation zwischen verschiedenen RCE-Instanzen wird durch den RCE-eigenen Communication Service realisiert. Dieser Dienst bietet verschiedene Kommunikationsprotokolle, wie zum Beispiel RMI (Remote Method Invocation), SOAP (Simple Object Access Protocol) oder RS (Remote OSGi Services) zur Verwendung an. Die Auswahl des Protokolls erfolgt abhängig vom Abstand der Kommunikationspartner:

- Bundles im selben RCE,
- mehrere RCE-Instanzen auf der gleichen Maschine,
- Intranet-Kommunikation oder
- Internet-Kommunikation

Der Anwendungsprogrammierer kann das Protokoll durch Konfiguration außerhalb seines Applikationscodes festlegen, je nachdem, in welcher Umgebung die verteilten Komponenten miteinander kommunizieren sollen.

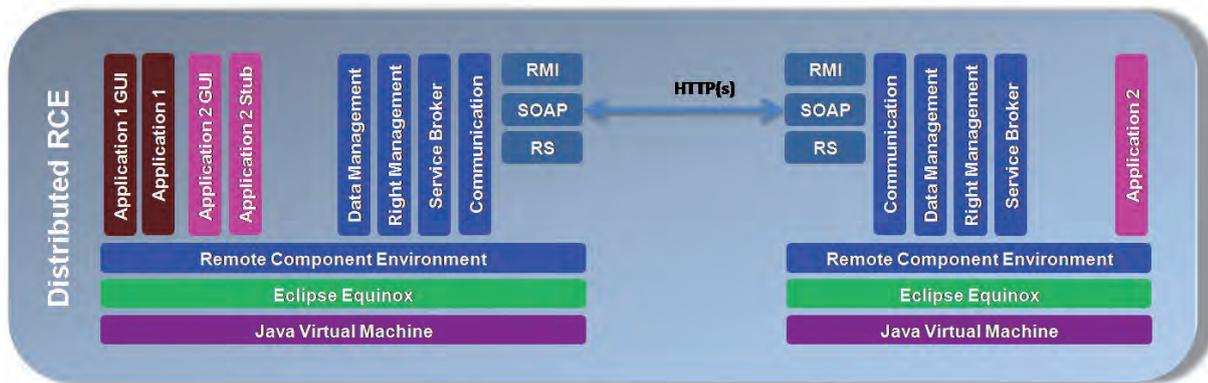


Abb. 4 - RCE Communication

## 5 Datenmanagement

Das Datenmanagement ist eine in das RCE integrierte Komponente zur Verwaltung von Dateien und Datenbanken in einem verteilten System. Mit seiner Hilfe ist es für den Anwendungsprogrammierer unerheblich, ob sich die Daten auf der lokalen Festplatte oder auf einem entfernten Rechner, der sich gar in einer anderen Organisation befinden kann. Mittels einer einheitlichen Schnittstelle wird auf die Datenbestände des RCE Netzwerks zugegriffen, wobei das in RCE integrierte Kommunikationssystem für die Auswahl des optimalen Kommunikationskanals sorgt, je nach Abstand der beteiligten Rechner und den Anforderungen an Geschwindigkeit und Sicherheit. Das in RCE integrierte Rechtesystem sorgt dafür, dass die Daten nur von hierfür autorisierten Benutzern gelesen oder überschrieben werden. Eine Versionsverwaltung sorgt für ein geordnetes Fortschreiben der Datenbestände.

## 6 Rechtemanagement

Das Rechtemanagement in RCE ist in zwei unabhängigen Teilen realisiert: Der Zugriff auf Dienste und Berechtigungen für Operationen auf Ressourcen. Bei der Vergabe und Überprüfung von Berechtigungen auf Ressourcen basieren auf RBAC (Role Based Access Control) [7], das es erlaubt, feingranular Rechte für Benutzer und definierte Rollen zu vergeben. Für die Kontrolle über die Ausführung einzelner Bundles wird die Security-Schicht von OSGi genutzt, mit der sowohl lokale (bundle-interne) als auch globale (systemweite) Ausführungsrechte spezifiziert werden können.

Die Authentifizierung gegenüber der Anwendung erfolgt dabei mittels X.509-Zertifikaten [8]. Dies umfasst sowohl das Anmelden an lokalen RCE-Instanzen über die grafische Benutzeroberfläche als auch die Authentifizierung gegenüber entfernten Instanzen ohne nochmalige Anmeldung mittels Proxy-Zertifikaten. Hiermit wird ein Single Sign-On realisiert.

## 7 Workflow

Mittels des Workflow-Basisdienstes ist es möglich, in RCE integrierte Komponenten zu einem Workflow zu koppeln. Zu jeder Komponente existiert eine Beschreibung in Form einer XML-Datei. Soll eine Komponente Teil eines Workflows sein, müssen in dessen Konfiguration so genannte Input- und Output-Properties definiert werden. Der so genannte Property Service übernimmt bei der Initialisierung des Plug-In die Generierung dieser Properties.

Eine Output-Property der einen Komponente wird mit einer Input-Property einer anderen Komponente verbunden, so dass zwischen Plug-Ins/Komponenten innerhalb eines Workflow typisierte Kommunikationskanäle entstehen. Dabei ist der Kommunikationsprozess Output-seitig als nicht blockierend und Input-seitig als blockierend realisiert. Die Daten werden demnach solange gepuffert, bis alle Nachrichten empfangen wurden und daraufhin in die ursprüngliche Reihenfolge gebracht.

Über eine grafische Benutzerschnittstelle können einzelne Komponenten, welche innerhalb des verteilten RCE-Systems integriert sind, zu einem Workflow verbunden werden, insofern sie die entsprechenden Input- und Output-Properties definiert haben.

## 8 Update

Ein Update Bundle erlaubt es, das System bei der Verfügbarkeit neuer Versionen der installierten Bundles auf den neuesten Releasestand upzudaten. Damit ist eine Administration verteilter RCE-Netze sehr einfach möglich.

## 9 Wrapper-Technologie

Mit Hilfe von so genannten Wrappern ist es möglich, dass externe Anwendungen in RCE integriert werden können. Über JNA-Schnittstellen (Java Native Access) ist es möglich, auch Programme, die nicht in Java implementiert worden sind, in das RCE Netz zu integrieren. Somit können bereits existierende Anwendungen innerhalb der RCE-Plattform verwendet werden.

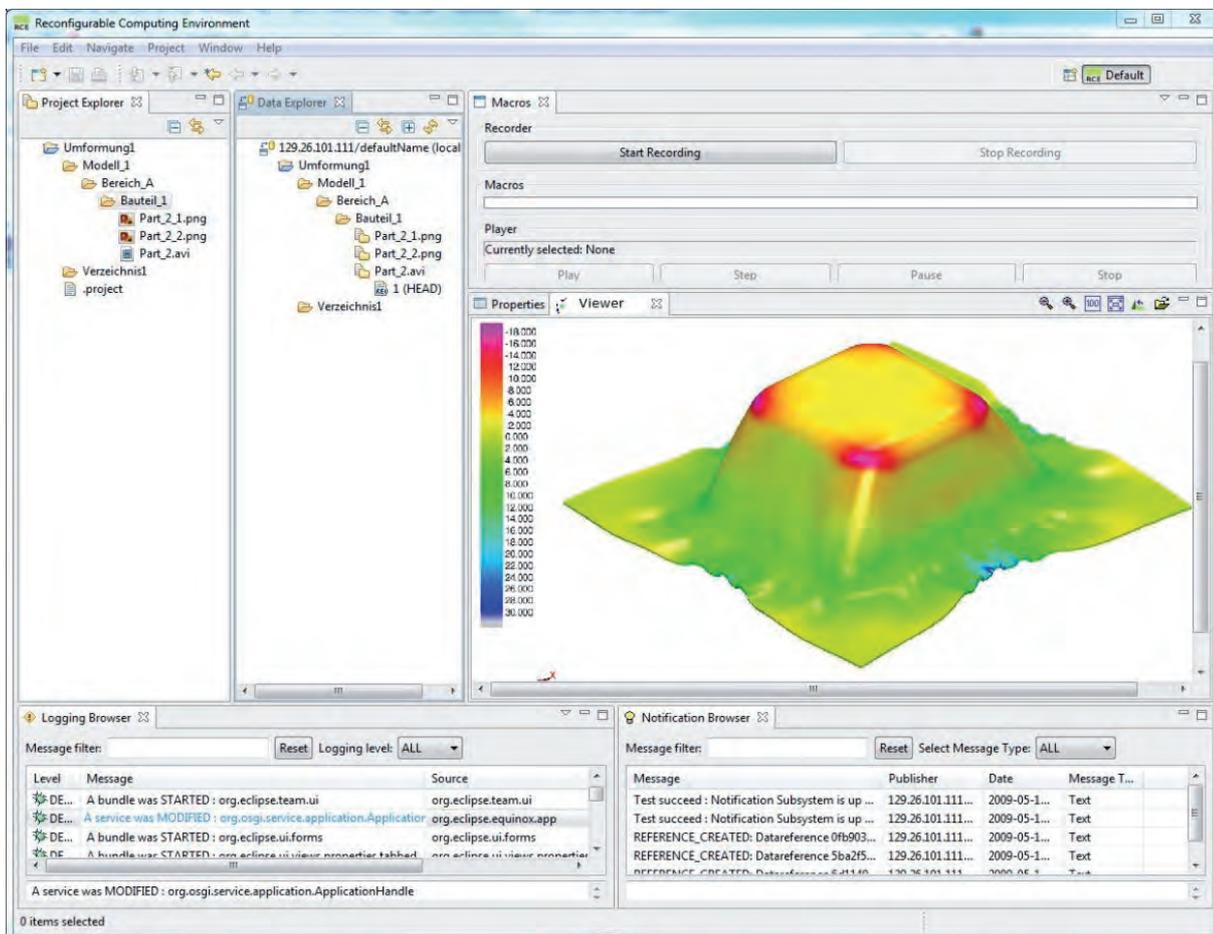


Abb. 5 – Tiefziehsimulation auf RCE-Basis

## 10 Grafische Benutzeroberfläche

RCE stellt eine grafische Benutzeroberfläche zur Verfügung, die einen Rahmen für die eigene Anwendungsentwicklung bietet. Sie beinhaltet Standard-Views, die in einer RCE-Applikation immer vorhanden sind. Darin enthalten sind sowohl einige Eclipse™ Standard-Views als auch ein Logging-Browser, in dem Logging-Meldungen organisiert angezeigt werden können, oder ein Notification-Browser, der RCE-netzweite Nachrichten anzeigt.

Für den Anwendungsentwickler steht außerdem mit DynTable ein Tabellen-Framework zur Verfügung, mit dem Tabellen mit übersichtlichen Benutzerschnittstellen und komfortablen Funktionalitäten entwickelt werden können.

Abb. 5 zeigt die Benutzeroberfläche von RCE am Beispiel einer Tiefzieh-Simulation aus dem Bereich der Verformungstechnik.

## 11 Applicationserver

Um die Kommunikation zwischen RCE und anderen Anwendungen, wie zum Beispiel Web-Portalen oder Grid-Infrastrukturen zu ermöglichen, wurde RCE um eine Webservice-Schnittstelle erweitert. Dazu wurde ein Applikationsserver in RCE integriert, in dem eine AXIS2 (Apache eXtensible Interaction System) Webservice-Engine [9] läuft. Hiermit erhalten Anwendungsentwickler die Möglichkeit, RCE-Funktionen über Webservices zu nutzen.

## 12 OGSA-DAI

Um in Grid-Umgebungen auf Ressourcen im RCE zugreifen zu können wurde eine OGSA-DAI-Anbindung [10] an das RCE-Datenmanagement realisiert. Damit ist es möglich, RCE in heterogene Systeme einzubetten.

## 13 RCE Anwendungen im Schiffbau

Im Forschungsprojekt SESIS [2] (Schiffentwurfs- und Simulationssystem) wurde eine modulare und flexible Plattform für das frühe Entwurfsstadium komplexer Schiffe entwickelt.

Zu diesem Zweck wurden verschiedene schiffbauliche Simulations- und Entwurfswerkzeuge in RCE integriert und über definierte Schnittstellen und entsprechende grafische Oberflächen miteinander verbunden.

Dem schiffbaulichen Anwender bietet SESIS folgende wesentliche Vorteile:

- Effektive Zusammenarbeit zwischen Werft und Zulieferern bereits im Entwurfsprozess durch sicheren Zugriff auf verteilte Daten und Anwendungen,
- Einbindung verschiedenster „historisch gewachsener“ Methoden in eine einheitliche moderne Oberfläche,
- Modularer Aufbau mit der Möglichkeit, Methoden dezentral zu entwickeln und in die Entwurfsumgebung einzubinden – der einzelne Nutzer erhält damit Zugriff auf die Wissensbasis externer Experten,
- Klare und übersichtliche Verwaltung der eingebundenen Methoden und Daten,
- Variabler Zugriff auf die benötigten Methoden, zentrale Lizenzverwaltung und bedarfsgerechte Abrechnung der Nutzung,
- Zentrale Wartung und Pflege der Softwarebasis sowie verteilte Entwicklung und Anpassung der Methoden durch dezentrale Experten.

Ziel von SESIS war es, die Softwareplattform RCE zu entwickeln und ausgewählte schiffbaulicher Methoden darin einzubinden. Nachstehend soll die Nutzung von RCE an verschiedenen Beispielanwendungen aus dem Schiffbau demonstriert werden. Eine Produktionssimulation auf einer Werft wurde von der Flensburger Schiff-

bau-Gesellschaft FSG [11] und dem Center of Maritime Technologies CMT [12] entwickelt. Die Simulation der Fahreigenschaften eines Schiffes und seiner Steuerung wurden von FSG, SAM Electronics [13] und der Technischen Universität Hamburg-Harburg [14] integriert.

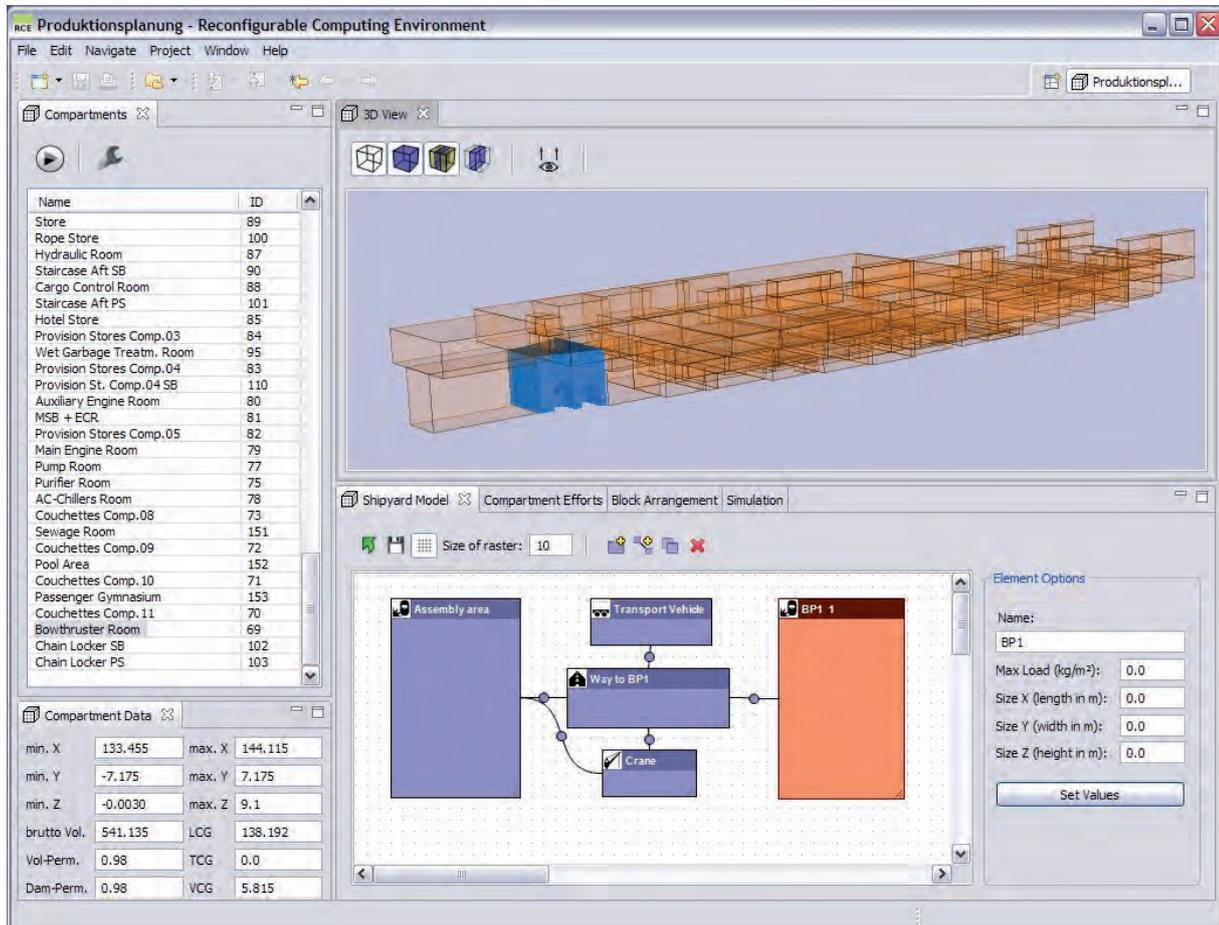


Abb. 6: Visualisierung von Werftdaten für die Produktionssimulation

### 13.1 Produktionsplanung einer Schiffwerft

Der Neubau eines Schiffes benötigt mehrere Monate und bedarf eines abgestimmten Zusammenspiels der Werft mit vielen Zuliefererfirmen. Da Flächen und Arbeitskräfte auf einer Schiffwerft sehr begrenzt sind, ist eine exakte Produktionsplanung notwendig. Durch eine simulationsgestützte Abschätzung der „Baubarkeit“ neuer Aufträge unter Berücksichtigung der bereits festgelegten Werftbelegung können Ecktermine, benötigte Ressourcen und eventuell notwendige Umbaumaßnahmen in der Werft geplant werden.

Die angewandte Simulationsmethode basiert auf einem Raummodell des zu bauenden Schiffes, das von der Werft mit einem Entwurfswerkzeug erstellt wurde. Mit Hilfe einer grafischen Benutzeroberfläche kann der Planer

- Räume aufgrund von Erfahrungswerten mit Bauaufwänden für verschiedene Gewerke belegen,
- Eine grobe Blockeinteilung und Reihenfolgenplanung vornehmen und
- Die kritischen Werftressourcen (Flächen, Arbeitskräfte nach Gewerken) definieren. Dieses Werftmodell kann anschließend zur Planung weiterer Aufträge abgelegt und wieder verwendet werden.

Die durch die Produktionsplanung berechneten Abläufe werden anschließend in werfttypischer Form (z.B. Gantt-Charts, Belegungstabellen, Kapazitätsgebirge) aufbereitet.

Abb. 6 zeigt eine Visualisierung der Räume eines Schiffneubaus und eine sehr einfaches Werfmodell. Aus dem verteilten Datenmanagement-System von RCE werden die benötigten Eckdaten für die Produktionsplanung von der Werft bereits in einem frühen Stadium den Planern zur Verfügung gestellt. So kann ohne Verzögerung damit begonnen werden, die Produktionsstraße für den Bau des Schiffes zu entwerfen und eine Ressourcenabschätzung abzuliefern. Dabei werden nur die Daten für das Planungsteam freigegeben, die für die jeweilige Planungsphase benötigt werden. Das in RCE integrierte Rechtemanagement sorgt für einen kontrollierten Zugriff auf die verschiedenen Daten.

Der Nutzerkreis für die Produktionsplanung sind Werften, die über keine eigene Simulationsanwendungen verfügen und ihre kritischen Ressourcen besser auslasten wollen. Mittels der Simulation kann die Planungssicherheit in frühen Entwurfsphasen entscheidend verbessert werden.

### 13.2 Antriebssimulation eines Schiffmodells

Die Simulation verschiedener Varianten der Antriebsanlage eines Schiffneubaus hilft bei der frühzeitigen Erkennung von Konzept- und Dimensionierungsfehlern. Für Teile des Gesamtsystems aus dem jeweiligen Kompetenzfeld der Werften bzw. Zulieferer existieren häufig Simulationscodes. Die übliche Programmierung von Ersatzmodellen für die Fremdkomponenten sowie der umständliche Datenaustausch werden durch eine direkte Kopplung der Simulationen über Rechnergrenzen hinweg ersetzt.

Schiffs- und Propellermodelle auf einem werftseitigen Rechner sind mit Modellen eines elektrischen Propellerantriebs inklusive der elektrischen Energieerzeugung und Bordnetzversorgung auf einem Rechner beim Zulieferer (Client) über das Internet koppelbar und gemeinsam simulierbar.

In der Simulationsvorbereitung stellt der Server auf der Werft die Schiffe und ihre Anlagenkonfigurationen und Betriebsfälle – im Wesentlichen die Ladefälle – bereit. Auf dem Client beim Zulieferer werden die elektrischen Komponenten ausgewählt, mit Daten versorgt sowie die Simulationsparameter vorgegeben. Während der Simulation sorgt der Client für die Synchronisierung der verteilten Simulation sowie den Datenaustausch. Sowohl auf dem Server als auch auf dem Client werden während des Simulationslaufes unabhängig voneinander Differentialgleichungssysteme iterativ im Zeitbereich gelöst.

Eine graphische Benutzerschnittstelle erlaubt es dem Ingenieur, Fahrten mit dem virtuellen Schiff zu simulieren. Ein digitaler Fahrhebel ermöglicht die Beschleunigung und ein Abbremsen des Schiffes. Währenddessen wird die Steuerung der bis zu 8 Generatoren des Schiffes durch die Zulieferersoftware simuliert. Auf der Werft wird das dynamische Beschleunigungsverhalten des Schiffes simuliert. Alle Messgrößen beider Systeme werden in der GUI aus Abb. 7 zusammengestellt.

Potenzielle Nutzer sind Werften, die über Simulationsmodelle von schiffbaulichen Komponenten verfügen sowie Zulieferer, die eigene Modellstrukturen von Antriebskomponenten verwenden wollen. Sie können mittels RCE Teilsimulationen aus völlig verschiedenen Wissensgebieten miteinander kombinieren. Damit kann sich jeder Projektpartner auf seinen Kompetenzbereich konzentrieren. Der direkte Austausch aktueller Entwurfsdaten ermöglicht eine schnelle Reaktion auf Design-Änderungen. Hiermit können Schwachstellen frühzeitig aufgedeckt werden und Schwachstellen eines Schiffentwurfs frühzeitig optimiert werden.

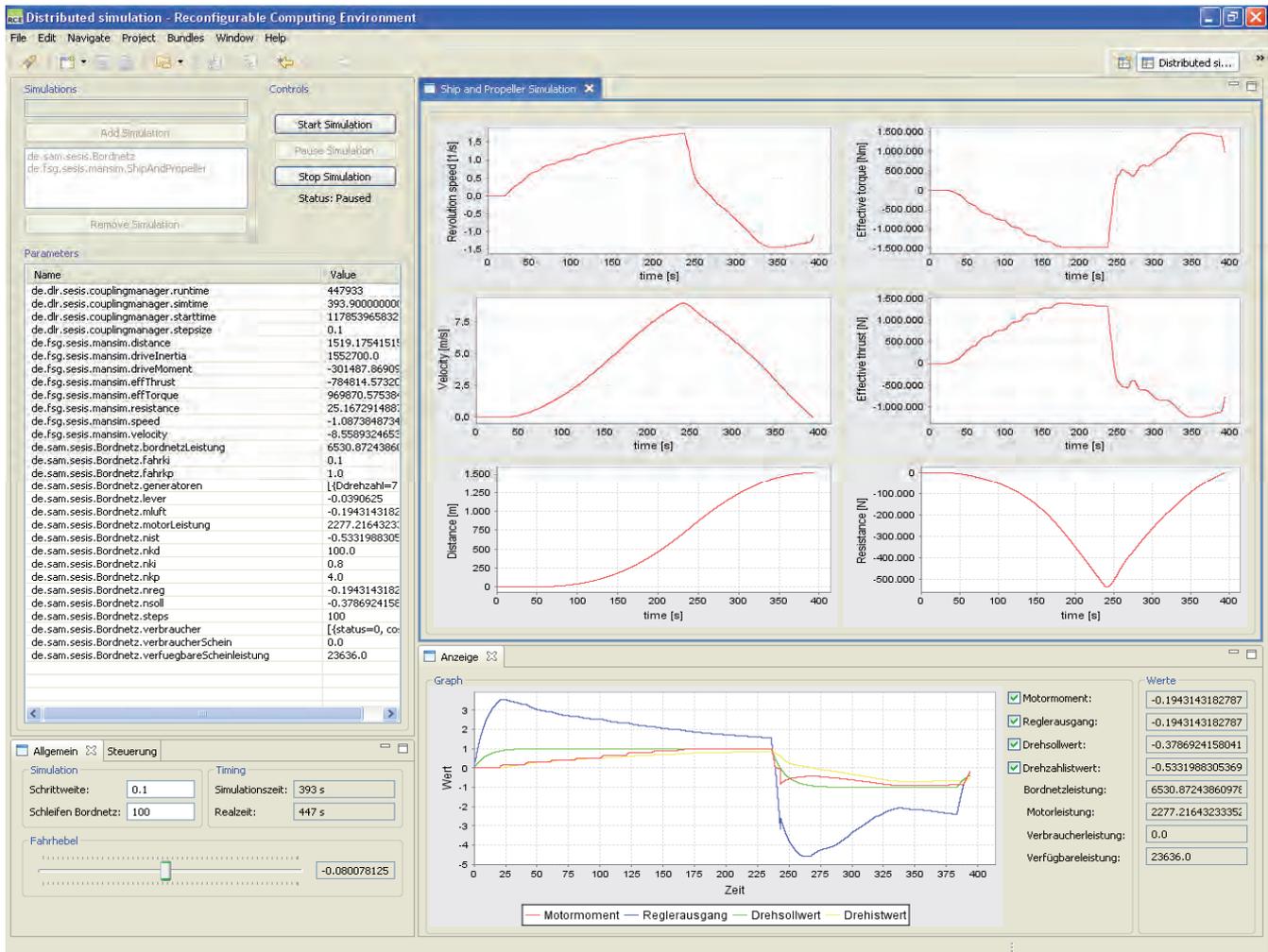


Abb. 7: Verteilte Simulation von Schiff und Antriebsanlage

## 14 RCE Anwendungen im Automobilbau

Im PartnerGrid [15], einem Verbundprojekt der Deutschen Grid-Initiative [16], entsteht eine Grid-basierte Kooperationsplattform für die Zusammenarbeit von kleinen und mittelständischen Unternehmen in den Bereichen Umform- und Gießtechnik. Ziel ist es, die Kooperationsprozesse, die sich aus den zunehmend arbeitsteiligen Entwurfs- und Produktionsprozessen ergeben, effizient zu unterstützen und dabei die flexible Nutzung von Ressourcen zu ermöglichen. Mit der PartnerGrid Plattform werden Anwendungsszenarien realisiert, die zur Entwicklung neuer IT-Services und Dienstleistungen bei den beteiligten Partnern führen.

Im Bearbeitungsworkflow beim Projektpartner GNS-Systems [17] werden sowohl interaktive als auch batchfähige Anwendungen benötigt. RCE dient als Software-Integrationsschicht, um Funktionalitäten der unterschiedlichen Systemkomponenten miteinander zu verbinden, da RCE sowohl lokale als auch Grid-Anwendungen einbinden kann. Das Datenmanagement verwaltet und versioniert die Berechnungsdaten und stellt sie über unterschiedliche Schnittstellen, wie z.B. OGSA-DAI [10] oder WebDAV [18] zur Verfügung. Ein von Fraunhofer IAO [19] entwickeltes webbasiertes Benutzerportal realisiert eine Projektabwicklung für technisch wissenschaftliche Aufträge und kollaboriert hierbei mit RCE-basierten Simulationsanwendungen.

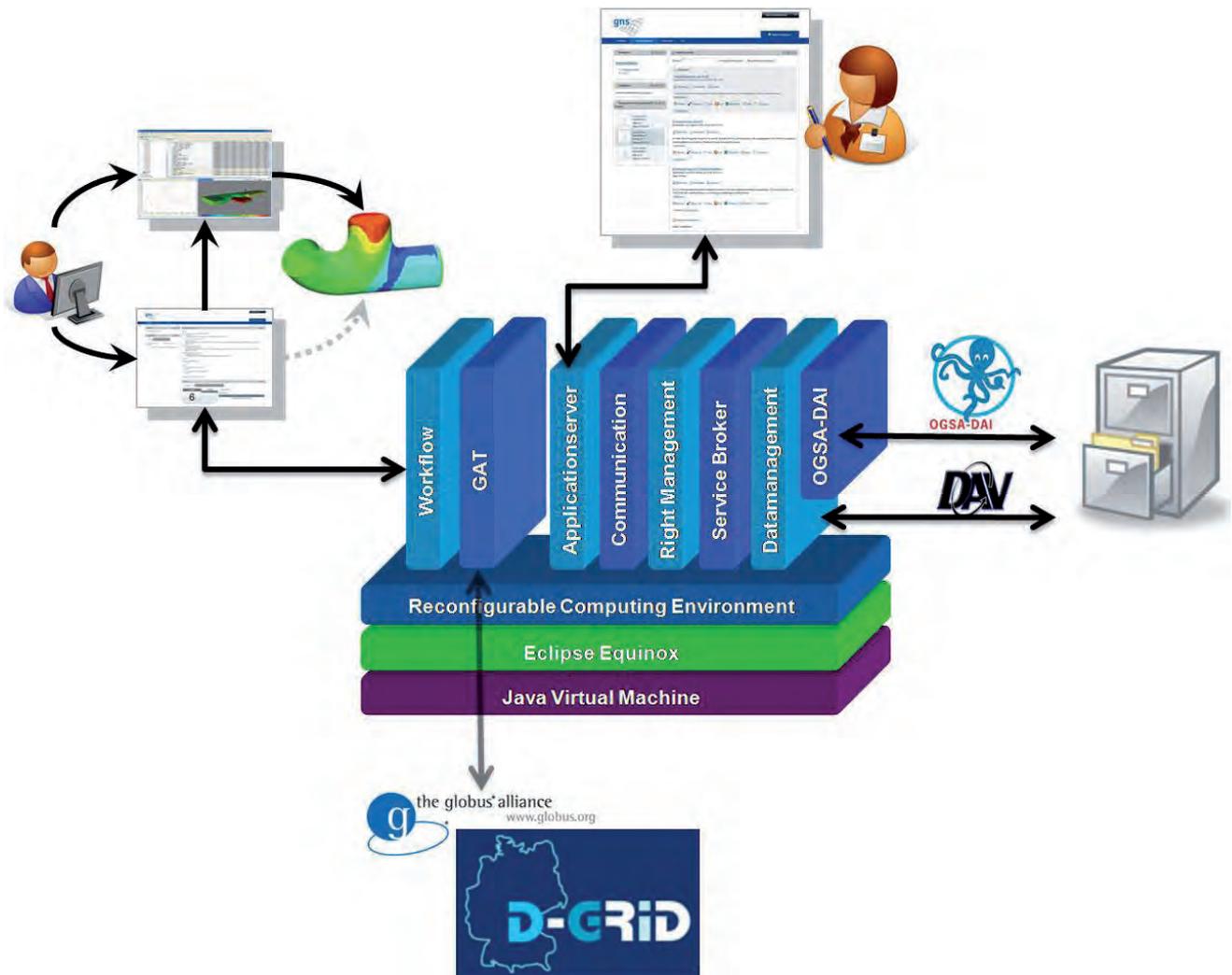


Abb. 7 - RCE innerhalb von PartnerGrid

Mehrere GNS-Anwendungen wurden in RCE integriert und können mit Hilfe der Workflow-Komponente zu einfachen Berechnungsworkflows verknüpft werden. Diese können teils lokal, teils mit Hilfe des Grid Application Toolkits GAT [20] auf der D-Grid Infrastruktur ausgeführt werden.

## 15 Ausblick

Fraunhofer SCAI sowie DLR SISTEC entwickeln RCE kontinuierlich weiter.

RCE wird bereits in mehreren Anwendungsszenarien in verschiedenen Industriezweigen eingesetzt, wie z.B. der Automobilbranche und der Schifffahrtsindustrie. Weitere Projekte sind geplant oder laufen bereits an.

Um der Herausforderung gerecht zu werden, die Softwareentwicklung nahe an den Anwender zu bringen, ist geplant, das Remote Component Environment im Sommer 2009 als Open Source Software zur Verfügung zu stellen. Es wird in einer dualen Lizenz veröffentlicht, wobei der kommerzielle Zweig von RCE Endnutzer-Support, Schulungen usw. durch Fraunhofer SCAI und das DLR beinhalten wird.

Fraunhofer SCAI bietet außerdem kommerzielle Tool-Entwicklung rund um RCE an. Darunter fällt die Entwicklung moderner Bedienoberflächen für Desktop-Anwendungen sowie die Entwicklung komplexer verteilter Anwendungen.

## 16 Literatur

- [1] RCE – Remote Component Environment, [www.rcenvironment.de](http://www.rcenvironment.de)
- [2] SESIS – Schiffentwurfs- und Simulationssystem, [www.sesis.de](http://www.sesis.de)
- [3] Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen SCAI, [www.scai.fraunhofer.de](http://www.scai.fraunhofer.de)
- [4] Deutsches Zentrum für Luft- und Raumfahrt DLR, Simulations- und Softwaretechnik (SISTEC), [www.dlr.de/sc](http://www.dlr.de/sc)
- [5] Eclipse™ Equinox, [www.eclipse.org/equinox](http://www.eclipse.org/equinox)
- [6] OSGi Alliance, Dynamic Module System for Java, [www.osgi.org](http://www.osgi.org)
- [7] RBAC, Role Based Access Control, [en.wikipedia.org/wiki/Role-based\\_access\\_control](http://en.wikipedia.org/wiki/Role-based_access_control)
- [8] RFC 5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, [tools.ietf.org/html/rfc3280/](http://tools.ietf.org/html/rfc3280/)
- [9] Axis2, Apache eXtensible Interaction System, [ws.apache.org/axis](http://ws.apache.org/axis)
- [10] OGSA-DAI, Open Grid Services Architecture, [www.ogsadai.org.uk](http://www.ogsadai.org.uk)
- [11] Flensburger Schiffbau-Gesellschaft, [www.fsg-ship.de](http://www.fsg-ship.de)
- [12] Center of Maritime Technologies e.V. (CMT), Hamburg, [www.cmt-net.org](http://www.cmt-net.org)
- [13] SAM Electronics, Hamburg, [www.sam-electronics.de](http://www.sam-electronics.de)
- [14] Technische Universität Hamburg-Harburg, [kontakt.tu-harburg.de/de/gen/M-6.html](http://kontakt.tu-harburg.de/de/gen/M-6.html)
- [15] PartnerGrid, [www.partnergrid.de](http://www.partnergrid.de)
- [16] Deutsche Grid-Initiative D-Grid, [www.d-grid.de](http://www.d-grid.de)
- [17] GNS-Systems GmbH, [www.gns-systems.de](http://www.gns-systems.de)
- [18] WebDAV, Web-based Distributed Authoring and Versioning, [www.webdav.org](http://www.webdav.org)
- [19] Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart, [www.iao.fraunhofer.de](http://www.iao.fraunhofer.de)
- [20] Grid Application Toolkit (GAT), [www.gridlab.org/gat](http://www.gridlab.org/gat)

## 17 Autoren

Ottmar Krämer-Fuhrmann, [ottmar.kraemer-fuhrmann@scai.fraunhofer.de](mailto:ottmar.kraemer-fuhrmann@scai.fraunhofer.de)  
Jürgen Klein, [juergen.klein@scai.fraunhofer.de](mailto:juergen.klein@scai.fraunhofer.de)

Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen SCAI  
Schloss Birlinghoven  
53754 Sankt Augustin  
[www.scai.fraunhofer.de](http://www.scai.fraunhofer.de)