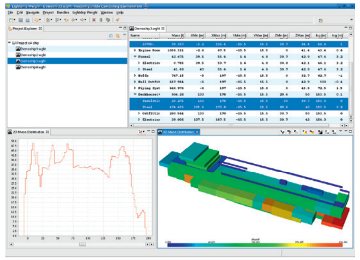
RoRo ferry by FSG and double hull tanker by Lindenau

RCE APPLICATION IN SHIP BUILDING INDUSTRIES

The Ship Design and Simulation System (SESIS)[4] is based on RCE. German ship yards (Flensburger Schiffbau-Gesellschaft[5] and Lindenau GmbH[6]) use SESIS to integrate consultants' and suppliers' tools into the early phase of the ship design process. Due to the architecture of RCE, distributed data and simulation facilities may be transparently shared.

A screenshot of the SESIS software is seen to the right. Here, SESIS is used to compute and visualize the weight of an empty ship, being designed at FSG, with 3-dimensional rooms colored to show the room content and indicate their approximate weight. Tables show the detailed weights of the selected components and graphs give an overview of the weight distribution over the total ship length.



Ship design components based on RCE

Several ship building scenarios have been integrated into SESIS. Overall, the design process of new ships, which are constructed in short time periods and involve integrating the knowledge of a large number of partners, has significantly been improved by RCE.

[4] www.sesis.de
[5] www.fsg-ship.de
[6] www.lindenau.de

RCE is developed by:

Deutsches Zentrum
DLR für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Fraunhofer
SCAI

CONTACT

Doreen Seider
Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR)
Simulation and Software Technology (SISTEC)
Distributed Systems and Component Software
Linder Höhe
51147 Köln | Germany
Tel +49 2203 601-3857 | Fax +49 2203 601-3070
doreen.seider@dlr.de

Ottmar Krämer-Fuhrmann
Fraunhofer Institute for Algorithms
and Scientific Computing SCAI
Schloss Birlinghoven
53754 Sankt Augustin | Germany
Tel +49 2241 14-2202 | Fax +49 2241 14-2181
ottmar.kraemer-fuhrmann@scai.fraunhofer.de

RCE
Remote Component Environment

DISTRIBUTED PLATFORM FOR THE INTEGRATION OF APPLICATIONS

BUILT ON
eclipse

RCE

THE REMOTE COMPONENT ENVIRONMENT

The Remote Component Environment (RCE)[1] is an all-purpose distributed platform for the integration of applications. Applications that are integrated into RCE have access to general-purpose software components such as a workflow engine, a privilege management, or a standardized interface to external compute and storage resources (e.g. Grid or compute clusters). By providing a unified environment with ready-to-use existing components RCE hides complexity and enables developers to concentrate on their application-specific logic.

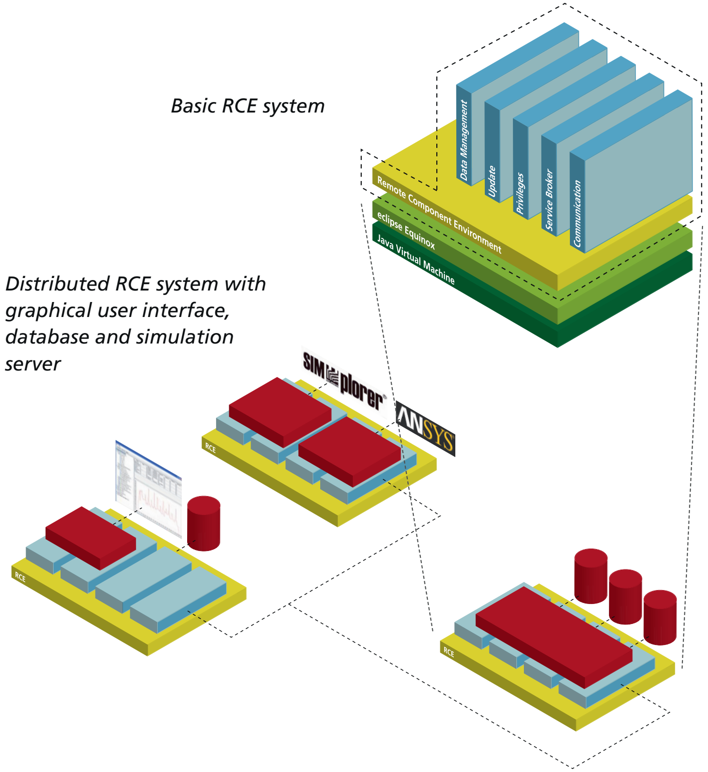
RCE is designed as a component-based system. It is build upon Equinox, an OSGi[2] implementation, and the Eclipse Rich Client Platform.[3] OSGi is a specification of a dynamic, modular platform that has been widely adopted within the Java software industry. The Eclipse Rich Client Platform builds the strong foundation of RCE that provides a substantial runtime environment, a graphical programming interface, and many already existing components. Thus, RCE was not build from scratch, but developed relying on a mature existing code base.

RCE can run in a distributed heterogeneous environment of Windows and UNIX systems. On every participating resource one instance of RCE is installed. These instances may have different characteristics due to the RCE components installed and the applications integrated (e.g. application server, data base server, or client with graphical user interface). RCE is designed in a way that its distribution is transparent to both users and developers of integrated applications.

RCE COMPONENTS

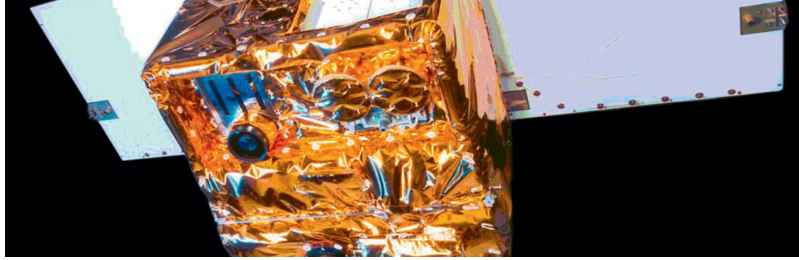
Among the software components that are provided by RCE are the following:

- The **communication component** provides uniform access to distributed RCE instances. As of now, the communication protocols RMI and SOAP are implemented.
- The **privilege management** allows the allocation of permissions at application and data level. The authentication is certificate-based and allows for single sign-on within the RCE context.
- The **workflow engine** couples integrated applications to executable workflows within the distributed system.
- The **data management** allows storing data sets within the distributed system in persistent and version-controlled form.
- A unified **Grid interface** provides access to heterogeneous Grid resources and High Performance Computing systems.



Basic RCE system

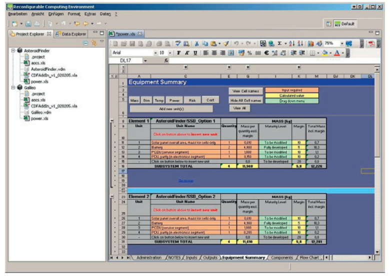
Distributed RCE system with graphical user interface, database and simulation server



BIRD satellite developed at DLR

RCE APPLICATION IN SATELLITE DESIGN

The German Aerospace Center (DLR) currently develops a Concurrent Engineering Facility (CEF) to improve the design process of space systems. A CEF is roughly characterized by four main parts: a standardized development process, work in interdisciplinary teams, an infrastructure to make this work possible, and a central tool to manage the data model of the space system.



Excel workbook integration within RCE

Whereas the first three parts are not directly affected by software systems, the last one needs to enable all members of the team to concurrently and efficiently work with the shared data. In so-called Phase-A studies usually a framework, based on Microsoft Excel spreadsheets, is used. This leads to problems with the underlying data model and its handling via Excel macros, which are unreliable and prone to errors.

Within a DLR-internal study these shortcomings have been consequently solved by developing a new object-oriented data model that is based on the RCE framework. Here, RCE automatically handles the data exchange between the Excel spreadsheets and the new data model, providing additional features like permission and version control.

[1] *www.dlr.de/sc/produkte/rce, www.scai.fraunhofer.de/rce.html, www.rcenvironment.de*

[2] *www.osgi.org*

[3] *www.eclipse.org*