Advances and Work in Progress in Aerospace Predesign Data Exchange, Validation and Software Integration at the German Aerospace Center

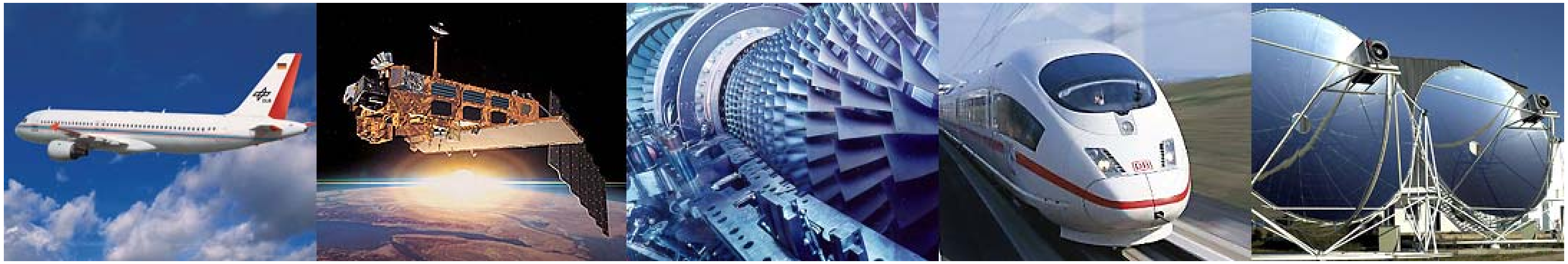
Arne Bachmann, Markus Kunde, Markus Litz, Daniel Böhnke, Stefan König

German Aerospace Center DLR, *Simulation and Software Technology*

Dept. Distributed Systems and Component Software

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

DLR
German Aerospace Center



- Research Institution
- Space Agency
- Project Management Agency

Locations and employees

6500 employees across
29 research institutes and
facilities at

- 13 sites.

Offices in Brussels,
Paris and Washington.

Guiding Principles – Vision

- DLR – one of Europe's leading public research institutions, setting trends in its aeronautics, space, transport and energy business areas
- DLR – in its space agency function, a force that shapes European space activities
- DLR – the umbrella organisation for the most effective and efficient project management agencies and offices

Guiding Principles – Mission

- To explore Earth and the Solar System; to conduct research into the preservation of the environment, into mobility and into public safety, and to address societal questions on behalf of public customers
- To bridge the gap between basic research and innovative applications and to transfer knowledge and research results to industry and the political sphere through mediation and consultation as well as through the provision of services
- To shape Germany‘s space commitment and represent its interests internationally as a governmental function
- To make a significant contribution towards enhancing Germany as a science and business location as well as to stimulate growth in the European region
- To train young scientists in order to enhance Germany's innovative capability

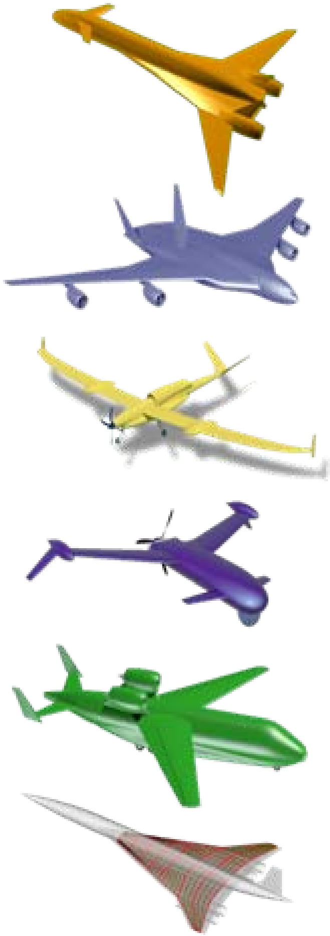
Guiding Principles – Approach

- Discipline-oriented institutes to support scientific work
- Matrix structure of programmatic control and technical management
- Support in the design of framework conditions for legal and public policy
- Operation of large research facilities and infrastructure for DLR’s research activities and missions as well as for customers and partners
- Consistent system of strategy, management and quality assurance
- Job-tailored, demand-oriented personnel management and systematic employee development
- Realisation of equal opportunities and support of work-life balance
- Contractually regulated partnerships with universities, industry, other research organisations and public customers

Overview

- Introduction
- Status as of 2008
- Limitations of current approach
- Recent experiences (2009, 2010)
- Hot topics
- Outlook

Introduction



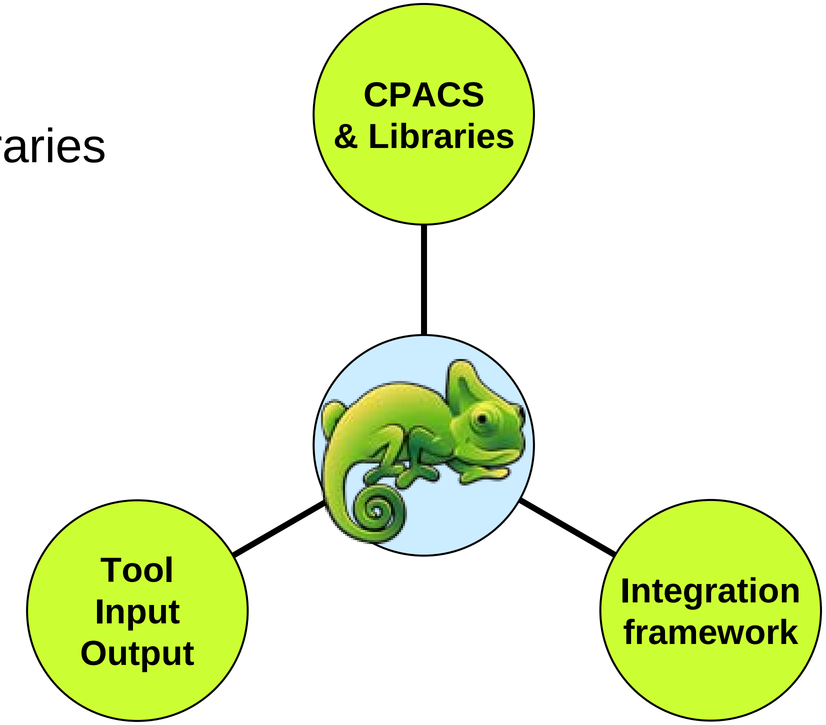
- Interdisciplinary cooperation within DLR
- Goals: Simulation, construction, assessment, experiments
- Global optimization of complex models
- Reuse and integration of existing disciplinary tools

The Chameleon Integration Environment

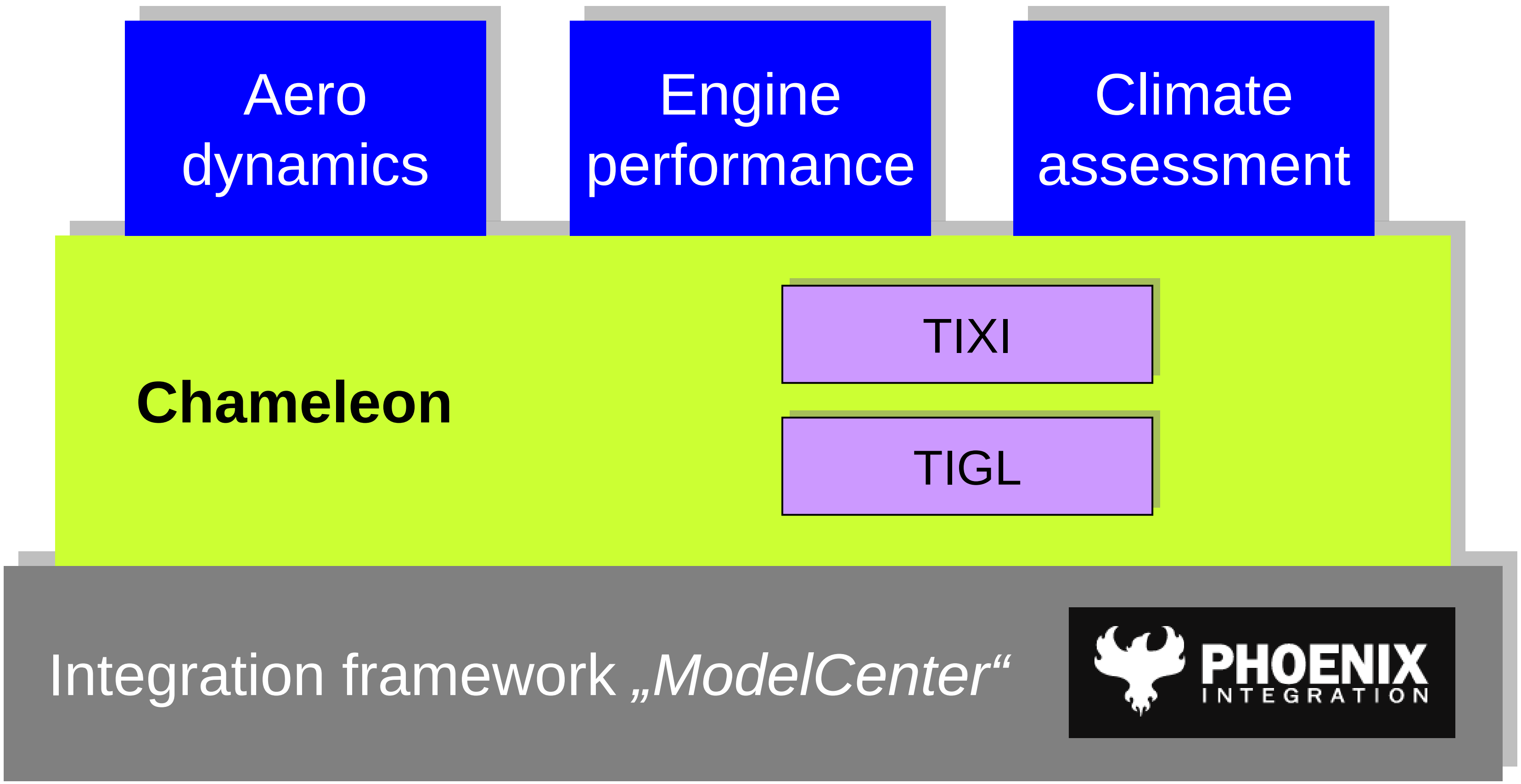
- A simulation and integration environment
- A software suite that enables you to
  - integrate any simulation tool
  - communicate data between tools
  - build up simulation processes and workflows
  - try out new ideas, experiment with your data and methods
- A principle, methodology, philosophy?

The Chameleon Idea

- Decouple all software parts of the simulation environment:
  - The disciplinary tools
  - The integration framework
  - Common data format
    - and their accessing libraries



Coarse picture of the Chameleon Architecture (2008)

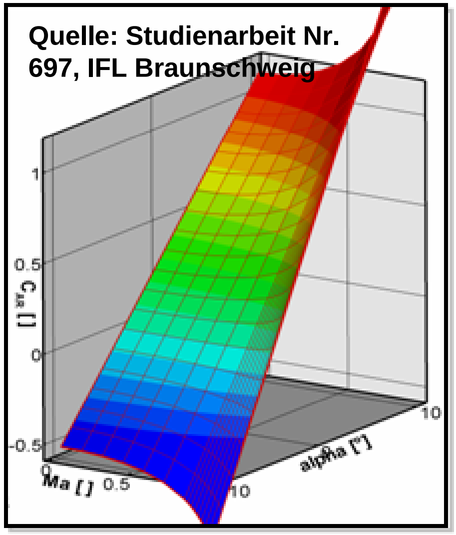


Advantages of Chameleon

- Really simple to
  - set up simulation processes
  - integrate your own tools into the simulation environment
- Usefulness increases
  - Interfacing with the common data format increases interoperability
  - Cooperations become easier to start with each newly added tool
  - Inclusion of many tools into *one* environment boosts productivity
  - Better reproducibility and validation capabilities with workflows

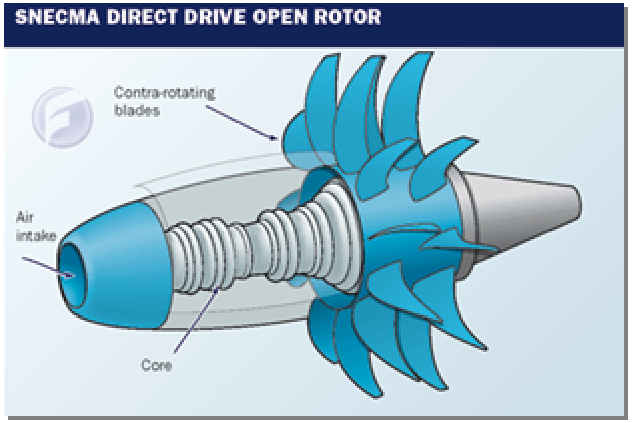
Examples of integrated predesign tools

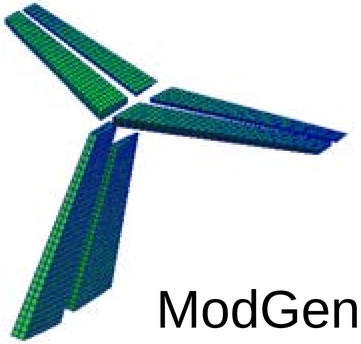
PANAM;
Mission simulation



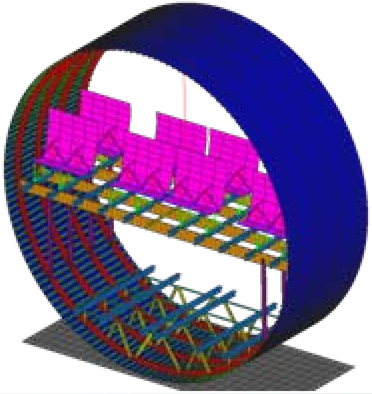
Lifting Line

HEIDI





ModGen



PAM-CRASH / SECT-MESH

TIVA / TIVA II

Technology Integration for the Virtual Aircraft

- DLR project (2005 – 2009)

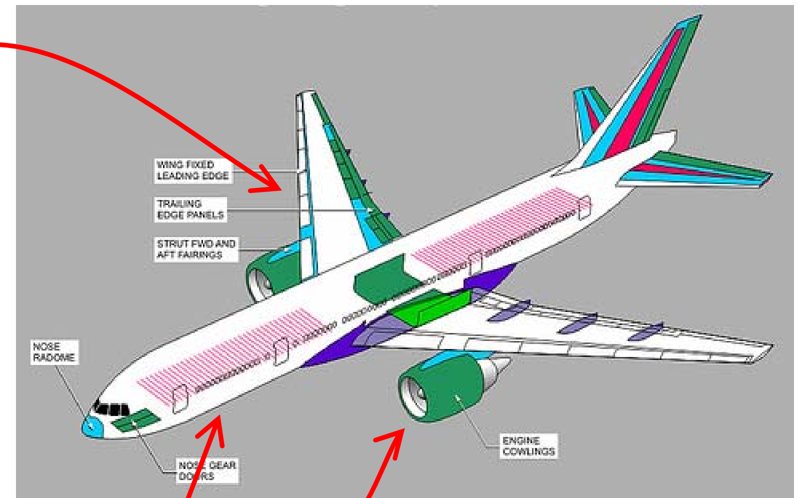
- **Goal**:
  Define and develop concepts and technologies to create and assess aircraft configurations in preliminary aircraft design

- Define a common data format (CPACS)
  Common Parametric Aircraft Configuration Schema

- Build up and integrate a toolkit of aviation tools to create simulations
  - Every tool communicates via CPACS data

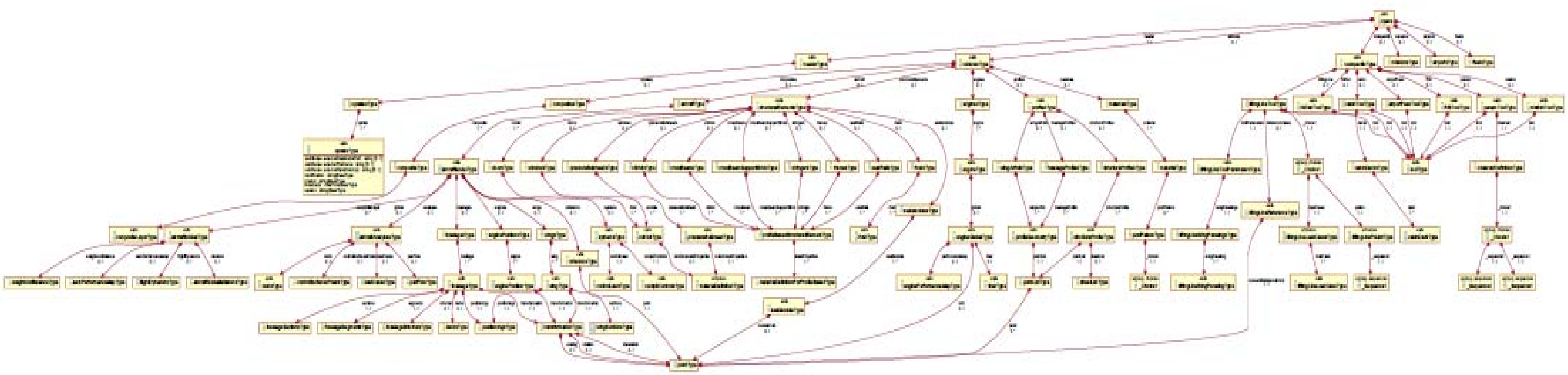
Common Parametric Aircraft Configuration Schema CPACS

- cpacs
  - header
  - global
  - engines
  - wings
    - sections
      - section
        - transformation
      - elements
        - airfoilUID
  - fuselages
  - missions



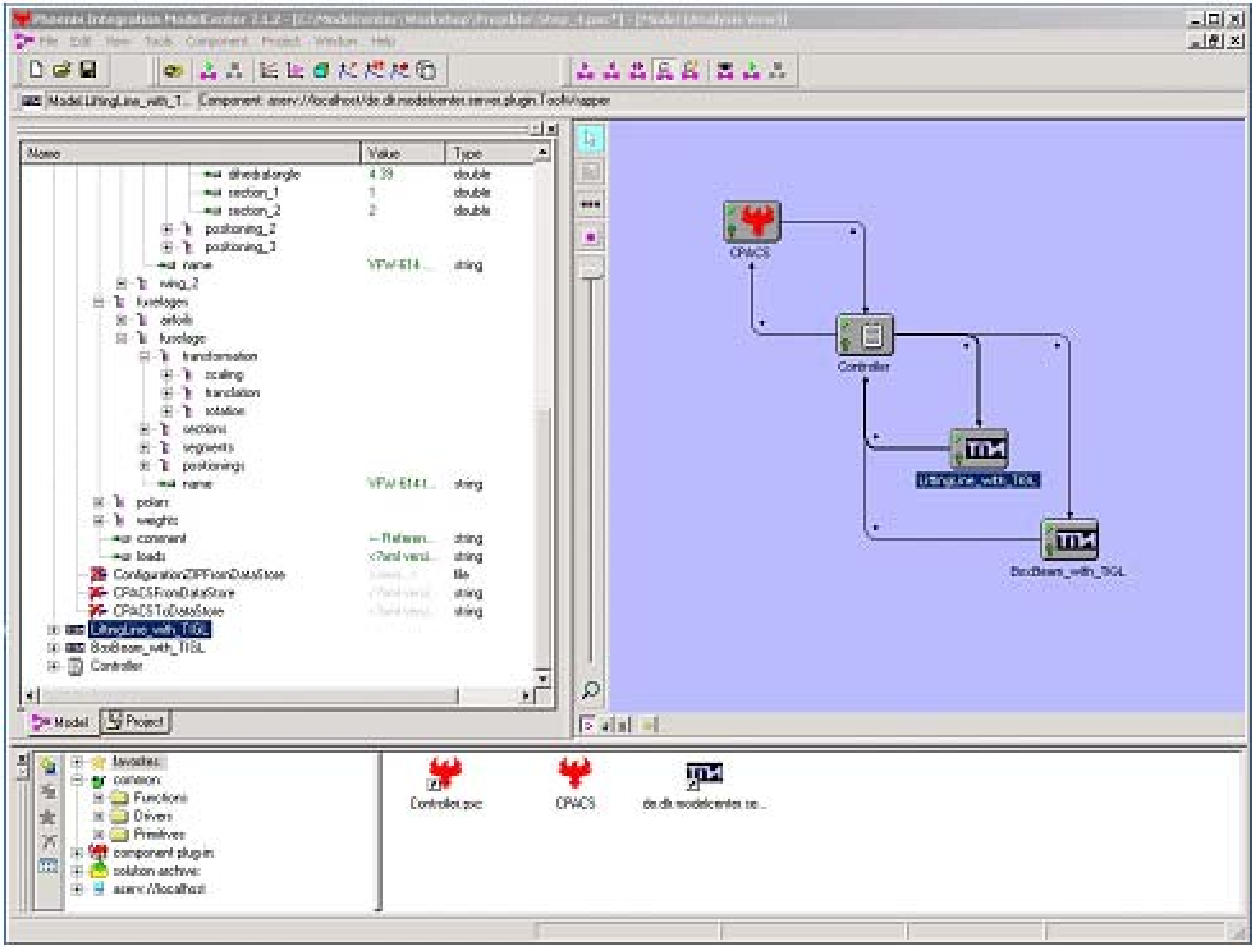
CPACS holds data concerning…

- airplane…
  - wings
  - fuselages
  - engines
  - systems
- missions
- airports
- fleets
- tools

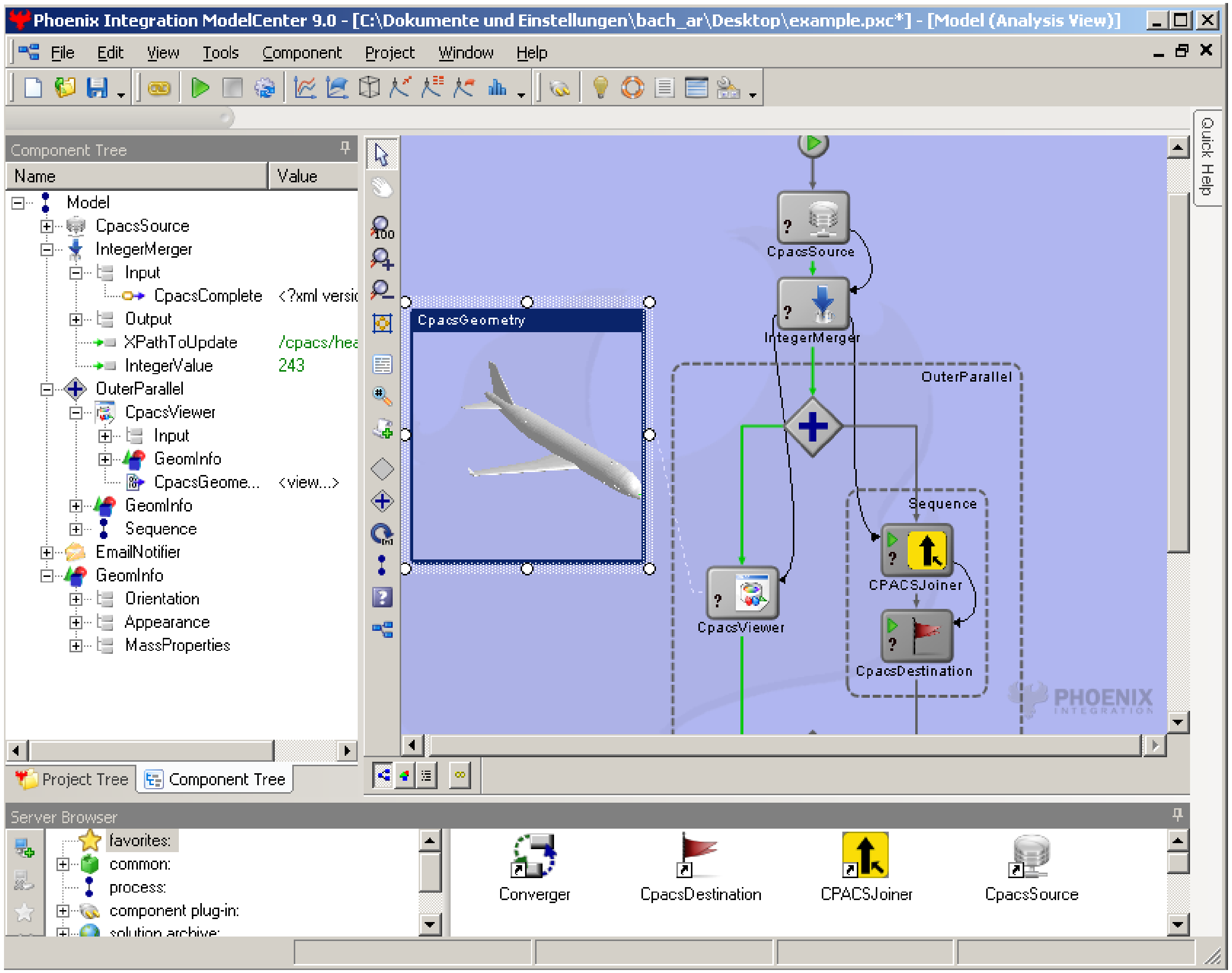


Status as of 2008

What it looked like:



What it looks now in 2010



Improvements since 2008

- Libraries
  - XML library allows for inclusion of other resources (even remote)
  - Geometric library creates watertight surfaces; more export formats
- Replaced script-controlled workflow by a data-driven approach
  - Leverages dynamic data flow scheduling for parallel tool execution
- Many more utility components
  - Email notifier (useful in long-running workflows)
  - Automatic bug report (one-click report to the developers)
  - Tool finder (in local network, grouped, categorized)
  - Data combinators (merge, split subtrees of complete dataset)

What we wanted to do after PDE 2008

- Define a data format
- Integrate tools via CPACS with each other
- Set up a workflow system
- Integrate mission control data into CPACS
- Geometry modeling for other aircraft parts
- Use our environment in other projects

What we wanted to do after PDE 2008

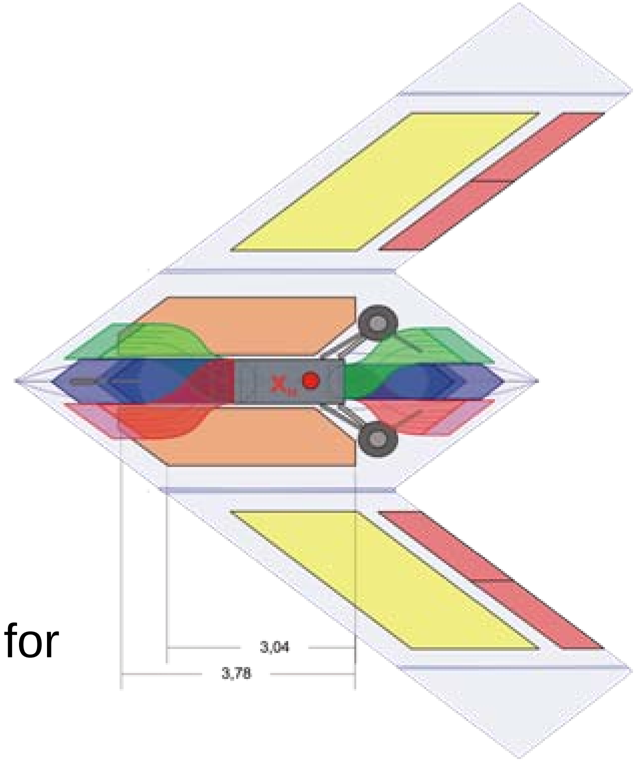
- Define a data format, current version is 1.2
- Integrate tools via CPACS with each other (15+ tools)
- Set up a workflow system (replaced central controller)
- Integrate mission control data into CPACS (since 1.0)
- Geometry modeling for other aircraft parts (e.g. engine nacelle)
- Use our environment in other projects (5+ projects)

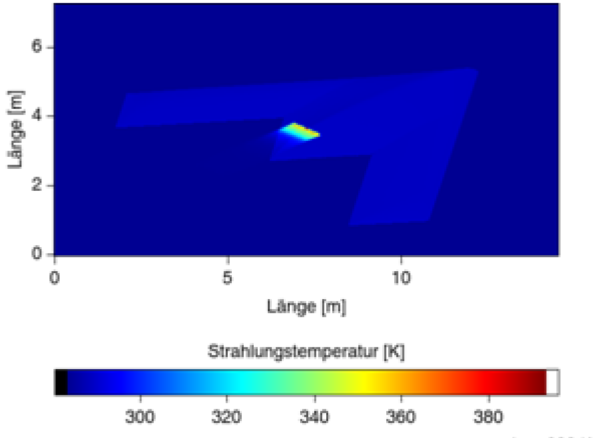
UCAV-2010
Unmanned Combat Air Vehicle

- DLR project (2007 – 2010)

- **Goal**:
  Numerical and experimental methods for the development and assessment of technologies for unmanned aerial vehicles

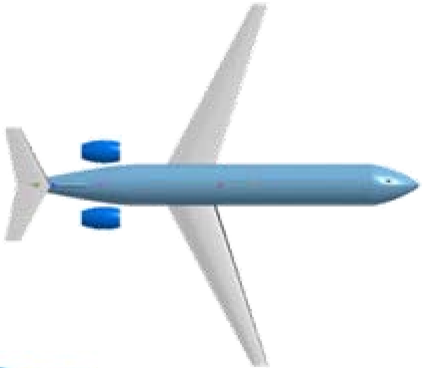
- Integrate infrared and radar signature tools
  - Extend the dataset for this purpose
  - Add new functions for the export of surface meshes

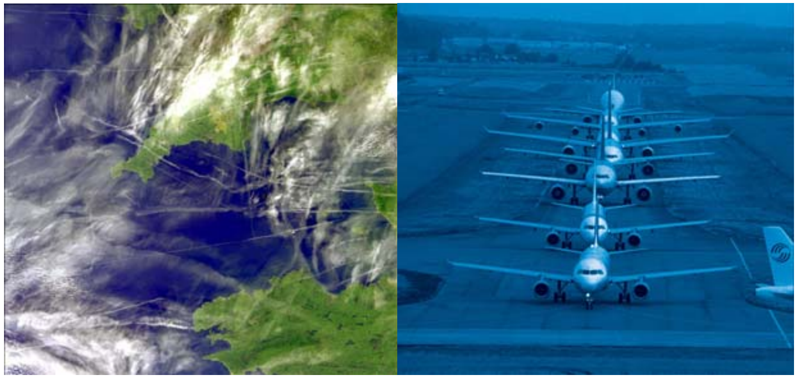




Four more projects…

- **Evaluation of innovative turbine engines (EVITA)**
- **Climate-compatible Air Transport System (CATS)**
- **Virtual Aircraft Multidisciplinary Analysis and Design Processes (VAMP)**
- **Integrated modelling of the air traffic system (IML2)**

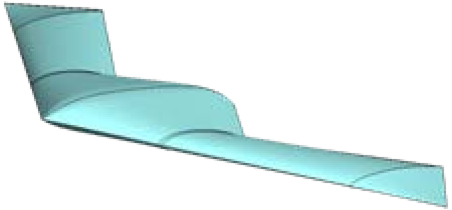




EVITA

TIM

VAMP



Airport2030

TIM II

Limitations of Chameleon when using the Integration Framework *ModelCenter*

- Client runs only on Windows
- Difficult to detect and handle errors
- Some license and support costs
- No integrated data management (yet, as of version 9)
- Difficult/cumbersome to extend by own ideas (views, components)

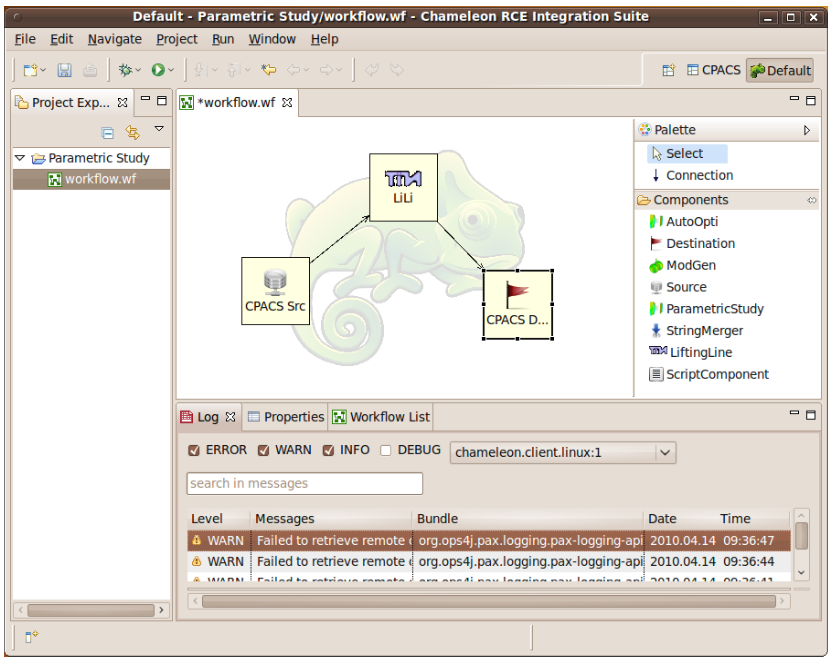
Recent experiences

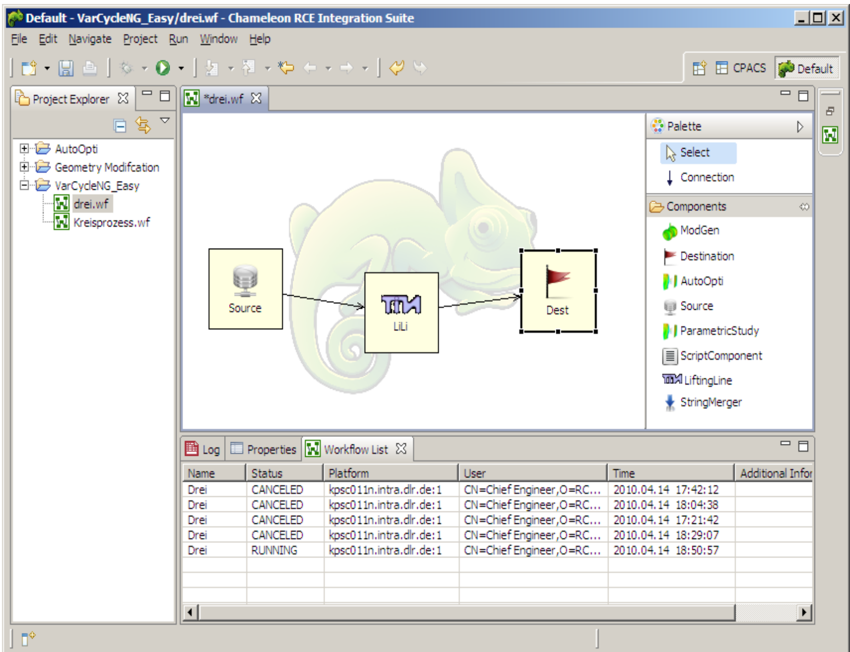
- Realization of the Chameleon ideas on top of another framework
  - Remote Component Environment (RCE) www.rcenvironment.de
  - Developed since 2006
  - Successfully deployed e.g. in ship construction predesign

- Advantages
  - Allows to run a workflow on a remote node
  - Allows to have local GUIs for remote components
  - Is free and open source
  - Has an integrated dynamic help
  - Is modular, extendable, stable
    - Based on Eclipse RCP / OSGi

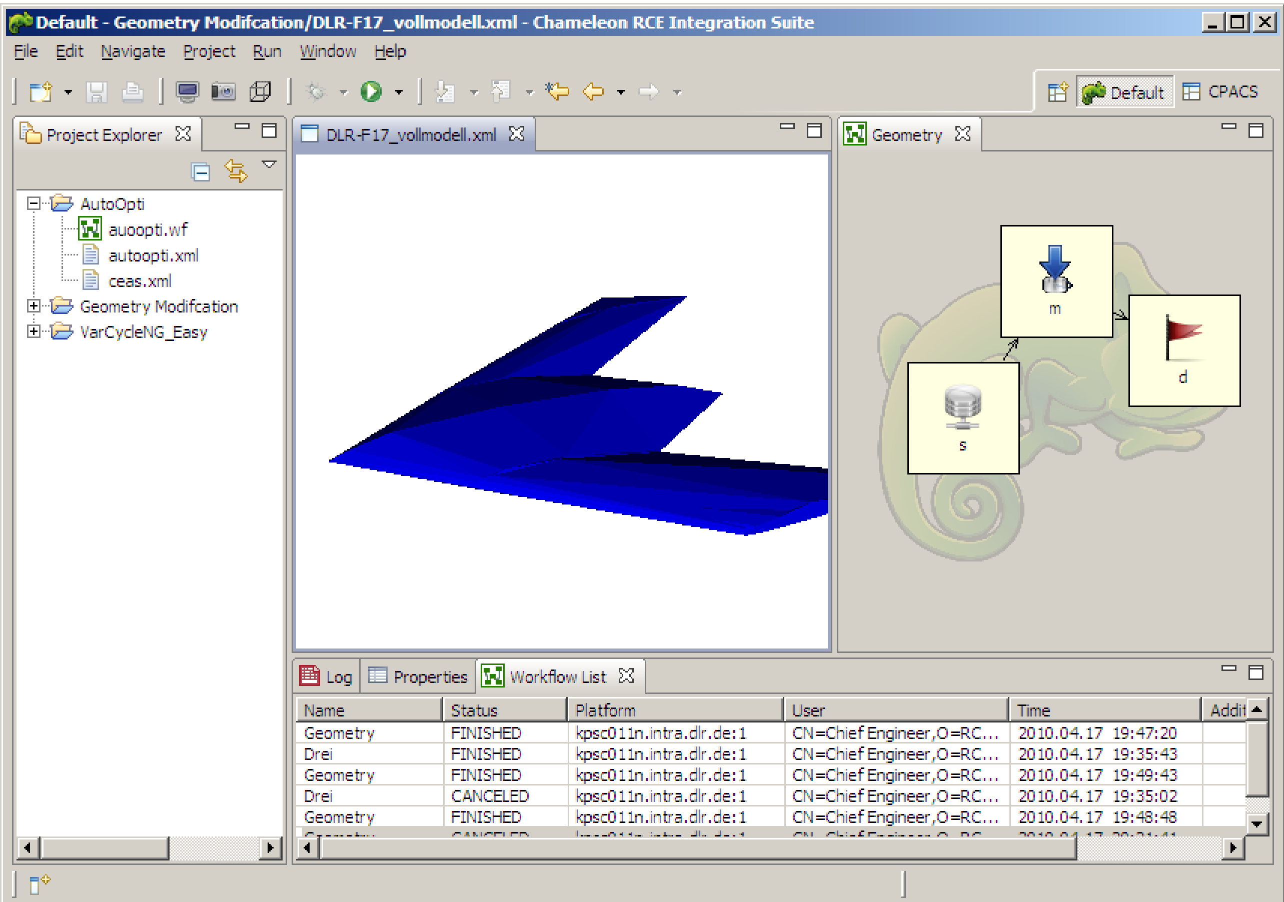
Operating System Independence

- Currently Linux and Windows supported





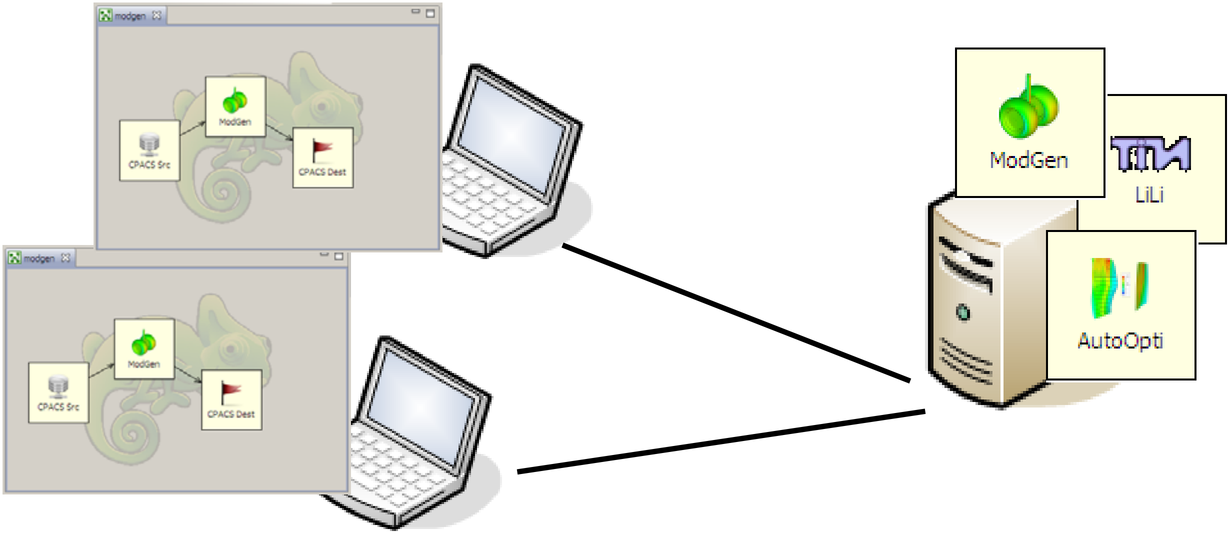
What Chameleon looks like on RCE



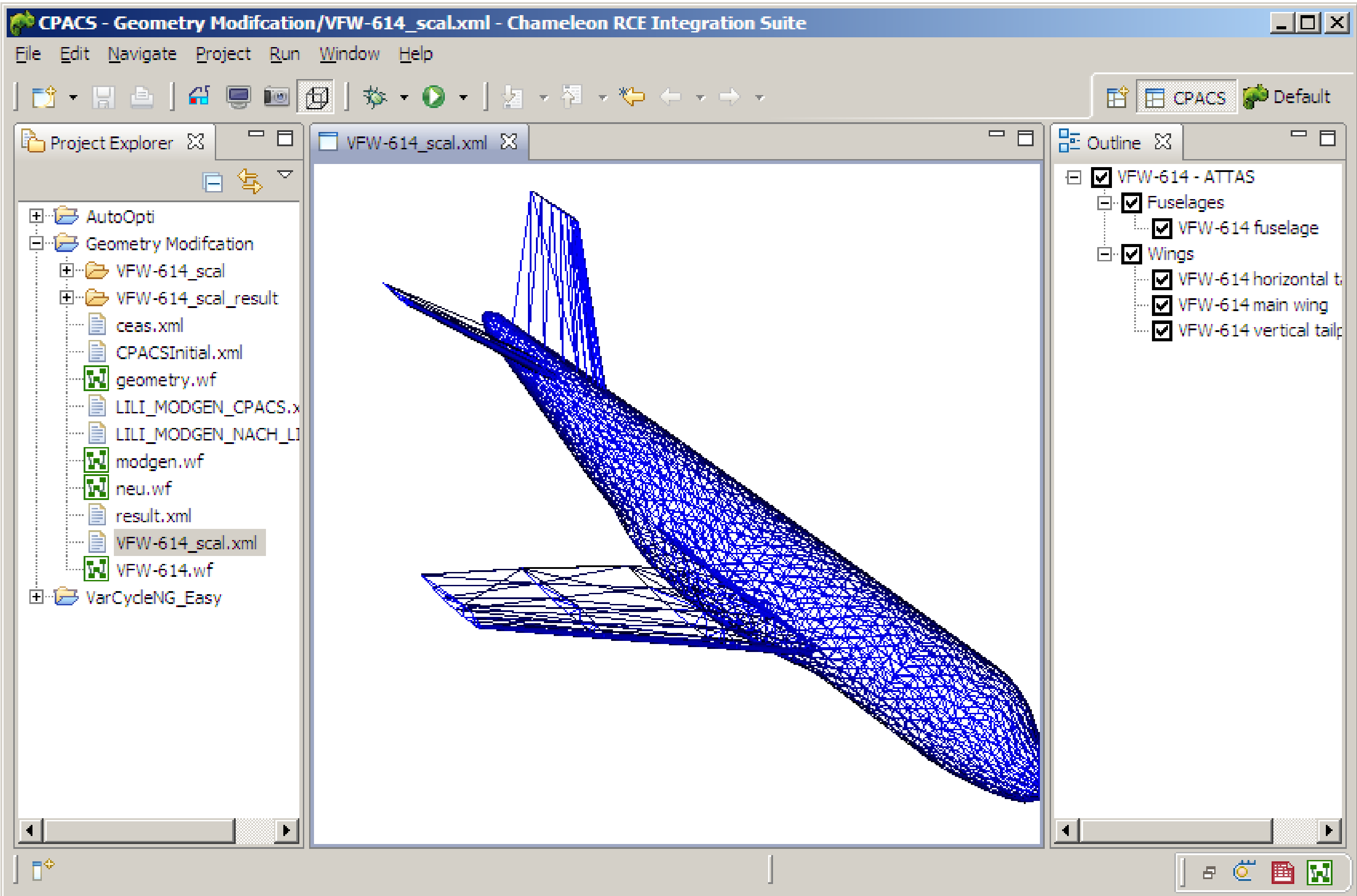
Name	Status	Platform	User	Time	Addit
Geometry	FINISHED	kpsc011n.intra.dlr.de:1	CN=Chief Engineer,O=RC...	2010.04.17 19:47:20	
Drei	FINISHED	kpsc011n.intra.dlr.de:1	CN=Chief Engineer,O=RC...	2010.04.17 19:35:43	
Geometry	FINISHED	kpsc011n.intra.dlr.de:1	CN=Chief Engineer,O=RC...	2010.04.17 19:49:43	
Drei	CANCELED	kpsc011n.intra.dlr.de:1	CN=Chief Engineer,O=RC...	2010.04.17 19:35:02	
Geometry	FINISHED	kpsc011n.intra.dlr.de:1	CN=Chief Engineer,O=RC...	2010.04.17 19:48:48	

Detachable workflows

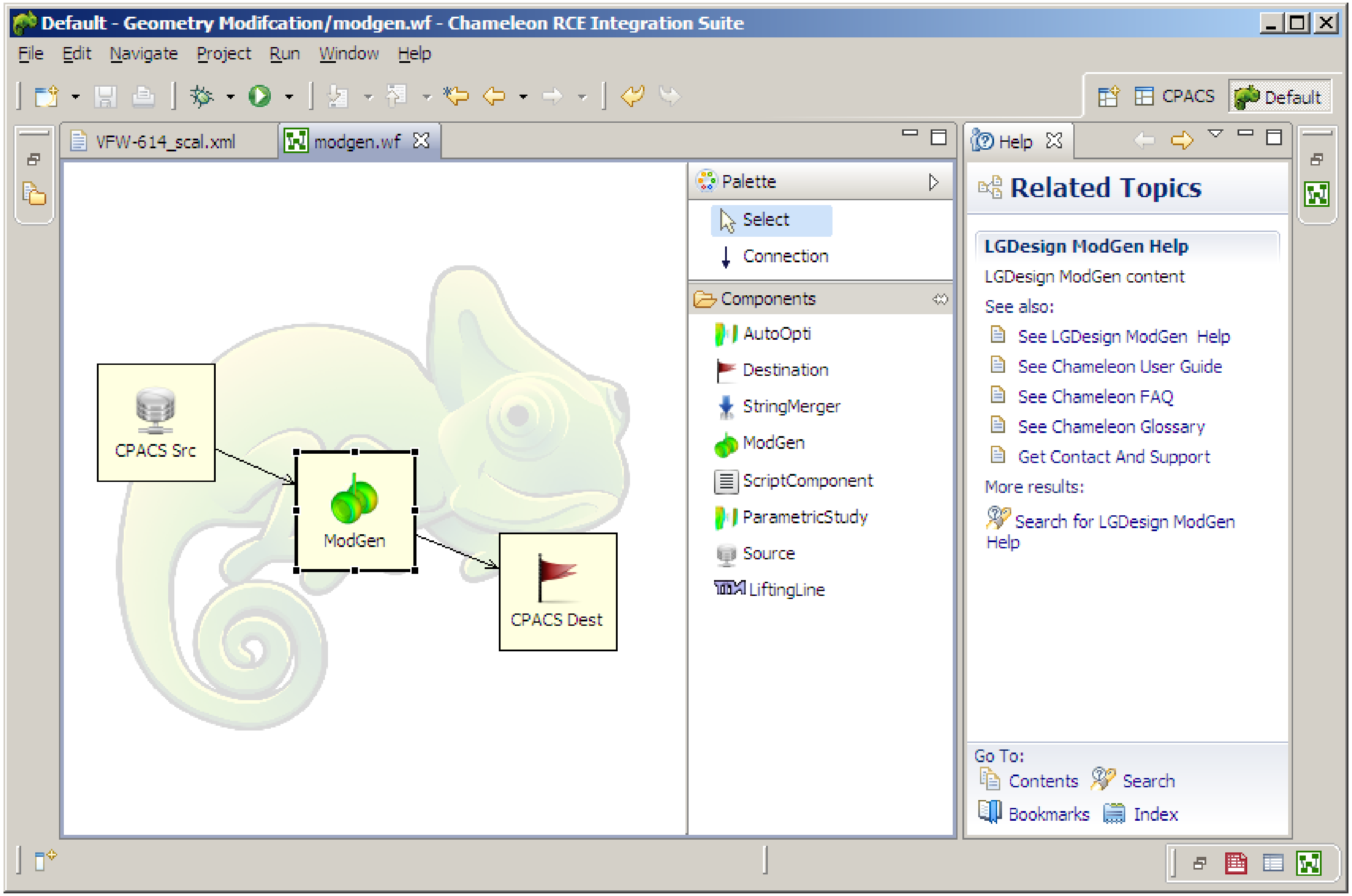
- Start a workflow on a remote server, configure locally
- Detach client (shut down computer)
- Re-attach (or attach other client) and monitor remote workflow



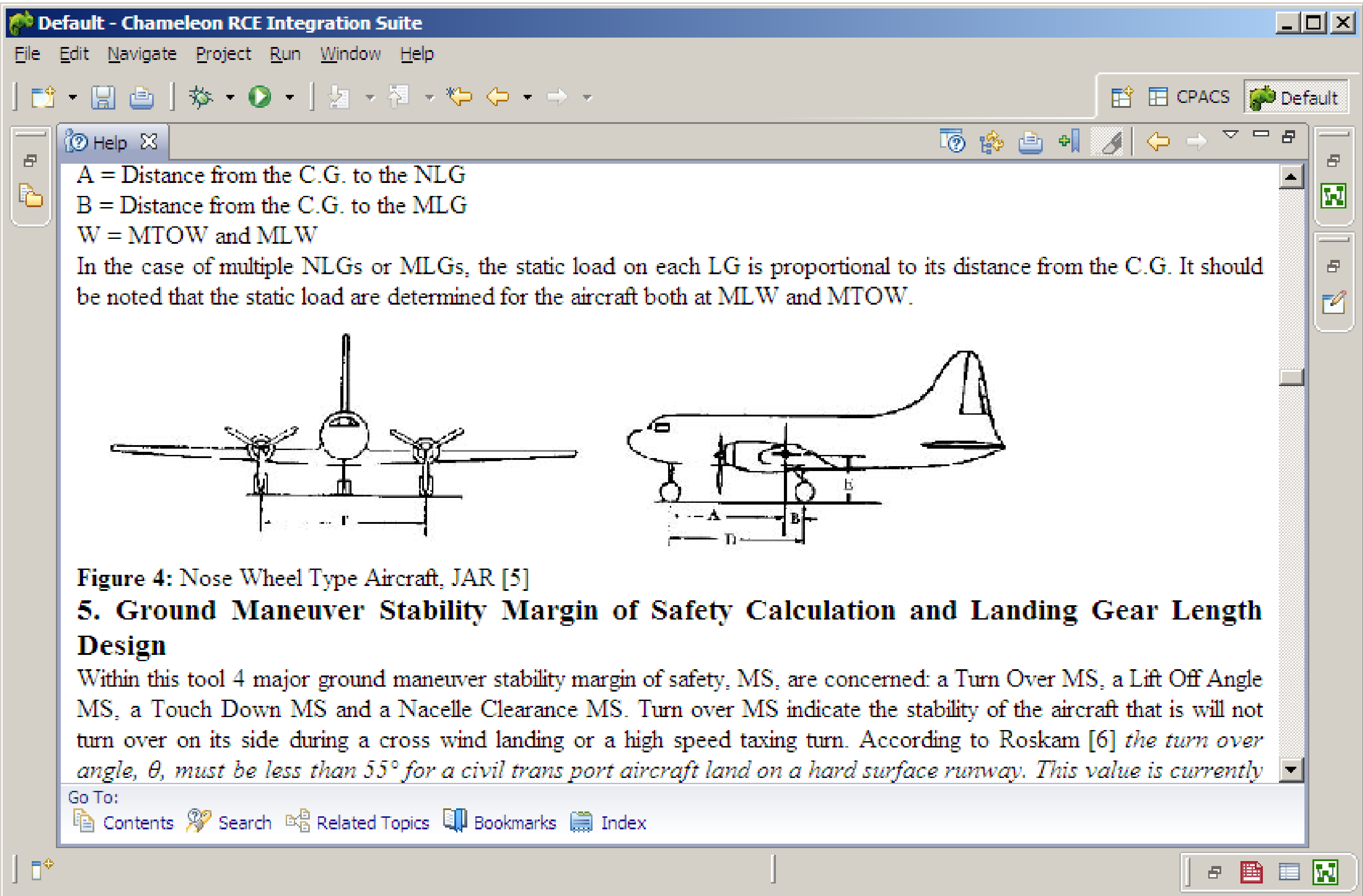
Example: Integrated viewer component for CPACS



Example: Dynamic help capabilities in action



Example: Dynamic help capabilities in action



A = Distance from the C.G. to the NLG
B = Distance from the C.G. to the MLG
W = MTOW and MLW
In the case of multiple NLGs or MLGs, the static load on each LG is proportional to its distance from the C.G. It should be noted that the static load are determined for the aircraft both at MLW and MTOW.

Figure 4: Nose Wheel Type Aircraft, JAR [5]

5. Ground Maneuver Stability Margin of Safety Calculation and Landing Gear Length Design

Within this tool 4 major ground maneuver stability margin of safety, MS, are concerned: a Turn Over MS, a Lift Off Angle MS, a Touch Down MS and a Nacelle Clearance MS. Turn over MS indicate the stability of the aircraft that is will not turn over on its side during a cross wind landing or a high speed taxing turn. According to Roskam [6] *the turn over angle, θ, must be less than 55° for a civil trans port aircraft land on a hard surface runway. This value is currently*

What we wanted to do after PDE 2008

- Define a data format, current version is 1.2
- Integrate tools via CPACS with each other (15+ tools)
- Set up a workflow system (replaced central controller)
- Integrate mission control data into CPACS (since 1.0)
- Geometry modeling for other aircraft parts (e.g. engine nacelle)
- Use our environment in other projects (5+ projects)

- … So yeah we did that.
- But: We started many more interesting research and development tasks!

What‘s hot?

- Böhnke, Daniel (2009): *Data integration in preliminary Airplane Design.* Master Thesis. Cologne.
  - Model-based conversion of CPACS data to STEP/Express.

- Böhnke, Daniel, Litz, Markus, Nagel, Björn , Rudolph, Stephan (2010) *Evaluation of Modeling Languages for Preliminary Airplane Design in Multidisiciplinary Design Environments.* DGLR Congress 2010. Hamburg

- König, Stefan (2010): *Untitled.* Master Thesis. Cologne.
  - Validation CPACS data in the Chameleon integration environment

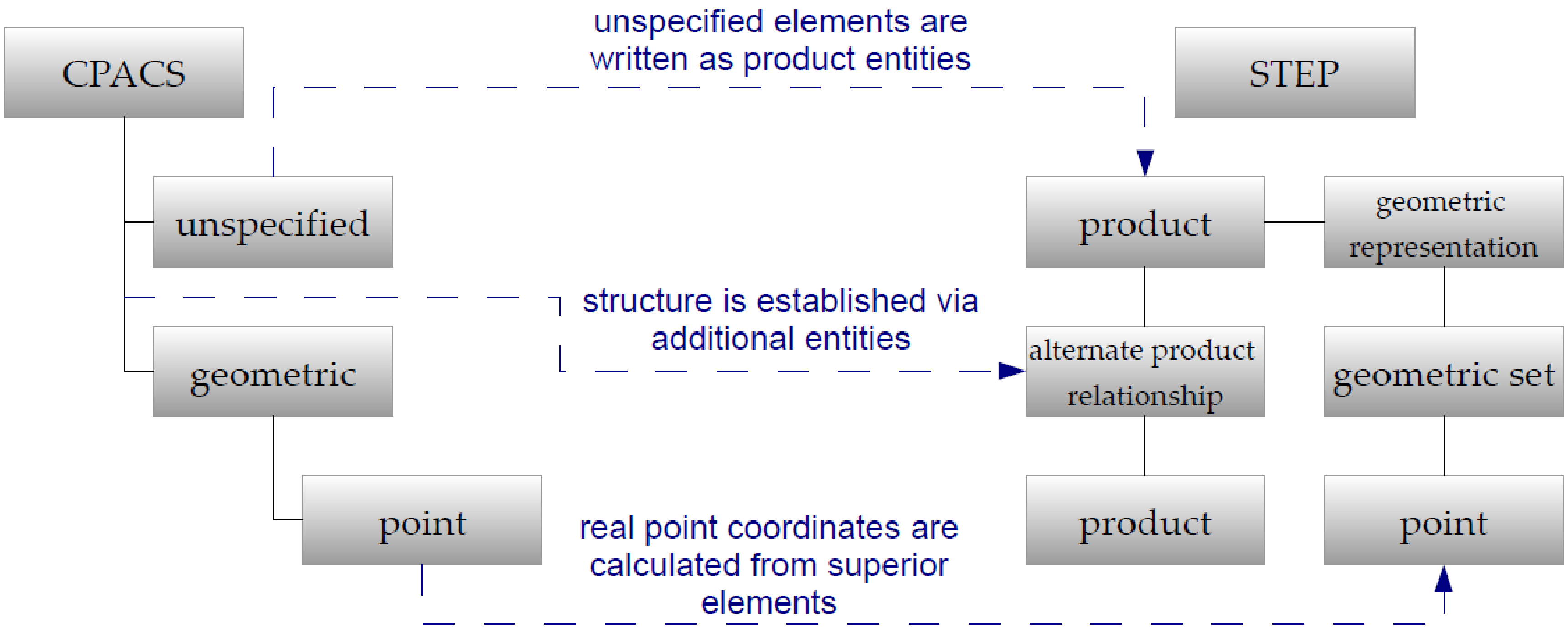
Modeling for conversion from CPACS to STEP

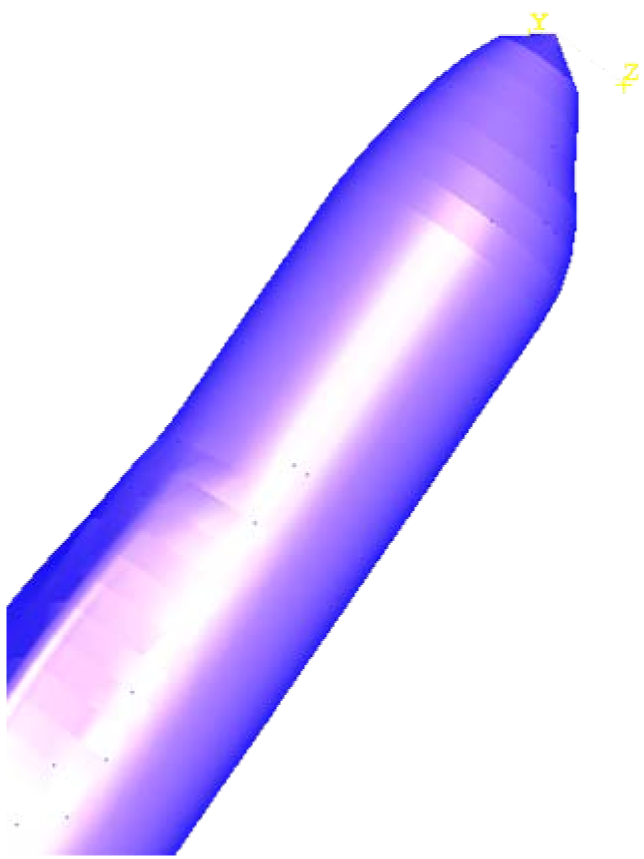
CPACS is the standard for holistic preliminary airplane design in the DLR

STEP is the standard for the exchange of information in the Industry

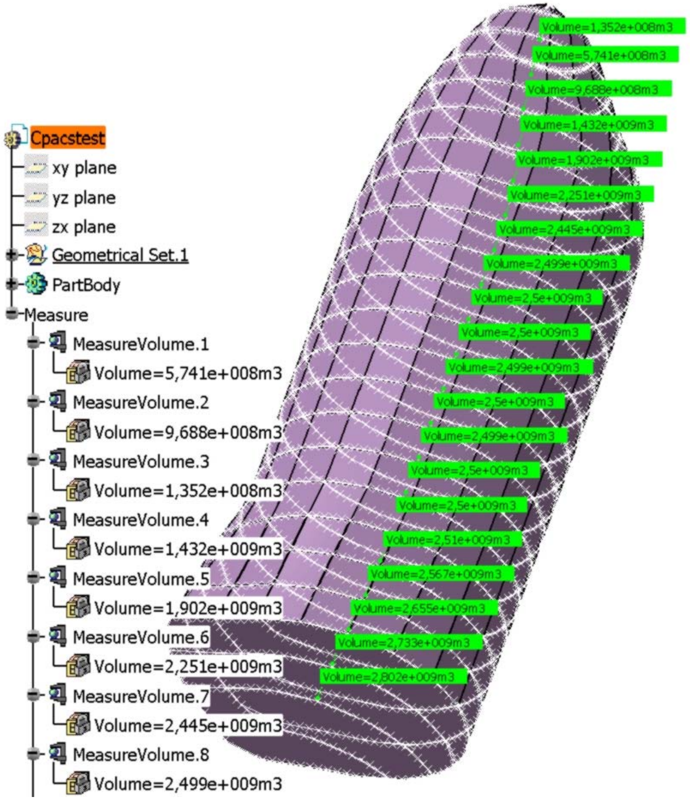
➡ **We need a CPACS to STEP conversion!**

Prototype using JAVA, JSDAI und JAXB





CPACS in TIGL



STEP in CATIA V5

CPACS validation

- Requirements:
  - Capability to validate CPACS data
  - Capability to verify dependencies within CPACS
  - Preferably during runtime
  - Demand for a DLR-wide concept

CPACS validation

- Possible solutions:
  - System immanent concepts
    - Documentation
    - Online help platforms
    - Semantic XML
  - System transcendental concepts
    - Integrated value checking
    - Tool-based semantics
    - Expert system

CPACS validation

- System immanent concepts
  - Integrated value checking: ModelCenter is capable of syntactically validating input parameters.
  - Tool-based semantics: For each tool in a workflow a validation checking instance could be integrated into the wrapping unit, the tool itself or the used library (TIGL/TIXI).
  - Expert system: A knowledge based database system which helps the user to link different tools and test various ranges of values.

CPACS validation

- System transcendental concepts
  - Documentation: DLR-wide documents describing test facts and their results
  - Online help platforms: wiki-like online platforms providing test facts
  - Semantic XML: external tools that check CPACS' XML file for invalid and inconsistent data (e.g. Schematron, XCSL or XML-Schema). Also there's the need to investigate the use of an ontology in combination with ontology-based query languages (e.g. OWL, RDF).

Thanks!

http://www.walle-derfilm.de/

Kontakt

Arne Bachmann
Abteilung Verteilte Systeme und
Komponentensoftware (SC-VK)
DLR Simulations- und Softwaretechnik
Köln-Porz / Braunschweig / Berlin
E-Mail: Arne.Bachmann@dlr.de
www.dlr.de/sc/abteilung/verteiltesysteme