# A Collaborative Environment for Flexible Development of MBS Software

**Manuel J. González Castro**

A thesis submitted for the degree of
Doctor Ingeniero Industrial

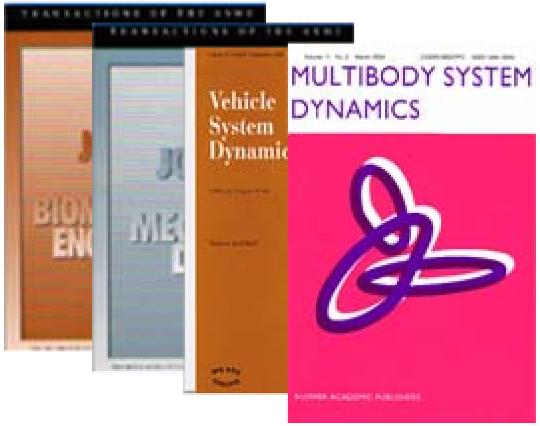
University of A Coruña

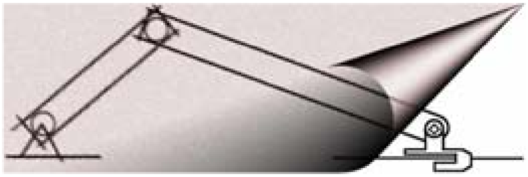
Ferrol, April 2005

# Outline

# Introduction

- MBS dynamics is an active research subject:
  - Many journal papers per year
  - Increasing number of conferences
- Many researchers working on open fields
- Development of new simulation methods:
  - Increase performance for real time
  - Handle complex non-linear aspects (contact-impact, friction, …)

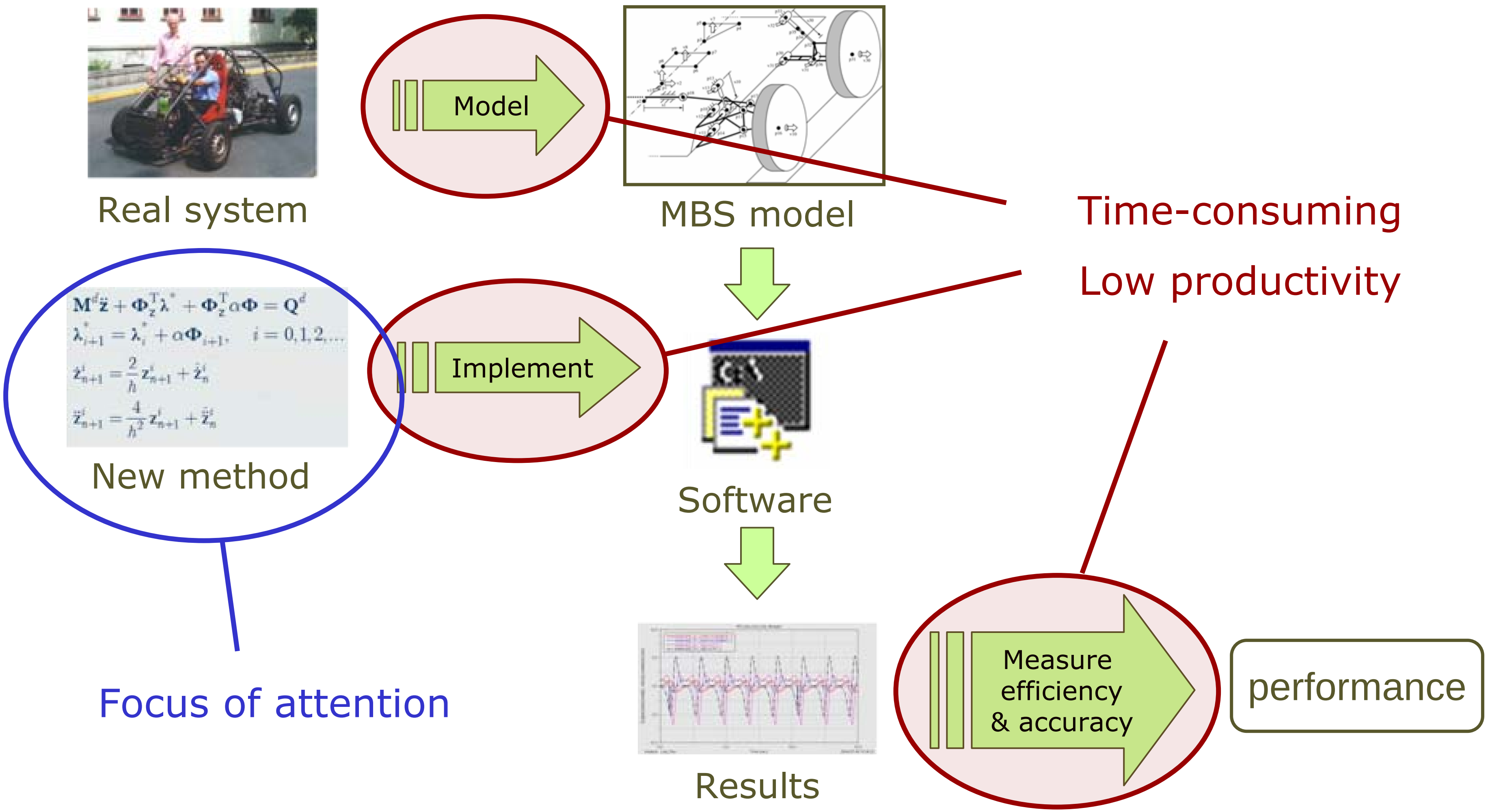


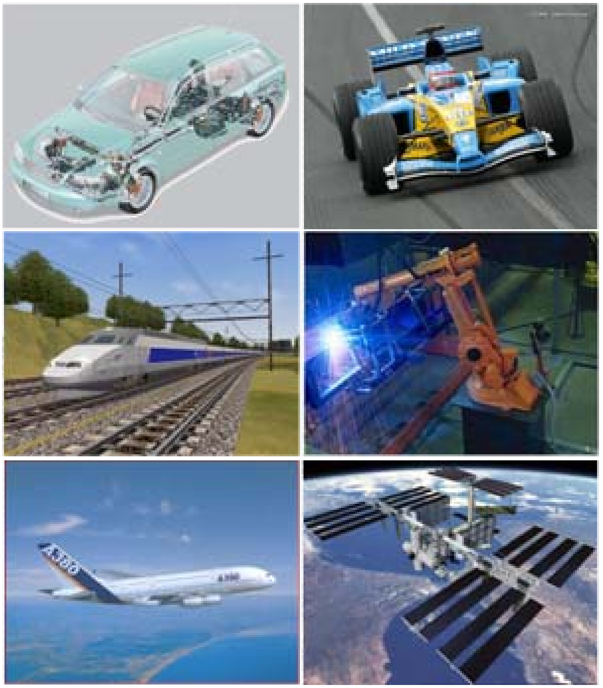
ECCOMAS Thematic Conference
Multibody Dynamics 2005
Madrid, June 2005

ASME

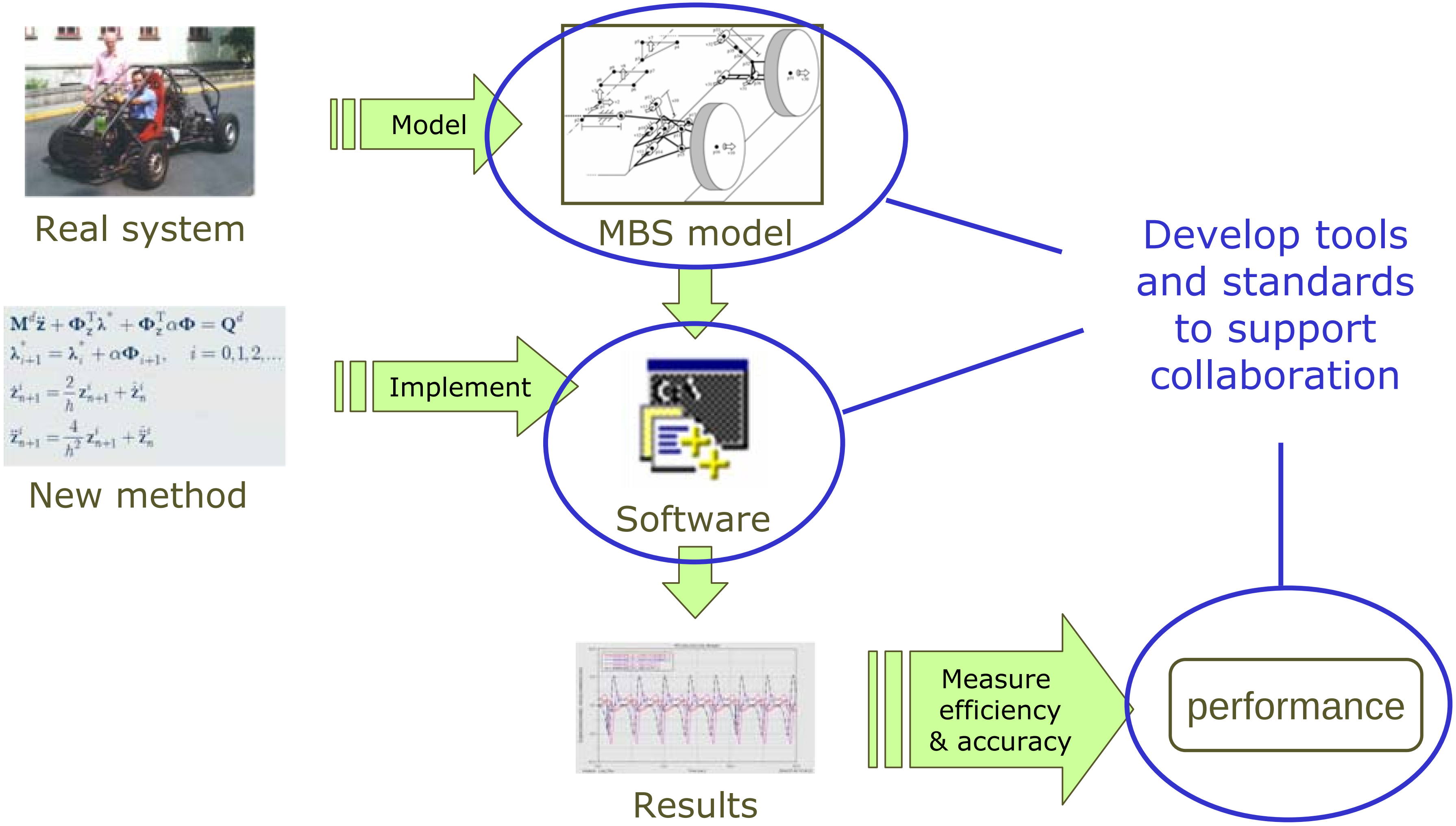
Fifth ASME International Conference on
Multibody Systems, Nonlinear Dynamics and Control
Long Beach, September 2005


Real system
Model
MBS model
Time-consuming
Low productivity
New method
Implement
Software
Focus of attention
Results
Measure
efficiency
& accuracy
performance

**Research MBS needs tools that support <u>collaboration</u>**



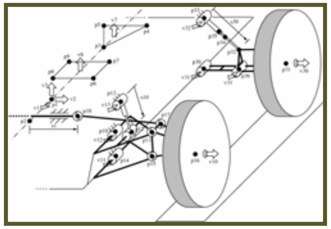
- Needed by scientific research:
  - Avoid duplication of efforts
  - Streamline research
- Needed by industry:
  - Products are very complex
  - Several teams must work together
- Needed by governments:
  - Requisite for some funding instruments


Real system
Model
MBS model
New method
Implement
Software
Develop tools and standards to support collaboration
Results
Measure efficiency & accuracy
performance

# Data models

Real system

Model



MBS model

$$M^d\ddot{z} + \Phi_z^T\lambda^* + \Phi_z^T\alpha\Phi = Q^d$$
$$\lambda^*_{i+1} = \lambda^*_i + \alpha\Phi_{i+1}, \quad i = 0,1,2,\ldots$$
$$\dot{z}_{n+1} = \frac{2}{h}z_{n+1} + \hat{\dot{z}}_n$$
$$\ddot{z}_{n+1} = \frac{4}{h^2}z_{n+1} + \hat{\ddot{z}}_n$$

New method

Implement

Software



Results

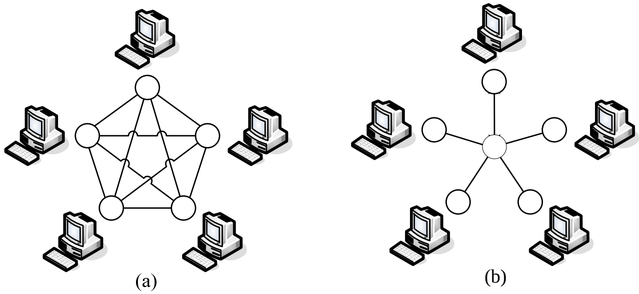
Measure efficiency & accuracy

performance

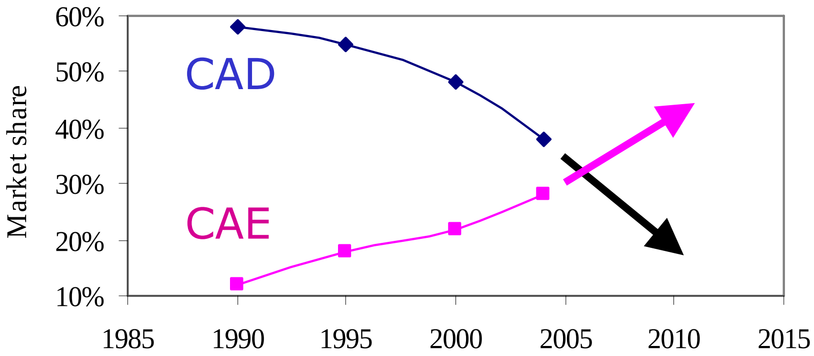
1. Introduction
2. **Data models**
3. Benchmarking
4. Simulation software
5. Conclusions

- Neutral data models are essential to exchange engineering product data
  - Avoid interoperability costs

- **There is no neutral data format for MBS**
  - Few users, one market leader
  - Interoperability costs are low




- The situation is changing
  - MBS community must address the problem as soon as possible




Evolution of CAD and CAE market shares

# Data models

- Engineering Product data
  - **STEP** (ISO 10303) is the current standard
  - Solves the exchange of CAD data
  - Currently being extended to CAE data: FEA, CFD, electronics, …
- Multibody systems
  - German standardization efforts in the 1990s: DAMOS-C, MechaSTEP
  - Commercial software uses **proprietary data formats**
- **XML** (eXtensible Markup Language)
  - Emerging technology, very successful in other fields
  - Very easy to use

## Data models

# Evaluation of commercial software

| Feature \ Preprocessor | | ADAMS v.2003 | SYMPACK v.8.6 | DADS v.9.6 | RecurDyn v.5.2 |
|---|---|---|---|---|---|
| Model database format | Primary | Binary file | Text files | Binary file | Binary file |
| | Secondary | Text file (.adm or .cmd) | - | Text file | - |
| Imports MBS models in other formats | | No | No | ADAMS (.adm) | ADAMS (.adm and .cmd) |
| Exports MBS models in other formats | | No | No | No | No |
| Formalism-independent modeling | | Yes | No | Yes | Yes |
| Sub-models | | No | Yes (only 1 level) | No | Yes (needs special preprocessing) |
| Units systems | | MLT based | Not MLT based | MLT based | MLT based |
| Units scope | | Global | Global | Global | Global |
| Parametric models | | Yes | Yes | Yes | Yes |

# Evaluation of STEP and XML

| | Feature | STEP | XML |
|---|---|---|---|
| Design | Support for object-oriented modeling | ••• | • |
| | Support for rule and constraint definition | ••• | •• |
| | Support for hierarchical structures in the data model | • | ••• |
| | Modeling language easy to learn | •• | ••• |
| | Availability of data models for product data | ••• | • |
| | Availability of data models for MBS | •• | • |
| | Modular and configurable data models | • | •• |
| Implementation | Code generation | ••• | • |
| | Human-readable physical file format | •• | ••• |
| | Support for hierarchical structures in the file format | • | ••• |
| | Quantity and quality of available supporting tools | • | ••• |
| | Cost of available supporting tools | • | ••• |
| | Quantity and quality of available documentation | •• | ••• |
| | Cost of available supporting documentation | • | ••• |

• good    •• medium    ••• good

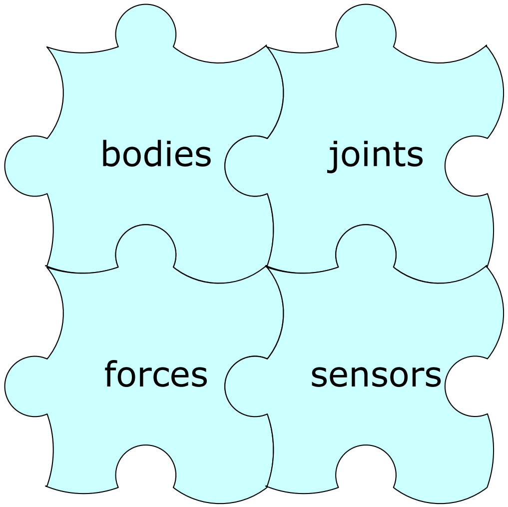
# XML-based data model

- Information is decoupled to facilitate data reuse:
  - Model
  - Analysis
  - Method

General structure

Pendulum

- Support for:
  - Sub-models
  - Units of measure
  - Parametric models
- Modeling language with a modular design
  - Easily extensible and configurable

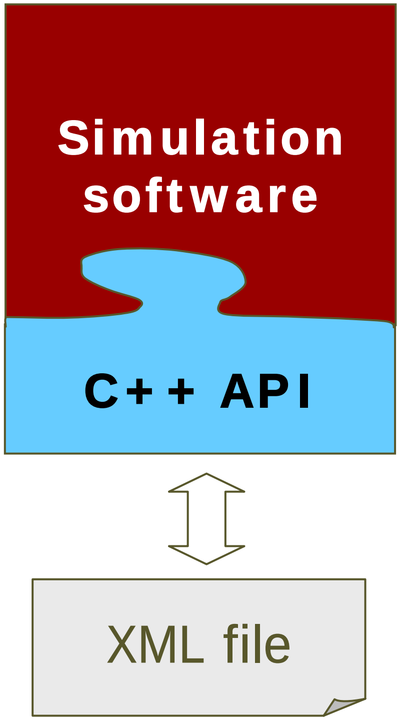


# Application Program Interface (API)

- C++ programming library to read/write XML data files
  - Simplifies file processing
  - Object-oriented

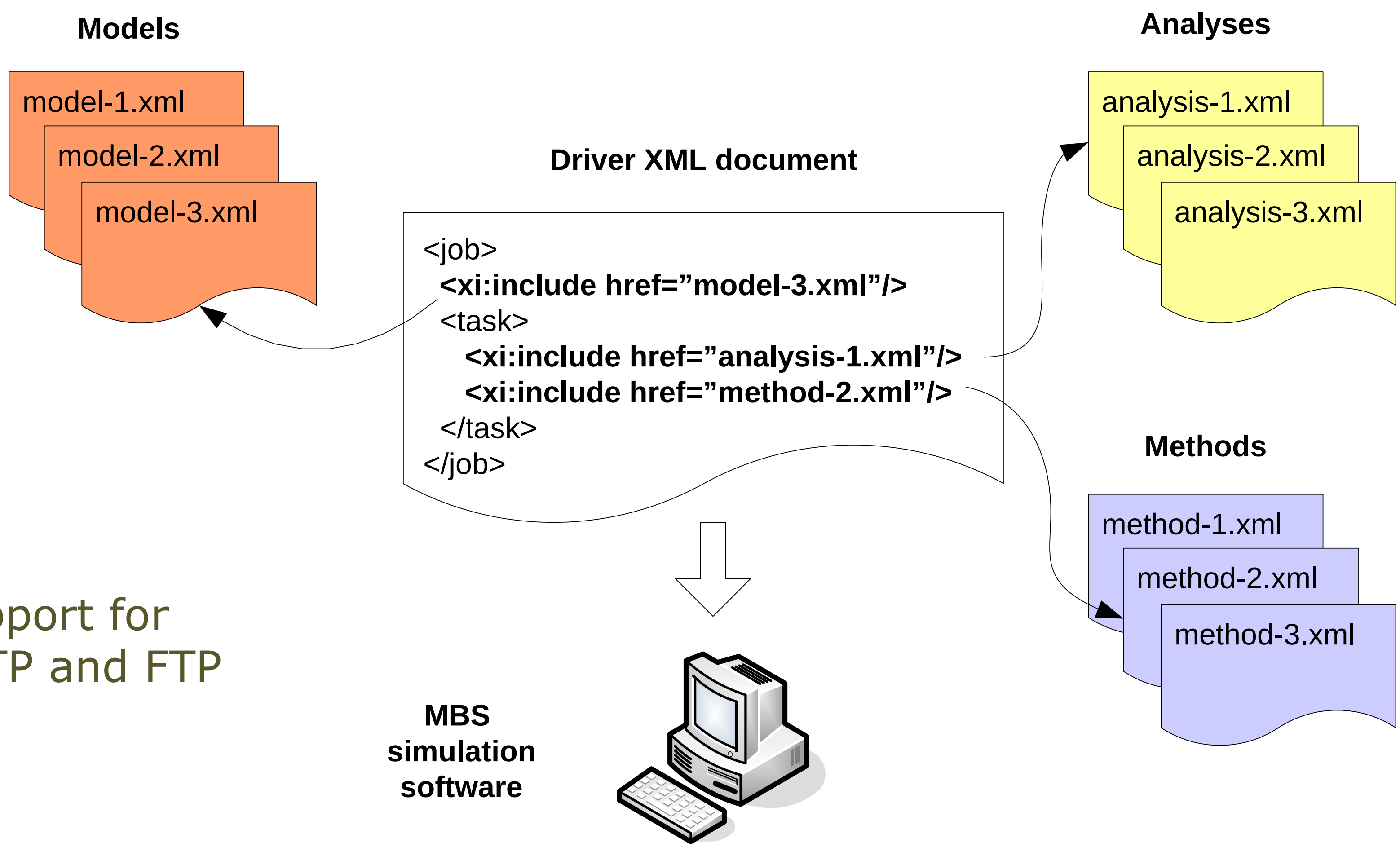
```
// create empty simulation job
Job job;

// read XML file
XmlReader reader("doc.xml");
reader.read(job);

// examine job content
Model* m = job->getModel();
// …
```

C++ for reading an XML simulation job



# Data models

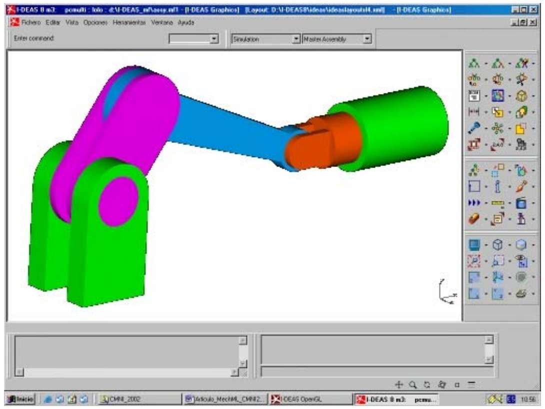
## Including system

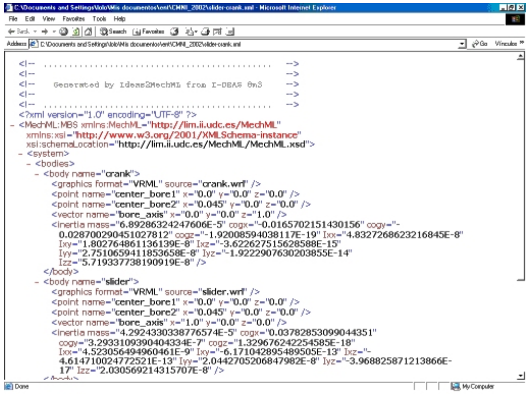



Support for
HTTP and FTP

# Data models

## Automatic generation of XML files





- Plug-in for I-DEAS (CAD/CAE/CAM system)
- The MBS is modeled in the pre-procesor
- The corresponding XML model is exported
- Due to I-DEAS limitations, joints and forces cannot be exported
  - Not useful
  - Serves as proof of concept of the idea

- Evaluation of commercial MBS software
  - Poor interoperability
  - Commercial data formats do not support collaboration
- Evaluation of STEP and XML as neutral data formats for multibody systems
  - STEP has better capabilities for design
  - XML seems to be much more easier to implement
- Prototype implementation of an XML-based data format for MBS
  - Simple yet powerful
  - Excellent capabilities for data exchange and reuse
  - XML proved to be a powerful, cheap and easy-to-use technology

# Data models

- STEP still has some important advantages
  - Large library of models (CAD, FEA, …)
  - Data models are more robust

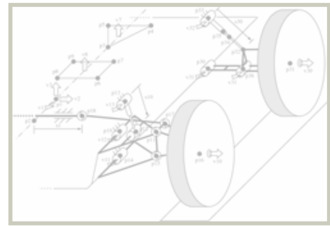
**An industrial- strength data model for MBS must use both STEP and XML**

- Some international efforts to merge STEP and XML are under progress
  - Apply them to MBS
  - Very interesting and promising field
  - Needs cooperation at international level

# Benchmarking

Model

Real system



MBS model

$$\mathbf{M}^d\ddot{\mathbf{z}} + \mathbf{\Phi}_z^T\lambda^* + \mathbf{\Phi}_z^T\alpha\mathbf{\Phi} = \mathbf{Q}^d$$

$$\lambda^*_{i+1} = \lambda^*_i + \alpha\mathbf{\Phi}_{i+1}, \quad i = 0,1,2,\ldots$$

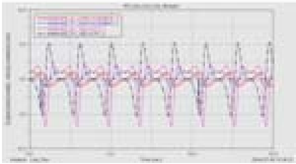
$$\dot{\mathbf{z}}^i_{n+1} = \frac{2}{h}\mathbf{z}^i_{n+1} + \hat{\dot{\mathbf{z}}}^i_n$$

$$\ddot{\mathbf{z}}^i_{n+1} = \frac{4}{h^2}\mathbf{z}^i_{n+1} + \hat{\ddot{\mathbf{z}}}^i_n$$

New method

Implement

Software



Results

Measure efficiency & accuracy

performance

# Benchmarking



- MBS researchers:

  *"I have developed a new simulation method.*

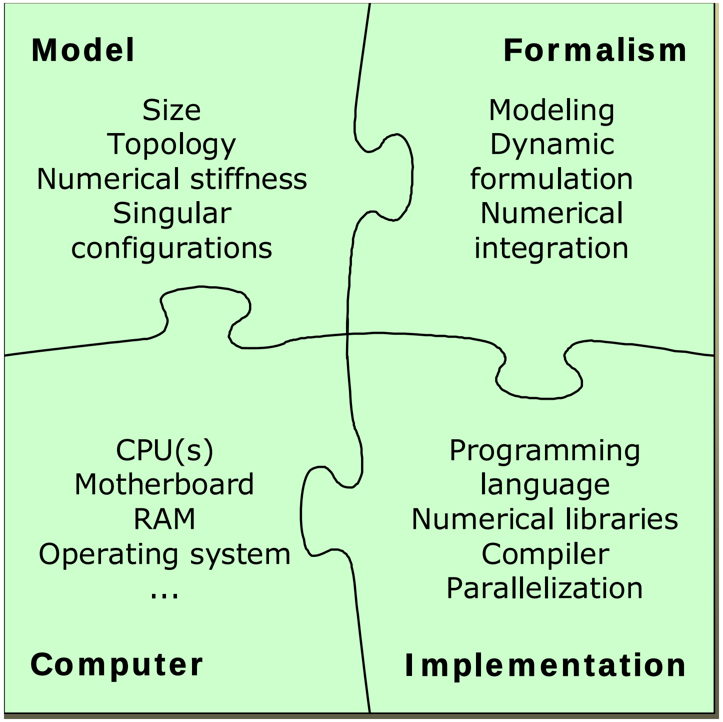
  *How good is it compared with others?"*

- MBS users:

  *"I need to simulate this system.*

  *Which method should I use?"*

- No easy answers
  - Efficiency depends on many inter-related factors
- Researchers report performance using different:
  - Models and analysis conditions
  - Accuracy in the solution
- Results are scattered and difficult to collect



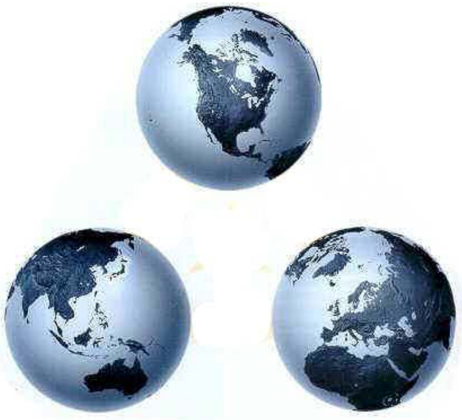

| Time-step | CPU time |
|---|---|
| 0.01 | 20 |
| 0.001 | 45 |

# Benchmarking



- System to measure performance:
  - Standard problem collection
  - Reference solutions
  - Clear procedure to measure efficiency

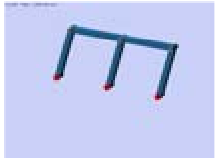
- System to share performance measures
  - Collect, organize and share information
  - Centralized
  - Public
  - WWW seems very appropriate

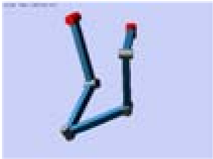


- Each problem describes the model, the analysis and measured coordinates

- Divided in categories
  - "Basic problems"
    - Small, isolate a particular characteristic
    - Need little time investment (important for a *standard* benchmark)
  - "Industrial applications"
    - Complex, real-life problems
    - Involving several complex phenomena together
    - Demonstrations for industry

# Problems in group A





| Code | Name | Characteristic |
|---|---|---|
| A00 | Double pendulum (2D) | Example, didactic problem |
| A01 | Double pendulum (3D) | High accelerations |
| A02 | Double four bar mechanism | Singular positions |
| A03 | Andrew’s mechanism | Very small time scale |
| A04 | Bricard’s mechanism | Redundant equations |
| A05 | Bicycle with rear suspension | Stiff system |

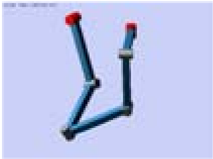
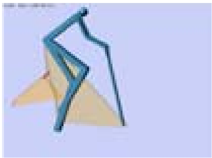
## Basic problems for rigid MBS

# Benchmarking

## Problems in group A





| Code | Name | Characteristic |
|---|---|---|
| **A00** | **Double pendulum (2D)** | **Example, didactic problem** |
| A01 | Double pendulum (3D) | High accelerations |
| A02 | Double four bar mechanism | Singular positions |
| A03 | Andrew's mechanism | Very small time scale |
| A04 | Bricard's mechanism | Redundant equations |
| A05 | Bicycle with rear suspension | Stiff system |

2 d.o.f.

Only gravity effects

Duration: 15 s

Measure end point coordinates

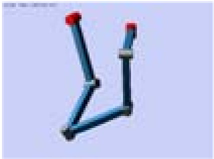
Example problem

# Benchmarking

## Problems in group A








| Code | Name | Characteristic |
|---|---|---|
| A00 | Double pendulum (2D) | Example, didactic problem |
| **A01** | **Double pendulum (3D)** | **High accelerations** |
| A02 | Double four bar mechanism | Singular positions |
| A03 | Andrew's mechanism | Very small time scale |
| A04 | Bricard's mechanism | Redundant equations |
| A05 | Bicycle with rear suspension | Stiff system |

6 d.o.f.

Gravity effects

Duration: 15 s

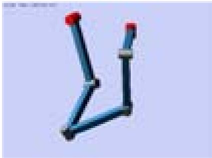
Measure end point coordinates

High accelerations
(chaotic movement)

Needs very accurate methods

# Benchmarking

## Problems in group A



| Code | Name | Characteristic |
|---|---|---|
| A00 | Double pendulum (2D) | Example, didactic problem |
| A01 | Double pendulum (3D) | High accelerations |
| **A02** | **Double four bar mechanism** | **Singular positions** |
| A03 | Andrew's mechanism | Very small time scale |
| A04 | Bricard's mechanism | Redundant equations |
| A05 | Bicycle with rear suspension | Stiff system |

1 d.o.f.

Gravity effects

Duration: 15 s

Measure coordinates of p1

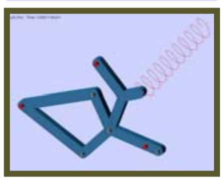
Singular configuration at horizontal position: 3 d.o.f.

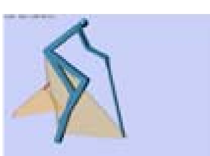
*Bayo and Avello, 1994*

# Benchmarking

## Problems in group A





| Code | Name | Characteristic |
|---|---|---|
| A00 | Double pendulum (2D) | Example, didactic problem |
| A01 | Double pendulum (3D) | High accelerations |
| A02 | Double four bar mechanism | Singular positions |
| **A03** | **Andrew's mechanism** | **Very small time scale** |
| A04 | Bricard's mechanism | Redundant equations |
| A05 | Bicycle with rear suspension | Stiff system |

1 d.o.f.

Applied torque

Duration: 0.15 s

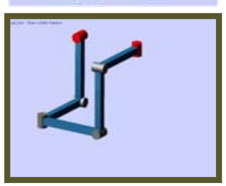
Measure coordinates of p4

Very small time scale

*Schiehlen, 1990*

# Benchmarking

## Problems in group A



| Code | Name | Characteristic |
|---|---|---|
| A00 | Double pendulum (2D) | Example, didactic problem |
| A01 | Double pendulum (3D) | High accelerations |
| A02 | Double four bar mechanism | Singular positions |
| A03 | Andrew's mechanism | Very small time scale |
| **A04** | **Bricard's mechanism** | **Redundant equations** |
| A05 | Bicycle with rear suspension | Stiff system |

1 d.o.f.

Gravity effects

Duration: 10 s

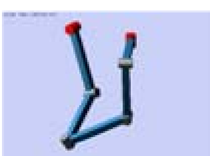
Measure coordinates of p3

Redundant equations (Grübler: 0 d.o.f.)

*García de Jalón & Bayo, 1994*

# Benchmarking

## Problems in group A



| Code | Name | Characteristic |
|---|---|---|
| A00 | Double pendulum (2D) | Example, didactic problem |
| A01 | Double pendulum (3D) | High accelerations |
| A02 | Double four bar mechanism | Singular positions |
| A03 | Andrew's mechanism | Very small time scale |
| A04 | Bricard's mechanism | Redundant equations |
| **A05** | **Bicycle with rear suspension** | **Stiff system** |

1 d.o.f.

Applied torque

Duration: 30 s

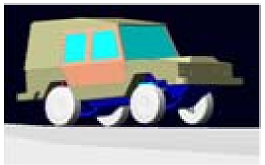
Measure coordinates of p1

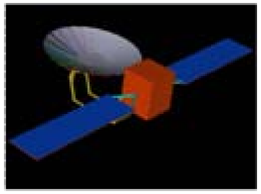
Stiff suspension spring

*Good & McPhee, 1999*

# Benchmarking

## Problems in group B



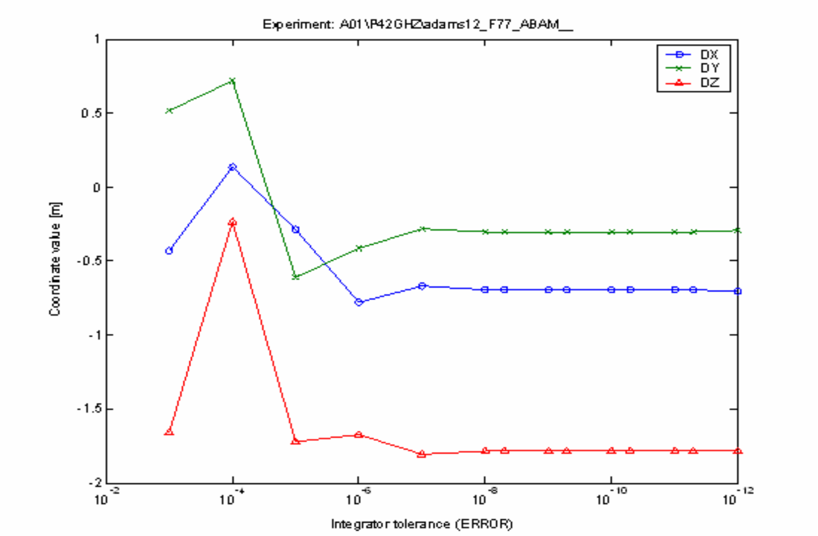


| Code | Name | Application |
|---|---|---|
| B01 | Iltis vehicle | Automotive |
| B02 | Dornier’s antenna | Aeroespace |
| B03 | Human body | Biomechanics |
| B04 | Puma robot | Robotics (serial) |
| B05 | Stewart platform | Robotics (parallel) |

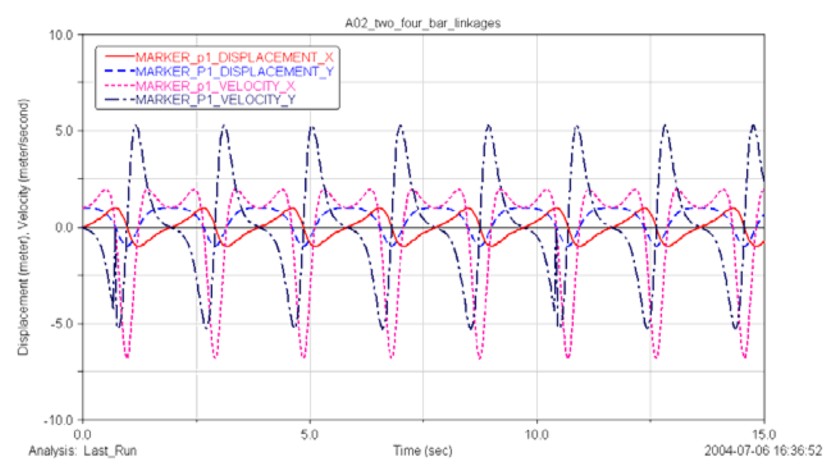
Industrial applications for rigid MBS

# Reference solutions – Group A

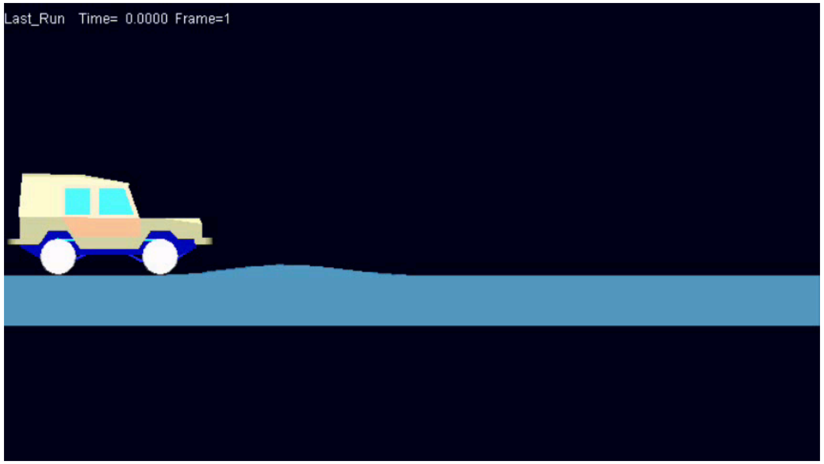
- Solved with ADAMS/Solver, some of them also with Matlab
- Different methods were used to ensure convergence to the right solution

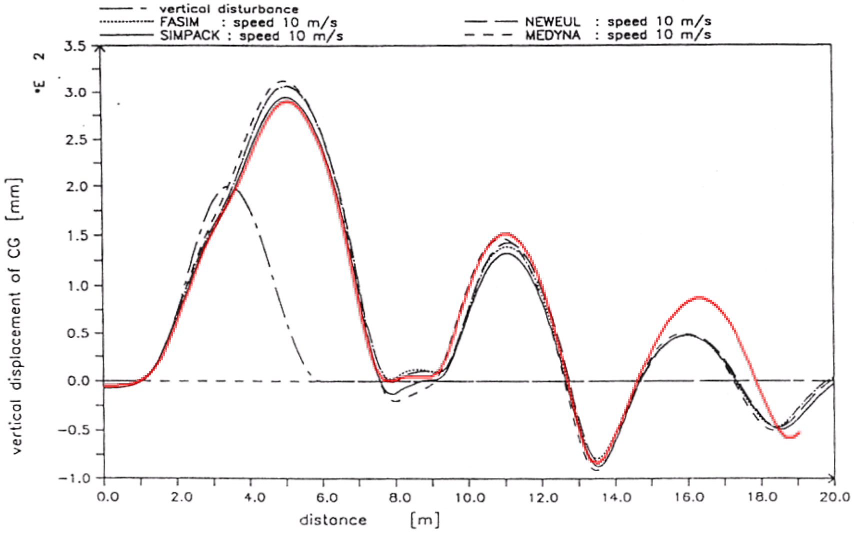



- The reference solution includes all the <u>time-history</u> of the measured coordinates



- Reference solutions difficult to find
- Example: Iltis benchmark
  - ADAMS solutions vs. published solutions (plots)
  - “Good” agreement, but…
  - Which one is the reference solution? The average?
- To be fair, more solvers should be used:
  - Simpack, Recurdyn, …



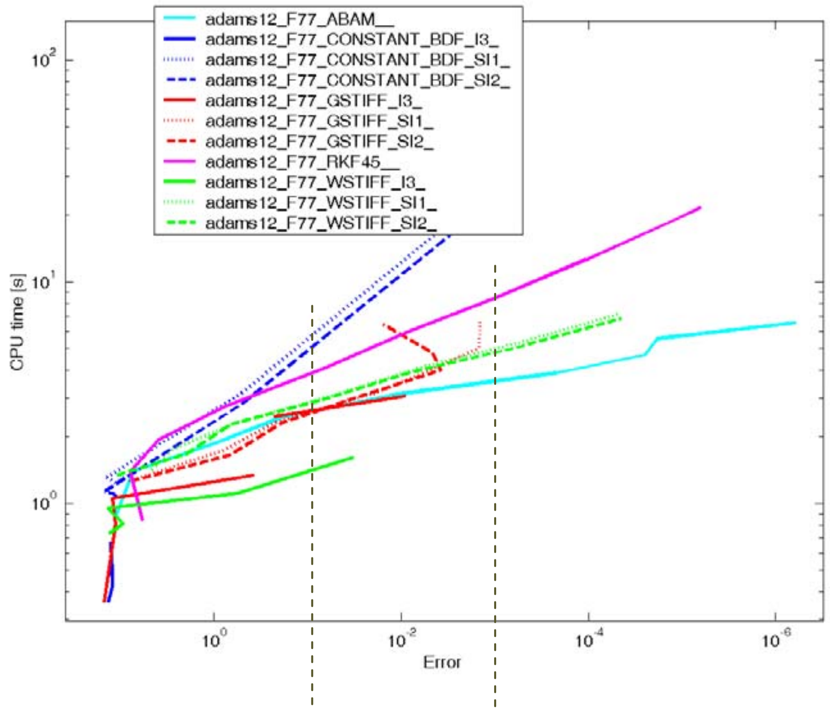
## Solve the problem as fast as possible within the required accuracy

- Accuracy is measured with L2-norm:

$$e_j(t_i) = \frac{\left| y_j(t_i) - y_j^{ref}(t_i) \right|}{y_j^w(t_i)}$$

$$e_{2,2} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\frac{1}{n}\sum_{j=1}^{n}\left(e_j(t_i)\right)^2}$$

- Reasonable error levels were determined from work-precision plots




e=0.1% (high)

e=10% (low)

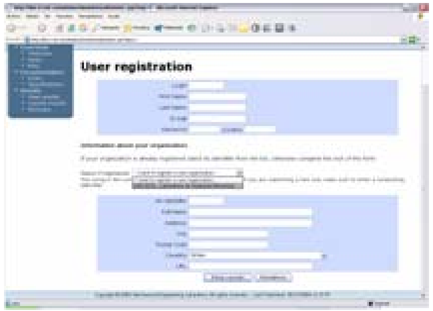
- Efficiency of a simulation is computed with the Software Performance Ratio (S.P.R.):

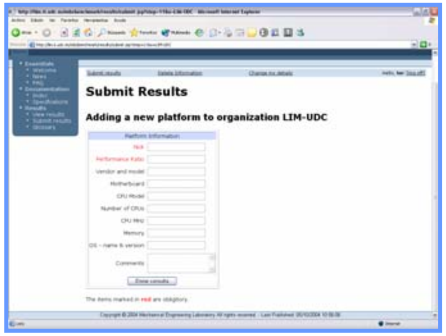
$$S.P.R._{test\ problem\ i} = \left(\frac{1}{H.P.R.}\right) \cdot \frac{simulation\ time_{test\ problem\ i}}{CPU\text{-}time_{test\ problem\ i}}$$

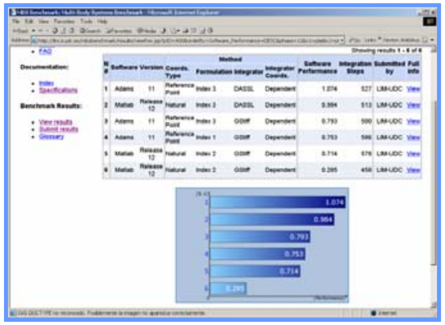
- Tries to remove dependency from:
  - Simulation duration
  - Computer

# Benchmarking

## Web site



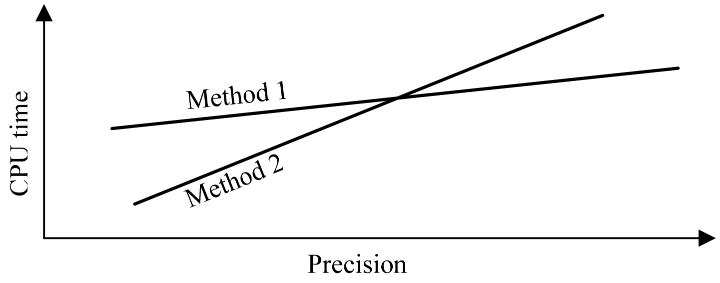



- Documentation
  - Specifications (HTML, PDF)
  - Reference solutions (numeric, plot, movie)
- Results submission
  - Only registered users (login required)
  - Detailed information about the simulator
  - Users can delete their results
- Results querying
  - Criteria and filters
  - HTML reports with graphic

http://lim.ii.udc.es/mbsbenchmark

- The benchmark has been applied to ADAMS/Solver
- Numerical experiments with different:
  - 11 simulation methods
  - 4 solver versions (release, programming language)
  - 2 computers (single-processors, dual-processor)
- Results:
  - Problem A05 is too easy
  - The rest of the problems are good benchmarks
  - The precision level is important



- Benchmark for MBS dynamics
  - Fully documented problems for rigid MBS
  - Simple procedure to measure efficiency
- Web-based system to manage performance data
  - Very useful to analyze information
  - Public, centralized, easy to use
- Application to a commercial software (ADAMS)
  - Validation of the proposed benchmark
  - Base-line results for future comparisons with other solvers

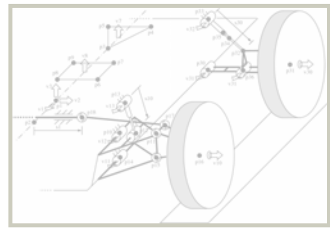
# Benchmarking

- Extend the problem collection
  - Find reference solutions for “Industrial applications”
  - Include other phenomena: flexibility, contact-impact, …
- Automate the benchmarking procedure
  - Useful to control quality of software releases
- Apply the benchmark to other simulation codes
  - Commercial
  - Academic

# Simulation software

Model

Real system



MBS model

$$\mathbf{M}^d\ddot{\mathbf{z}} + \mathbf{\Phi}_z^T\boldsymbol{\lambda}^* + \mathbf{\Phi}_z^T\alpha\mathbf{\Phi} = \mathbf{Q}^d$$

$$\boldsymbol{\lambda}^*_{i+1} = \boldsymbol{\lambda}^*_i + \alpha\mathbf{\Phi}_{i+1}, \quad i = 0,1,2,\ldots$$

$$\dot{\mathbf{z}}'_{n+1} = \frac{2}{h}\mathbf{z}'_{n+1} + \hat{\dot{\mathbf{z}}}'_n$$

$$\ddot{\mathbf{z}}'_{n+1} = \frac{4}{h^2}\mathbf{z}'_{n+1} + \hat{\ddot{\mathbf{z}}}'_n$$

New method

Implement

Software



Results

Measure efficiency & accuracy

performance

# Simulation software

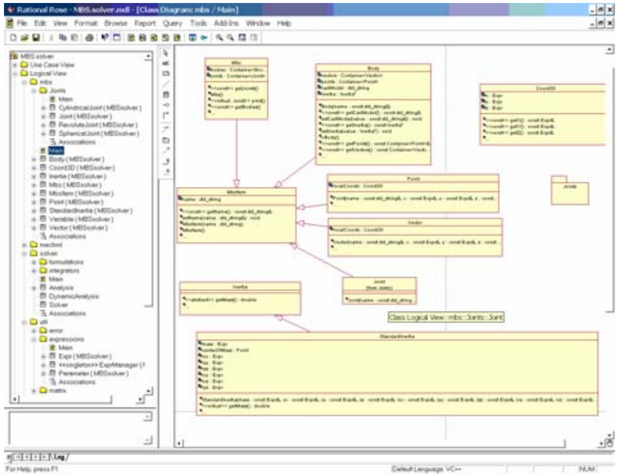
- Usually, engineers do not use software design techniques
  - Code developed ad-hoc to solve a particular problem
  - Bad programming style, code difficult to reuse
- General MBS simulation software can become very complex
  - Needs methods for software engineering
  - Needs collaboration between programmers

# Simulation software

- Design a general-purpose, generic MBS simulation software
  - Not tied to a particular formulation
  - Support for multiple simulation methods
  - Modular and extensible
- Select the right tools and techniques
  - Development environment
  - Programming language, numerical libraries, …
- Deploy the system and train colleagues

# Simulation software — Evaluation of CASE tools

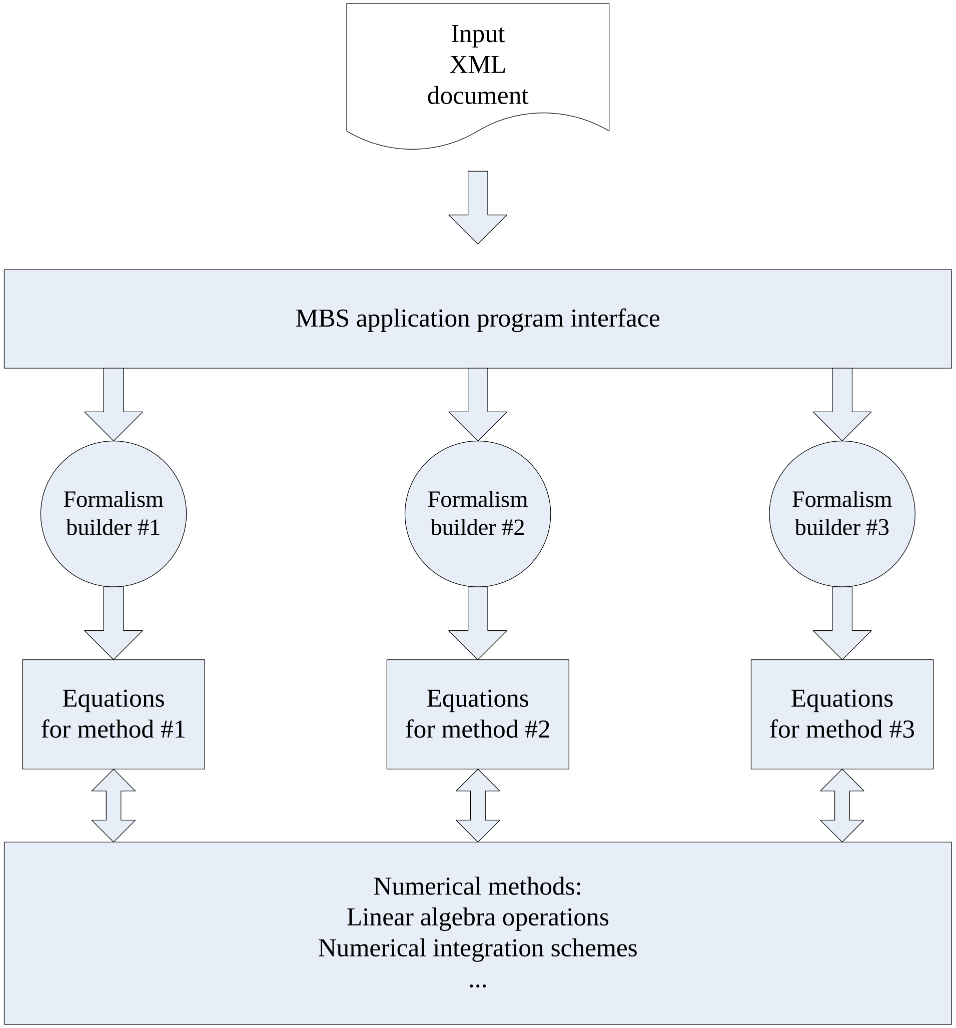
- CASE = Computer Aided Software Engineering
  - Code is generated from graphical models
- Evaluation of a commercial tool:
  - Very difficult, needs too customization
  - Not adequate for academic environments
- An open source project host is best suited:
  - Source control
  - Bug tracking, task management, …
  - After evaluation, Berlios was selected



IBM Rational Rose


Berlios

| | |
|---|---|
| License | Open Source (GPL) |
| Programming language | C++ / Fortran |
| Compiler | Visual Studio 7/ GNU GCC 3 |
| Source control system | SNV |
| Documentation tool | Doxygen |
| Design tool | Poseidon UML |
| Management | XPlanner |
| Project Host | Berlios |
| XML parser | Libxml |
| Visualization System | OSG / OpenGL Performer |
| Math kernel (BLAS) | ATLAS |


Input
XML
document
MBS application program interface
Formalism
builder #1
Formalism
builder #2
Formalism
builder #3
Equations
for method #1
Equations
for method #2
Equations
for method #3
Numerical methods:
Linear algebra operations
Numerical integration schemes
...

# Simulation software

- Design and skeleton of the program (“base classes”) is finished
- General facilities
  - XML input/output and friendly error reports
- Formalisms
  - Support only global formulations based on natural coordinates
  - Library of joints
  - Automatic constraint generation for joints
- Numerical methods (solvers)
  - Matrix class library to wrap different:
    - Data structures: dense and sparse
    - Linear solvers (TAU, PHIPHACS, Harwell library, …)
- Generic interface for integrators

# Conclusions

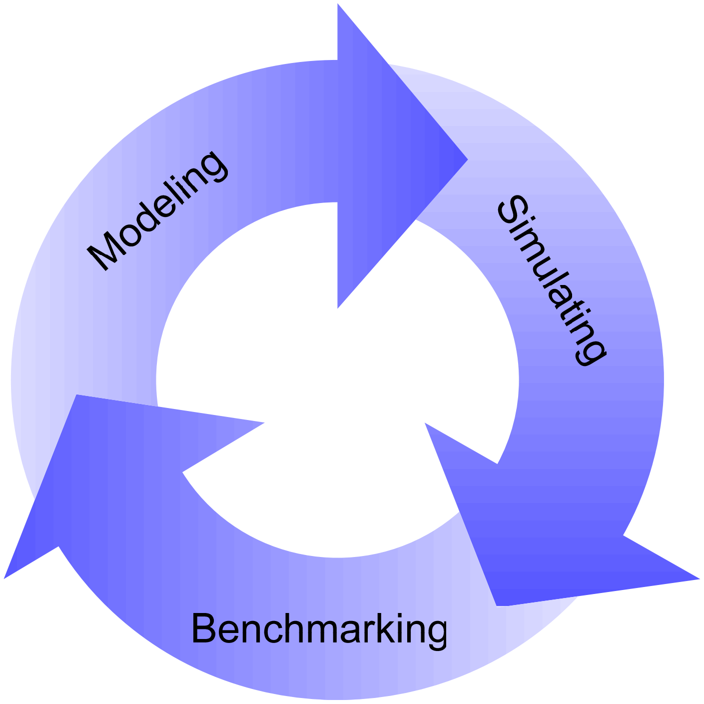
1. Introduction
2. Data modeling
3. Benchmarking
4. Simulation software
5. **Conclusions**

# Conclusions

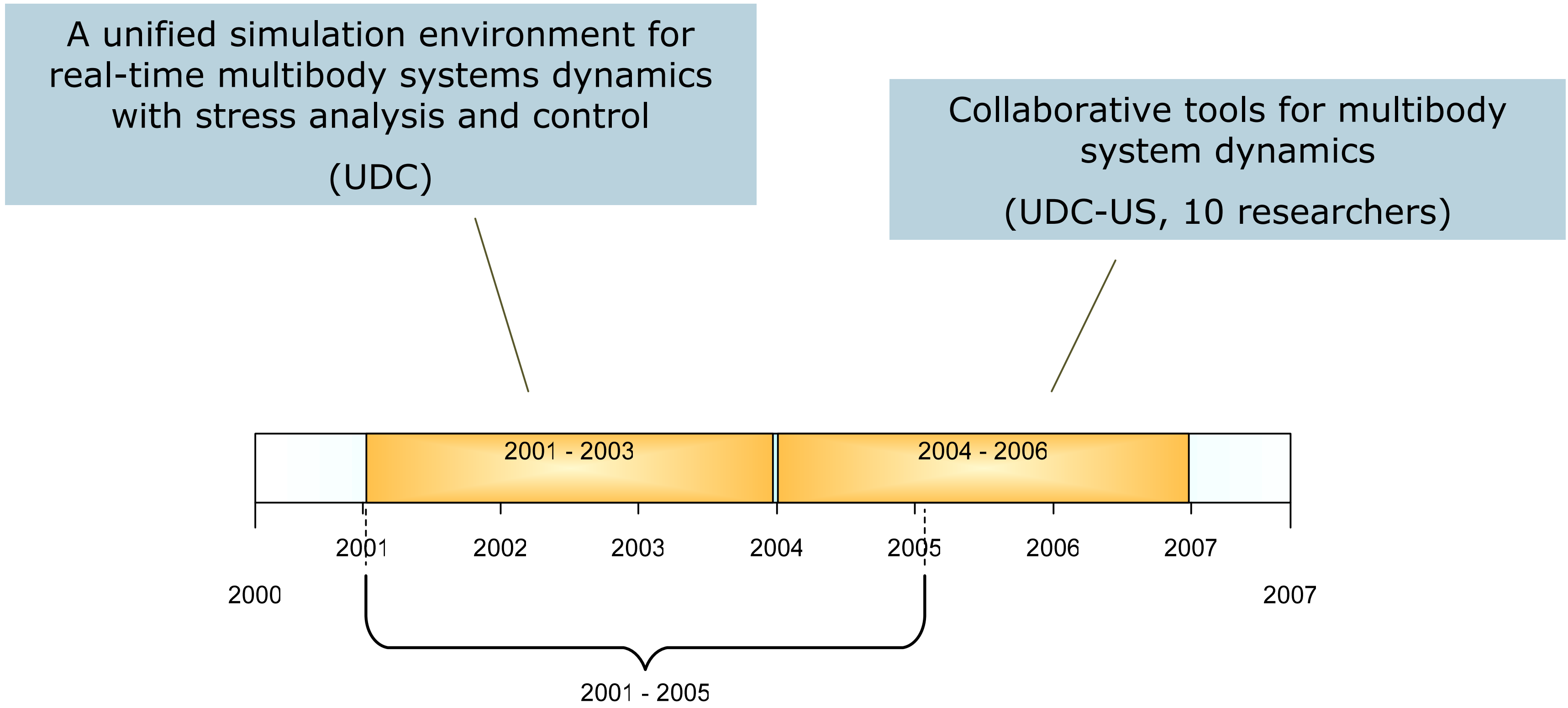
- The barriers to collaboration in MBS dynamics have been studied
- Extensive review of the state of the art
- Evaluation and selection of tools and technologies

- Solutions have been proposed to:
  - Neutral data format
  - Benchmarking system
  - Simulation software

- Prototypes have been proposed for all the systems



# Future work


A unified simulation environment for real-time multibody systems dynamics with stress analysis and control
(UDC)
Collaborative tools for multibody system dynamics
(UDC-US, 10 researchers)
2001 - 2003
2004 - 2006
2001
2002
2003
2004
2005
2006
2007
2000
2007
2001 - 2005