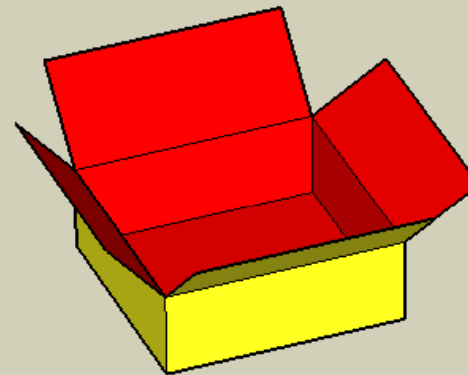




Università degli Studi di Trieste
www.units.it
Via S. Ivo, 2 - Tel. +39-040-250.7111 - Piazza Europa, 1 - I-34127 Trieste, ITALIA

MOSE
Molecular Simulation Engineering
enabling innovation



CAPEOPENTOOLKIT.NET

Maurizio Fermeglia, mauf@mose.units.it

Marco Parenzan, marco.parenzan@capeopentoolkit.net

Agenda

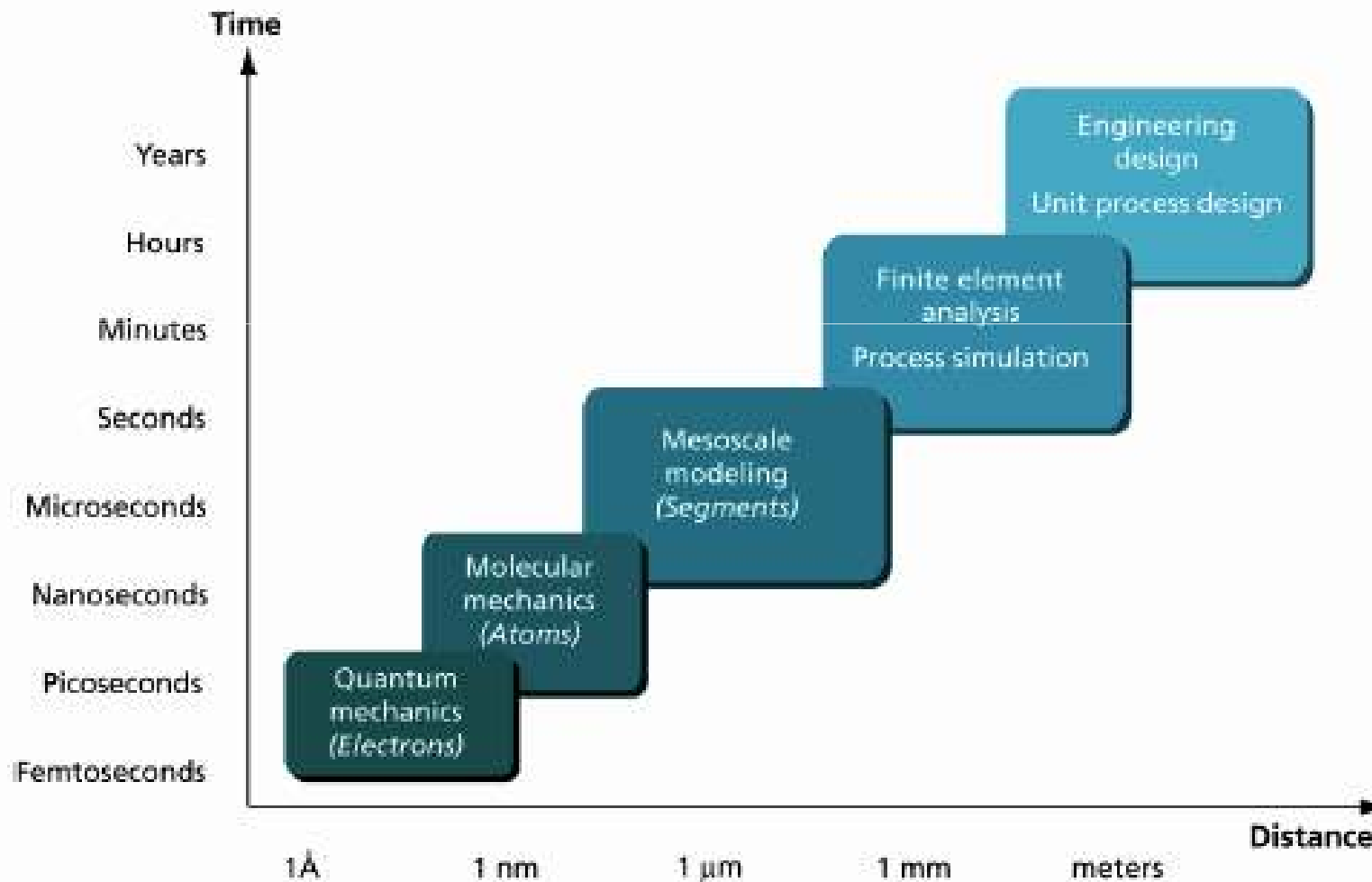


- MOSE
- Cape Open Toolkit .net
- Software Demo
- Futures
- Conclusions

MOlecular Simulation Engineering

- MOSE is a research structure aiming at developing computer-based techniques
- Research sectors
 - Process simulation: process synthesis, design, modeling for chemical, biochemical, energy production
 - Material Sciences: thermo physical properties for materials, polymer technology and nanoscience/nanotechnology
 - Life Sciences: drug-receptor interactions, drug-design, QSAR, drug-delivery...
 - Data-base and Web Applications development

Multiscale modeling



Cape Open Toolkit Team

- ▣ Maurizio Fermeglia (Director)
- ▣ Marco Parenzan (Architect)
- ▣ Marco Carone (Developer)
- ▣ Letitia Toma (Tester)

Capeopentoolkit.net

Motivation

- A chemical engineer have few computer coding tools in hand
 - All the issues related to .NET interoperability, OOP, Data access, are totally obscure to chemical engineers
 - The discussions about IUnknow and IDispatch are not comfortable for chemical engineers
 - Chemical engineers perhaps have been exposed to some language (VB in most cases or FTN) and have very little experience with interfaces and methods
 - We don't know if it's correct or not: this is what it is (and what we have, perhaps in Italy, but probably everywhere else)
- Better approach:
 - Computer geeks prepare the infrastructure part
 - Chemical engineer code the computing part

Objectives

- Simplify the development in CapeOpen:
 - ▣ minimizing the infrastructure code needed on a per-unit basis
 - ▣ allowing non-computer scientists to develop code using large choice of different languages (strongly typed or dynamic languages)
- Develop new Framework containing:
 - ▣ New API
 - New base library for infrastructural code
 - ▣ New component wizard
 - New registration tools

Not only a new wizard...
...an entire NEW FRAMEWORK!

Steps

- Test of the VB6 wizard-generated code
- Migration of VB6 wizard-generated code in C#/VB.NET
- Separation of computing code from infrastructural code
- Development of a library that encapsulates all the original COM interfaces
- Migration of current sample sources to the new platform

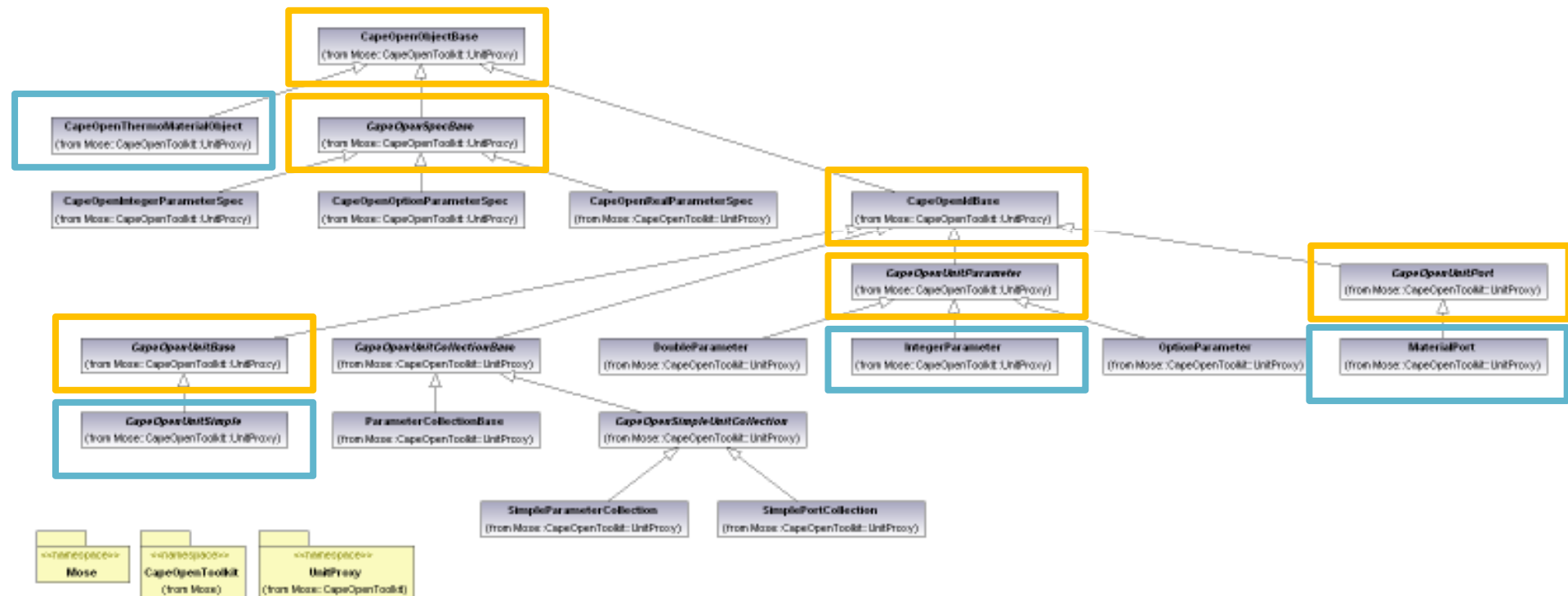
The new framework...class based

- Many times implementation of interface methods is a repetitive work due to infrastructure needs and not a specific issue needs
- Class inheritance allows the development of new classes inheriting code from parent classes and minimizing the work needed to code a fully functional class

The new framework...strongly typed

- Strong Typing is the ability for a compiler to check all types compatibility at compile time (early binding)
- This allows avoiding runtime problems such as “Invalid Cast Exceptions” due to an extensive usage of Variant (VB) or void* (C/C++)
- A better experience in coding is an extended usage of Intellisense in development tools such as Visual Studio .NET that makes coding a faster and more productive experience
- Strongly typed unit classes that allow more precise, productive, “intellisensed” and scientific code typing without the usage of “less scientific” type casting

The new framework... architecture



Generated by UModel

www.allova.com

The new framework...component registration

- We have tried implementing .NET 2.0 registration-free COM support in our project
 - ▣ It does not support “Implemented Category” registry entry
 - ▣ We cannot use this technology
- Metadata needed by registry for COM registration have been included as metadata in unit assembly
- A new registration tool builds registration info directly from assembly and directly from its installation location
- Registration is automatically executed so no more registry knowledge is needed

Capeopentoolkit.net

Software Demo

The new toolkit

- CapeOpenToolkit.net 1.0
 - ▣ Bin
 - ▣ Docs
 - ▣ Samples
 - Dummy Unit
 - Dummy Unit VB
 - Splitter Unit
 - Splitter Unit VB
 - Reactor Unit
 - WAR Unit
 - ▣ Sources
 - UnitProxy
 - UnitWizard
 - RegUnitAsm
 - Components

The samples

- Dummy (VB, C#)
- Splitter (VB, C#)
- Reactor (C#)
- WAR (C#)

The new project

- Custom base class
- <unitsource>.<extension>
 - ▣ OnCalculate
- <unitsource>.Designer.<extension>
 - ▣ OnInitialize
 - ▣ Strongly Types Ports and Parameters' values
- MsBuild/Visual Studio.NET 2005 project

The new wizard: UnitWizard

- Datatypes
- Typed parameters and ports
- Richer code generation
- Multilanguage target

The new registration utility: RegUnitAsm

- <project properties>\AssemblyInfo.<extension>
 - ▣ CapeOpen specific attributes for RegUnitAsm.exe
- Generates <unitname>.reg file based on assembly location
- Supports multi-unit assemblies

Capeopentoolkit.net

Futures

Cape Open from Dynamic Languages

- IronPython is the implementation of the famous Open Source language on .NET platform
- We are experimenting the support in creation of units in IronPython
- Two ways:
 - ▣ Full class implementation
 - ▣ Calculate only

Some other ideas in development

- Metrics is our first approach to something a computer geek can call “very strong” typing
 - ▣ The idea is avoiding usage of integer and double (types that make us thinking of an adimensional UM)
 - ▣ We want using a new type system made of (in a physical terms) dimensional UMs
 - ▣ Technically speaking, this is all about “Value Types”
 - Non performance penalties with primitive types (Integer, Double)
- Code Generation tools
 - ▣ Creating Units outside Visual Studio
- Web Services support Proxy
 - ▣ Distributing computing around Internet

CAPEOPENTOOLKIT.NET

Conclusions

We have developed a new framework

- Motivation
 - A chemical engineer have few computer coding tools in hand
 - Better approach:
 - Chemical engineer code the computing part
- Objectives
 - Simplify the development in CapeOpen
 - Develop a new framework
 - Strongly Typed
 - Class Based
- The new Toolkit
 - New API, with shared base library
 - New Unit Project Template
 - New richer Wizard with new simpler registration tool
 - New samples
- Toolkit will be soon available @ <http://www.capeopentoolkit.net/>

QUESTIONS?

Maurizio Fermeglia, mauf@mose.units.it

Marco Parenzan, marco.parenzan@capeopentoolkit.net