MRC 2007
CHUT 07
CONTEXT 07
Roskilde, Denmark
August 2007

Modeling Context in Software Reuse

Eduardo Cruz ecrs@cin.ufpe.br
Vaninha Vieira vvs@cin.ufpe.br
Eduardo Almeida esa2@cin.ufpe.br
Silvio Meira srlm@cin.ufpe.br
Ana Carolina Salgado acs@cin.ufpe.br
Patrick Brézillon brezil@poleia.lip6.fr

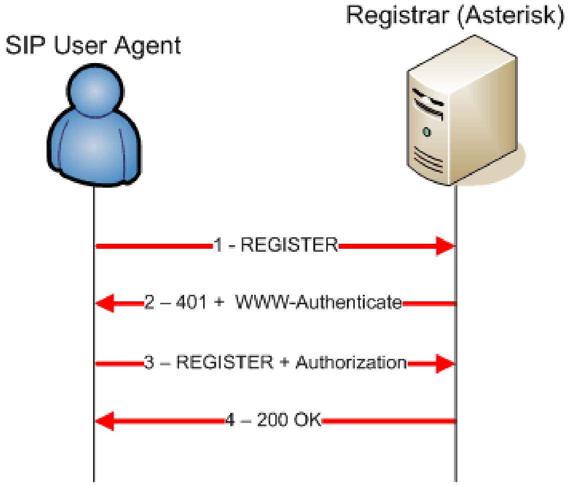
VIRTUS IMPAVIDA
Centro de Informática U·F·P·E
LIP 6

Outline

- **Motivation and Problem Statement**
- Our Proposal
- Modeling Context in Software Reuse
- Examples of Use
- Final Considerations
- Perspectives

Software Reuse

- Main purpose
  - ✓ No need to write from scratch every time
  - ✓ Developers can use pre-built components
- Benefits
  - ↓ developer’s efforts and software costs
  - ↑ software quality
- Example: Security Systems
  - Authentication
  - Authorization

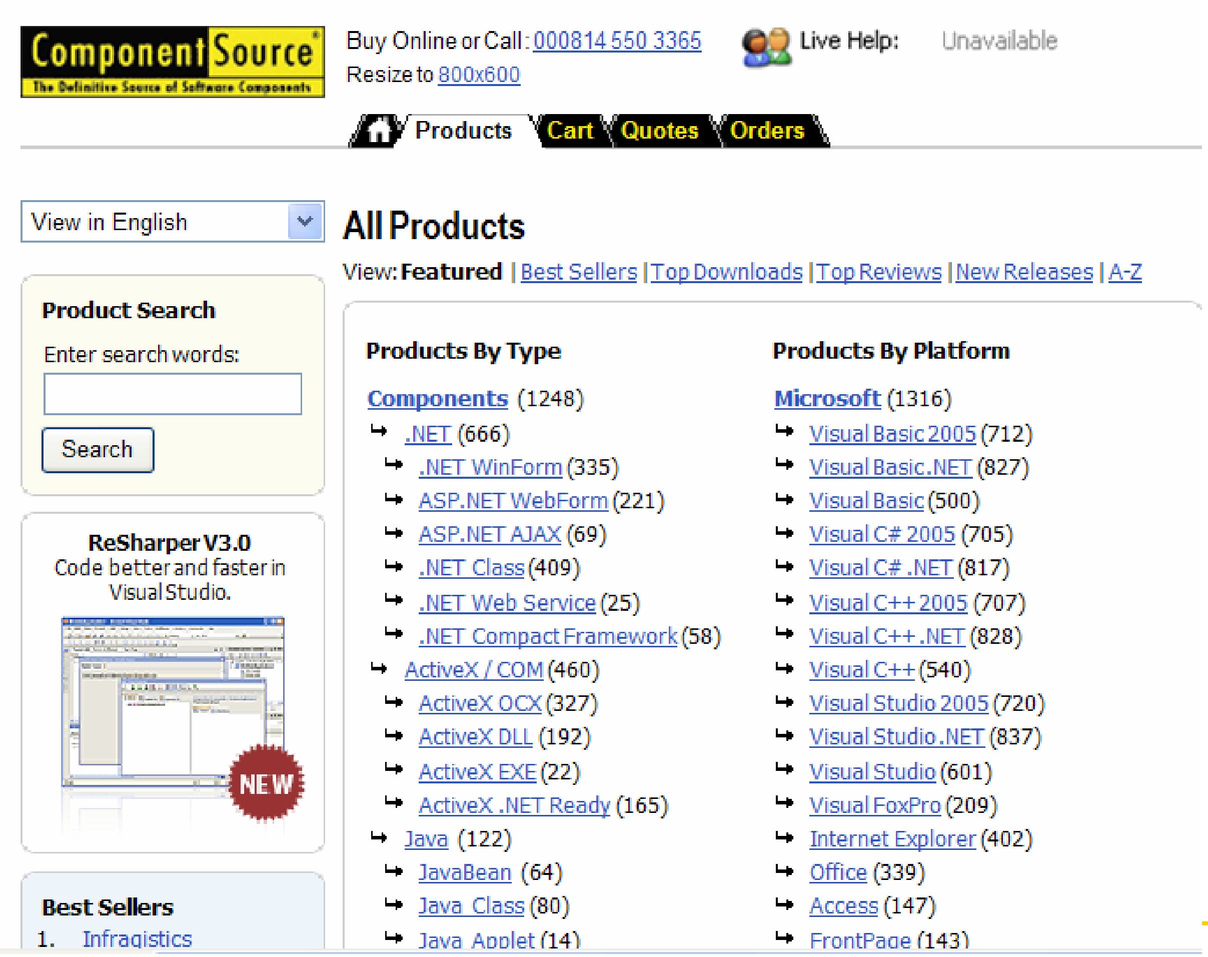


Components Repositories

- A large number of software components are now available in repositories
- There exists already a component market

ComponentSource.com

- Pioneer in the market of reusable software **components** (founded in 1995)



ComponentSource® The Definitive Source of Software Components

Buy Online or Call: 000814 550 3365 Live Help: Unavailable

Resize to 800x600

View in English

All Products

View: **Featured** | Best Sellers | Top Downloads | Top Reviews | New Releases | A-Z

Product Search

Enter search words:

Search

ReSharper V3.0
Code better and faster in Visual Studio.

Best Sellers

1. Infragistics

Products By Type

Components (1248)
- .NET (666)
  - .NET WinForm (335)
  - ASP.NET WebForm (221)
  - ASP.NET AJAX (69)
  - .NET Class (409)
  - .NET Web Service (25)
  - .NET Compact Framework (58)
- ActiveX / COM (460)
  - ActiveX OCX (327)
  - ActiveX DLL (192)
  - ActiveX EXE (22)
  - ActiveX .NET Ready (165)
- Java (122)
  - JavaBean (64)
  - Java Class (80)
  - Java Applet (14)

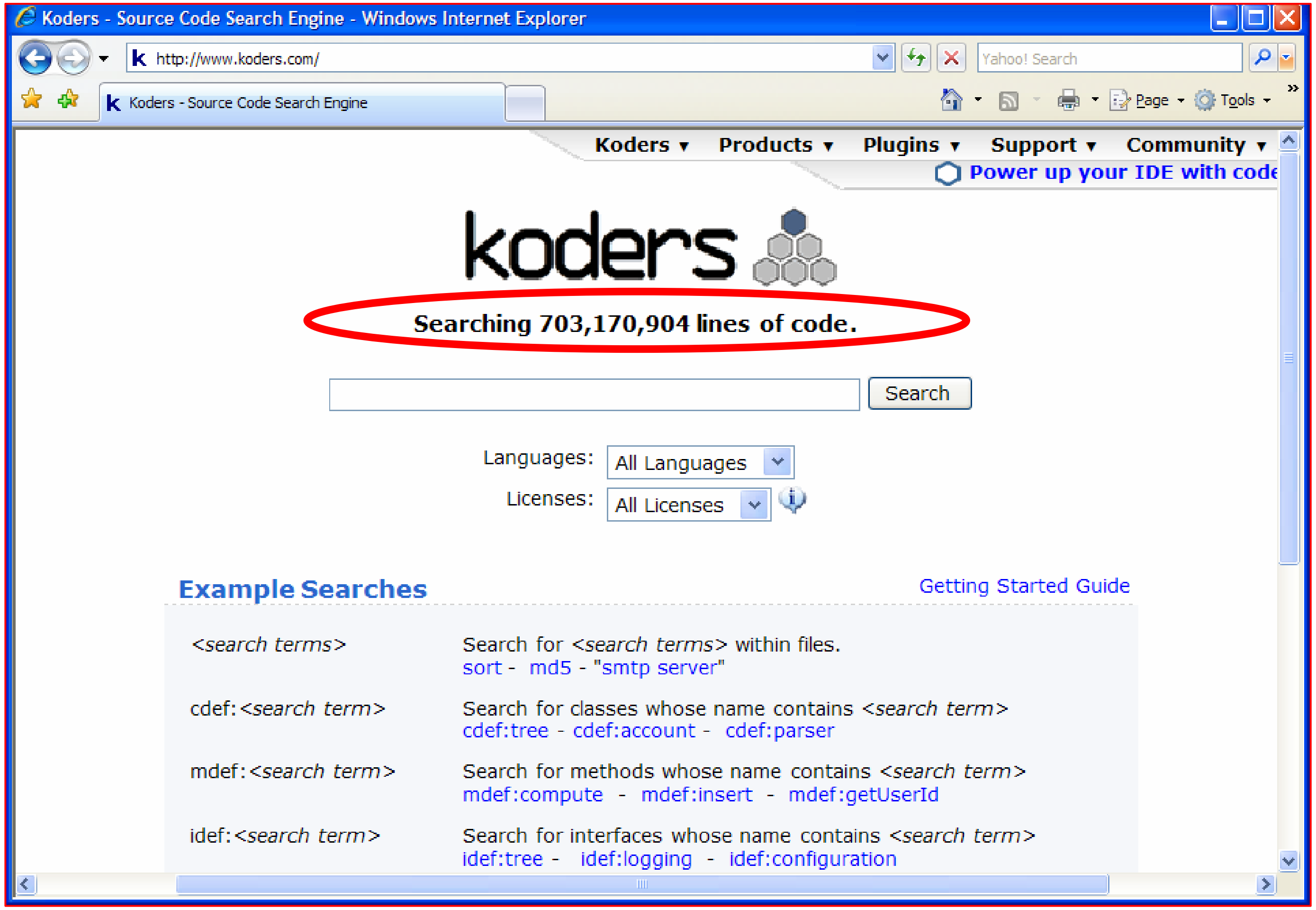
Products By Platform

Microsoft (1316)
- Visual Basic 2005 (712)
- Visual Basic .NET (827)
- Visual Basic (500)
- Visual C# 2005 (705)
- Visual C# .NET (817)
- Visual C++ 2005 (707)
- Visual C++ .NET (828)
- Visual C++ (540)
- Visual Studio 2005 (720)
- Visual Studio .NET (837)
- Visual Studio (601)
- Visual FoxPro (209)
- Internet Explorer (402)
- Office (339)
- Access (147)
- FrontPage (143)

- Thousand of components can be assembled in different contexts

Koders.com

- User can have access to 700 million lines of code



Koders - Source Code Search Engine - Windows Internet Explorer
http://www.koders.com/
Yahoo! Search
Koders - Source Code Search Engine
Page
Tools
Koders
Products
Plugins
Support
Community
Power up your IDE with code
koders

Searching 703,170,904 lines of code.

Search

Languages: All Languages

Licenses: All Licenses

Example Searches

Getting Started Guide

<search terms> Search for *<search terms>* within files.
sort - md5 - "smtp server"

cdef:*<search term>* Search for classes whose name contains *<search term>*
cdef:tree - cdef:account - cdef:parser

mdef:*<search term>* Search for methods whose name contains *<search term>*
mdef:compute - mdef:insert - mdef:getUserId

idef:*<search term>* Search for interfaces whose name contains *<search term>*
idef:tree - idef:logging - idef:configuration

Problem with Components Repositories

- More components you have, more difficult to find the right one you need
- Components are stored out of their contexts of use
  - Component developer's context may be different from component user's context

Components must be searched and assembled not only by their content, but also by their context of use

Outline

- Motivation and Problem Statement
- **Our Proposal**
- Modeling Context in Software Reuse
- Examples of Use
- Final Considerations
- Perspectives

B.A.R.T. Search Engine

- B.A.RT. - Basic Asset Retrieval Tool
- Asset = any reusable software artifact
  - ✓ Components, documents, source code, …
- Currently do not consider the context of use

CEManTIKA Approach

Contextual

Elements

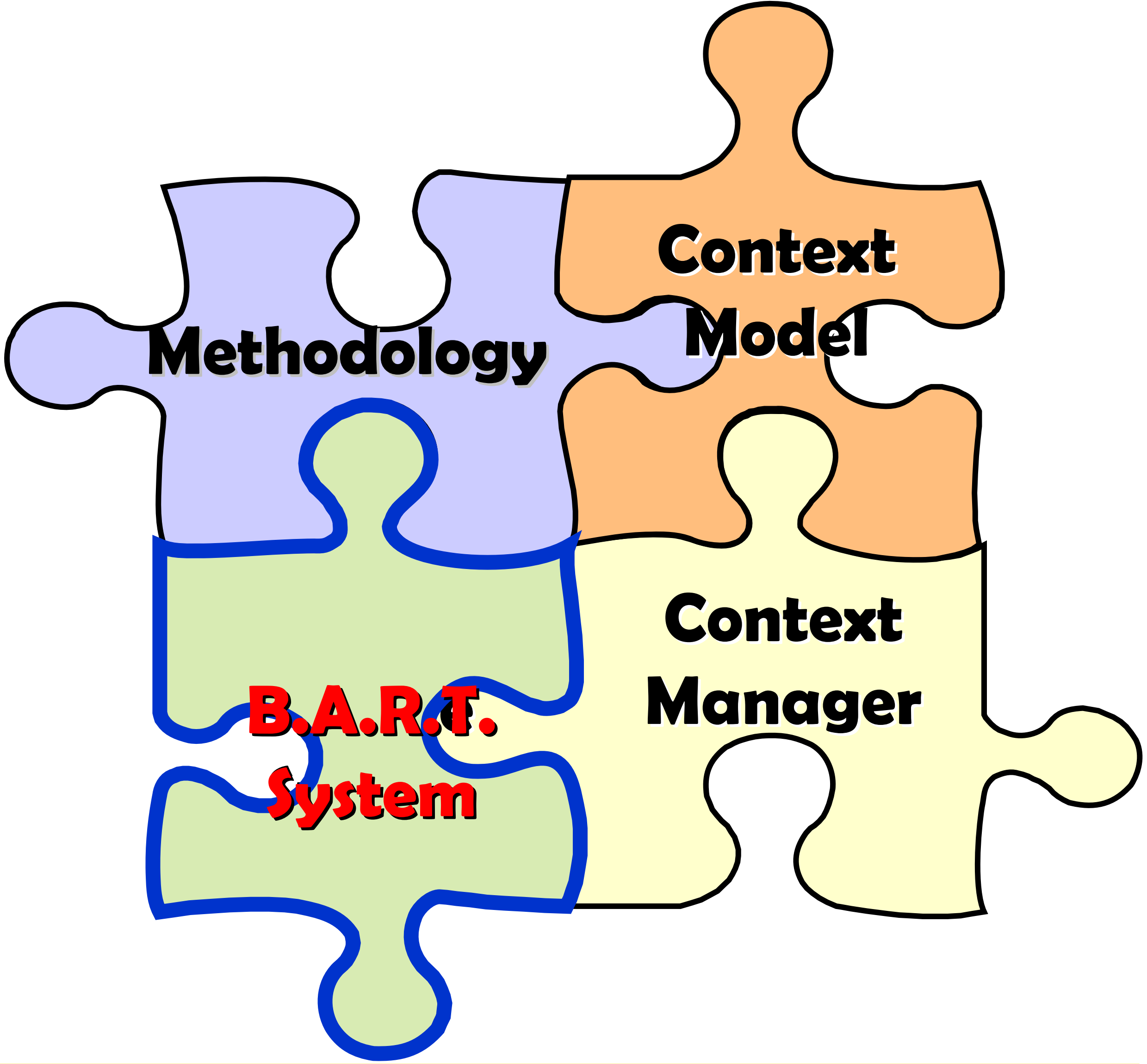
Management

Through

Incremental

Knowledge

Acquisition



Methodology

Context Model

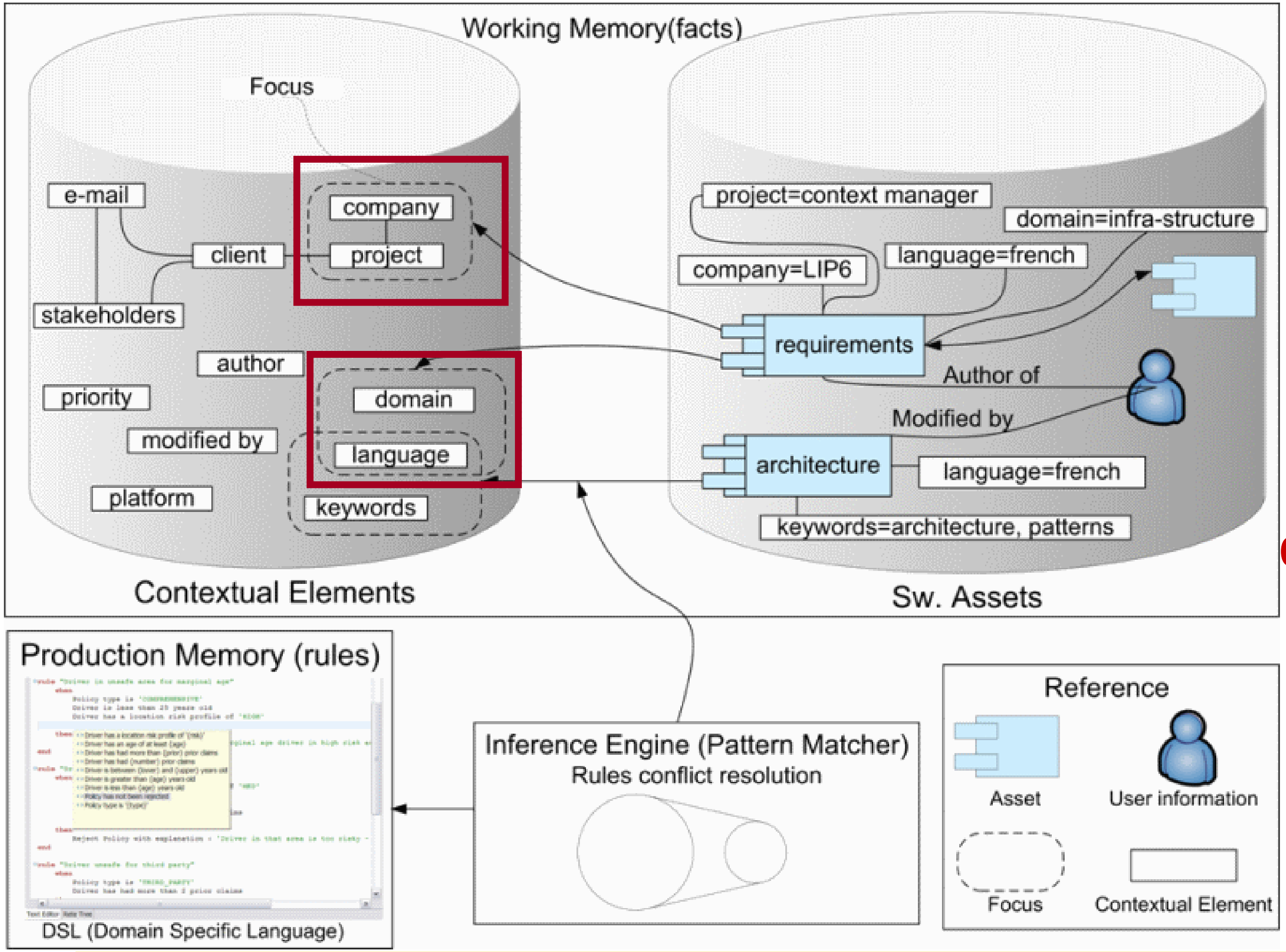
B.A.R.T. System

Context Manager

Our Goals

- Make Context Explicit
- Apply context to improve software reuse
- Change the paradigm {query, results} for {query, context, results}

Assets repository x Contextual Elements



Contextual elements are stored apart and related to software assets depending on the focus

Outline

- Motivation and Problem Statement
- Our Proposal
- **Modeling Context in Software Reuse**
- Examples of Use
- Final Considerations
- Perspectives

Context in Software Reuse

- Context is identified by Contextual Elements (CE)
- CEs in this domain are related to
  - Tasks
  - Roles
  - Assets
  - User Environments

Focus on Tasks

- Consider the different tasks in the software development process
  - ✓Software design
  - ✓Software implementation
  - ✓Software test
  - ✓Software documentation
- different tasks, different contexts' needs

Focus on the Users' Roles

- Identify the Role of the developer while realizing a given task
  - ✓ Project manager
  - ✓ Software Architect
  - ✓ Software Developer
  - ✓ Configuration Manager
  - ✓ Software Quality Engineer
  - ✓ Software Tester
- Different roles, different contexts and activities

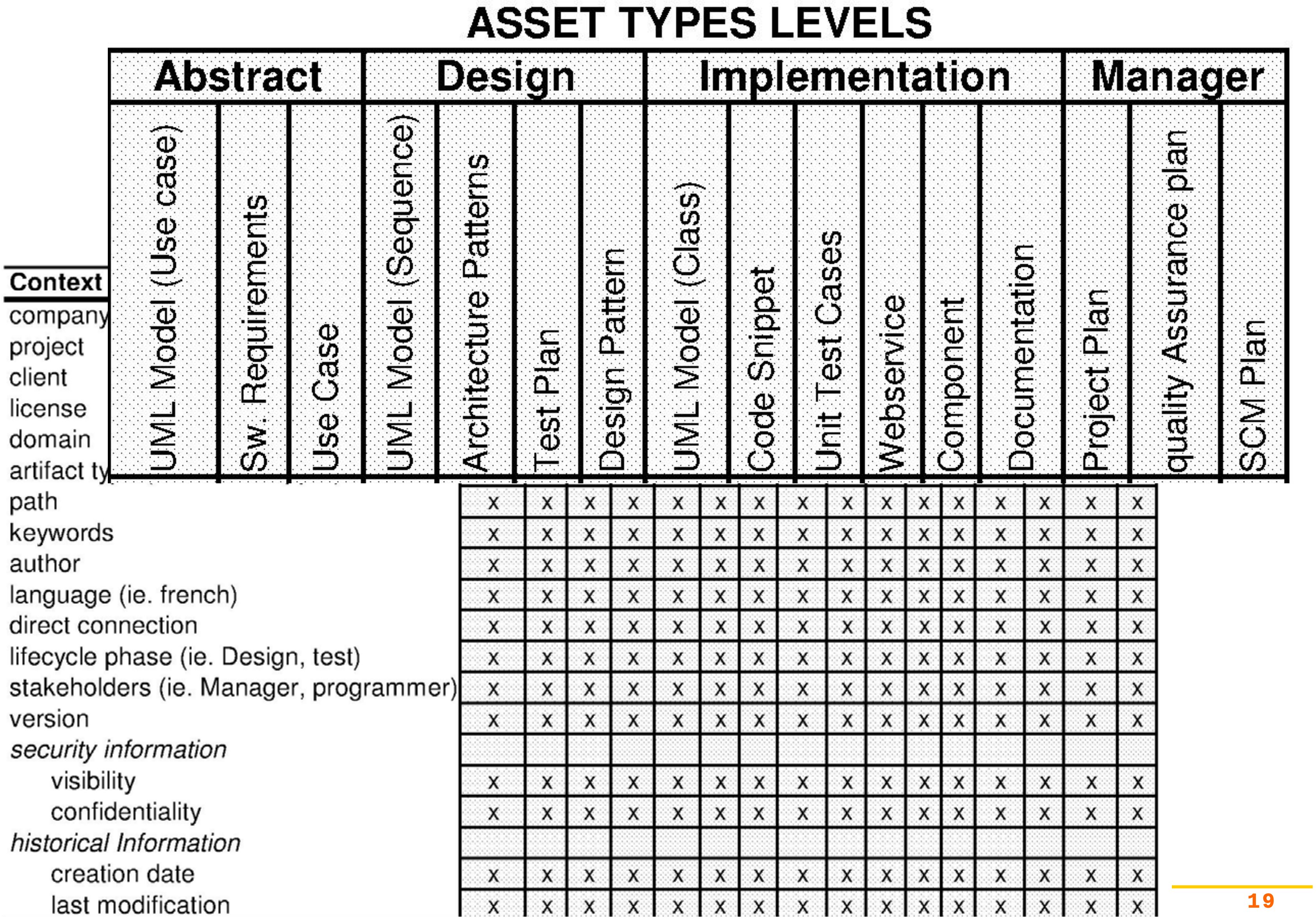
Focus on Assets

- Describe the category of the software artifact
  - ✓ Software component
  - ✓ use case document
  - ✓ java source code
  - ✓ Webservice
  - ✓ xml document
  - ✓ Frameworks
  - ✓ Windows DLL
- Different assets category imply in different usages

Focus on the User Environment

- Environment used to develop the asset
  - ✓Programming IDE (Eclipse)
  - ✓Text Editor
  - ✓Web interface
  - ✓Modeling Tool
- Different environments, different requirements

CEs associated to Assets



ASSET TYPES LEVELS

	Abstract			Design				Implementation						Manager		
Context	UML Model (Use case)	Sw. Requirements	Use Case	UML Model (Sequence)	Architecture Patterns	Test Plan	Design Pattern	UML Model (Class)	Code Snippet	Unit Test Cases	Webservice	Component	Documentation	Project Plan	quality Assurance plan	SCM Plan
company																
project																
client																
license																
domain																
artifact ty																
path	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
keywords	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
author	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
language (ie. french)	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
direct connection	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
lifecycle phase (ie. Design, test)	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
stakeholders (ie. Manager, programmer)	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
version	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
security information																
visibility	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
confidentiality	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
historical Information																
creation date	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
last modification	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Outline

- Motivation and Problem Statement
- Our Proposal
- Modeling Context in Software Reuse
- **Examples of Use**
- Final Considerations
- Perspectives

Example 1 {Query; Context; Results}

- Query = “Authentication”
- Identified Contextual Elements:
  - ✓ User Role = Programmer
  - ✓ User Environment = Eclipse IDE
  - ✓ User Task = Debugging Software
  - ✓ Programming language=Java
  - ✓ Project=JContactPhone
  - ✓ License=GPL
  - ✓ Domain=Mobile
- Results = ***source code components*** related to **authentication** written in ***Java Mobile (J2ME)*** with ***GPL license***

Example 2 {Query; Context; Results}

- Query = “queen”
- Identified Contextual Elements:
  - User Role = Architect
  - User Environment = Borland Together
  - User Task = Modeling a class diagram
  - UML Diagram=Class
  - Project=ChessGame
  - Domain=Games
  - GameType=BoardGame
- Results = ***class diagrams*** related to the ***games domain*** about the **queen** piece in a ***Chess Game***

Outline

- Motivation and Problem Statement
- Our Proposal
- Modeling Context in Software Reuse
- Examples of Use
- **Final Considerations**
- **Perspectives**

Final Considerations

- Software reuse and assets repositories must be seen in a human centered approach
  - consider user’s context of use
  - improve the interaction between the user and the system
  - Better and more appropriate search results
- Some information can be implicitly extracted
  - ex. User environment, programming language
- Some information must be explicitly asked
  - ex. User role in the task execution

Lessons learned

- Two parts in the context
  - ✓ a domain-dependent part represented by the contextual elements
    - Software reuse
  - ✓ a domain-independent part with the context manager and all its mechanisms
    - CEManTIKA

Perspectives / Future Work

- Specify the integration between B.A.R.T. and the CEManTIKA manager
- Specify scenarios of use to enable the definition and modeling of as much CEs as possible in different situations
- In near future, implement the context-aware search in B.A.R.T.
  - ✓ Evaluate with industry partners

MRC 2007

CHUT 07 CONTEXT 07

Modeling Context in Software Reuse

Eduardo Cruz ecrs@cin.ufpe.br
Vaninha Vieira vvs@cin.ufpe.br
Eduardo Almeida esa2@cin.ufpe.br
Silvio Meira srlm@cin.ufpe.br
Ana Carolina Salgado acs@cin.ufpe.br
Patrick Brézillon brezil@poleia.lip6.fr