

# IS-COTS: a help to COTS Products Integration



**LIUPPA**

Philippe ROOSE

LIUPPA (University of Pau/France)

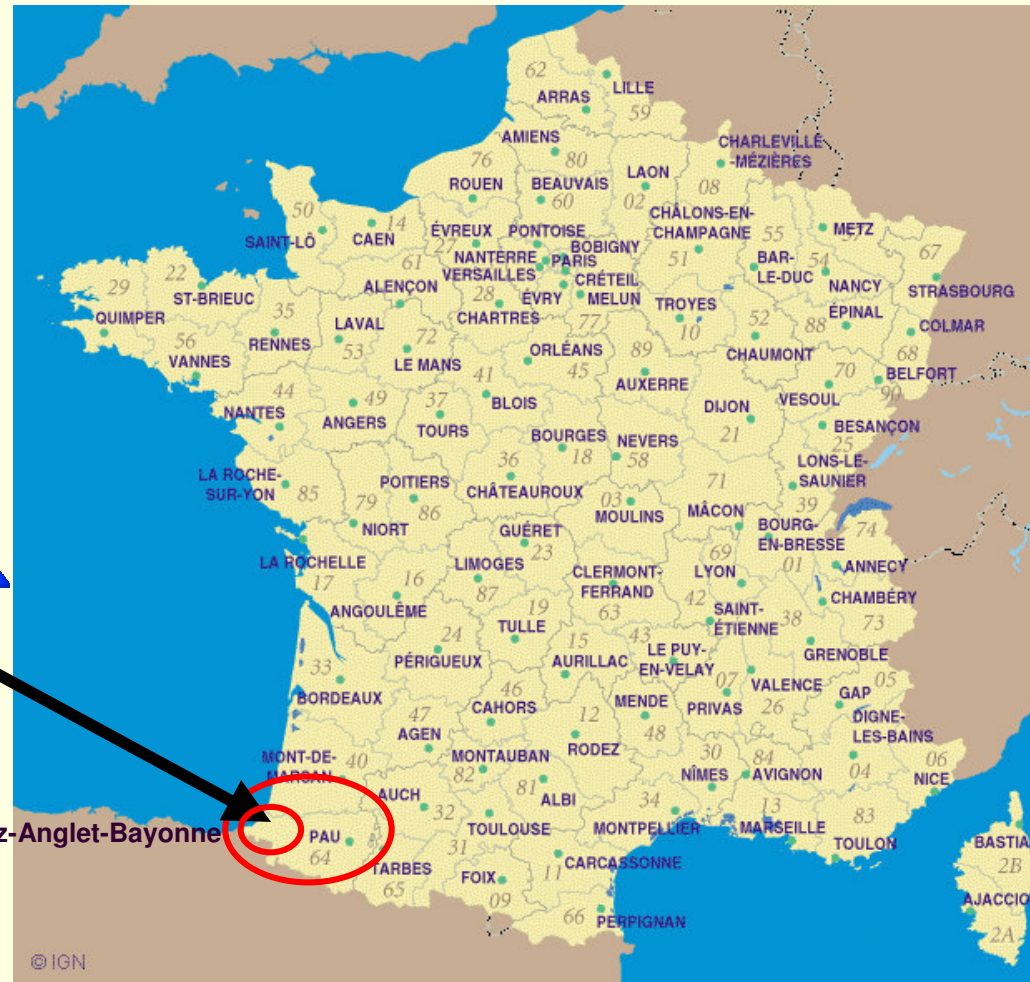
# Introduction – LIUPPA (France)



LIUPPA

Public Research Center  
of the University of Pau

Biarritz-Anglet-Bayonne



LIUPPA - Philippe ROOSE / IOTSDM'08 (Madrid)

# Subject

---

- COTS Based Software applications
  - **a help to COTS Products Integration**
- Acknowledgement
  - An increasing interest in the use of COTS
- In Return
  - An important loss of control over the systems onto which COTS are deployed
    - A strong dependence on COTS editors

# Subject

- How to reduce these difficulties
  - With a set of knowledge about COTS
    - Functionality
    - Integration Environment
    - Execution Environment
- Problem
  - How to get useful information
    - Availability, accessibility, validity, etc.
- A solution
  - Prototyping the COTS based application
    - Experimentation to collect information

# Proposition

---

- How

- With tool to help to collect and to manipulate such information.
- Helping developers with the integration of COTS.

# Proposition

---

## ■ Solutions

- An ID Card for each COTS (ICOTS)
- An Information System (IS) for COTS
- An approach to use the IS with several actors
  - Give useful information for the integration of COTS

## ■ Gains

- Quick collection of information
- Help with data consolidation
- Capitalization on experiences
- Help with COTS integration

# A COTS Product : Commercial-Off-The-Shelf

---

- A software component
  - With several copies (not unique)
  - Source code available or not (*black, grey, white boxes*)
  - Sold, rented, freely provided by a vendor
  - With periodic updates having increasing functionality whereas some other functionality becomes deprecated.
  - Propose one (or several) services as an essential part for another application

# The use of a prototype

---

## Problems with collecting information

- Volume (600 risky conditions or combination of entry parameters)
- Spreading, loss of accessibility
- COTS based developers often say that they need to prototype their application to integrate one or several COTS in order to:
  - Experiment and test the veracity of explanations given for the COTS about its services.
  - Collect qualitative and quantitative information
  - Collect information on compatibility, execution environment, etc.
  - Validate information and ensure the processes' validity with correct test beds.



# Useful Information

---

- We propose a “Client/Provider” relationship

- Why?

- To write code in order to implement the integration of the COTS
  - To run the application with the COTS

# Useful Information for using COTS

---

- All software needed to execute the COTS
  - Depends on the COTS
  - Depends on the needed service
- All software not permitted to execute the COTS
  - Depends on the COTS
  - Depends on the needed service
- Documentation explaining the various cases study.

# Useful Information for the Integration of COTS

---

- All software needed for the implementation
  - Software development tools, API, etc.
- Programming languages
- Associated documentation
  - Examples of source code

# We need some information

- 
- Le diagramme de base de données relationnelles illustre la structure de données pour le projet de gestion de la production et des connaissances. Les tables sont représentées par des rectangles bleus, et les relations sont indiquées par des lignes noires avec des cardinalités.
- Tables et leurs attributs :**
- Acteurs** : IdActeurs, Pswrd, Nom, Prenom
  - Acces** : IdActeurs, IdProduct, IdVersion
  - CotsProducer** : IdProducer, Name
  - CotsProduct** : IdProduct, IdProducer, CotsName, Type, TypeFournisseur, EnsembleProducts
  - CotsVersion** : IdProduct, IdVersion, NumScInt, Langue, License, Integrale
  - AssociationServices** : IdProduct, IdVersion, IdService, IdProductAssocie, IdVersionAssocie, IdServiceAssocie
  - Services** : IdProduct, IdVersion, IdService, NomService, IdSousService, Integrale
  - ScenIntServices** : IdProduct, IdVersion, IdService, NumScIntService, Valide, Notes
  - EnvDevServiceBaseConnaissances** : IdProduct, IdVersion, IdService, NumScIntService, IdBaseConnaissances
  - ListeConnaissancesProduit** : IdProduct, IdVersion, IdBaseConnaissances
  - ScenExec** : IdProduct, IdVersion, NumScExec, TypeLienExec, Valide, Notes
  - Dependencies** : IdProduct, IdVersion, NumScExec, IdProductDepend, IdVersionDepend, TypeDepend
  - EnvDevBaseConnaissances** : IdProduct, IdVersion, NumScInt, IdBaseConnaissances
  - ScenInt** : IdProduct, IdVersion, NumScInt, Valide, Notes
  - ScenExecEnvDev** : IdProduct, IdVersion, NumScInt, IdProductDepend, IdVersionDepend, Notes
  - EnvDevService** : IdProduct, IdVersion, IdService, NumScInt, IdDev, Notes
  - EnvDev** : IdProduct, IdVersion, NumScInt, IdDev, Notes
  - LangProg** : IdProduct, IdVersion, NumScInt, IdDev, Langage
  - OutilsDev** : IdProduct, IdVersion, NumScInt, IdDev, IdProductDev, IdVersionDev
  - BaseConnaissances** : IdBaseConnaissances, Notes
- Relations et Cardinalités :**
- Acteurs** (1) à **Acces** (∞)
  - Acces** (∞) à **CotsProduct** (∞)
  - CotsProduct** (1) à **CotsVersion** (∞)
  - CotsVersion** (1) à **AssociationServices** (∞)
  - AssociationServices** (∞) à **Services** (∞)
  - Services** (∞) à **ScenIntServices** (∞)
  - ScenIntServices** (∞) à **EnvDevServiceBaseConnaissances** (∞)
  - ListeConnaissancesProduit** (∞) à **ScenExec** (∞)
  - ListeConnaissancesProduit** (∞) à **BaseConnaissances** (∞)
  - ScenExec** (1) à **Dependencies** (∞)
  - Dependencies** (∞) à **EnvDevBaseConnaissances** (∞)
  - EnvDevBaseConnaissances** (∞) à **ScenInt** (∞)
  - ScenInt** (1) à **ScenExecEnvDev** (∞)
  - ScenExecEnvDev** (∞) à **EnvDevService** (∞)
  - EnvDevService** (∞) à **EnvDev** (∞)
  - EnvDev** (1) à **LangProg** (∞)
  - EnvDev** (1) à **OutilsDev** (∞)
  - LangProg** (1) à **OutilsDev** (∞)

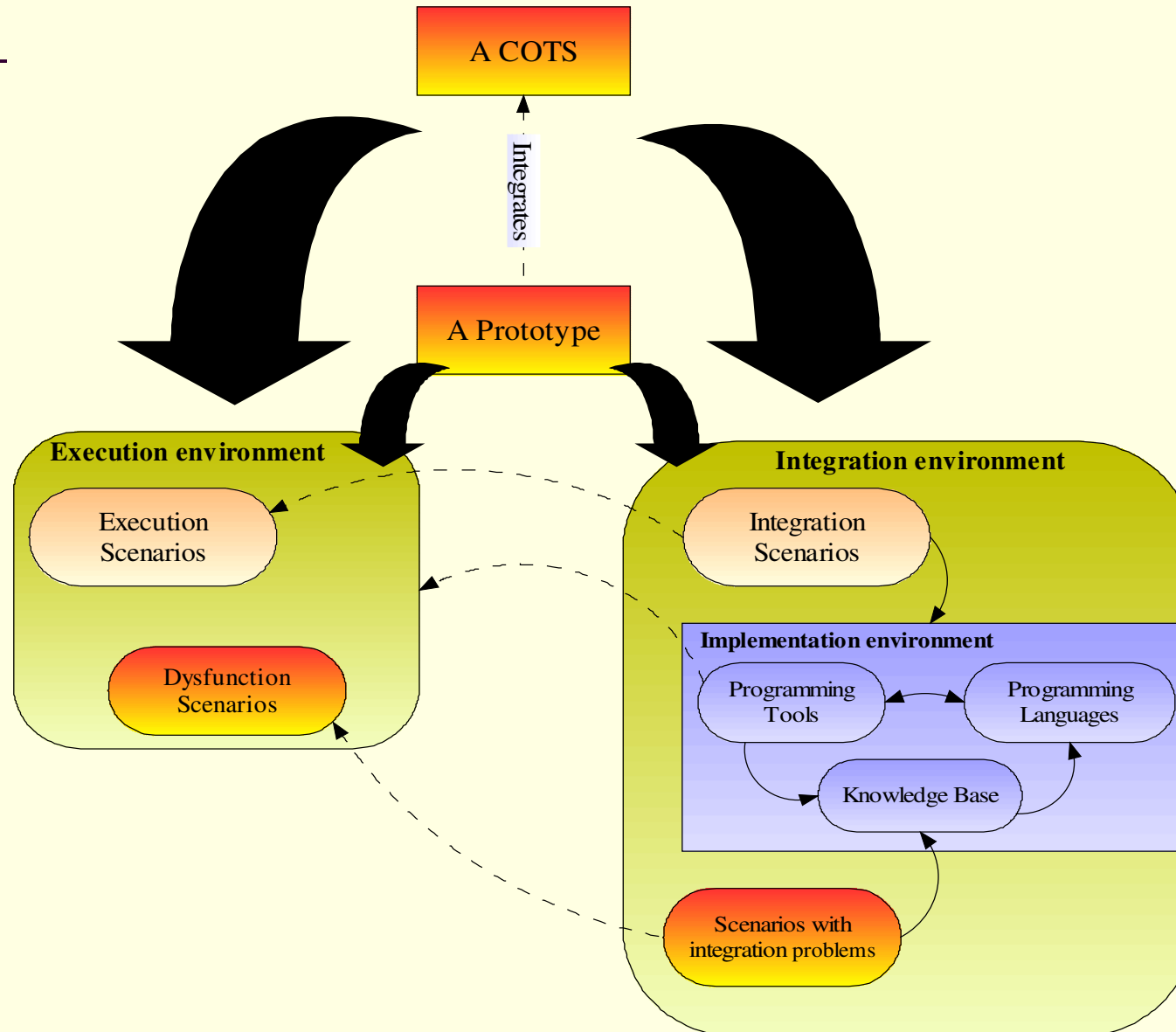
## Data Model

# Environment and scenarios

---

- Each scenario is a particular case to evaluate.
- To describe a prototype or a COTS, we will identify three environments with reports production (*see figure next slide*):
  - Execution environment,
  - Integration environment,
  - Implementation environments.

# Environment and Scenarios



# Report (1) on execution environment

---

- Set of software needed to the execution of each COTS product.
- List of incompatible software.
- List of compatible operating systems common to all COTS products.

# Report (2) on integration environment

---

- List of software needed for integration.
- Services provided by selected COTS product and those remaining to develop to complete the integration.
- Documents linked to each COTS service.



# Report (3) on failure

---

- List of non-compatibilities for each COTS product.
- Identification of the COTS products responsible for non-compatibility in an assemblage.

# Actors (actor 1 : FourCo)

---

- **1) *COTS provider and designer is called FourCo***
  - Classically, one must dissociate the manufacturer of a product from the distribution circuit or the vendor.
  - Most of the time, they represent two distinct entities.
  - It is not relevant to know who provides the product and who gets the information needed by the client.
  - The only difference should be the quality of the service provided by the various distribution circuits.
  - This criterion is important for the selection of the COTS vendor, but has no impact on the quality of the service nor the choice of the product.

# Actors (actor 2 : COTS User)

---

## ■ 2) *COTS User*

- Has to fish for information.
- For those that are not easily retrieved, user will have to search for them with evaluation and tests realized via prototypes.
- This means that the "COTS User" will
  - Start the first implementation process activity (technological survey).
  - Initialize the communication process with FourCo
  - Go on communicating with it during all the application's development process.
  - Get some help from a new actor called 'COTS Community'.

# Actors (actor 3 : COTS Community)

---

## ■ **3) COTS Community**

- Represents the set of developers with at least one COTS based application development experience.
- The history of their development will help to compare estimates with real costs and risks of the COTS users managed project.
- Play an important part, by enriching the reports associated with each COTS product.

# Prototype: An Environment to capture information

---

## ■ The Prototype

- Records the ID Card with information linked to the COTS product and to the COTS based application.
- The ICOTS becomes available, and is open to new ICOTS downloading.
- IS-COTS enables the identification of all 'COTS Products' services that are needed for the COTS based application.
- Identifies the interoperable services (*those that are not interoperable are not interesting*)
- Helps to:
  - input prerequisites, development tools, code samples, and identified failures.
  - target the set of software needed for the development, the execution of the prototype as well as possible failures in the prototype execution.

# Prototype: Provider Part

Système d'Informations pour COTS ( Raphaël MICHEL)

**Définir l'Application**

Modifier le Scénario d'Intégration d'un Service

Ajouter une Documentation

Ajouter un environnement de développement d'un Service

Supprimer

**Identifiez votre application**

**Fiche ICOTS**

**Editeur : Microsoft**

Editeur: Microsoft Id Editeur: ms

**Produit : Ms Outlook**

Produit: Ms Outlook Id Produit: ms:MsOutl

Catégorie: Produit

**Vers. 2003**

Version: 2003 Id Version: ms:MsOutl:2003fr

Langue: Fr

Intégrable

License: Payante

**Service : Gestion des contacts**

Service: Gestion des contacts Id Service: ms:MsOutl:2003fr:Gest\_Cc

Intégrable

☒ Intégrable ☐ NON Intégrable

**Scénario d'Intégration <specifique contacts> du Service : "Gestion des contacts"**

Id Scénario Intégration d'un service: ms:MsOutl:2003fr:Gest\_Cont:specifique c

Service: Gestion des contacts

basé sur le Scénario d'Intégration d'un produit: ms:MsOutl:2003fr:Par microsoft

Editeur: Microsoft

Produit: Ms Outlook

Version: 2003

Scénario de Fonctionnement (permettant l'Intégration du Service)

Notes

Active X Data Objects Library

Windows Serveur

Visual Studio

SQL Server

Project Serveur

Project

IIS

Outlook ObjectLibrary

Data Access Components

.Net Frameworks

Windows

Définition de l'Application

Télécharger / Retirer les fiches ICOTS

Sélectionner des COTS

Sélectionner les services des COTS

Rapports

Demander des informations

Fournir des informations

# Prototype: User Part

Système d'Informations pour COTS ( Raphaël MICHEL)

**Définir l'Application**

Modifier l'environnement d'exécution

Ajouter une dépendance

Supprimer

**Définir l'Application**

Télécharger / Retirer les fiches ICOTS

Sélectionner des COTS

Sélectionner les services des COTS

Rapports

Demander des informations

Fournir des informations

**Identifiez votre application**

Perso\_RM

IPPOP1

IPPOP1/V.0.5/Française

Manipulation Grille GRAI

Messagerie

Gestion de Projet

Gestion des utilisateurs

Perso:Ippop1:0.5:Scénario Principal

OS/XP

XP/V.Professionnel SP2/Française

OS/Windows 2000

Windows 2000/V.Service Pack 2/Fr

**Fiche ICOTS**

Editeur : Perso\_RM

Produit : IPPOP1

Vers. 0.5

Version 0.5 Id Version Perso:Ippop1:0.5

Langue Française NON Intégrable

License Gratuit

Scénario nécessaire à l'exécution du produit : Scénario Principal

Scénario d'exécution Perso:Ippop1:0.5:Scénario Principal

Scénario permettant l'exécution du produit

Notes

Scenario prioritaire pour l'environnement d'exécution du projet IPPOP

# Conclusion

---

- The prototype has been implemented with .NET Technology for Windows.
- It is downloadable at the following URL
  - <http://www.iutbayonne.univ-pau.fr/~roose/pub/recherche/SICOTS>
- It allows improvement of the collection of information, facilitates and accelerates the access to information for the integration.
- It assists evaluations thanks to its reports and allows information capitalization by the COTS Products User Community.

**Access to information is precious. It is important to capitalize it.**

- **Limits**
  - Interoperability problems for ICOTS : it needs a normalization of exchanges and data.
  - We have only addressed COTS products with an integration point of view. This does not cover the wide spectrum of selection, evaluation, qualification, risks and costs steps.



# Current/Future works

---

- Specialization in Multimedia oriented COTS
  - ID-Card for MM-COTS
  - Specification of interactions
  - Automatic generation of connectors (even transitively) with QoS constraints
  - Dynamic links
  - Preservation of the QoS : dynamic reconfiguration

# Questions

---

