

---

# Context Aware Deployment for Mobile Users

Chantal Taconet, Erik Putrycz, Guy Bernard

GET / INT - France Evry

<http://picolibre.int-evry.fr/projects/sdi/>

<http://www-inf.int-evry.fr>

COMPSAC '03, Dallas, November 2003



COMPSAC'03

---

## Introduction

- **Deployment**
  - ◆ All activities that make software available to its users
  - ◆ After the development of the software
- **Mobile Users**
  - ◆ They change of location
  - ◆ They change of terminals
  - ◆ In the same time:
    - they have preferred applications
    - they dislike software installations
- **Context Aware**
  - ◆ Variable execution contexts
- **Just in time deployment**
  - ◆ Allows to automatically take execution context into account



COMPSAC'03

## Outline

---

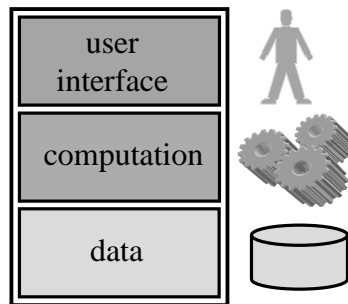
- Application and hypothesis
- Multi-component application assembly and deployment descriptors
- Deployment process
- Discovery service
- Results
- Conclusion and perspectives

## Application and hypothesis

---

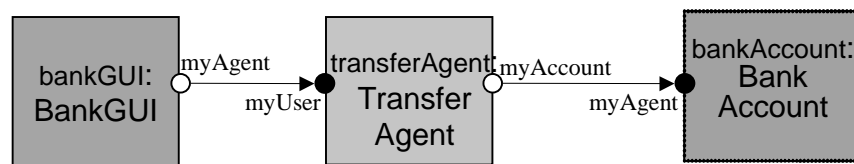
- Distributed applications such as
  - ◆ bank transfer application
  - ◆ shopping application
  - ◆ on site medical emergency applications
- Multi-component applications
  - supported by component middleware
- One component type - several implementations
- Parts of the application may be placed out of the terminal
- Mobile users have access to their application descriptions
- Application Service Providers furnishes a deployment infrastructure

## Components of an application



## Application assembly descriptor

lists component **instances** with their **types** and their **connections**



- required interface
- provided interface

## Bank Assembly Descriptor

---

```
<assembly applicationname="SDI.Bank">
  <component instanceName="bankGUI"
    type="sddemo.bank.BankGUI"/>
  <component instanceName="transferAgent"
    type="sddemo.bank.TransferAgent"/>
  <component instanceName="bankAccount"
    type="sddemo.bank.BankAccount"/>
  <connection from="bankGUI.myAgent"
    to="transferAgent.myUser" type="synchronous"/>
  <connection from="transferAgent.myAccount"
    to="bankAccount.myAgent" type="synchronous"/>
</assembly>
```



COMPSAC'03

## Abstract deployment descriptor (content)

---

- **component lifecycle:**
  - ◆ already instantiated (*entity*)
  - ◆ to instantiate and destroy (*session*)
  - ◆ to instantiate and leave until completion (*process*)
- **component location constraints:**
  - ◆ on terminal
  - ◆ co-located with an other component
- **component implementation and component instances**
- discovery rules:**
  - ◆ constraints
  - ◆ preferences
- **component instantiation initial arguments**



COMPSAC'03

## Abstract deployment descriptor (example)

```
<deployment applicationname="SDI.Bank"
  <component name="bankGUI" lifecycle="session" entry="true">
    <location> <onhost name="{terminal}"/> </location>
    <implementation><discovery>
      <preference value="language="{userpref.language}"/>
    </discovery></implementation>
    <instantiation initarguments="currency="{userpref.currency}"/>
    <entry method="start"/>
  </component>
  <component name="transferAgent" lifecycle="process"/>
  <component name="bankAccount" lifecycle="entity">
    <instance> <discovery><constraint
      value="RIB="{userpref.sdidemo.bank.accountRIB}"/>
    </discovery></instance>
  </component>
</deployment>
```



COMPSAC'03

## Deployment and relevant context

- relevant context includes
  - ◆ user preferences (language) ,
  - ◆ network and geographical location,
  - ◆ service availability,
  - ◆ system load and network bandwidth,
  - ◆ terminal capabilities
- relevant context may be
  - ◆ specific to an application (abstract deployment descriptor)
  - ◆ generic



COMPSAC'03

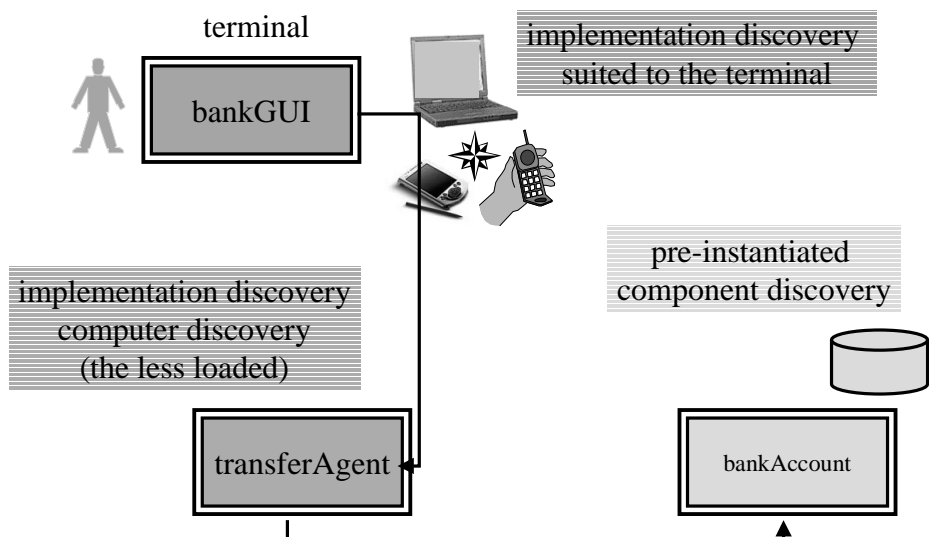
## Deployment input

- Application assembly and deployment descriptors (*written by the application developer*)
- Deployment context (*extracted by the deployment middleware*)
- Deployment infrastructure (*Application Service Provider*)
- Deployment policies (*deployment middleware*):
  - ◆ implementation suited to the terminal
    - user interface capabilities
    - memory and processing capabilities
    - software availability
  - ◆ computer suited to an implementation



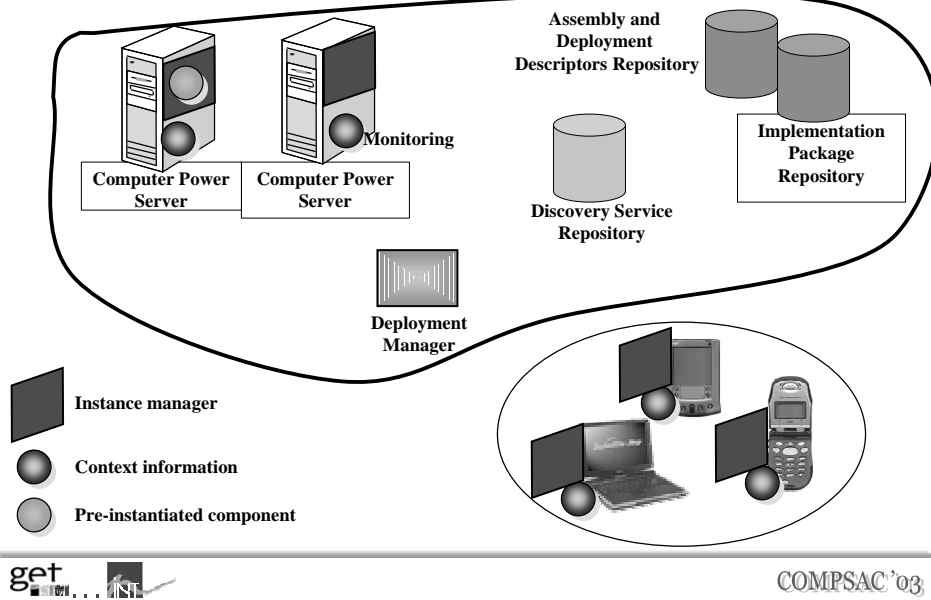
COMPSAC'03

## SDI Deployment choices



COMPSAC'03

## Deployment Infrastructure



## Discovery service (Trader CORBA)

- predefined service types (*package, instanceManager*)
  - ◆ typed **properties list**
  - ◆ inheritance of **service type**
- register typed offers (*systematic registration in the discovery service of pre-instantiated components, packages, instanceManager*)
- query with **constraint** expression and **preferences** expression in *OCL (OMG Constraint Language)*

## Discovery contract example

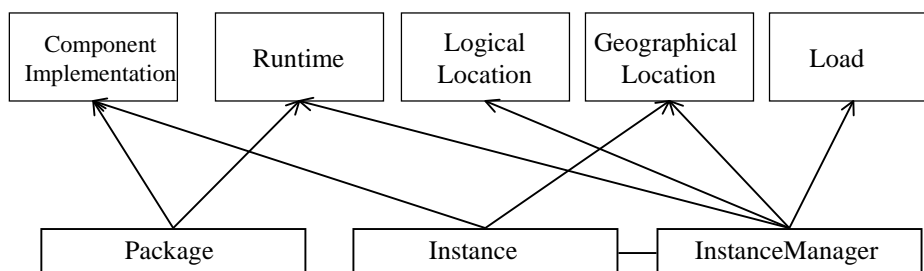
```
service Package:core::ComponentType, core::Runtime {  
    interface "IDL:omg.org/CORBA/Object:1.0";  
    property string name ;  
    property string url ;  
    property string entryPoint ;  
};
```

- Service types are defined with discovery contracts  
a contract generates the complex code to access the discovery service



COMPSAC'03

## Discovery Service types Inheritance



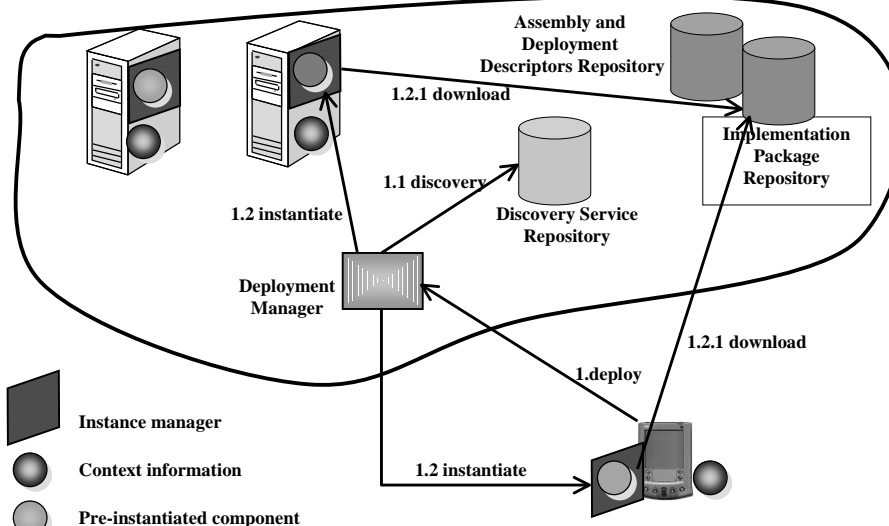
COMPSAC'03



## Generic search functions

- **discovery constraints and preferences dynamically generated**
  - ◆ **find a package for a given terminal**
    - >package list
  - ◆ **find an instance manager for a given package**
    - > instance manager list ordered by their respective load

## Deployment collaboration



## Implementation

---

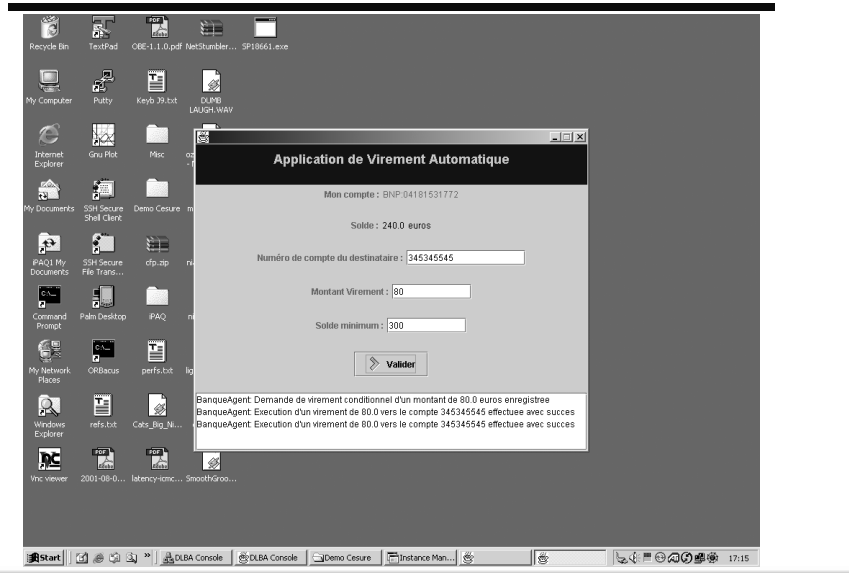
- CORBA middleware
- CORBA trading service
- OpenORB CORBA and trading service implementation
- OpenORB Service Type Language to define discovery contracts
- Java language
- multi OS (tested with Windows, Windows CE, linux)
- iPAQ 3600 and PC intel 450MHz as terminals
- PC intel 933MHz as Computer power
- 802.11 wireless network

## Bank GUI for terminal iPaq

---



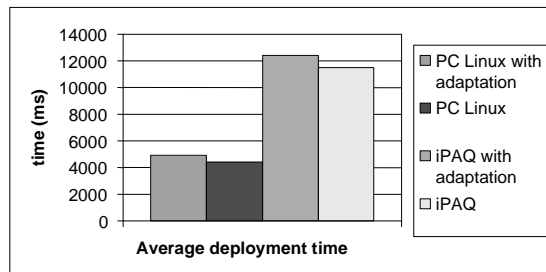
## Bank GUI for a desktop PC



COMPSAC'03

## Deployment measurements

- deployment evaluation on a desktop PC and an iPAQ
- low adaptation overhead :
  - ◆ adaptation = instance manager and packages found by the trader
  - ◆ without adaptation = packages and instance managers fixed



COMPSAC'03

## Deployment time analyze

terminal	total	Adaptation	UI instantiation (on the deployment manager)	UI instantiation (on the instance manager)	UI download (on the instance manager)
<b>IPAQ 3600</b>	<b>12.4</b>	<b>0.9</b>	<b>10.7</b>	<b>5.5</b>	<b>1.1</b>
<b>PC Intel 450MHz</b>	<b>5.0</b>	<b>0.5</b>	<b>3.5</b>	<b>3.3</b>	<b>&lt;0.1</b>

time in seconds

**Most of the deployment time is in instantiating the graphical interface**

**Not in downloading software, not in discovery requests**



COMPSAC'03

## Conclusion

- A first deployment prototype called SDI (Smart Deployment Infrastructure)
- Integrated with load balancing and load monitoring
  - ⇒ INT participation with OMG RFP on Load Balancing
- Implementation available as open source <http://picolibre.int-evry.fr/projects/sdi>
- Context aware and just in time multi-component application deployment
- Definition of a description language for context-aware deployment



COMPSAC'03

## **Perspectives**

---

- SDI in *CCM (CORBA Component Model)* CCM includes a deployment manager but with no context-adaptation
- Automatic registration and un-registration of resources in the CCM middleware
- Context aware deployment may also modify the structure of an application, e.g. add non functional components
- Context aware deployment and reconfiguration of the structure during the application

---

**Thank you for your attention**

**Questions**

## Level of adaptations

---

- **adaptation during the execution of the application**
  - ◆ reflexive middleware or applications
  - ◆ context-aware middleware or applications
  
- **SDI: adaptation during the installation of the application**
  - ◆ **context-aware deployment :**
    - install what you need in the execution context
    - discover available services
  - ◆ **reconfiguration of the deployment because of events**

## CCM Assembly descriptor

---

- **assembly descriptor**
- **+ deployment information :**
  - ◆ colocation of instances
  - ◆ URLs of archives of components
  - ◆ how to register instances (but with fixed values of properties)
  - ◆ query of instances (with fixed constraints)

## SDI Deployment descriptors

- assembly descriptor and deployment descriptor separated
- abstract deployment descriptor:
  - ◆ life cycle information about the instances (*installed* or *toInstall*)
  - ◆ if *toInstall* : instance location constraint or co-location constraint
  - ◆ variables related to the deployment context
- contextual deployment descriptor (dynamically generated with context information)
- concrete deployment descriptor (after the deployment)
  - ◆ for withdrawal of the assembly
  - ◆ for monitoring and reconfiguration of the assembly

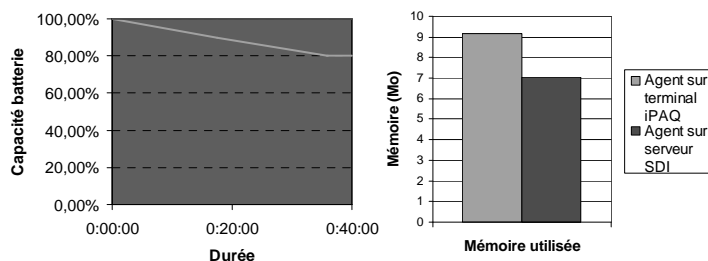


COMPSAC'03

## Adaptation et gain à l'exécution

### Terminal mobile : iPAQ (ressources limitées)

- Comparaison sur deux scénarios
  - 1<sup>er</sup>: *Agent installé sur terminal (iPAQ)*
  - 2<sup>ième</sup>: *Choix de l'hôte de l'agent par SDI (choix = serveur SDI)*
- Mesures
  - ◆ Capacité batterie pendant l'exécution
  - ◆ Mémoire utilisée sur l'iPAQ



COMPSAC'03