

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

TITRE DU MÉMOIRE

**MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN SCIENCES ET TECHNOLOGIES DE L'INFORMATION**

**PAR
PRÉNOM ET NOM DE L'AUTEUR**

DÉCEMBRE 2013

Ce mémoire a été évalué par un jury composé des personnes suivantes :

Dr. Prénom et Nom Président du jury

Dr. Prénom et Nom Membre du jury

Dr. Prénom et Nom Directeur de recherche

Mémoire accepté le : 20 novembre 2013

Dédicace

Remerciements

Table des matières

Liste des figures	ii
Liste des tableaux	iii
Liste des abréviations, sigles et acronymes	iv
Résumé	v
1 Introduction	1
1.1 CORBA	2
2 État des connaissances	3
2.1 Vue d'ensemble de l'architecture de CORBA	4
2.1.1 Description des interfaces	5
A Interface IDL	7
Bibliographie	7

Liste des figures

2.1	Architecture du bus	4
2.2	Contenu d'un IOR	5

Liste des tableaux

2.1	Exceptions possibles pour chaque fonction	6
-----	---	---

Liste des abréviations, sigles et acronymes

API Application Programming Interface

BOA Basic Object Adapter

CCM CORBA Component Model

CDR Common Data Representation

CORBA Common Object Request Broker Architecture

Résumé

Les producteurs de logiciels ont besoin de contrôler l'accès et l'utilisation des logiciels qu'ils fournissent afin d'obtenir une compensation pour leur utilisation. La méthode la plus utilisée est la licence. Pour s'assurer que le contrat entre les producteurs et les clients de logiciels est respecté il est nécessaire de mettre en place une architecture matérielle et logicielle. Dans le cadre de CORBA, l'OMG a publié une spécification pour un service de licence. Notre but est la conception d'un tel service conforme à la spécification CORBA. Les objectifs à atteindre dans l'élaboration de ce service sont une capacité d'évolution du service et la transparence pour l'utilisateur. Plusieurs recherches ont été effectuées et publiées sur la conception de différents services pour CORBA, mais aucune ne porte sur le service de licence. Notre contribution comble donc cette lacune. La conception du service est décrite en détail. L'utilisation de techniques de génie logiciel adéquates permet à notre service de prendre en compte de nouvelles licences et de nouveaux formats de licences postérieurs à la création du service, ce qui répond au critère d'évolution. Après évaluation du service, nous avons constaté, par rapport aux tests de contrôle, une augmentation de la durée de traitement d'une application cliente, en fonction de la taille de la licence. Cependant, cette augmentation est imperceptible à l'échelle humaine. Notre service est donc sans impact significatif sur le démarrage et l'utilisation de l'application cliente, ce qui répond au critère de transparence.

Abstract

Software producers need to control access and use of their software products to obtain compensation for their usage. Licensing is the most used method. To be sure that the contract between software producers and clients is respected, it is necessary to set up a hardware and software architecture. For CORBA, OMG has published a license service specification. Our goal is to design such a service and to stay in conformance with the specification. The set goal is to design an evolvable service which is transparent to the end user. Some research has been done and published on other CORBA services designs, but nothing has been done on a license service. Thus our contribution fills the gap. The design is described in details. The use of sound software engineering techniques allows the service to be able to take care of new licenses and new license formats that could be developed later on. This answers the evolution criteria. After evaluation, we have observed, compared with the control test, an increase in treatment time from a client application, based on license size. But, this increase is not perceptible at the human scale. The service does not significantly impact the start and use of client applications, which answers the transparency criteria.

Chapitre 1

Introduction

Nous allons montrer différentes méthodes dans les quelques lignes qui suivent pour que vous ayez tous les éléments utiles pour écrire votre document. Regarder aussi les différents titres et comment ils apparaissent dans la table des matières.

Pour commencer un nouveau paragraphe laissez une ligne entre le nouveau et l'ancien.

Pour écrire en *italique*.

Pour changer de police : la **sans**, la **télétype**, les petites majuscules CE TEXTE EN PETITE MAJUSCULE.

Pour écrire « mise en œuvre », les guillemets et l'espace insécable.

N'oubliez pas de mettre un espace insécable avant ?, !, :, ;.

Voici différentes listes :

- achat de licences en groupe (économie d'échelle),
- allocation des licences non utilisées aux utilisateurs qui en ont réellement besoin (diminution des achats).

1. `obtain_producer_specific_licence_service`
2. `start_use`
3. `check_use`
4. Requête au système de licence
5. Transmission au service d'événement (**push**)
6. Notification d'un événement (**push**)
7. `check_use`

licences prérequis : où une licence n'est accordée que si une autre a déjà été accordée précédemment.

licences corequises : où un ensemble de licences doivent toutes coexister pour autoriser le fonctionnement d'une application.

licences exrequises : où un ensemble de licences peuvent être accordées seulement si d'autres ne sont pas utilisées.

licences génériques dépendantes : où les dépendances d'un ensemble de licences sont décrites à travers des expressions de contraintes.

1.1 CORBA

Exemple de citation, pour la bibliographie, et de références, vers d'autres sections ou chapitre, vers des tableaux, vers des figures.

CORBA [2] (*Common Object Request Broker Architecture*) est un standard [3, 5, 8] réalisé par l'OMG (*Object Management Group*) (référence vers le chapitre 1) dans le but de permettre l'inter-opérabilité entre des systèmes objets répartis et ce quelque soit les architectures mises en jeux (support réseaux, machines clientes et serveurs, systèmes d'exploitation, langages de programmation). L'OMG [9, 6] est un organisme composé de plus 800 membres intéressés dans le développement objet. On retrouve de grandes entreprises comme IBM [8, 7], Microsoft [1, 4], Hewlett Packard, etc.

CORBA est avant tout une architecture que l'on appelle OMA (*Object Management Architecture*) (figure 2.1). Celle-ci se compose de l'ORB (*Object Request Broker*), de services objets (tableau 2.1), d'utilitaires communs, d'interfaces de domaines et d'interfaces applicatives.

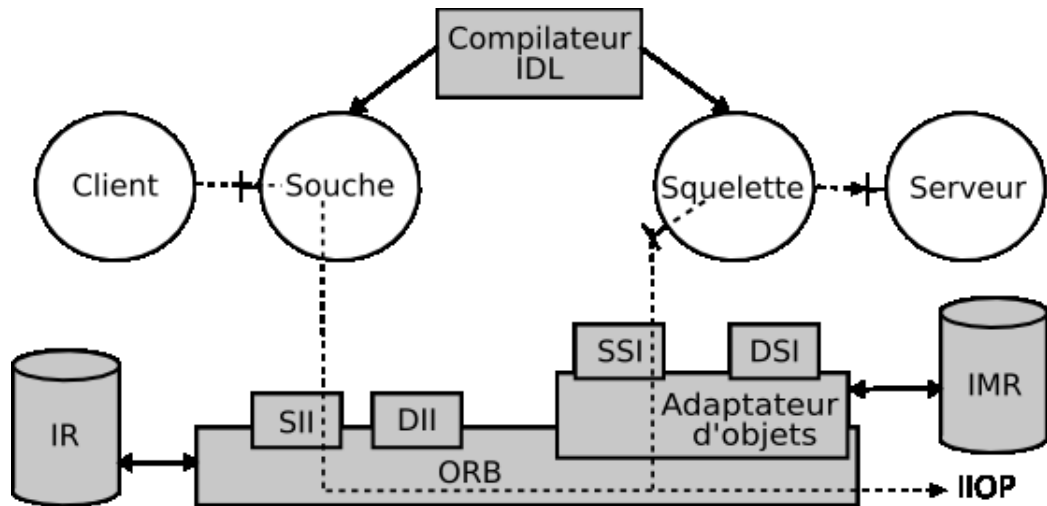
Chapitre 2

État des connaissances

CORBA est un vaste terrain de recherche, allant de la conception de services tel que le service d'événement [2] par exemple, à des réalisations temps réels [3]. Nous avons trouvé que l'entreprise Segue a développé un service de licence appelé SilkMeter [2]. Ce service est offert en tant que produit commercial, le code source n'est pas ouvert et aucune publication n'a été effectuée le concernant. L'interface IDL est fournie avec la projection en C++ et java. Ce produit était disponible pour les plate-formes Windows et Unix, et supporte Orbix et Visibroker. Actuellement, dans l'offre de produits de la compagnie Segue, plus aucune mention de ce produit n'est disponible sur le site web.

2.1 Vue d'ensemble de l'architecture de CORBA

FIGURE 2.1 – Architecture du bus



Insertion d'un contenu *verbatim*. Ce qui est écrit dans votre fichier source latex est écrit tel quel dans le fichier généré.

FIGURE 2.2 – Contenu d'un IOR

```

Repo Id:   IDL:GenericServer:1.0

IIOP Profile
  Version:  1.0
  Address:   inet:localhost.localdomain:12122
  Location:  corbaloc::localhost.localdomain:12122//20524/1117736168/
%5f0
          Key:  2f 32 30 35 32 34 2f 31 31 31 37 37 33 36 31 36 /20524/1
11773616
                  38 2f 5f 30                                8/_0

Multiple Components Profile
  Components:  Native Codesets:
                normal: ISO 8859-1:1987; Latin Alphabet No. 1
                wide:  ISO/IEC 10646-1:1993; UTF-16, UCS Transformation
Format 16-bit form
          Key:  (empty)

```

2.1.1 Description des interfaces

Il est clairement indiquée dans la spécification que toutes les interfaces sont obligatoires pour toutes les implémentations. En effet, ne pas respecter les interfaces réduirait l'utilité du service, car il ne serait plus portable.

Une sous sous section

Ce genre de sous-section n'apparaît pas dans la table des matières.

Les tableaux et figures peuvent être déplacés par latex, sur une page à part ou en fin de chapitre, s'il n'arrive pas à l'intégré dans du texte. Cela arrive lorsqu'il y a trop d'image ou de tableaux par rapport au texte.

TABLE 2.1 – Exceptions possibles pour chaque fonction

Opérations	Exceptions
obtain_producer_specific_licence_service	InvalidProducer, InvalidParameter
start_use	InvalidParameter, ComponentNotRegistered
check_use	InvalidParameter
end_use	InvalidParameter

Annexe A

Interface IDL

Exemple avec des tabulations :

```
<?xml version="1.0" encoding="UTF-8"?>
<license name="UQOLicenseServiceClient" version="0.1">
  <permit>
    <username value="aymeric"/>
  </permit>
</license>
```

Exemple avec des espaces :

```
<?xml version="1.0" encoding="UTF-8"?>
<license name="UQOLicenseServiceClient" version="0.1">
  <permit>
    <username value="aymeric"/>
  </permit>
</license>
```


Bibliographie

- [1] BANSAL, A. Shared objects for the object disoriented! <http://www-106.ibm.com/developerworks/library/l-shobj/>, April 2001.
- [2] BASTIDE, R., PALANQUE, P., SY, O., AND NAVARRE, D. Formal specification of CORBA services : experience and lessons learned. In *Proceedings of the conference on Object-oriented programming, systems, languages, and applications* (2000), ACM Press, pp. 105–117.
- [3] GAMMA, E., HELM, R., JOHNSON, R., AND VLISSIDES, J. *Design patterns : elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., 1995.
- [4] JOHNSON, A. libxml++. <http://libxmlplusplus.sourceforge.net>, May 2005.
- [5] NARASIMHAN, P., MOSER, L. E., AND MELLIAR-SMITH, P. Using interceptors to enhance CORBA. *Computer* 32, 7 (July 1999), 62–68.
- [6] OBJECT MANAGEMENT GROUP. *Licensing service specification*. Object Management Group, April 2000.
- [7] PILHOFFER, F. *Design and Implementation of the Portable Object Adapter*. PhD thesis, Fachbereich Informatik, Johann Wolfgang Goethe-Universität, June 1999.
- [8] SCHMIDT, D. C., AND CLEELAND, C. *Applying a pattern language to develop extensible ORB middleware*. Cambridge University Press, 2001, pp. 393–438.
- [9] VINOSKI, S. New features for CORBA 3.0. *Communications of the ACM* 41, 10 (October 1998), 44–52.