

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université de Batna
Faculté des sciences de l'ingénieur
Département d'informatique

Université de Batna
Rue Chahid Boukhlouf Mouhamed El-hadi
05000 Batna www.univ-batna.dz

Thèse de Magistère

Thème

**Définition d'un style architectural pour la
description de systèmes logiciels à base de
composants de type COTS, selon une
approche « services WEB »**

présentée par

Kamel MANSOURI

pour obtenir le diplôme de Magistère spécialité Informatique

Soutenue publiquement le 13 Novembre 2005 devant le jury composé de :

Dr. M. Benmouhammed	M.C. Université de Constantine, Président,
Dr. M.S.Kheriddine	C.C. Université de Batna, Rapporteur,
Dr. A.Zidani	M.C. Université de Batna, Examineur,
Dr. M.K.Kholladi	M.C. Université de Constantine, Examineur.

SEPTEMBRE 2004

ABSTRACT

The development of big software applications is oriented toward the integration or interoperation of existing software components (like COTS and legacy systems) . This tendency is accompanied by a certain number of drawbacks for which classical approaches in software composition cannot be applied and fail. COTS-based systems are built in ad-hoc manner and it is not possible to reason on them no more it is possible to demonstrate if such systems satisfy important properties like Quality Of Service and Quality Attributes.

The recent works issued in web field allow the definition and the use of a complex web service architecture. Languages such as WSFL, XLANG and BPEL4WS support these architectures called Services Oriented Architectures.

The definition of software systems using these languages benefits some existing technical solutions such as SOAP, UDDI, etc., that permit the distribution, the discovery and the interoperability of web services.

However, these languages do not have any formal foundation. One cannot reason on such architectures expressed using such languages: properties cannot be expressed and the system dynamic evolution is not supported.

On the other hand, software architecture domain aims at providing formal languages for the description of software systems allowing to check properties (formal analyses) and to reason about software architecture models.

In this work, we proposes a formalisation of COTS-based system (their structure, their behaviours) using architectural styles. The ADL used is π -ADL (based on the π -calculus, supporting style description). The memory will also present our approach consisting in refining an abstract architecture to an executable and services-oriented one.

Keywords : COTS-based system, software architecture, styles, services oriented architecture.

Table des matières

CHAPITRE I. INTRODUCTION	1
I.1 Introduction a la problématique	3
I.2 Pourquoi les systèmes à base de COTS ?	3
I.3 Les COTS et les services WEB	3
I.4 L'approche centrée architecture	4
I.4.1 Présentation	4
I.4.2 Les enjeux des travaux sur l'architecture logicielle	5
I.4.2.1 Processus de conception devenu inefficace	5
I.4.2.2 Evolution vers un processus centré architecture	5
I.4.3 La notion de style architectural	6
I.4.3.1 Langage de Description Architecturale (LDA)	7
I.5 Problématique	7
I.6 Organisation du document	7
CHAPITRE II. ETAT DE L'ART	9
II.1 Les systèmes à base de COTS	10
II.1.1 Définition du COTS	10
II.1.2 Comparaison entre COTS et composant	11
II.1.3 Classification de COTS	11
II.1.4 Modélisation des systèmes à base de COTS	11
II.1.4.1 UML (Unified Modeling Language)	12
II.1.4.2 Les Langages de Description Architecturale (LDA)	12
II.1.5 Technologies et normes pour l'intégration de composant	12
II.1.5.1 SUN (EJB)	13
II.1.5.2 Microsoft (COM/DCOM)	14
II.1.5.3 OMG (CORBA)	14
II.1.6 Problèmes d'intégration des COTS	15
II.1.7 Bilan	16
II.2 Les services WEB	18
II.2.1 Pourquoi les services WEB dans notre approche?	18
II.2.2 Définition d'un service WEB	18
II.2.3 Langage de description de service WEB	19
II.2.3.1 La Grammaire de WSDL	19
II.2.4 SOAP	20
II.2.4.1 Définition	20
II.2.4.2 Grammaire SOAP	20
II.2.5 UDDI	21
II.2.5.1 Définition	21
II.3 Les workflow	22
II.4 Quelque workflow	24
II.4.1 WSFL d'IBM	24
II.4.1.1 Définition	24
II.4.1.2 Un exemple WSFL	24
II.4.1.3 Présentation générale de la syntaxe de WSFL	26
II.4.2 XLANG de Microsoft	27
II.4.2.1 Définition	27
II.4.2.2 Les actions XLANG	28
II.4.2.3 Les actions de contrôle XLANG	28
II.4.2.4 un exemple XLANG	30
II.4.3 BPEL4WS de IBM, Microsoft et BEA	30
II.4.3.1 Définition	30
II.4.3.2 Structure de BPEL4WS	31
II.4.3.3 Traitement d'erreurs	31

II.4.3.4 Exemple	31
II.4.4 Autres langages.....	34
II.4.4.1 WSCI de SUN	34
II.4.4.2 WSCL de HP	34
II.4.4.3 BPMLde BPMI.....	35
II.4.5 Bilan	35
II.5 Le serveur Biztalk	37
II.5.1 Introduction.....	37
II.5.2 Le moteur BizTalk Server	39
II.5.3 Relier les applications.....	40
II.5.3.1 Envoyer et recevoir des Messages: les adaptateurs.....	40
II.5.3.2 Traitement des messages : les pipelines.....	41
II.5.3.3 Choix des Messages: les Souscriptions (subscriptions)	42
II.5.4 Définition d'un Processus Métier.....	42
II.5.4.1 Orchestration	43
II.5.4.2 Le moteur des règles de gestion	45
II.5.5 Bilan	45
II.6 Les langages de description architecturale	47
II.6.1 Définition	47
II.6.1.1 Style.....	48
II.6.2 Présentation de quelque ADL	49
II.6.2.1 UNICON-2.....	49
II.6.2.2 AESOP.....	52
II.6.2.3 ARMANI	55
II.6.2.4 π -ADL.....	58
II.6.3 Bilan des ADL.....	60
II.6.4 Le langage formel π -CALCUL	62
II.6.4.1 Introduction :	62
II.6.4.2 Concept de Mobilité	62
II.6.4.3 Syntaxe	63
II.6.4.4 Sémantique	64
II.6.4.5 Le π -calcul asynchrone	64
II.6.4.6 Le π -calcul polyadique	65
II.6.4.7 Le π -calcul d'ordre supérieur.....	65
II.6.4.8 Bilan.....	65
CHAPITRE III. PROPOSITIONS.....	67
III.1 Rappel de la problématique	69
III.2 Proposition d'une architecture de référence.....	70
III.3 Définition des services principaux fournis par UDDI-COTS.....	71
III.4 Description des éléments de l'architecture.....	72
III.4.1 Service de publication (UDDI).....	72
III.4.2 Service Contrôle de processus (UC).....	75
III.4.3 Service orientation (OR)	77
III.4.4 Représentation externe du COTS	78
III.4.5 L'attachement des composants.....	82
III.5 Génération de code (π -ADL à XLANG)	83
III.6 Bilan	85
III.7 Formalisation d'un exemple avec nos styles architecturaux	87
III.7.1 Description de l'exemple.....	87
III.7.2 Création de l'architecteur du système	87
III.7.2.1 Projection du système selon la topologie	87
III.7.2.2 Description du système en π -ADL	88
III.7.2.3 Raffinement du système.....	89
III.7.3 Bilan	93
CHAPITRE IV. BILAN GENERALE	95
IV.1 Bilan de travail.....	97
IV.2 Conception de système	97

IV.3 Le raffinement	98
IV.4 Conclusion.....	98
REFERENCES BIBLIOGRAPHIQUES	101
ANNEXE A : DESCRIPTION DU STYLE	107
ANNEXE B : L'UC DE L'EXEMPLE EN XLANG	115
ANNEXE D : PUBLICATION REALISEE	125