



Configuration Interface for MESSage ROUTing

Cahier des Charges

Date : 05/04/07
Version : 1.1
Statut : diffusable

BAGNARD Natacha
Auteurs : **FOROT Julien**

Table des révisions

<i>Version</i>	<i>Date</i>	<i>Modifications</i>
0.1	17/01/07	Création du document
0.2	01/02/07	Modification du document suite à une lecture de Jérôme Camilleri
0.3	08/02/07	Modification du document suite à une lecture de Jérôme Camilleri (fiche CDC_Camilleri_08022007.odt)
0.4	01/03/07	Modification du document suite à une réunion (fiche reunion_01032007.odt)
0.5	14/03/07	Modification du document suite à une lecture de Jérôme Camilleri (fiche CDC_Camilleri_13032007.odt)
1.0	20/03/07	Validation du document par Jérôme Camilleri
1.1	05/04/2007	Modification suite au 1er audit avec Martine Tasset (fiche Audit1_04042007.odt)

Table des matières

1 Introduction.....	4
1.1 But du document.....	4
1.2 Portée du document.....	4
1.3 Définition, Acronymes et Abréviations.....	4
1.4 Organisation du document.....	7
2 Description générale.....	7
2.1 Contexte.....	7
2.2 Analyse de l'existant.....	8
2.2.1 Généralités.....	8
2.2.2 Fonctionnalités.....	9
2.2.3 Description technique.....	10
3 Analyse des besoins.....	10
3.1 Utilisateurs du futur système.....	10
3.2 Besoins fonctionnels.....	10
3.2.1 Acteur.....	11
3.2.2 Cas d'utilisation.....	11
3.3 Besoins non fonctionnels.....	13
3.3.1 Refonte de l'architecture.....	13
3.3.2 ESB supportés.....	13
3.3.3 Composants implémentés.....	14
4 Contraintes.....	15
4.1 Exigences de documentation.....	15
4.2 Exigences de qualité.....	15
4.3 Contraintes temporelles et ressources.....	16
4.4 Contraintes techniques.....	16
4.4.1 Environnement de développement.....	16
4.4.2 Environnement logiciel de l'application	16
4.4.3 Framework.....	16
5 Évolutions envisageables.....	16

1 Introduction

1.1 But du document

Ce document a pour but de définir le problème soumis par le client et auquel le projet logiciel doit apporter une solution. Il devra décrire précisément les besoins (fonctionnels et non fonctionnels) et exigences des utilisateurs futurs, les fonctionnalités attendues ainsi que les contraintes techniques. Une série de discussions, de réunions et de réflexion avec le MOAd permettra de rédiger ce cahier des charges.

1.2 Portée du document

Ce cahier des charges, une fois validé par le client, servira de base à la spécification et à la conception du produit à livrer.

Ce document est destiné:

- au MOAd : Jérôme Camilleri
- à la consultante : Martine Tasset
- au jury du Master 2 Pro GI pour l'évaluation du stage
- à l'équipe projet : Natacha Bagnard et Julien Forot

Ce document permettra d'avoir une définition unique et précise du produit/prototype. Il servira de base :

- à l'évaluation du produit final.
- à la rédaction des plans de tests d'acceptation.
- à la rédaction du plan de développement logiciel.

1.3 Définition, Acronymes et Abréviations

Le tableau suivant donne une définition des acronymes et abréviations spécifiques à ce projet.

Terme	Définition
BC	Les B inding C omponents fournissent des connections pour des services externes à l'environnement JBI (protocoles de communications – ftp, soap, ..) grâce à un mécanisme de normalisation. Le BC normalise le message qui lui vient de l'extérieur sous format WSDL 1.1 ou WSDL 2.0.

Terme	Définition
Eclipse	Eclipse est un environnement de développement (IDE) Open Source principalement destiné au langage JAVA et fonctionnant à base d'extensions (plugins).
EMF	Eclipse Modeling Framework est un framework Eclipse qui permet de générer une partie du code d'un programme à partir de son modèle de données.
ESB	<p>L'Enterprise Service Bus ou ESB est une technologie informatique intergicielle permettant à des applications hétérogènes d'interagir au travers de services standards qu'elles mettent à disposition.</p> <p>Elle s'appuie sur les standards suivants :</p> <ul style="list-style-type: none"> • Services Web (SOAP, WSDL, etc.) • Connecteurs d'applications JCA • Langage XML.
JB1	<p>Java Business Integration est une norme édictée dans la JSR 208 dans le cadre du Java Community Process.</p> <p>Le problème initial est l'intégration de données en provenance de sources différentes au sein d'un Système d'informations composé d'applications disparates. Les ESB font partie des solutions existantes.</p> <p>JB1 est une spécification normalisant ces intégrations via un jeu d'API permettant à tout fournisseur les respectant, de pouvoir se connecter à un container JB1 pour échanger des messages avec le reste du SI.</p>
GEF	Graphical Editing Framework est un framework Eclipse qui permet de générer une partie du code d'une interface graphique à partir du modèle de celle-ci.
GMF	Graphical Modeling Framework est un framework Eclipse fournissant un pont entre les frameworks EMF et GEF.
Package JB1	Archive Zip contenant toutes les informations nécessaires au déploiement et au fonctionnement d'une application JB1.
Petals	<p>Petals (ObjectWeb) est un produit Open Source ESB qui implémente la norme JB1 (Java Business Integration : cf spécification JSR 208).</p> <p>Petals est un serveur d'applications Java. Il fournit un support des composants suivants : SOAP Binding, JMS, Mail, FTP, XQuare, POJO, XSLT, CSV, Forward, BPEL Orchestra.</p>
Plugin	Application étendant les capacités et les fonctionnalités d'une autre application.

Terme	Définition
SA	Service Assembly. Ensemble des Service Unit et des informations concernant les connexions entre les composants.
SE	Les Service Engine fournissent des services de transformations et de « logique business » aux autres composants. Ne sont pas directement en contact avec des composants extérieurs au JBI, contrairement aux BC.
ServiceMix	ServiceMix (Apache) est un produit Open Source ESB qui implémente de la norme JBI (Java Business Integration : cf spécification JSR 208). ServiceMix est un composant léger qui peut être intégré à n'importe quel serveur d'applications Java. Ses fonctionnalités se rapprochent de celles d'un broker ou d'un serveur d'intégration. Il fournit un support des composants suivants : BPEL, JCA, Transformation / XSLT, XSQL, FTP, HTTP, JMS, RSS, Email.
SOA	SOA est le sigle de Service Oriented Architecture ou Architecture Orientée Services. La notion de SOA est un modèle d'interaction applicative qui met en œuvre des services (composants logiciels). Le service est une action exécutée par un « fournisseur » (ou « producteur ») à l'attention d'un « client » (ou « consommateur »).
SOAP protocol	Simple Object Acces Protocol. Il s'agit d'un modèle décrit par une recommandation du W3C. Celui-ci permet l'appel de méthodes sur un objet distribué à travers le réseau Internet. Les paramètres nécessaires à l'appel sont transférés via le protocole HTTP par un message XML si bien qu'aucun problème de Firewall ou Proxy ne se pose.
SU	Service Unit. Ensemble des fichiers utilisés par une instance d'un composant : fichiers de configuration du composant et fichiers annexes utilisés par le composant (feuille de style XSLT par exemple pour une transformation).

Terme	Définition
Workflow	<p>Un workflow est un flux d'informations au sein d'une organisation, comme par exemple la transmission automatique de documents entre des personnes.</p> <p>On appelle « workflow » (« flux de travail ») la modélisation et la gestion informatique de l'ensemble des tâches à accomplir et des différents acteurs impliqués dans la réalisation d'un processus métier. Le terme de « workflow » pourrait donc être traduit en français par « gestion électronique des processus métier ». De façon plus pratique, le workflow décrit le circuit de validation, les tâches à accomplir entre les différents acteurs d'un processus, les délais, les modes de validation, et fournit à chacun des acteurs les informations nécessaires pour la réalisation de sa tâche.</p>
XML	Extensible Markup Language. Métalangage extensible dérivé de SGML permettant la structuration des données.
XSL	Extensible Stylesheet Language. Spécifications du W3C en vue de permettre une création de documents HTML ou XML qui sépare le fond et la forme, pour une clarté et une maintenance des sites plus simples. C'est un descendant de CSS.
XSLT	XSL Transformations : Langage dédié à la transformation de données XML, faisant partie de XSL.

1.4 Organisation du document

La suite de ce document donne une vue générale de l'application logicielle qui fait l'objet de ce travail. Il donne la description des besoins et fonctions du produit demandé.

Le paragraphe 2 décrit le contexte dans lequel le projet se déroule et l'application existante. Il décrit également les utilisateurs visés par cette application.

Le paragraphe 3 détaille les besoins fonctionnels et non fonctionnels de l'application.

Le paragraphe 4 traite des contraintes de développement et de maintenance.

Le paragraphe 5 dresse une liste non exhaustive des évolutions envisagées/envisageables pour l'application.

2 Description générale

2.1 Contexte

Ce projet est réalisé dans le cadre du projet BSOA (Bull Service Oriented Architecture). BSOA est une plate-forme middleware Open Source développée par Bull R&D. Celle-ci devra

permettre la mise en oeuvre de systèmes d'informations flexibles grâce à l'approche SOA (Service Oriented Architecture). BSOA comprend entre autres un workflow de personnes (Bonita), un workflow de services Web inter-applicatifs (Orchestra), une console d'administration, un IDE et un portail Web (ExoPlateforme) ; le tout étant déployé sur le serveur d'applications J2EE JOnAS.

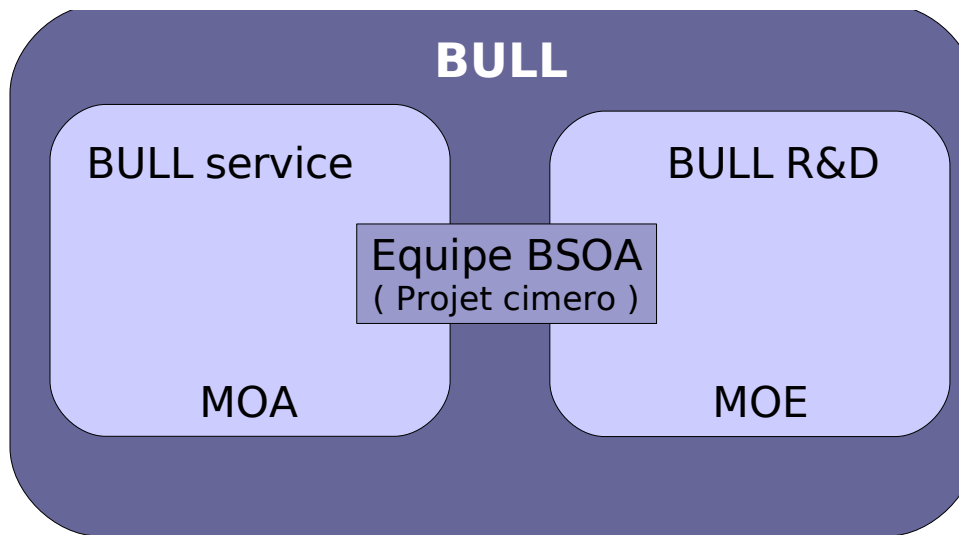


Illustration 1: Contexte du projet CIMERO

Dans le cadre du projet BSOA, deux anciens Master2 Pro GI ont développé une solution logicielle, CIMERO (Configuration Interface for MESSage ROuting), permettant de faciliter l'utilisation de ServiceMix et la configuration graphique d'un flux de messages. Le but de ce stage est de réaliser une nouvelle version de CIMERO.

2.2 Analyse de l'existant

2.2.1 Généralités

CIMERO se compose actuellement de 2 plugins distincts : l'éditeur graphique (editor) et le plugin ServiceMix.

CIMERO editor offre à l'utilisateur la possibilité de générer un fichier de configuration sous format XML ou un package JBI sous forme d'archive, représentant tout deux le graphe créé par l'utilisateur mais sous une forme différente. Via un menu contextuel, il est possible de déployer le package JBI généré sur un serveur ServiceMix, si celui-ci est lancé.

CIMERO ServiceMix permet de lancer à partir de l'environnement Eclipse un serveur ServiceMix. L'utilisateur peut ainsi démarrer ou arrêter ServiceMix.

2.2.2 Fonctionnalités

Le projet CIMERO développé par deux anciens étudiants Master2 Pro Génie Informatique possède déjà de nombreuses fonctionnalités. Le diagramme de cas d'utilisation suivant présente les

fonctionnalités proposées par Cimero¹.

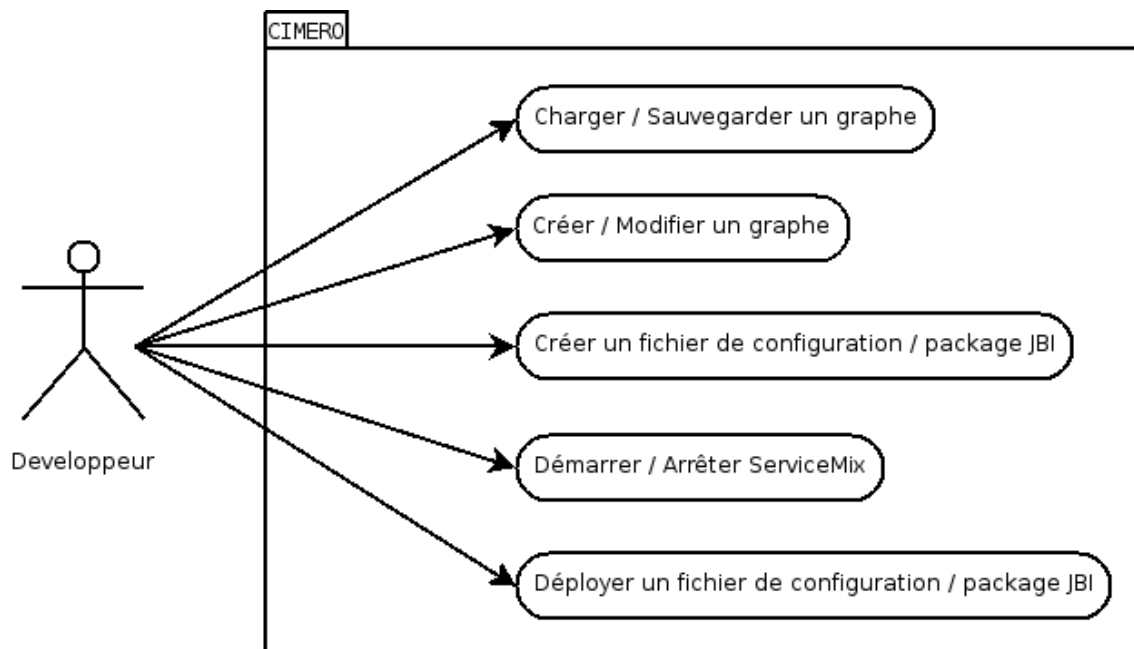


Illustration 2: Use Case général de CIMERO version 1

a) Acteurs

Le développeur représente tous les utilisateurs qui désirent créer ou éditer un graphe destiné à la configuration de ServiceMix. Toutes les actions présentées ci-dessus lui sont accessibles.

b) Cas d'utilisation

➤ Créer un graphe et modifier un graphe

L'utilisateur peut créer une nouvelle représentation du flux sous la forme d'un graphe. Il peut ensuite le modifier en y ajoutant, paramétrant ou supprimant des composants. Il peut ajouter des composants disponibles dans une boîte à outils, modifier les propriétés d'un composant précédemment inséré ou choisir de le supprimer.

➤ Charger / sauvegarder un graphe

Il est possible de sauvegarder un graphe. Son état actuel (les composants, la configuration, etc...) est sauvegardé et il est possible, par la suite, de charger ce graphe et ainsi retrouver son état au moment de la sauvegarde.

➤ Créer un fichier de configuration/package JBI

L'utilisateur, après avoir édité un graphe, peut décider de créer le fichier de configuration ou un package JBI correspondant à ce graphe. Les fichiers générés sont spécifiques à ServiceMix.

➤ Démarrer / arrêter ServiceMix

¹ Voir Cahier des charges de la version précédente : CahierDesChargesV2.5.sxw.

L'utilisateur peut démarrer (et arrêter) ServiceMix sans quitter son environnement de développement (Eclipse). Une fois ServiceMix démarré, il est possible de déployer des fichiers de configuration.

➤ **Déployer un fichier de configuration/package JBI**

L'utilisateur peut déployer un fichier de configuration ou un package JBI qu'il a généré au préalable pour un flux donné. Les services déployés sont alors actifs.

2.2.3 Description technique

Ce logiciel fonctionne sous l'IDE Eclipse 3.1. Il est écrit en langage JAVA à l'aide du JDK 1.4. Il fonctionne quelque soit l'OS utilisé : Windows, Debian...

3 Analyse des besoins

3.1 Utilisateurs du futur système

Les utilisateurs finaux de CIMERO sont des designers de flux. On distingue 2 catégories principales d'utilisateurs cibles pour le logiciel :

- **Les utilisateurs novices** : Ils utilisent un outil comme CIMERO pour découvrir JBI. Ils n'utilisent pas les fonctions avancées du logiciel.
- **Les utilisateurs expérimentés** : Ils utilisent un outil comme CIMERO pour produire des applications JBI qu'ils modifieront ensuite manuellement si besoin. Des fonctionnalités avancées décrites dans la partie suivante seront mises à leur disposition.

3.2 Besoins fonctionnels

La nouvelle version de CIMERO propose de nouvelles fonctionnalités. Le diagramme des cas d'utilisation suivant présente l'ensemble des fonctions qui seront accessibles à l'utilisateur, les décorations précisent le niveau minimum de compétence des utilisateurs cibles. Les fonctionnalités concernant les fichiers de configurations (déployer et générer) ne seront plus disponibles car ces fichiers ne sont pas standards.

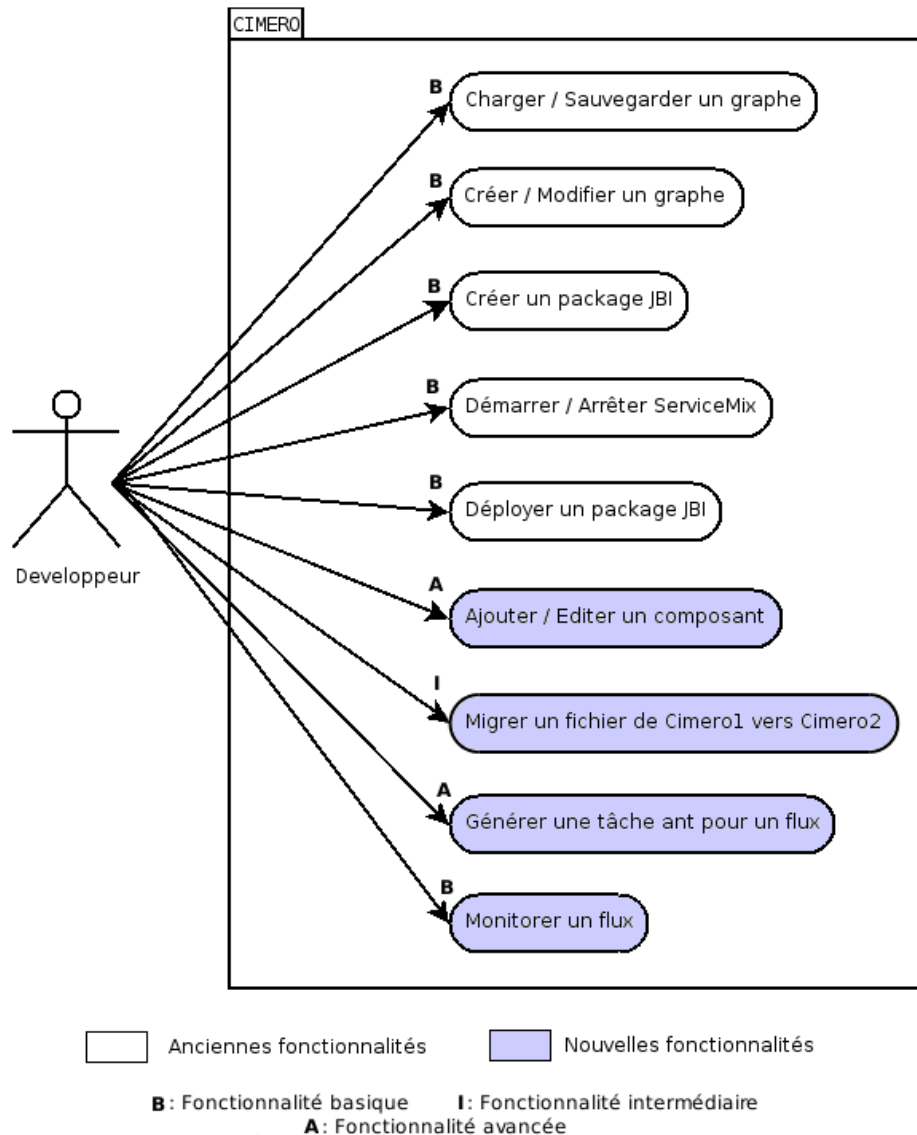


Illustration 3: Use Case général de CIMERO version 2

3.2.1 Acteur

Le type d'acteur concerné ne change pas. Voir section 2.2.1, page 9 de ce document.

3.2.2 Cas d'utilisation

Les cas d'utilisation conservés de la version précédente sont décrits section 2.2.2, page 9 de ce document.

➤ Ajouter/Éditer un composant

L'utilisateur pourra ajouter de nouveaux composants standards en décrivant ses propriétés, grâce à un outil graphique, inclus dans CIMERO. Ces propriétés permettront de construire un modèle du composant. Ceux-ci seront ensuite inclus dans la palette et utilisables dans un graphe.

Lors de l'utilisation d'un de ces composants dans un graphe, les valeurs des propriétés du modèle devront être définies.

L'utilisateur aura également la possibilité d'éditer un composant déjà présent dans la palette via cet outil graphique, ceci afin de permettre au logiciel de gérer l'évolution des composants existants comme l'ajout de nouvelles propriétés, la suppression de propriétés, etc ...

Une livraison partielle de cette fonctionnalité pourra être proposée dans la mesure où les aspects techniques ne sont pas totalement maîtrisés.

➤ **Migrer un fichier de CIMERO Version 1 vers CIMERO Version 2**

L'utilisateur pourra importer un fichier XML correspondant à la traduction d'un fichier CIMERO Version 1 (.cimero) dont il souhaite générer le .cimero2 associé. Le fichier XML devra être de la forme attendue et utiliser des composants disponibles dans CIMERO, sous peine de ne pas pouvoir être importé. Si ces conditions sont vérifiées l'utilisateur pourra alors visualiser le graphe correspondant et l'éditer.

Le logiciel proposera un mode dégradé dans le cas où le fichier est correct mais que des composants ne sont pas supportés par CIMERO : Représentation graphique commune à tous les composants non reconnus, aucune visualisation des propriétés et donc pas d'édition de leurs valeurs possible.

➤ **Générer une tâche Ant pour un flux**

L'utilisateur pourra choisir de générer une tâche Ant. Voici un exemple de la tâche ant qui devra être générée pour le graphe suivant :

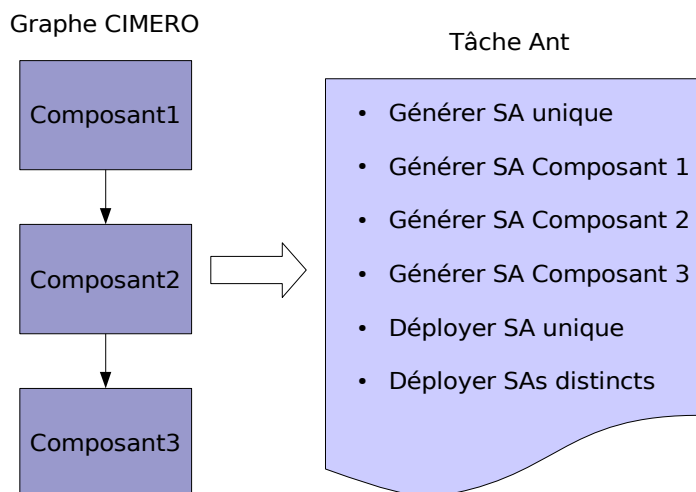


Illustration 4: Exemple de tâche Ant pour un graphe donné

✓ *Générer SA unique*

Cette cible permet de créer un package JBI correspondant à un SA unique, contenant les Sus

des différents composants.

✓ *Générer SA Composant n*

Cette cible permet de créer un package JBI correspondant à un SA contenant un SU, qui instancie le composant n. Les packages sont indépendants les uns des autres.

✓ *Déployer SA unique*

Cette cible permet de déployer le package JBI généré grâce à la cible « Générer SA unique ».

✓ *Déployer SAs distincts*

Cette cible permet de déployer les packages JBI générés grâce aux cibles « Générer SA Composant n ». L'utilisateur devra modifier manuellement la tâche Ant, si il souhaite préciser sur quel serveur chaque package doit être déployé, sinon tous les packages seront déployés sur l'ESB défini par défaut.

➤ **Monitorer un flux**

L'utilisateur aura la possibilité de passer en mode test. Il pourra alors visualiser, sur le graphe CIMERO, les échanges de messages dans l'application JBI déployée sur le bus grâce à des capteurs intégrés à l'ESB ServiceMix. Pour rendre cette fonction accessible aux autres ESB, il faudra implémenter des capteurs spécifiques pour chacun d'eux. Les tests et la mise au point des applications JBI développées grâce à CIMERO seront donc facilités par cet outil, car il permettra de suivre le parcours des messages et leurs contenus, et donc de repérer les composants qui posent problème.

3.3 Besoins non fonctionnels

3.3.1 Refonte de l'architecture

Une refonte de l'architecture globale du logiciel est nécessaire, pour des questions de maintenabilité et d'évolutivité. Le framework GMF devra être utilisé dans le cadre de cette refonte.

3.3.2 ESB supportés

La première version de CIMERO est entièrement dépendante de l'ESB ServiceMix. Bull souhaite pour la deuxième version de l'application être le plus indépendant possible de l'ESB, pour des raisons d'évolutivité et de standardisation, et pour avoir, par exemple, la possibilité d'utiliser un autre conteneur JBI, comme PETALS. La nouvelle version de CIMERO doit permettre de supporter un nouvel ESB avec un coût minimum de développement : simple ajout d'un nouveau module. Le logiciel sera donc le plus générique possible, il s'appuiera sur les spécifications JBI¹.

1 JBI 1.0 - JSR 208

3.3.3 Composants implémentés

a) Composants standards ServiceMix

Les composants ServiceMix suivants seront supportés par CIMERO :

➤ Binding Components

<i>Nom du composant</i>	<i>Description</i>
servicemix-http	Nécessaire pour la communication avec les Services Web
servicemix-jms	Nécessaire pour la communication avec les destinations JMS
servicemix-ftp	Utilisé pour accéder à un serveur ftp (lire/écrire) ou surveiller un répertoire ftp
servicemix-quartz	Permet de planifier des actions dans le temps

➤ Service Engines

<i>Nom du composant</i>	<i>Description</i>
servicemix-saxon	Permet de transformer un document XML suivant les règles du fichier de transformation XSLT
servicemix-drools	Routeur utilisant un fichier de règles Drools
servicemix-eip:content-based router	Routeur sur le contenu (règles XPATH)
servicemix-eip:wire-tap	Envoie une copie du flot sur une deuxième sortie
servicemix-eip:content-enricher	Enrichit le message à partir d'une source (par exemple un WS)
servicemix-eip:split-aggregator	Collecte les messages qu'il reçoit et renvoi l'agrégation en fonction d'une propriété définie
servicemix-eip:xpath-splitter	Reçoit un message et le renvoi en plusieurs parties (grâce à des règles XPATH)
servicemix-eip:message-filter	Filtre les messages reçus grâce à un prédicat XPATH

b) Composants non-supportés dans le futur CIMERO

Les composants non-standards ne seront pas supportés dans la nouvelle version de

CIMERO. Voici la liste des composants supportés par ServiceMix mais qui ne seront plus supportés par CIMERO :

- RSS flow
- Email

Cependant, si une version standard de ces composants est proposée par la suite, ils pourront être ajoutés grâce à l'outil d'ajout de composants.

c) Packaging du produit final

Le plugin développé devra être disponible sous deux packaging différents :

- plugin Eclipse classique
- version standalone

4 Contraintes

4.1 Exigences de documentation

Dans l'optique d'une maintenabilité facilitée, une documentation aussi complète et précise que possible sera rédigée et permettra à une future équipe de développeurs de reprendre et d'améliorer ce projet.

Liste des documents qui seront attachés à ce projet :

- Cahier des charges
- Plan d'assurance qualité logicielle
- Plan de développement logiciel
- Dossier de spécifications externes
- Dossier de conception globale
- Dossier de conception détaillée
- Plan de tests
- Jeu de tests
- Manuel utilisateur

4.2 Exigences de qualité

L'application sera développée en suivant les recommandations faites dans le Plan d'Assurance Qualité Logicielle. Les principaux facteurs requis sont :

- Portabilité
- Maintenabilité

4.3 Contraintes temporelles et ressources

Le projet durera 9 mois. Il sera composé de deux périodes :

- une période à mi-temps (3 mois) : le mardi, mercredi et jeudi de début janvier à fin mars
- une période à plein temps (6 mois) : de début avril à fin septembre.

L'équipe de développement sera composée de deux personnes et le projet devra être terminé avant le 30 septembre.

4.4 Contraintes techniques

4.4.1 Environnement de développement

Le plugin sera écrit en langage JAVA à l'aide du JDK 1.5, avec l'IDE Eclipse.

4.4.2 Environnement logiciel de l'application

Ce logiciel devra fonctionner sous l'IDE Eclipse 3.3. Le fonctionnement du logiciel sera indépendant de l'OS.

4.4.3 Framework

L'application sera développée en utilisant le framework GMF d'Eclipse.

5 Évolutions envisageables

Le module de support de l'ESB Petals pourra être implémenté pour permettre également à l'utilisateur de créer des applications JBI pour Petals avec CIMERO. Grâce à l'architecture mise en place pendant le développement du logiciel, le module s'intégrera facilement à l'application.