

Extension de passerelles OSGi™ pour les domaines de la distribution électrique: Modèles et outils

F. Baude², A. Bottaro¹, J.M. Brun⁴, A. Chazalet³, A. Constancin⁴, D. Donsez³; L. Gurgen¹, Ph. Lalanda³, V. Legrand², V. Lestideau³, S. Marié⁴, C. Marin³, A. Moreau⁵, V. Olive¹

¹ France Télécom R&D, ² INRIA-CNRS i3S, ³ LSR IMAG, ⁴ Schneider Electric, ⁵ Trialog
pise@imag.fr

RESUME

L'exploitation des données collectées par les capteurs enfouis dans les entreprises offre de nouvelles opportunités économiques aux équipementiers. Cependant, la mise en place des services associés est actuellement encore faite de manière empirique. Le projet PISE¹ s'est intéressé à proposer des méthodes et des outils pour la mise en place flexible et évolutive de ces services sur un parc de passerelles à grande échelle.

MOTS CLES

OSGi, plate-forme à services, architecture orientée service (SOA), passerelle, déploiement et administration à large échelle.

1. INTRODUCTION

La prolifération d'équipements communicants dans de nombreux domaines industriels et l'apparition d'Internet offrant des possibilités de connections simples et à bas coût favorisent la création de services distants à forte valeur ajoutée [1][14] utilisant les données enfouies dans les équipements (une tension sur un appareil de mesure électrique ou une température d'une pièce par exemple [15]).

Il apparaît néanmoins que l'état de l'art actuel ne permet pas la mise en place rapide de services performants et simples à administrer, indispensables à la satisfaction conjointe des opérateurs et des clients. Les solutions existantes reposent en effet sur des technologies Internet « classiques » :

- Utilisation d'une architecture client-serveur avec l'usage de navigateurs standardisés accédant directement à l'équipement ou au routeur de supervision des équipements via le protocole http,
- Utilisation d'un serveur applicatif générique de type J2EE qui concentre l'information de gestion des équipements et interroge directement soit les équipements via des protocoles spécifiques soit un routeur offrant une connexion Internet.

Ces architectures sont inadaptées aux enjeux car elles répondent au besoin de gestion à distance d'équipements mais avec plusieurs limitations importantes :

- Elles sont surtout adaptées à de petits parcs d'équipements (solutions non « *scalables* ») car tous les traitements sont localisés sur les serveurs d'application,
- Elles ne fonctionnent correctement que sur des environnements de type Intranet où le réseau lui-même fournit une qualité de service suffisante en terme de rapidité, de fiabilité et de sécurité,
- Elles nécessitent des développements importants (et non réutilisables) coté serveur puisqu'il n'existe pas d'infrastructure déchargeant le programmeur des aspects non fonctionnels liés à l'exploitation de données enfouies distantes.

Le projet PISE¹ ambitionne de résoudre ces limitations. La section 2 de cet article présente l'architecture sur laquelle sont basés nos travaux et décrit les objectifs du projet PISE. La section 3 donne une présentation des verrous technologiques à lever, ainsi que des solutions mises en œuvre. La section 4 présente les éléments de l'architecture de la passerelle et de son environnement de développement et d'exploitation. La section 5 positionnera ce projet par rapport à d'autres travaux. Enfin, l'article conclut sur quelques perspectives.

2. PROJET PISE

2.1 Architecture

Aujourd'hui, de nouvelles tendances illustrées par la figure ci-dessous se dessinent pour satisfaire les besoins d'exploitation à distance d'équipements, à savoir :

- l'utilisation de passerelles Internet intelligentes et normalisées entre les équipements industriels et les serveurs des opérateurs. Ces passerelles abritent des logiciels applicatifs qui s'exécutent au plus près des équipements. Ceci permet de réaliser des services nécessitant des échanges fréquents et rapides avec les équipements, de réduire les besoins et les coûts de communication, et globalement d'améliorer la fiabilité de l'offre de services.
- l'utilisation de serveurs gérés par les opérateurs et dont le rôle est d'administrer un parc de passerelles Internet. L'administration inclut la configuration technique des passerelles et la gestion des applicatifs qu'elles abritent. Les serveurs peuvent être reliés aux serveurs des équipementiers fournissant des applicatifs spécifiques.

¹ PISE est un projet RNRT 2003
<http://www.telecom.gouv.fr/rnrt/rnrt/projets/PISE.htm>

- la définition de nouvelles technologies de communication, telles que la médiation [16] entre passerelles et serveurs pour satisfaire aux critères de qualité industrielle, ainsi que de nouveaux modèles de programmation distribuée par objets ou composants [2].



Figure 1. Architecture de passerelles.

Le projet PISE a adopté cette architecture centrée autour de passerelles Internet. Ce concept est apparu récemment et des premières tentatives de normalisation ont vu le jour. En particulier, l'Alliance OSGi [3] regroupant 70 participants majeurs a standardisé, sous la forme d'APIs Java, la gestion du cycle de vie des applicatifs présents sur la passerelle. Les applicatifs peuvent ainsi être traités comme des composants logiciels exécutés au sein d'une infrastructure spécifique dès lors que certaines adaptations sont apportées [4] à l'application pour qu'elle soit vue comme un ensemble de composants interagissant. Cette infrastructure demeure néanmoins très insuffisante puisqu'elle ne prend en charge aucun aspect non fonctionnel autre que la gestion du cycle de vie. Le projet PISE est naturellement focalisé sur ces aspects non fonctionnels.

2.2 Objectifs du projet PISE

Le projet est orienté par les besoins spécifiques (hétérogénéité des équipements, hétérogénéité des bus de terrain qui permettent de communiquer avec ces équipements, installation avec une durée de vie importante) des domaines complémentaires de la distribution électrique et de la domotique/immotique à savoir :

- Fourniture de services métiers à forte valeur ajoutée
- Capacité à embarquer des services tiers (provenant de fournisseur tiers)
- Administration distante d'un grand parc d'équipements et des services qu'ils hébergent.

Il convient de séparer l'aspect services métier et technique : Les exigences non fonctionnelles considérées par le projet PISE recouvrent la sécurité, la distribution, la gestion des ressources et la journalisation des évènements.

Pour prendre en compte les contraintes des domaines de distribution électrique et de la domotique/immotique, notamment une installation à durée de vie importante et des contraintes de continuité de service, l'infrastructure logicielle OSGi, avec ses fonctions de chargement / déchargement dynamique, est un choix fondateur pour répondre aux besoins de mise à jour d'une partie

de l'application (correction de bug, installation de nouveaux services ou suppression de services existants suite à un changement de contrat) sans interrompre l'application entière.

L'administration distante d'un grand parc d'équipements et des services qu'ils hébergent est un sujet actif car il tente de combiner avec succès des techniques éprouvées d'administration système et réseau (SNMP), d'applications (JMX), d'une part et des intergiciels à composants sécurisés, mobiles, adaptables d'autre part. L'objectif est de fournir des services non fonctionnels et des outils de déploiement, de supervision, qui soient flexibles et d'usage transparent aux programmeurs de services fonctionnels.

L'objectif de ce projet est de concevoir une solution intégrée :

- répondant aux besoins concernant les aspects non fonctionnels,
- fournissant l'outillage permettant l'administration et le déploiement d'applications OSGi à grande échelle.

Il propose donc de concevoir, développer et valider une infrastructure logicielle pour passerelles Internet sécurisées capable d'accueillir dynamiquement des services techniques et métier, ainsi qu'un outil d'administration distante pouvant gérer un parc comportant des milliers de passerelles.

Cette infrastructure est compatible avec la norme OSGi et repose sur un modèle à composants favorisant une approche modulaire pour le développement des services et, surtout, prenant en charge de façon transparente et flexible les aspects techniques liés à la sécurité, la distribution, la gestion des ressources et la journalisation. A ce jour, l'Alliance OSGi n'a pas encore standardisé de modèle à composants permettant de traiter ces aspects non fonctionnels ; le projet PISE sera force de proposition en la matière.

Comme nous allons le voir par la suite, les technologies mises en œuvre portent réellement sur toute la chaîne de vie de l'installation, depuis la modélisation des applications jusqu'à la gestion de celles-ci à l'exécution.

3. VERROUS TECHNOLOGIQUES ET SOLUTIONS PROPOSÉES

Les verrous technologiques majeurs à lever dans ce projet et les solutions apportées par PISE sont les suivants:

3.1 Définition du modèle à composants dédié orienté service

Le développement d'application dans PISE s'inscrit à la fois dans l'ingénierie des composants, des services et des modèles. PISE considère deux verrous importants au niveau de la modélisation et du développement des applications. Le premier verrou est l'utilisabilité des modèles mis à la disposition des architectes et des développeurs des applications qui sont généralement des experts métiers non informaticiens. Le modèle doit prendre en compte à la fois les contraintes spécifiques au domaine métier (par exemple, la distribution électrique) et les contraintes technologiques de la plateforme cible (par exemple, l'usage du service OSGi WireAdmin, ...). Il faut dès lors aboutir à un niveau de description suffisamment précis pour une modélisation fine

englobant néanmoins les deux domaines. Le second verrou concerne l'intégration des applications existantes (« legacy ») qui est mal gérée par la plupart des modèles à composants (tels que JavaBeans, EJB, CCM, NET).

L'un des enjeux du projet est de permettre une utilisation flexible des services non fonctionnels. Contrairement à de nombreux modèles à composants, le projet PISE vise à ce que les services métier puissent installer ou désinstaller dynamiquement des services non fonctionnels durant leur cycle de vie: interface web, transfert de données via des protocoles standards (FTP, HTTP, WS-*, SMTP, JDBC...). A cet effet, une architecture orientée services (SOA Service-Oriented Architecture) [6] est privilégiée, appuyée par le concept des conteneurs extensibles dynamiques dans l'infrastructure OSGi.

L'approche suivie dans le projet PISE [5] consiste en l'utilisation de l'Ingénierie Dirigée par les Modèles [7][8] **Erreur ! Source du renvoi introuvable.**, pour faciliter d'une part le développement d'un outil de modélisation des applications métier et d'autre part le développement de ces applications orientées services.

L'environnement de développement des applications est constitué d'un outil de modélisation graphique, d'un dépôt d'implémentations de services métier et de services non-fonctionnels et d'un générateur de code. L'outil de modélisation graphique permet de construire une application en sélectionnant, dans le dépôt d'implémentations, les services correspondant à cette spécification. La composition de ces services est ensuite réalisée par la génération du code réalisant la « glue » nécessaire.

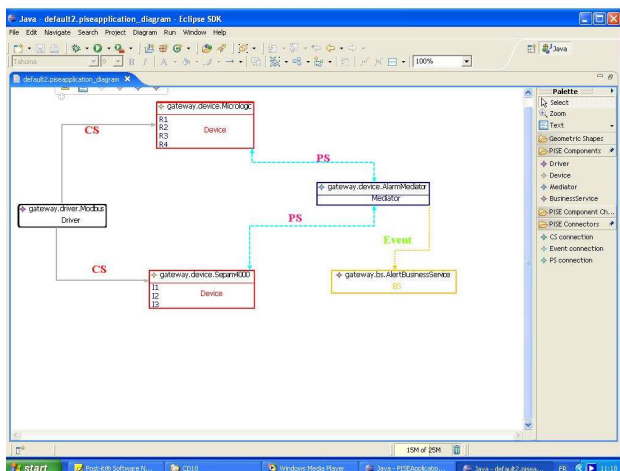


Figure 2: Outil de modélisation et de génération des applications.

3.2 Traitement de la distribution

Les applications d'entreprises peuvent lier des équipements appartenant à plusieurs réseaux capillaires sous-jacents aux plateformes de services. Par conséquent, certains traitements peuvent impliquer des équipements sur des plateformes distinctes. Rendre la topologie de plateformes distribuées transparente aux applications et conserver l'efficacité des traitements intra-plateforme est un des objectifs du projet [9].

Les différents middlewares orientés services existant aujourd'hui tels UPnP, Web Services, Bonjour, SLP, JINI, CORBA permettent de distribuer la découverte et la communication sur les

réseaux de plateformes et d'équipements IP. Afin de rendre la topologie transparente, les actions de découverte de composants distants sont effectuées à la suite des actions de découverte locale, dans le container approprié fourni par le middleware sous-jacent à notre modèle. La représentation locale et distante des services est rendue quasiment identique au développeur par génération dynamique de *stubs*. Ces objets générés sont utilisés par les composants demandeurs de services. Mises à part certaines contraintes devant être prises en compte pour l'écriture d'interfaces de services distribuables, le développeur a essentiellement pour tâche de spécifier la valeur d'un attribut booléen sur chaque composant demandeur et fournisseur de services distribués afin que la découverte et la communication distribuées soient mises en œuvre au moment de l'exécution des composants.

La version 4 (R4) de la spécification OSGi normalise une partie du traitement de la distribution pour la technologie UPnP. L'un des défis du projet PISE est de montrer comment parachever les limites de la transparence du modèle OSGi. Le but ultime est de rendre le développement d'applications distribuées identique au développement d'un simple objet Java dans la philosophie des POJOs (*Plain Old Java Objects*, littéralement "les bons vieux objets Javas").

3.3 Traitement de la sécurité

Un enjeu fort du projet est la définition d'un composant de sécurité qui soit utilisable par les différents « bundles » OSGi, indépendamment de leur nature. Un des principaux besoins exprimés concerne l'installation et la mise à jour sécurisée de « bundles » sur les passerelles. La solution retenue consiste à enrichir l'intergiciel des passerelles d'un mécanisme ad hoc qui vérifie par l'intermédiaire de certificats et de permissions que la politique de sécurité est correctement respectée (authentification des intervenants et autorisation notamment). Le projet PISE vise également à permettre aux « bundles » de protéger les services qu'ils exposent contre les accès non autorisés. Pour cela un composant de bas niveau implémente le "Conditional Permission Admin" de la spécification R4 OSGi. Son API permet d'associer des permissions à des bundles selon certaines conditions : signature java, chemin d'accès, etc., et de les modifier dynamiquement. Un ensemble de permissions a par ailleurs été défini pour répondre aux besoins de sécurité spécifiques des plates-formes métier. Enfin, un mécanisme de log garde une trace des accès aux services afin de pouvoir suivre à posteriori l'historique des opérations effectuées.

3.4 Gestion d'un parc de plusieurs milliers de passerelles

Administrer un parc formé de passerelles hébergées sur du matériel et des systèmes plus ou moins récents, fortement hétérogènes, dont les capacités de stockage et de traitement peuvent éventuellement s'avérer insuffisantes, sporadiquement reliées au réseau extérieur via une connexion non fiable et non forcément permanente, pose de réels problèmes de mise en œuvre et d'intégration, exacerbés par la contrainte du passage à très grande échelle.

L'ensemble de ces aspects requiert des solutions d'administration qui concilient répartition, asynchronisme (pour avoir du parallélisme dans l'exécution des opérations), sécurité et bien

entendu facilité d'utilisation. Le projet a mis en œuvre un outil d'administration et de déploiement à grande échelle se basant sur :

- JMX, car c'est un standard d'administration d'applications, utilise déjà couramment dans l'administration distante d'une passerelle OSGi [17].
- ProActive, qui permet des communications de groupes typées asynchrones. Un parc de passerelles sera considéré alors comme un groupe typé : Dès lors, l'administration par JMX d'un groupe de passerelles est une opération qui s'exécute en parallèle.

Le projet propose dès lors un ensemble d'outils pour l'opérateur, qui lui permettent à distance non seulement de déployer les services d'un prestataire (emballer, installer, démarrer, mettre à jour, reconfigurer, retirer) mais aussi de superviser le fonctionnement d'un nombre éventuellement important de passerelles Internet. Ces processus sont représentés sur la figure 3.

Par ailleurs, le projet propose un mécanisme *déclaratif* d'interrogation des données provenant des différents équipements spécifiques à la distribution électrique (voir figure 4). A l'aide des requêtes "SQL-like" l'utilisateur peut définir, quels équipements à interroger, à quel moment, et pendant quelle durée.

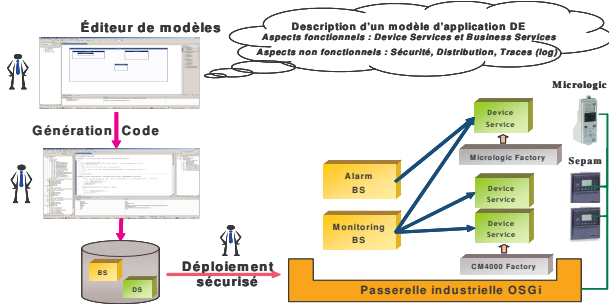


Figure 3: environnement de développement et de déploiement PISE

L'interrogation n'adresse pas seulement l'acquisition des données mais aussi l'exécution de certaines opérations sur les données captées. Ces opérations incluent des filtres, des opérations arithmétiques, des agrégations, etc. Les opérations peuvent être exécutées à différents niveaux de l'architecture ce qui permet de distribuer la charge d'évaluation des requêtes, ainsi de faciliter le passage à la grande échelle [10].

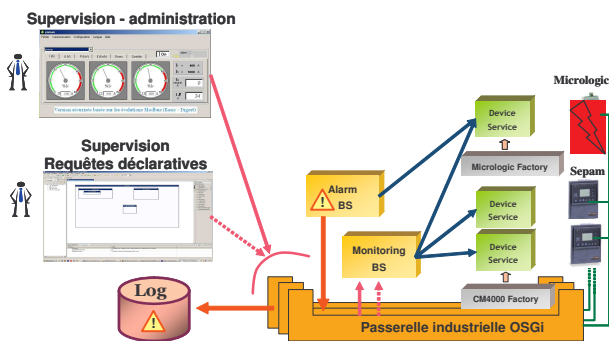


Figure 4: Environnement d'exécution des passerelles PISE

3.5 Validation expérimentale

Afin de valider l'infrastructure dans le domaine de la distribution électrique, il sera fait usage de plates-formes métier complètement outillées.

Par ailleurs, la solution de l'administration et du déploiement à large échelle sera validée par l'émulation d'un grand parc de passerelles OSGi sur une grille de machines (par exemple Grid'5000). Le démarrage des passerelles appartenant à ce parc sera effectué par des techniques de soumission de processus sur clusters ou grilles [11].

4. TRAVAUX RELATIFS

Le projet ITEA Robocop vise un but similaire mais dans le domaine des équipements électroniques mobiles (PDAs, téléphones portables...), logiquement caractérisé par des besoins non fonctionnels différents (gestion de la consommation électrique par exemple).

Il existe par contre des travaux plus généraux sur la notion de composants qui seront utilisés dans le projet. On peut par exemple citer les travaux autour du « CORBA Component Model », ainsi que les études liées au modèle de composants hiérarchiques Fractal (en partie menés au sein du projet RNTL ARCAD).

REMMOC [12] est un modèle pionnier dans l'utilisation transparente de middlewares orientés services distincts. Tout protocole de découverte et de communication distante peut être pris en compte dans l'architecture modulaire de REMMOC. La reconfiguration dynamique de la plateforme en fonction des protocoles effectivement utilisés sur le réseau est l'aspect principal mis en avant dans ces travaux. Un module de l'architecture est responsable de la "découverte" des protocoles utilisés. La reconfiguration réagit sur les événements émis par ce module. La vision de l'interopérabilité est similaire à la notre : Les fournisseurs et demandeurs de services sont écrits indépendamment des protocoles de découverte et de communication effectivement utilisés et une syntaxe de description générique joue le rôle de pivot commun. Cependant, l'adaptation dynamique de la disponibilité des services requis et la continuité entre communication locale et distante ne sont pas adressées dans le projet REMMOC.

MSD [13] est un middleware permettant l'interaction transparente d'applications préexistantes exposant et requérant des services dans des protocoles distincts. Chaque réseau est administré par un Gestionnaire (Network Manager) lié à autant de modules de découvertes que de protocoles pouvant être utilisés sur le réseau comme UPnP, SLP... Ces entités relaient les événements de découverte et les appels protocolaires entre modules de découverte sur un même réseau et des ponts permettent de les relayer entre les réseaux. La vision introduite par MSD est fondamentalement différente. Tous les fournisseurs et demandeurs de services préexistent et sont développés avec leurs protocoles propres. L'architecture MSD agit telle une passerelle entre les réseaux topologiques et entre les technologies de découverte. La transparence repose sur l'utilisation massive des capacités réseaux.

5. CONCLUSION ET PERSPECTIVES

Le projet PISE est une première étape dans la compréhension de l'ingénierie des services M2M (« Machine to Machine »). Ce

projet a permis de jeter des bases formelles et technologiques pour d'autres projets reprenant les travaux de PISE. A titre d'exemples, le projet ITEA ANSO s'intéresse à l'ingénierie des services autonomiques dans le contexte de la domotique et le projet ITEA SODA s'intéresse à la modélisation et la composition des services fournis par les équipements d'automatique industrielle. Des parties des résultats du projet PISE seront relâchées à destination de la communauté open-source (ObjectWeb, Apache, ...) autour d'OSGi et plusieurs propositions d'extension sont déjà soumises à l'Alliance OSGi.

6. REFERENCES

- [1] International Telecommunication Union, "The Internet of Things, Executive Summary", ITU Internet Reports 2005, November 2005
http://www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summary.pdf
- [2] C. Szyperski, *Component Software - Beyond Object-Oriented Programming, 2/E*, Addison-Wesley / ACM Press, 2003, ISBN 0-201-74572-0 (first edition in 1997, ISBN 0-201-17888-5)26.
- [3] OSGi Alliance, "OSGi Service Platform Specification (4d Release)," Octobre 2005, <http://www.osgi.org>.
- [4] Cervantes H., Hall R., "A Framework for Constructing Adaptive Component-based Applications: Concepts and Experiences", 7th Symposium on Component-Based Software Engineering (CBSE), Mai 2004, Edinburgh, Ecosse.
- [5] Marin C., Lalanda Ph., Donsez D., "A MDE Approach for Power Distribution Service Development" Actes de 3rd International Conference on Service Oriented Computing 2005 (ICSOC05), Amsterdam, The Netherlands, December 12-15, 2005
- [6] Huhns M. N., Singh M. P.. *Service-Oriented Computing: Key Concepts and Principles*. IEEE Internet Computing, 9:75–81, 2005.
- [7] Selic B.. *The Pragmatics of Model -Driven Development*. IEEE Software, pages 19–25, 2003.
- [8] Seidewitz E., *What Models Mean*. IEEE Software, pages 26–32, September 2003.
- [9] Bottaro A., Gérodolle A., Lalanda P., "Pervasive Spontaneous Composition", First IEEE International Workshop on Service Integration in Pervasive Environments, Lyon, France, 2006
- [10] Gurgun L., Labbé C., Olive V., and Roncancio C.. *A scalable architecture for heterogeneous sensor management*. In MDDS'05, DEXA Workshops, Denmark, 2005.
- [11] Baude F., Baduel L., Caromel D., Contes A., Huet F., Morel M. et Quilici R., "Programming, Composing, Deploying for the Grid", in "GRID COMPUTING: Software Environments and Tools", J. C. Cunha and O. F. Rana (Eds), Springer Verlag, January 2006.
- [12] Grace P., Blair G. S., Samuel S., "ReMMoC, "A Reflective Middleware to Support Mobile Client Interoperability", Proceedings of International Symposium on Distributed Objects and Applications, November 2003
- [13] Raverdy P.-G., Issarny V., "Context-aware Service Discovery in Heterogeneous Networks", Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2005), June 2005
- [14] Lalanda Ph., *E-Services Infrastructure in Power Distribution*. IEEE Internet Computing, May-June 2005.
- [15] Akyildiz I. F. et al. *A Survey on Sensor Network Applications*. IEEE Communication Magazine, 2002.
- [16] Wiederhold G., *Mediators in the architecture of future information systems*. IEEE Computer, 25(3):3849, 1992.
- [17] Frénot S., Stefan D., *Instrumentation de plates-formes de services ouvertes – Gestion JMX sur OSGi, Mobile and Ubiquitous computing (Ubimob2004)*, 2004, ACM Press.