

Présentation  
du  
Projet de Recherche et Développement  
Generic System

Nicolas Feybesse

2008-2009

# Table des matières

|       |   |    |
|-------|---|----|
| 1     | Description du projet de recherche.....   | 3  |
| 1.1   | Contexte et état de l'art.....  | 3  |
| 1.2   | Objectifs et contraintes.....   | 4  |
| 1.3   | Difficultés techniques.....   | 5  |
| 1.4   | Résultats obtenus.....  | 6  |
| 1.5   | Article de presse dès 2007.....   | 6  |
| 2     | Travaux effectués sur Generic System 1 (début 2007 à fin 2009).....               | 7  |
| 2.1   | Analyses.....   | 7  |
| 2.1.1 | Analyse conceptuelle.....   | 7  |
| 2.1.2 | Analyse organique.....  | 8  |
| 2.1.3 | Analyse fonctionnelle.....  | 10 |
| 2.2   | Prototypes de conception.....   | 13 |
| 2.2.1 | Generic System Tracker (à partir de Novembre 2007).....                           | 13 |
| 2.2.2 | Generic System Web : (à partir de Juillet 2008).....                              | 14 |
| 2.2.3 | Generic System CRM (à partir de Mai 2008).....                                    | 15 |
| 2.2.4 | Generic System Intermediation (de Novembre 2010 à Août 2011 Partnersfit.com)..... | 15 |
| 3     | Travaux effectués sur Generic System 2 (à partir de Juin 2009).....               | 16 |
| 3.1   | Analyses.....   | 16 |
| 3.1.1 | Analyse conceptuelle.....   | 16 |
| 3.1.2 | Analyse organique.....  | 16 |
| 3.1.3 | Analyse fonctionnelle.....  | 17 |
| 3.2   | Prototypes de conception.....   | 17 |
| 4     | Conclusion.....   | 17 |
| 5     | Ressources affectées au projet.....   | 18 |
| 5.1   | Temps passé et tâches réalisées par chaque ingénieur (début 2008 – fin 2009)..... | 18 |
| 5.2   | Matériel.....   | 27 |
| 6     | Annexe : Generic System 2 - Analyse Conceptuelle.....                             | 28 |

# 1 Description du projet de recherche

Le projet Generic System est un projet de recherche et développement initié officiellement en Janvier 2007 par la société Middleware Factory sur la base de travaux réalisés par Nicolas Feybesse durant les années 2003 à 2007.

L'entreprise Middleware Factory qui réalise des prestations de service dans le monde des systèmes d'informations, plus précisément dans l'univers des serveurs d'application. Middleware Factory a souhaité capitaliser son savoir faire en concevant une application innovante basée sur une nouvelle façon de représenter (modéliser) l'information.

Ce projet de recherche peut être scindé en deux phases de recherche et développement :

- Generic System V1 dont le modèle de représentation de l'information innovant est construit sur la base d'un modèle d'information classique dit « relationnel » et qui avait pour but de valider l'utilisabilité et l'intérêt opérationnel de la nouvelle organisation de l'information que propose Generic System. Les efforts de recherche et développements de Middleware Factory sur Generic System V1 ont commencé officiellement en 2007 pour se terminer progressivement à la fin de l'année 2009.
- Generic System V2 dont le modèle de représentation de l'information a été nettement amélioré et enrichi et qui s'abstrait de tout recours à une base de donnée relationnelle. Les efforts de recherche et développement sur cette version ont commencé en Août 2009 et sont toujours en cours.

## 1.1 Contexte et état de l'art

Dans l'univers des systèmes d'informations, les applications logicielles s'articulent généralement autour de la représentation de l'information dans des systèmes persistants et concurrentiels. Une énorme majorité de ces systèmes appartiennent à la catégorie des Systèmes de Base de Données Relationnels (SGBDR) dont l'illustre représentant est la société Oracle qui commercialise un système de bases de donnée du même nom et qu'il est inutile de présenter.

D'autres produits concurrents existent et forment avec Oracle, la catégorie des systèmes dits « relationnels » par le simple fait de respecter une norme commune d'exploitation des informations qu'ils contiennent : le langage « SQL ».

Les serveurs d'application sur lesquels Middleware Factory a construit son expertise en terme de services, proposent d'exploiter les systèmes d'information relationnels par le biais de langages de dernière génération, les langages dits « orientés objet » tels que Java (ou C#).

Les paradigmes de la programmation orienté objet et ceux des systèmes persistants relationnels sont malheureusement parfois orthogonaux, ce qui complexifie fortement et très rapidement leur cohabitation.

La mission des serveurs d'application revient dans bien des cas à masquer cette complexité et offrir des services simplificateurs pour concevoir et développer des logiciels qui bénéficieront des bienfaits de chacun des deux paradigmes : la concurrence et la persistance des systèmes relationnels ainsi que la conception objet des langages orientés objet.

L'immense succès du framework open-source « Hibernate » et des normes « JPA » ou « EJB3 »

dans l'univers des serveurs d'applications JEE n'est que la conséquence des services simplificateurs qu'ils proposent dans l'exploitation des informations persistantes relationnelles par des langages orientés objet.

## **1.2 Objectifs et contraintes**

L'objectif de Generic System a été dans sa première version de proposer un système de représentation objet élaboré, cohérent, capable de gérer de manière générique différents modèles de données, dont la persistance et la concurrence d'accès à l'information soit naturelle et simple lors de l'exploitation des informations par un langage orienté objet, en éliminant tout recours au langage SQL.

L'objectif de la seconde version est de proposer un système de représentation objet complet (UML), persistant et concurrentiel sans aucun recours à une base de donnée relationnelle.

Les contraintes auxquelles nous avons souhaité nous plier afin de proposer un moteur d'information qui puisse être profitable et porteur d'innovations tangibles, sont les suivantes :

Le moteur de représentation doit être objet : Les modèles de données que le moteur d'information permet de représenter sont dits « objet ». La norme dans ce domaine à laquelle on tentera de se conformer est UML2 (Diagrammes de classe)

Le moteur doit être simple et cognitif : Les APIs (Applcation Programming Interfaces) qui traitent l'information doivent être simples et les plus naturelles possible quant à leur utilisation. Elles doivent se conformer autant que possibles aux mécanismes cognitifs. La représentation de l'information doit être pour sa part aussi canonique que possible.

Le système doit être restucturable

La structure de l'information doit elle-même pouvoir être manipulée selon des procédés aussi similaires que possible à ceux ceux qui manipulent l'information elle-même. (Concept directement issu du domaine de l'intelligence artificielle et dont le respect constitue une innovation significative car il rend le système d'information complètement évolutif).

Le système doit être persistant : Le moteur d'information doit être pourvu d'un mécanisme de sauvegarde de l'information, et doit ainsi, pouvoir être arrêté, puis être redémarré plus tard sans perte d'information. Il doit expressément posséder la capacité de persister tout modèle d'information, en particulier les modèles contenant des cycles (graphes).

Le système doit être cohérent : De la même manière qu'une base de donnée relationnelle, le moteur d'information doit proposer à cet effet une gestion transactionnelle et une gestion de contraintes d'intégrité élaborée. Les contraintes d'intégrité classiques du monde relationnel ainsi que d'autres inhérentes au monde objet doivent être respectées afin de garantir la cohérence et la stabilité du système.

Le système doit être concurrentiel : Plusieurs utilisateurs doivent pouvoir accéder à l'information en même temps. Le moteur doit proposer à cette effet une résolution des conflits concurrentiels cohérente et acceptable.

Le système doit être restaurable : En cas de crash, le système doit pouvoir se rétablir avec une perte d'information aussi minime que possible et revenir à un état cohérent.

Le système doit être performant : Une partie du moteur doit être multi-threadée et doit répartir la charge sur les processeurs disponibles.

### **1.3 Difficultés techniques**

Aucune société n'a, à ce jour, su mettre au point un tel système, ou en tout cas, su lui assurer un franc succès. La maturité des projets du marché reste faible, et leur utilisation ne décolle pas significativement dans l'environnement professionnel.

Rien n'impose pourtant l'utilisation d'une base de donnée relationnelle pour gérer un système d'information, hormis le fait qu'elles se soient fortement démocratisées ces dernières années.

Le fondateur d'Hibernate, Gavin King, était de cet avis que les bases de données relationnelles étaient incontournables du fait de leur démocratisation, et il avait sans doute raison étant donné le succès d'Hibernate qui s'en est suivi. Mais l'article dans lequel il s'est exprimé date de 2004, et ceux qui l'ont lu, devraient désormais se méfier de sa date de péremption.

On trouve aujourd'hui beaucoup d'autres articles qui étayent cette opinion que concernant les bases de données relationnelles, et aussi persistantes soient-elles, leur utilisation ne le sera pas.  
(<http://www.infoq.com/news/2007/06/orm-obms-debate>)

L'analyse des produits existants, qui sont généralement regroupés sous l'appellation OODMS (Object Oriented Database Management System), nous permet de penser que ces solutions sont dans de nombreux cas

- peu opérationnelles (défauts dans la résolution technique des difficultés)
- peu utilisables (défauts dans la résolution conceptuelle des difficultés)

Ces solutions sont finalement peu évolutives et peu adaptées aux besoins des entreprises car elles ne proposent pas systématiquement et encore moins simultanément la résolution des contraintes exposées précédemment.

Très souvent, c'est la persistance de graphe avec cycles qui leur fait défaut et à notre connaissance, aucune ne propose le mécanisme le plus innovant de Generic System qui consiste à adresser l'information et la structure de l'information sur le même plan ce qui permet ainsi de découpler la structure de l'information du mécanisme de persistance.

Que ce soit d'ordre technique ou conceptuelle, la résolution correcte des difficultés passe par une importante remise en question des choix qui sont faits habituellement, particulièrement ceux issus du monde relationnel. Dans Generic System, l'organisation de l'information, la désignation des objets, la gestion de la concurrence d'accès et le mécanisme de persistance ont fait l'objet de choix très innovants, souvent influencés par des découvertes récentes. Enfin ils ont bénéficié de nombreux ajustements successifs grâce à l'élaboration de nombreux prototypes de conception avant d'être définitivement adoptés.

### **1.4 Résultats obtenus**

Je prétends avoir découvert au cours de l'année 2003 les prémices d'une solution innovante de modéliser l'information qui a permis à Middleware Factory et son équipe, en procédant à de nombreuses recherches et en réalisant de nombreux prototypes de conception successifs de mettre au point un moteur d'information opérationnel, cohérent, persistant, concurrentiel, orienté objet,

restructurable à chaud, restaurable et plus cognitif que ce qui se fait jusqu'à présent.

Notre démarche n'a pas été de nous conformer directement aux tentatives de normalisation des outils de type OODBMS mais plutôt de conserver une certaine indépendance dans nos choix, de manière à ne pas être trop influencé par une façon de procéder qui, à notre sens, paraît assez laborieuse dans l'obtention de résultats tangibles dans ce domaine depuis quelques années.

## 1.5 Article de presse dès 2007

Lors du lancement officiel de Generic System en 2007, nous avons fait paraître un article de presse diffusé dans le magazine mensuel « Entreprendre » (mensuel N°215 Octobre 2007). Celui-ci a consacré une demi page au projet Generic System dans un dossier sur les nouvelles solutions informatiques.

### Dossier : ENTREPRENDRE N° 215 Octobre 2007

## Middleware Factory Intégrateur en logiciel libre

*Créée en 2005, Middleware Factory, propose une réelle expertise des serveurs d'applications open source Jboss. Avec le nouveau concept Generic System, c'est le « Tout Paramétrable » à portée des PME-PMI.*

*Interview de Nicolas FEYBESSE, Directeur général.*

**Quelles sont les activités de Middleware Factory ?**



Middleware Factory est une Société de Service en Logiciel Libre spécialisée sur les architectures Java/J2EE Open Source. Elle propose son expertise des serveurs d'applications JBOSS ainsi que sur tous les projets satellites Open Source les plus pointus tels que Hibernate, EJB3, Seam, JSF, Facelets... Middleware Factory est aussi un organisme de formation reconnu par lequel sont passés de nombreux consultants afin de parfaire leur connaissance des serveurs d'applications J2EE. Middleware Factory entretient l'expertise de ses consultants, par le développement interne d'une solution globale de gestion de l'information qu'elle propose aux PME : le Generic System. Cette plateforme, de type SAAS (Software As A Service), ayant bénéficié de plusieurs années de recherche et développement, met en

musique les outils open source les plus avancés du marché.

**Quel est le concept de Generic System ?**

Le client paramètre son métier via une interface graphique en quelques clics, et bénéficie à l'issue de cette étape d'un outil de gestion d'information Web, simple et opérationnel. L'application a construit automatiquement un modèle de représentation des données et propose une gestion simple de celles-ci dans le respect des règles métiers paramétrées. Cela permet de répondre immédiatement aux besoins les plus urgents. Le client affine ensuite son paramétrage au fur et à mesure qu'il exploite l'outil et en fonction des améliorations suggérées par les utilisateurs. A l'instar de l'industrie pharmaceutique, il est désormais possible de « générer » des applications extrêmement complexes à moindre coût. Les utilisateurs sont impliqués dès le départ dans ce concept. Ils construisent l'application par un paramétrage itératif. **C'est une redéfinition du rôle des utilisateurs ?**

Middleware Factory, en réussissant son



pari d'isoler les aspects techniques des aspects fonctionnels dans son système d'information, positionne le métier du client au centre des préoccupations et redéfinit effectivement le rôle de l'utilisateur en l'impliquant davantage. Middleware Factory propose à ses clients (parmi lesquels figurent DEXIA, CALYON, SNCF, Finance Active...) une nouvelle méthodologie : le paramétrage itératif et les assiste dans l'élaboration de leurs applications.

*Interview : APP*



Technocentre de Suresnes  
51 Rue de Verdun  
92150 Suresnes  
Tél / Fax : 01 72 89 25 30  
info@middlewarefactory.com  
www.middlewarefactory.com

## **2 Travaux effectués sur Generic System 1 (début 2007 à fin 2009)**

### **2.1 Analyses**

#### **2.1.1 Analyse conceptuelle**

Avec Generic System, Middleware Factory veut parvenir à mettre au point un socle applicatif extrêmement novateur où stockage et traitement des données obéissent à une seule et même logique, la logique objet.

La spécificité inédite du noyau de Generic System 1 réside, dans sa capacité à servir d'embrayage au sens presque mécanique entre le modèle Objet et le modèle Relationnel : le modèle Objet peut être structurellement modifié sans que la structure du modèle Relationnel ne le soit.

Une telle souplesse s'appuie sur la conception d'un « méta-modèle », c'est à dire sur un modèle de données capable de représenter d'autres modèles de données. Chaque modèle Objet y est considéré comme un jeu de données respectant les règles fixées par le méta-modèle. Une modification de la structure de ce modèle Objet se traduit alors, comme pour n'importe quel autre jeu de données, par une altération des enregistrements de la base, dont la structure, définie par le méta-modèle relationnel, reste inchangée.

D'apparence extrêmement simple, ce méta-modèle est le fruit d'un travail de conception et d'une réflexion continue de plusieurs années, permise par l'élaboration de nombreux prototypes de conception.

Ce découplage entre la structure de l'information du modèle objet qui est dynamique et celle du modèle relationnel qui reste figée constitue une innovation majeure dans l'univers des systèmes d'information dans de nombreux domaines.

C'est un univers dans lequel, l'abstraction est omniprésente, et le concept d'auto-référence, si complexe à appréhender, et comme l'avaient prédit nombre de chercheurs en théorie des systèmes renommés (1), est porteur d'importantes avancées tant sur le plan économique que sur le plan scientifique avec de réelles perspectives dans le domaine de l'intelligence artificielle notamment.

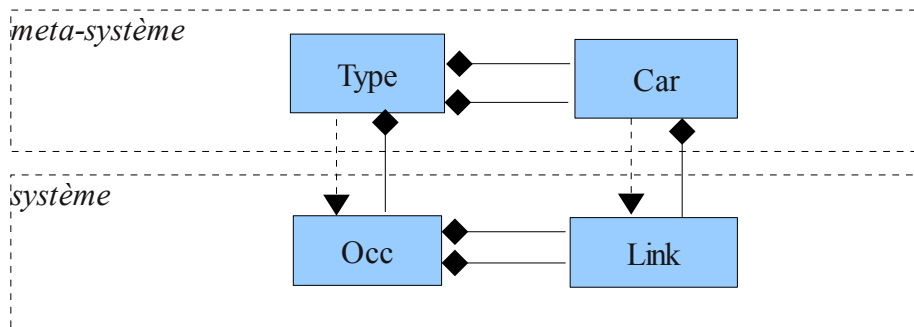
Par exemple, on pourrait concevoir un système qui, à l'instar du « rêve » de l'être humain, restructurerait les informations qu'il a accumulées, pour gagner en espace, et/ou en accessibilité (cf Generic System 2 – Concept de standard et d'information flottante).

Dans un système d'information relationnel, un tel mécanisme n'est pas simplement réalisable du fait que la structure de l'information n'est pas modifiable de manière transactionnelle et concurrentielle.

De même il n'est pas impossible qu'un système d'information puisse un jour découvrir lui-même la structure de l'information qu'il doit appréhender en procédant à des comparaisons structurelles sur celle-ci. Cette idée est directement inspirée des mécanismes cognitifs et de la capacité que possède la pensée humaine à procéder à des « analogies » lorsqu'elle appréhende une information d'une typologie nouvelle.

(1) [Douglas\\_Hofstadter - Gödel, Escher, Bach: les brins d'une guirlande éternelle \(1979\)](#)

## Modèle canonique simplifié de Generic System 1



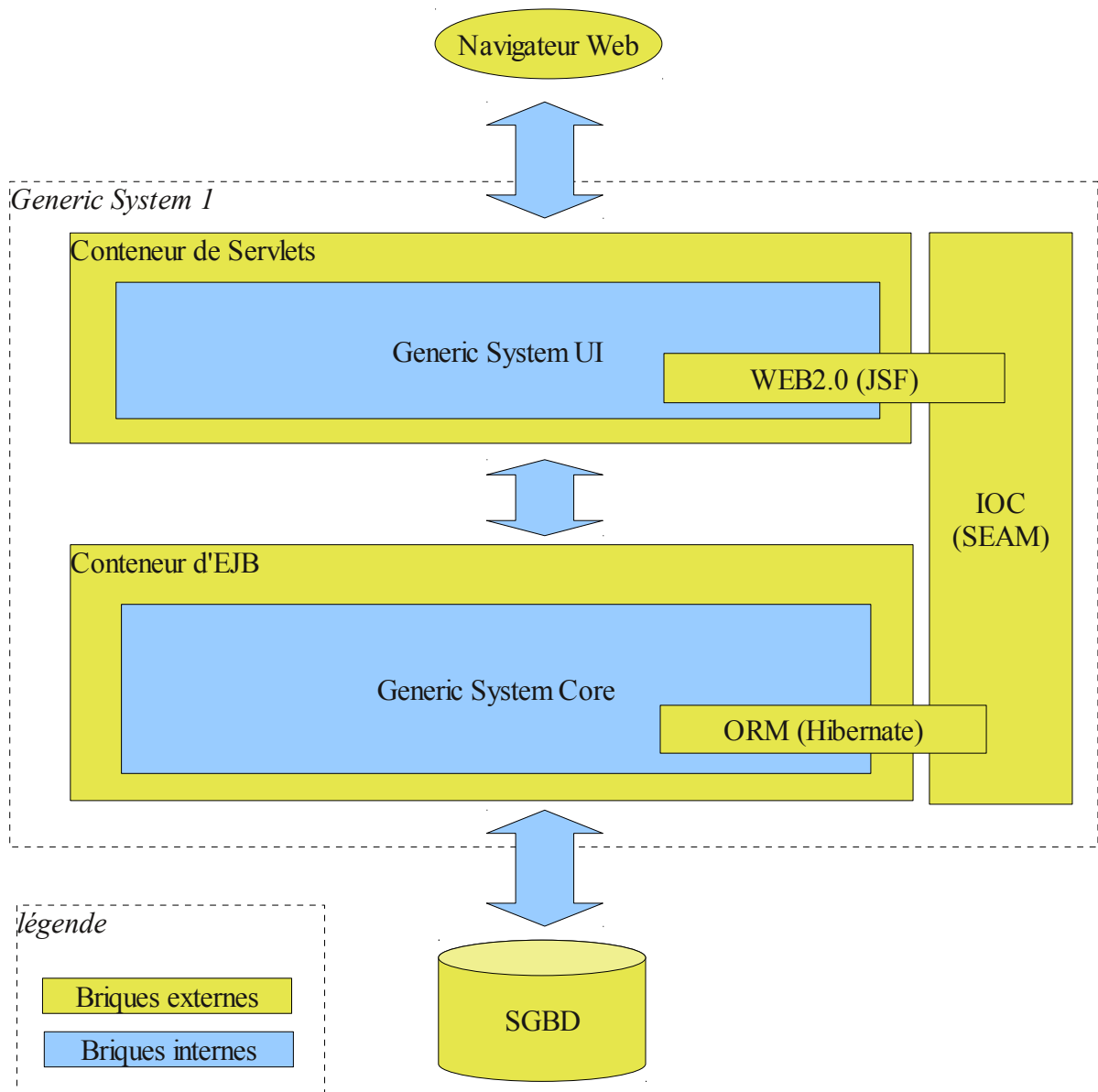
- La classe principale est l'occurrence (**Occ**)
- Toute occurrence a un type d'occurrence (**Type**)
- Un type rassemble un ensemble d'occurrences (celles du même type)
- Un type est lui même une occurrence du meta-système
- Un link (**Link**) s'appuie sur une ou deux occurrences (une occurrence de base et éventuellement une occurrence cible) pour le compte d'une caractéristique (**Car**)
- Une occurrence possède un ensemble de links qui s'appuie sur elle en tant qu'occurrence de base et un autre en tant qu'occurrence cible
- Une caractéristique (**Car**) rassemble un ensemble de links
- Une caractéristique est elle-même un link du meta-système
- Les caractéristiques sans cible sont des attributs, celles avec une cible sont des relations.
- Les links sans cible sont les valeur d'attributs, celles avec une cible les liens réels d'une relation
- Le modèle peut s'étendre aux relation n-aire théoriquement.

Pour plus de détails, on peut se référer ici à l'annexe : Generic System 2 – Analyse conceptuelle qui reprend, précise et enrichit un grand nombre de concepts déjà présents dans Generic System 1.

### 2.1.2 Analyse organique

Le modèle Objet possède deux représentants principaux, les environnements Java/JEE et Dot Net. Le monde Java, plus mûr et très utilisé dans le monde professionnel, repose sur des briques open-source et bénéficie d'une communauté très active. C'est donc tout naturellement que Middleware Factory, forte de l'expertise de ses consultants dans le domaine Java, et de sa connaissance approfondie des serveurs d'application JEE a choisi de développer Generic System en s'appuyant sur une plate-forme JEE.





## 2.1.3 Analyse fonctionnelle

### Generici System Core

Generic System Core est une couche d'abstraction qui structure les informations selon un modèle objet et assure leur persistance dans un système relationnel. Cette couche s'appuie sur un ensemble de frameworks innovants de l'univers Java/JEE pour bénéficier de services éprouvés dans leur spécialisation technique.

La conception des objets « entité » et leur mapping avec une structure de base de donnée relationnelle constitue une partie essentielle du noyau de Generic System 1.

Différentes modélisations physiques ont été expérimentées grâce aux nombreuses possibilités de mapping d'Hibernate. Malheureusement, plus le meta-modèle est canonique, plus les requêtes SQL générées sont nombreuses et tournent autour d'un jeu restreint de tables avec des jointures multiples. Même si en pratique, les différentes expérimentations ont donnés d'assez bon résultats en terme de cohérence et de performance, il est évident qu'une telle utilisation des bases de données relationnelles est peu académique mais surtout peu scalable (à moins de concevoir un moteur relationnel spécialisé).

Enfin, il n'a pas été réellement possible de gérer les relation n-aires, dans le sens où la structure physique de la base de donnée relationnelle reste dépendante du nombre « n » maximum des relations n-aires que l'on souhaite représenter. Plus précisément, nos recherches ont démontré qu'il est sans doute possible de réaliser un tel découplage, mais au prix de l'utilisation d'un modèle physique beaucoup moins performant parce qu'entraînant l'exécution de nombreuses jointures supplémentaires dans les requêtes générées.

Ces deux difficultés ont largement participé à la décision de s'abstraire du recours à une base de données relationnelle et d'un ORM dans la seconde version de Generic System.

### Les objets systèmes.

Comme nous l'avons compris, Generic System Core gère un graphe d'objets qui portent en leur sein, toute l'information que l'on souhaite persister.

Un certain nombre d'entre eux, peuvent voir leur existence garantie, sans qu'il n'y ait eu aucune intervention utilisateur. Leur cycle de vie est géré différemment : ils sont déclarés directement dans le code source puis détectés et montés au démarrage du système. Ces objets sont dits « systèmes » et peuvent être utilisés par les extensions de Generic System Core au sein de traitements, en toute sécurité puisque leur existence est garantie.

Generic System Core agit avec ces objets système tel un bus auquel se connecteraient un ensemble de services basés sur de l'information persistante. Ces objets « système » constituent une sorte de « paramétrage » directement réalisé dans le code source de l'application.

Le découpage modulaire naturel du code source java en archives (fichier « jar ») permet ainsi de monter tel ou tel paramétrage en fonction de la présence ou non de telle ou telle archive et de mettre à disposition de l'application des services et fonctionnalités résultant de leur simple déploiement.

Comme nous l'avons dit, Generic System Core s'appuie sur un certain nombre de frameworks afin de bénéficier d'un certain nombre de services indispensables dans la résolution technique de

problématiques complexes inhérentes à la problématique de découplage de persistance qu'il tente de résoudre.

### **Le framework de persistance : Hibernate**

Illustre représentant des frameworks dit « ORM » (Object Relation Mapping), Hibernate / EJB3 assure la génération de la structure de base de donnée ainsi que les ordres SQL envoyé au SGBDR en respectant un « mapping » relationnel / objet » défini dans le noyau. Un ORM fournit aussi la gestion d'un cache d'objet, avec une gestion de concurrence d'accès de type « optimiste » que Generic System utilise abondamment.

Enfin, pour des raisons de performance, nous avons monté dans un second temps, un cache dit de « second niveau » centralise, concurrentiel et transactionnel (Jboss Cache).

### **Le framework d'injection IOC (Inversion Of Control) : SEAM**

Ce framework a été sélectionné pour bénéficier de ses nombreux services intégrés et éprouvés en terme de

- gestion transactionnelle,
- gestion du cache Hibernate (conversations longues avec la session Hibernate et gestion de la concurrence d'accès de type optimiste)
- gestion du cycle de vie des objets « system » contenues dans les différentes archives (modules) de Generic System.
- gestion d'un bus de communication entre les différentes couches (notamment avec la couche interface utilisateur et le framework de présentation Web JSF).

Son utilisation nous a permis et pour de nombreuses difficultés connexes à notre problématique principale de bénéficier d'une première résolution technique relativement efficace et exhaustive. Elle nous a permis ainsi de nous concentrer sur la résolution conceptuelle et technique de la problématique centrale que cherche à résoudre Generic System : concevoir un système d'information persistant objet découplé structurellement du modèle relationnel.

### **Le SGBDR (Système de Gestion de Bases de Données Relationnelles)**

Adressé de manière transactionnelle par des ordres SQL générés par l'ORM, le SGBDR assure la persistance et les propriétés ACID du système de bases de données.

L'ORM Hibernate / EJB3 assure une compatibilité (relativement transparente) avec de nombreux SGBDR du marché. Il est à noter que Generic System n'utilise que les fonctionnalités les plus simples des SGBDR (pas de procédure stockée, pas de Triggers..) et fonctionne bien avec un niveau d'isolation de transactions dit « REPEATABLE\_READ ». Nos expérimentations se sont généralement faites avec un SGBDR de type embarqué (HSQLDB, H2...), dont les performances sont accrues.

L'indépendance de Generic System envers le SGBDR fait partie intégrante de l'objectif de Generic System. L'utilisation d'une base de donnée relationnelle est rapidement devenue celle d'un service comme n'importe quel autre. Celle-ci n'a pas eu un grand intérêt, ni sur la forme concernant nos recherches, ni sur le fond concernant son utilité réelle (cf Generic System 2).

Beaucoup des difficultés conceptuelles et techniques ont été résolues en dehors de la base de données relationnelle, souvent volontairement afin de s'abstraire de celle-ci par la suite.

## **Generic System UI**

Generic System UI est un module qui permet de se brancher sur Generic System Core pour que l'application dans son ensemble puisse être accédée par une interface graphique de type Web 2.0. La présence de ce module est obligatoire à toute application Web.

Il a servi de prototype de conception pour la prise en compte dans le noyau de nombreuses fonctionnalités inhérentes aux interfaces graphiques et manquantes jusqu'alors (gestion native de structures arborescentes pour les composants graphiques, indexage des informations, ordonnancement et filtrage de résultats de requêtes, paramétrage contextuel ...).

Ce module s'appuie sur le framework JSF et d'autres bibliothèques compatibles avec ce framework de présentation : Ajax4JSF, Richfaces pour le Web 2.0, et la bibliothèque de gestion des templates Facelet.

Il est intéressant de signaler ici, que le modèle MVC a été partiellement enfreint dans Generic System au profit d'un mécanisme bien plus innovant et complètement orienté « composants ». Aucune navigation de page web n'est réellement proposé par Generic System : tout se fait par rafraichissement partiel (de type AJAX) d'une seule et me page Web. Il y a cependant un mécanisme astucieux qui permet l'indexation par les moteurs de recherche du résultat des rafraichissements partiels de cette page web, afin de ne pas condamner la visibilité de nos interfaces sur Internet.

La conception de ce module s'est avérée indispensable, puisque sa présence est requise par tout prototype nécessitant une interface utilisateur. La conception des différents prototypes de conception de Generic System 1 nécessitaient des interfaces utilisateurs et ils sont donc en partie dépendant de ce module mais aussi envers le noyau de Generic System bien évidemment.

## **2.2 Prototypes de conception**

### **2.2.1 Generic System Tracker (à partir de Novembre 2007)**

Le module Tracker, est le premier prototype de conception que nous avons développé et expérimenté à partir de Novembre 2007 pendant la conception même de Generic System 1. Destiné au suivi d'anomalies de Generic System lui-même, il définit un premier exemple de modèle d'information dont Generic System a permis la gestion.

Le caractère auto-référent de ce module a fait de lui un module dont la mise au point fut extrêmement sensible car similaire à la typologie auto-référente des problèmes rencontrés dans le noyau : le moindre dysfonctionnement dans l'exploitation de ce module nous empêchait d'y référencer ce même dysfonctionnement.

Le module a permis de référencer les différentes tâches à réaliser dans Generic System, de leur associer une priorité, un statut, un genre, de les assigner à des utilisateurs, de connaître les versions de Generic System affectés, d'en connaître la description et leur date de résolution etc ...

Ce prototype de conception a permis au noyau d'acquérir la capacité de gérer n'importe quel modèle (diagramme de classe UML) qui contiendrait des entités, des attributs, des relations d'association, d'agrégation et de composition ainsi que de gérer leur cardinalité.

A ce stade, la gestion de l'héritage et des relations n-aires sont encore manquantes.

Ce prototype peut être consulté directement en ligne sur <http://www.genericsystem.com>

GENERIC SYSTEM 6.4 GA LIVE DEMO - TRACKER | Tâche | Tâches non résolues | GS-01622 - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils Aide

http://www.genericsystem.com/tracker/type-tache/taches-n... Fichier ja

GENERIC SYSTEM 6.4 GA ...

middleWARE

WEB TRACKER

Connexion

**Tâche(s)**

- Tâches non résolues
- Toutes les tâches
- Tâches résolues

**Priorité(s)**

- Liste des Priorité(s)

**Statut(s)**

- Liste des Statut(s)

**Genre(s)**

- Liste des Genre(s)

**Commentaire(s)**

- Tous les commentaires

**Version(s)**

- Toutes les versions

**Tâches non résolues** Rechercher

| Référence | Date             | Description       | Etat       | Priorité  | Assigné       | Genre         |
|-----------|------------------|-------------------|------------|-----------|---------------|---------------|
| GS-00038  | 25 oct. 2007 11  | Revue des mes     | NON RÉSOLU | OPTIONNEL | nfeybesse     | AMELIORATION  |
| GS-00975  | 16 juil. 2008 13 | Concevoir la for  | NON RÉSOLU | MAJEUR    | nfeybesse     | NOUVELLE FONC |
| GS-00978  | 16 juil. 2008 13 | Permettre le pai  | NON RÉSOLU | CRITIQUE  | nfeybesse     | NOUVELLE FONC |
| GS-00979  | 16 juil. 2008 13 | Rectifier l'envoi | NON RÉSOLU | CRITIQUE  | nfeybesse     | AMELIORATION  |
| GS-01050  | 31 août 2008 21  | Refondre l'envo   | NON RÉSOLU | CRITIQUE  | nfeybesse     | AMELIORATION  |
| GS-01295  | 28 sept. 2008 1  | Concevoir un m    | NON RÉSOLU | BLOQUANT  | nfeybesse     | AMELIORATION  |
| GS-01622  | 12 déc. 2008 17  | Généraliser les   | NON RÉSOLU | MAJEUR    | ckhoury       | AMELIORATION  |
| GS-01623  | 12 déc. 2008 17  | Refondre la ges   | NON RÉSOLU | MAJEUR    | nfeybesse     | AMELIORATION  |
| GS-01634  | 16 déc. 2008 12  | Mettre les mess   | NON RÉSOLU | MAJEUR    | nfeybesse     | NOUVELLE FONC |
| GS-01890  | 07 avr. 2009 13  | Lorsqu'un type    | NON RÉSOLU | OPTIONNEL | Non renseigné | ANOMALIE      |
| GS-01891  | 07 avr. 2009 13  | Lors de l'édition | NON RÉSOLU | MAJEUR    | Non renseigné | ANOMALIE      |
| GS-02501  | 19 janv. 2010 1  | Mise en place d   | NON RÉSOLU | MAJEUR    | nfeybesse     | NOUVELLE FONC |
| GS-02504  | 19 janv. 2010 1  | Mise en place d   | NON RÉSOLU | MAJEUR    | nfeybesse     | NOUVELLE FONC |

**Tâche GS-01622**

**Description \***

Généraliser les conteneurs sur toute la page

**Etat \***

NON RÉSOLU

**Priorité \***

MAJEUR

**Createur**

ckhoury

**Assigné**

ckhoury

**Version(s) affectée(s) \***

- ☐ GENERIC SYSTEM 6.0 GA
- ☐ GENERIC SYSTEM 6.0 RC1
- ☐ GENERIC SYSTEM 6.0 RC2
- ☐ GENERIC SYSTEM 6.1 GA
- ☐ GENERIC SYSTEM 6.1 RC1
- ☐ GENERIC SYSTEM 6.2 GA

Terminé

GENERIC SYSTEM 6.4 GA LIVE DEMO - NEXT GENERATION MANAGEMENT SYSTEM - Mozilla Firefox

EichierEditionAffichageHistoriqueMarque-pagesOutilsAide

http://www.genericsystem.com/Google

Java SE 6PFGS PF (local)JmxPw GeneratorSecurity for Win...RichFaces DemoGeneric SystemJSF 2.0GS 2.0

Tâches non résolues

Toutes les tâches

Tâches résolues

Priorité(s)

Créer un(e) Priorité

Liste des Priorité(s)

Statut(s)

Créer un(e) Statut

Liste des Statut(s)

Genre(s)

Créer un(e) Genre

Liste des Genre(s)

Commentaire(s)

Tous les commentaires

Version(s)

Créer un(e) Version

Toutes les versions

|          |                     |                       |            |           |               |                |
|----------|---------------------|-----------------------|------------|-----------|---------------|----------------|
| GS-00038 | 25 oct. 2007 11:36  | Revue des messag      | NON RESOLU | OPTIONNEL | nfeysesse     | AMELIORATION   |
| GS-00975 | 16 juil. 2008 13:46 | Concevoir la foncti   | NON RESOLU | MAJEUR    | nfeysesse     | NOUVELLE FONCT |
| GS-00978 | 16 juil. 2008 13:52 | Permettre le param    | NON RESOLU | CRITIQUE  | nfeysesse     | NOUVELLE FONCT |
| GS-00979 | 16 juil. 2008 13:53 | Rectifier l'envoi de  | NON RESOLU | CRITIQUE  | nfeysesse     | AMELIORATION   |
| GS-01050 | 31 août 2008 21:26  | Refondre l'envoi de   | NON RESOLU | CRITIQUE  | nfeysesse     | AMELIORATION   |
| GS-01295 | 28 sept. 2008 12:3  | Concevoir un moye     | NON RESOLU | BLOQUANT  | nfeysesse     | AMELIORATION   |
| GS-01622 | 12 déc. 2008 17:26  | Généraliser les cor   | NON RESOLU | MAJEUR    | okhoury       | AMELIORATION   |
| GS-01623 | 12 déc. 2008 17:26  | Refondre la gestion   | NON RESOLU | MAJEUR    | nfeysesse     | AMELIORATION   |
| GS-01634 | 16 déc. 2008 12:44  | Mettre les message    | NON RESOLU | MAJEUR    | nfeysesse     | NOUVELLE FONCT |
| GS-01890 | 07 avr. 2009 13:52  | Lorsqu'un type a pi   | NON RESOLU | OPTIONNEL | Non renseigné | ANOMALIE       |
| GS-01891 | 07 avr. 2009 13:53  | Lors de l'édition d'u | NON RESOLU | MAJEUR    | Non renseigné | ANOMALIE       |
| GS-02501 | 19 janv. 2010 11:11 | Mise en place de te   | NON RESOLU | MAJEUR    | nfeysesse     | NOUVELLE FONCT |
| GS-02504 | 19 janv. 2010 11:11 | Mise en place de la   | NON RESOLU | MAJEUR    | nfeysesse     | NOUVELLE FONCT |

Tâche GS-01295

Description \*

Concevoir un moyen de permettant aux composants Faceted JSF d'être simplement paramétrés simplement et à volonté.

Etat \*

NON RESOLU

Priorité \*

BLOQUANT

BLOQUANT

CRITIQUE

MAJEUR

OPTIONNEL

TRIVIAL

Commentaire

Créateur

nfeysesse

Assigné

nfeysesse

Version(s) affectée(s) \*

☐ GENERIC SYSTEM 6.0 GA

☐ GENERIC SYSTEM 6.0 RC1

☐ GENERIC SYSTEM 6.0 RC2

☒ GENERIC SYSTEM 6.1 GA

Version corrigée

GENERIC SYSTEM 6.2 GA

17/10/2008

Création : 28 sept. 2008 10:30 par nfeysesse

Dernière modification : 17 nov. 2008 10:31 par thomas version 50

Copyright © 2005-2010 Middleware Factory - Tous droits réservés

Terminé

## 2.2.2 Generic System Web : (à partir de Janvier 2008)

Ce module est un prototype de conception qui nous a permis que Generic System aie les capacités d'un CMS (Content Management System) et persiste ainsi des informations de toute nature : images, fichiers multimedia, textes formatés etc.

Nous en avons profité pour élaborer le site Internet de Middleware Factory à l'aide de ce module.



Ce prototype peut être consulté directement en ligne sur <http://www.middlewarefactory.com>

Il faut bien entendu s'identifier et posséder les droits pour bénéficier des fonctionnalités de CMS.



## 2.2.3 Generic System CRM (à partir de Mai 2008)

Ce module est un prototype de conception qui permet de se servir de generic System tel un CRM (Customer Relationship Management). Il a permis la conception au sein du noyau de la gestion de tous types de relations qu'on peut trouver dans un diagramme de classe UML (héritage, association, composition, agrégation...) mais surtout leur manipulation s'est effectuée dans le cadre d'une application exploitée par des utilisateurs non informaticiens, ce qui a fortement accentué les contraintes techniques pour assurer une réelle souplesse d'utilisation. Par la suite, nous nous en sommes servi pour faire le suivi commercial des clients de Middleware Factory avec succès pendant les années 2009-2010.

Ce prototype peut être consulté directement en ligne sur <http://www.middlewarefactory.com>, à condition de s'identifier et de posséder les droits.

La capture d'écran sera réalisée dans 1 seconde

Middleware Factory | Middleware Factory Leader en France dans le RIA - Mozilla Firefox

Echier Edition Affichage Historique Marque-pages Outils Aide

http://www.middlewarefactory.com/web/accueil/middleware-factory/leader-ria

Java SE 6 PF GS PF (local) Jmx Pwd Generator Security for Win... RichFaces Demo Generic System JSF 2.0 GS 2.0

MAILING WEB CRM USER MAIL PROSPECTEUR

**Entreprise(s)**

- + Créer un(e) Entreprise
- Toutes les entreprises

**Contact(s)**

- + Créer un(e) Contact
- Tous les contacts

**Fonction(s)**

- + Créer un(e) Fonction
- Liste des Fonction(s)

**Secteur d'activité(s)**

- + Créer un(e) Secteur d'activité
- Liste des Secteur d'activité(s)

**Action(s)**

- Toutes les actions

**Mail(s)**

- Liste des Mail(s)

**Environnement technique(s)**

- + Créer un(e) Environnement technique
- Liste des Environnement technique(s)

**Environnement fonctionnel(s)**

- + Créer un(e) Environnement fonctionnel
- Liste des Environnement fonctionnel(s)

**Adresse(s)**

- Liste des Adresse(s)

**Toutes les entreprises**

Rechercher +

| Référence       | Standard      | Secteur d'activité |
|-----------------|---------------|--------------------|
| AGF             | 0153001803    | ASSURANCE          |
| AIR FRANCE      | Non renseigné | ENERGIE & TRANSI   |
| ALLIANCE HEALTH | 0140805100    | INDUSTRIE PHARM.   |
| ALSTOM          | 0141492000    | ENERGIE & TRANSI   |
| ALTUGLAS        | 0178662300    | INDUSTRIE,PRODUX   |
| AMEXIO          | 0155581818    | SSI                |
| ANEO            | 0146442466    | SSI                |
| AON FRANCE      | 0158757575    | ASSURANCE          |
| APEC            | Non renseigné | SECTEUR PUBLIC     |
| AREVA           | 0134966000    | ENERGIE & TRANSI   |
| ARTSYS          | 0143671949    | SSI                |
| ASPHERIA        | 0134634949    | EDITEUR DE LOGIC   |
| ASSU2000        | 0148101500    | ASSURANCE          |
| ATOS EUROMEXT   | 0173030303    | SSI                |
| ATOS ORIGIN     | Non renseigné | SSI                |
| AUDITEC         | 0153246666    | EDITEUR DE LOGIC   |
| AVIVA FRANCE    | 0176625000    | ASSURANCE          |

1 2 3 >>>

**Entreprise ATOS EUROMEXT**

**Contacts**

|                   |                   |                            |   |
|-------------------|-------------------|----------------------------|---|
| DEWERSE DOMINIQUE | ddewerse@myx.com  | RESPONSABLE D'EXPLOITATION | ✖ |
| BISSORE           |                   | ASSISTANTE DSI             | ✖ |
| HERSFELD GIL      | ghersfeld@myx.com | RESPONSABLE DES ETUDES     | ✖ |
| CROWS MICHEL      |                   | ASSISTANTE DSI             | ✖ |

**Informations générales**

Standard: 0173030303 ✖ + Adresse(s): ✖ + Secteur d'activité: SSI ✖

Création: 03 nov. 2008 16:19 par ychami Dernière modification: 03 nov. 2008 16:19 par ychami version 45

Copyright © 2005-2010 Middleware Factory - Tous droits réservés

Terminé

## **2.2.4 Generic System Intermediation (de Novembre 2010 à Août 2011 Partnersfit.com)**

Tentative avortée de commercialiser des services basés sur l'inter-médiation entre SSII pour rapprocher les besoins et les ressources de cette catégorie d'entreprise, le module est un prototype de conception qui nous a permis de mettre au point le noyau pour pouvoir être exploité dans une application à but commercial et industriel.

La mise au point du noyau et du module interface graphique a canalisé nos efforts pour couvrir de nouveaux besoins (exploitation asynchrone des informations et gestion de meta-informations / propriétés héritables notamment), au détriment des efforts de commercialisation des services eux-même et de la communication autour de ceux-ci.

Depuis, malgré le fait que le module avait atteint les objectifs qu'on s'était fixé en terme de fonctionnalités, nous n'avons pas souhaité investir d'avantage dans la commercialisation des services qu'il proposait, pensant que cette action commençait à s'écarter de notre souhait premier de mettre au point un système d'information tout objet.

Cette décision a été prise dans un contexte où un succès commercial restait malgré tout hypothétique. Nous avons préféré concentrer nos efforts sur la conception d'une deuxième version de Generic System que nous espérons encore plus pertinente que la première.

En ligne pendant plus d'un an sur <http://www.partnersfit.com>, ce module n'est désormais consultable qu'au sein des locaux de Middleware Factory.

Partners Fit - La plateforme d'intermédiation des professionnels de l'IT - Mozilla Firefox

FichierÉditionAffichageHistoriqueMarque-pagesOutilsAide

http://localhost:8080/partnersfit.com/profils/profitype/tousprofils

Google

Java SE 6PFGS PF (local)JmxPwd GeneratorSecurity for Win...RichFaces DemoGeneric SystemJSF 2.0GS 2.0

Bonjour ThomasDéconnexion

partnersfit

ACCUEILVOTRE COMPTEMISSIONSPROFILSNOS ENGAGEMENTSNOUS CONTACTERMAILERPROSPECTEURNOS CONDITIONS

Profils disponibles

Soumettre un profil

Voir tous les profils

Voir vos profils disponibles

Voir vos profils en attente

Vos profils archivés

Tous les profils en attente

Tous les profils inactifs

Diffusez en quelques instants les profils de vos consultants à plus de 1500 partenaires référencés !

Voir tous les profils

Rechercher

| Titre   | Disponibilité | Tarif | Mobilité      | Maj. le       |
|---|---------------|-------|---------------|---------------|
| CONSULTANTE AMOA / CHEF DE PROJET                                   | 23 août 2010  | 470   | ILE-DE-FRANCE | 23 août 2010  |
| CP / CONSULTANT / FORMATEUR DRUPAL PHP                              | 23 août 2010  | 450   | ILE-DE-FRANCE | 14 août 2010  |
| ARCHITECTE - DBA ORACLE / 20 ANS D'EXPÉRIENCE / ORACLE, UNIX        | 07 août 2010  | 460   | TOTALE        | 07 août 2010  |
| DEVELOPPEUR PHP, AJAX, C#, DOTNET, JAVA, ...                        | 05 août 2010  | 370   | ILE-DE-FRANCE | 05 août 2010  |
| INGÉNIER D'ÉTUDE  |               |       | TOTALE        | 05 août 2010  |
| SENIOR IT PROJECT   |               |       | TOTALE        | 05 août 2010  |
| TECHNICIEN D'EXP  |               |       | ILE-DE-FRANCE | 05 août 2010  |
| INGÉNIER SYSTÈME  |               |       | ILE-DE-FRANCE | 05 août 2010  |
| ADMINISTRATEUR  |               |       | TOTALE        | 02 août 2010  |
| INGÉNIER DE PRO   |               |       | TOTALE        | 02 août 2010  |
| INGÉNIER D'ÉTUDE  |               |       | ILE-DE-FRANCE | 29 juil. 2010 |
| INGÉNIER D'ÉTUDE  |               |       | TOTALE        | 29 juil. 2010 |
| CONSULTANTE SU  |               |       | ILE-DE-FRANCE | 27 juil. 2010 |
| DEVELOPPEUR JAV   |               |       | ILE-DE-FRANCE | 26 juil. 2010 |
| PACKAGEUR MSI - ADMINISTRATEUR SYSTÈME - TECHNICIEN HELP DESK - HOT | 22 juil. 2010 | 450   | ILE-DE-FRANCE | 22 juil. 2010 |
| DEVELOPPEUR PHP JUNIOR  | 22 juil. 2010 | 240   | TOTALE        | 22 juil. 2010 |
| HOMOLOGATEUR - RECEPTEUR  | 22 juil. 2010 | 280   | VOR PROFIL    | 22 juil. 2010 |
| INGÉNIER D'ÉTUDE / CORREL - GENOS - R&D                             | 18 juil. 2010 | 350   | TOTAL E       | 18 juil. 2010 |

Création : Profil (Etape 1 / 6)

Titre \*

INGÉNIER JAVA / JEE

Suivant

Partners Fit est une marque déposée de la société Gold Faces - Copyright © Gold Faces 2009-2010 - Tous droits réservés

Terminé

Partners Fit - LA PLATEFORME D'INTERMÉDIATION DES PROFESSIONNELS DE L'IT - Mozilla Firefox

Eichier Edition Affichage Historique Marque-pages Outils Aide

http://localhost:8080/partnersfit.com/missions/missiontype/toutesmissions

Java SE 6 PF GS PF (local) Jmx Pwd Generator Security for Win... RichFaces Demo Generic System JSF 2.0 GS 2.0

Bonjour Thomas [002103] [Completweb [002103]] [Déconnexion]

ADMIN

Missions

Mission

Proposer une mission

Voir toutes les missions

Voir vos missions

Toutes les missions

Saisissez une proposition - là où il y a plus de références !

Attributs

Double Long

Date String

File Boolean

Associations

Compositions

Héritage

Conteneurs

Editer les propriétés du Filtre [Id:2135] Voir toutes les missions [Id:2370]

| Clé                                | Valeur                                     | Type    | Action              |
|------------------------------------|--|---------|---------------------|
| abstractInheritanceDisabled        | <input checked="" type="checkbox"/>        | Boolean |                     |
| activated                          | <input type="checkbox"/>                   | Boolean |                     |
| adUrl                              | 728x90_middleware.gif                      | String  |                     |
| admin                              | <input type="checkbox"/>                   | Boolean |                     |
| applicationCopyright               | Middleware Factory                         | String  |                     |
| applicationFooter                  | Copyright © 2005-2010 Middleware Factory - | String  |                     |
| applicationKeyword                 | Generic System                             | String  |                     |
| applicationSubtitle                | NEXT GENERATION MANAGEMENT SYSTEM          | String  |                     |
| applicationTitle                   | GENERIC SYSTEM 6.4 GA LIVE DEMO            | String  |                     |
| attributeClassName                 | String                                     | String  |                     |
| attributeClassNamePropertyType     | AttributeClassName                         | String  |                     |
| columnSize                         | 100  | Long    |                     |
| columnNumber                       | 1  | Long    |                     |
| comparatorExpression               | #{occ2.get('mission-                       | String  | Supprimer propriété |
| createTimestampPropertyType        | DateTime                                   | String  |                     |
| cssStyle                           | text-shadow: #666 1px 1px 1px;             | String  | Supprimer propriété |
| dated                              | <input type="checkbox"/>                   | Boolean |                     |
| defaultSelected                    | <input checked="" type="checkbox"/>        | Boolean | Supprimer propriété |
| defaultSelectedInheritanceDisabled | <input checked="" type="checkbox"/>        | Boolean |                     |
| displayEditor                      | true                                       | String  |                     |
| displayExcelExport                 | false                                      | String  |                     |
| displayPrinter                     | false                                      | String  |                     |

Insérer

Insérer [Id:2380] Insérer [Id:2380]

Terminé

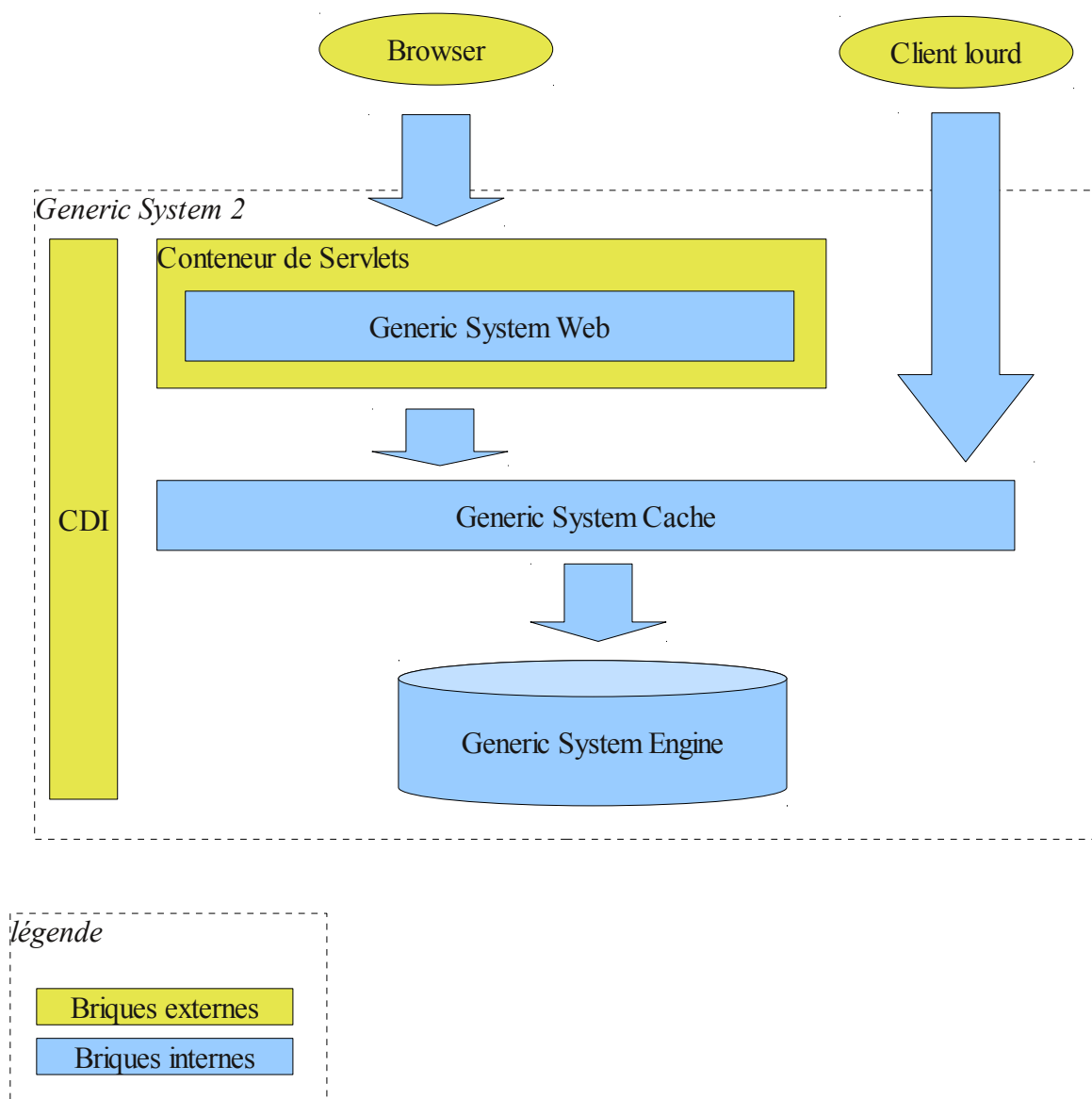
### 3 Travaux effectués sur Generic System 2 (à partir de Juin 2009)

#### 3.1 Analyses

##### 3.1.1 Analyse conceptuelle

Cf annexe Generic System V2 – Analyse conceptuelle

##### 3.1.2 Analyse organique



### **3.1.3 Analyse fonctionnelle**

TODO

### **3.2 *Prototypes de conception***

Plusieurs prototypes de conception sont en cours de réalisation (2011) et seront décrits dans cette section au cours de cette même année.

## **4 Conclusion**

Les qualités révélées par l'approche proposée par Generic System nous permettent d'envisager des progrès très significatifs dans l'exploitation des système d'information, que ce soit d'un point de vue économique mais également scientifique.

Aujourd'hui, nous continueront d'investir dans Generic System, qui à notre sens, ne désigne pas seulement un système d'information, mais aussi une méthode très innovante de représentation de l'information, dont il est la première expérimentation.

Désormais, nous allons parfaire la mise au point de la seconde version de Generic System 2 et par la suite, concentrer nos effort sur sa diffusion et son exploitation.

## **5 Ressources affectées au projet**

### ***5.1 Temps passé et tâches réalisées par chaque ingénieur (début 2008 – fin 2009)***

**Nicolas Feybesse**

**Chef du projet de Recherche & Développement**

*Depuis Janvier 2007*

*Diplôme : Ingénieur ENSI Bordeaux*

Affectation permanente au projet de recherche à 75 % du temps (rapports du serveur de version des sources consultable sur site).

Conception et développement de Generic System 1 et Generic system 2

- Analyses conceptuelle, organiques et fonctionnelle, recherche et modélisation mathématique
- Développement et mise au point du noyau et des prototypes, mise en œuvre de méthodes innovantes et de concepts dérivés de l'intelligence artificielle, tests et jeux d'essai.

**Thierry Bouhnik**

**Ingénieur Recherche & Développement Java/JEE**

*du 05/11/2007 au 08/02/2008*

*Diplôme : Mastère en Sciences et Technologies du Logiciel*

**Mission :**

Recherche et développement sur Generic System : socle applicatif permettant d'isoler le paramétrage fonctionnel (règles métiers) du modèle de base de données et de gérer l'accès concurrentiel et distant à l'information :

- Personnalisation totale de l'IHM par paramétrage : construction d'un arbre de conteneurs de composants JSF représentant une page Web et dont le paramétrage se fait par des EJB Entity appelés récursivement.
- Création de nouveaux composants JSF pour personnaliser par paramétrage l'affichage et le filtrage de listes d'entités
- Enrichissement du mapping objet/relationnel EJB3 par annotations
- Correction d'anomalies dans les liaisons JSF / EJB Session
- Utilisation systématique des mécanismes d'inversion de contrôle (IOC) du framework SEAM
- Intégration de composants J2EE AJAX (Client Riche)
- Mise en place et configuration d'un Server Proxy
- Sécurisation du serveur Web Apache : génération d'un certificat et prise en compte du celui-ci par le protocole sécurisé HTTPS
- Réalisation d'un Webservice dans Generic System afin de valider l'adresse Email des nouveaux comptes utilisateurs.
- Mise en place de la fonction Reverse Proxy d'Apache pour générer des URLs bookmarkables (pretty URLs)
- Adaptations pour rendre les URLs bookmarkables (pretty URLs)
- Mise en cache d'une partie des pages Web
- Interventions régulières sur la base de données (restructurations, indexations, contraintes d'intégrité référentielle, de non nullité et d'unicité).

Environnement : Generic System 1 UML, Java 5, J2EE, JBoss 4 (EJB3), Hibernate 3, Eclipse, Webservices, Linux (Kubuntu), JSF 1.2, Expressions EL, Apache, VirtualHosts, Proxy Squid, Framework Seam, Richfaces, Ajax4JSF, Design Patterns (IOC, MVC...), XML, SVN, LOG4J, ANT, HSQLDB, SQL.



**Mission :**

Recherche et développement sur Generic System : socle applicatif / meta-application permettant la réalisation d'applications Web de gestion d'informations, « Clé en main » par un simple paramétrage.

- Développement de composants J2EE/ AJAX riches à l'aide de Richfaces, Ajax4JSF, Javascript, AJAX, JSF, Facelet (Arbres, tableaux avec possibilités de filtrage et de tri, drag and drop). Site visible sur <http://www.genericsystem.com>
- Cloisonnement logique de l'information de Generic System à partir de la notion de « Compte »
- Filtrage Hibernate, modification des composants JSF/Facelet pour la gestion des comptes, et la répartition des données sur chacun d'eux
- Conception d'un écouteur de phases (JSF/SEAM) pour faciliter la correction d'anomalies afin de les chronométrer et d'évaluer la performance de chacune d'elles
- Migration des scripts de compilation, packaging et déploiement de ANT vers MAVEN 2
- Gestion des dépendances (versions des bibliothèques) et configuration de plugins MAVEN2
- Restructuration modulaire de Generic System
- Elaboration d'une stratégie de tests (JUnit) et configuration de la génération automatique de la documentation de l'API (JavaDoc).
- Enrichissement et modifications du Mapping Relationnel/Objet EJB3 par annotations pour la gestion des images (icônes)
- Prise en charges des icônes dans l'interface graphiques.
- Réinstallation et reconfiguration du serveur SVN
- Paramétrage complet d'un nouveau module de gestion de factures (Diagramme de classe UML2)
- Participation à la réalisation d'un Webservice d'interrogation paramétrable du Generic System
- Configurations multiples en XML (Faces-config.xml, page.xml, persistence.xml, Web.xml,...)

Environnement : Generic System 1, UML 2, JAVA/J2EE, JDK6, JSF 1.2, Richfaces 3.2, Ajax4JSF, JBOSS 4 (EJB3), Hibernate 3, Eclipse 3.4, Linux (Kubuntu), JavaBeans, Framework Seam 2, Expressions EL, Design Patterns (IOC, MVC...), Webservices, XML, XHTML, SVN, LOG4J, Maven 2, m2eclipse, HSQL, SQL

**Guillaume Gorson Tanguy**

*du 6 Octobre 2008 au 31 Octobre 2008*

*Diplôme : Mastère Professionnel IUP Génie Informatique*

**Ingénieur R&D JAVA/J2EE**

**Mission :** Recherche et développement sur Generic System 1 : socle applicatif / meta-application permettant la réalisation d'applications Web de gestion d'informations, « Clé en main » par un simple paramétrage (Live Demo : <http://www.genericsystem.com>)

- Analyse d'un besoin d'un client fictif industriel pour tester Generic System en condition réelle
- Analyse des écart entre les besoins et les capacités du noyau de Generic System
- Conception des fonctionnalités manquantes
- Développement et paramétrage fonctionnel de l'application (modèle des données, règles métiers, droits et workflow)
- Conception et développement de nouveaux composants Web Generic System en JSF/Facelet
- Amélioration du Framework GS (prise en compte d'annotations personnalisées JDK5 pour le paramétrage des objets systèmes)
- Optimisations dans la gestion du paramétrage dans le cache de second niveau d'Hibernate (configuration de JBoss Cache en mode transactionnel)
- Amélioration de la persistance des paramètres utilisateurs de l'IHM de Generic System par l'utilisation de Cookies

Environnement technique : Generic System 1, UML 2, JAVA/J2EE, JDK6, JBOSS 4 (EJB3), Hibernate 3, Eclipse 3.4, Linux (Kubuntu), JavaBeans, JSF 1.2, Richfaces 3.2, Ajax4jsf, Facelet, Framework Seam 2, Expressions EL, Design Patterns (IOC, MVC...), Webservices, XML, XHTML, SVN, LOG4J, Maven 2, m2eclipse, HSQL, SQL

**Mission :**

Recherche et développement sur Generic System 1 : socle applicatif / meta-application permettant la réalisation d'applications Web de gestion d'informations, « Clé en main » par un simple paramétrage. Live demo : <http://www.genericsystem.com>

- Industrialisation des tâches de compilation, packaging et déploiement avec Maven 2.
- Configuration de la gestion des repositories (Releases et Snapshots) et, des politiques de mise à jour des librairies.
- Configuration de la gestion des dépendances des librairies, de leur version et de leur scope. Elimination des dépendances cycliques.
- Configuration de plugins dans Maven 2 : *maven-compiler-plugin* et du *maven-ejb-plugin* pour compiler et packager les EJBs.
- Conception UML (diagramme de classe) à partir d'un cahier des charges pour un client industriel virtuel.
- Paramétrage de l'application (utilisation d'Expressions EL)
- Enrichissement et modification du Mapping Relationnel/Objet EJB3 par annotations (Modification des relations de composition et d'association OneToMany vers ManyToOne).
- Enrichissement du framework Generic System (Conception de nouvelles annotations JDK5).
- Conception d'un composant Seam d'analyse des classes par introspection afin de prendre en compte les nouvelles annotations
- Factorisations diverses dans le noyau du Generic System (Utilisation intensive des types paramétrés (Generics) et de classes internes, mise en place d'un héritage multiple d'implémentation).
- Maintenances corrective et évolutive sur les pages JSF/Facelet.
- Développement de nombreux composants JSF /Richfaces / AJAX d'arbres, de menu contextuel et de drags and drops dans l'administration de Generic System
- Tests unitaires

Environnement : Generic System 1, UML 2, JAVA (JDK 6), J2EE, JBoss 4 (EJB3), Hibernate 3, Eclipse, Linux (Kubuntu), JavaBeans, JSF 1.2, Framework Seam 2, Expressions EL, Design Patterns (Singleton, MVC, Factory), XML, XHTML, SVN, LOG4J, Maven 2, HSQL, SQL, Richfaces 3.2, Ajax4jsf

## Oussama Tabbakh

Du 01 Avril 2009 au 26 Mai 2009

diplôme : *Ingénieur en Génie Logiciel*

*Institut National des Sciences Appliquées et de Technologies (INSAT) de Tunis*

### **Mission :**

Recherche et développement sur Generic System 1 : socle applicatif / meta-application permettant la réalisation d'applications Web de gestion d'informations, « Clé en main » par un simple paramétrage. Live demo : <http://www.genericsystem.com>

- Configuration du plugin Maven2 *maven-war-plugin* pour packager le fichier WAR destiné au conteneur de servlets.
- Configuration du *maven-ear-plugin* pour packager les EAR, déclaration de librairies en « modules » visibles à la fois dans les Servlets et les EJBs.
- Pour l'ensemble de ces plugins : personnalisation des répertoires utilisés par défaut.
- Configuration d'Eclipse *Ganymede* et installation des Plug-in *m2eclipse* et du plug-in SVN *subversive*.
- Intégration de tâches ANT à Maven 2 (configuration du plug-in *maven-antrun-plugin*)
- Enrichissement et modification du Mapping Relationnel/Objet EJB3 par annotations (Modification des relations de composition et d'association OneToMany vers ManyToOne).
- Enrichissement du framework Generic System (Conception de nouvelles annotations JDK5).
- Développement d'un composant Seam d'analyse des classes par introspection afin de prendre en compte les nouvelles annotations
- Génération d'Emails personnalisés sur la base de templates (enrichissement de modèles de mails par analyse d'expressions EL)
- Réalisation et configuration des envois en mode asynchrone

Environnement : Generic System 1, UML 2, JAVA (JDK 6), J2EE, JBoss 4 (EJB3), Hibernate 3, Eclipse, Linux (Kubuntu), JavaBeans, JSF 1.2, Framework Seam 2, Expressions EL, Design Patterns (Singleton, MVC, Factory), XML, XHTML, SVN, LOG4J, Maven 2, HSQL, SQL, Richfaces 3.2, Ajax4jsf

**Pierre Carré**

**Ingénieur R&D JAVA/J2EE**

*du 16 Novembre 2009 au 15 Février 2010*

*Diplôme : Ingénieur Développement Logiciel CFA AFTI Orsay*

Recherche et développement sur Generic System 2.

Mise en place de 2 architectures de test en environnement JEE du nouveau noyau MVCC de Generic System réalisé par l'équipe de recherche et de développement :

- Mise en place et configuration des environnements :
  - JEE5 : JBOSS 5.1, EJB3, Seam 2.1, JSF2.0, Richfaces 3.3, Maven 2
  - JEE6 : JBOSS 6.0, EJB3, Weld 1.0 (CDI / JSR 299), JSF2.0, Maven 2
- Réalisation de tests pour concevoir les API EJB3 et JTA du noyau MVCC dans ces deux environnements.
  - Développement des couches implémentant les interfaces *javax.persistence.EntityManager* et *javax.transaction.Transaction*
- Développement de tests de du moteur en environnement Multi-threadé
  - Contrôle des synchronisations avec TestNg
  - Utilisation de l'API *java.util.Concurrent*
- Conception et réalisation au sein du noyau d'un gestionnaire d'archivage des Snapshots (sauvegardes) automatiques générés par le système d'information selon une fonction logarithmique avec des coefficients paramétrables.
- Développement d'une interface Web d'administration complète du système sur la base d'une arborescence d'objets en JSF/Richfaces
  - Visualisation complète de l'ensemble des objets et des relations du moteur
  - Mise à disposition des opérations CRUD sur l'ensemble des objets / relations.

Recherche et développement sur Generic System 1 :

- Conception et développement d'une couche intermédiaire afin d'isoler le graphe du système d'information (Modèle) de l'interface graphique (Vue)
  - Génération d'un arbre de composants graphiques préalablement à la génération des composants JSF par Facelet.
- Conception, paramétrage et développement d'une plateforme de génération d'Emails pour le module de Prospection de Generic System.
- Conception et réalisation d'un module statistique sous la forme de Webservices (comptage de clics et de connexions des prospects).

Environnements : Generic System 1, JAVA/J2EE, JBOSS 4.2, 5.1, 6.0, JSF 1.2 & JSF 2.0, Facelet, Jboss Seam, Hibernate, EJB3 Maven 2, JDK6, MySQL, SQL, Eclipse 3.5, SVN, Linux (Kubuntu), XML, Webservices.

## **5.2 Matériel**

Au cours de la conception de Generic System 2, nous avons eu besoin de tester les performances du nouveau noyau en environnement multi-threadé et cela sur une machine possédant un processeur multi-coeurs, afin de concevoir des algorithmes permettant de répartir convenablement les traitements sur les différents cœurs disponibles lorsque la charge augmente.

Nous avons acquis une machine réputé performante possédant un processeur de type Intel Core I7, et possédant 4 coeurs (8 coeurs hyperthreadés) afin d'accompagner la conception du noyau de Generic System 2.

Nous exploitons cette machine sur la base d'un système de type Linux avec une JVM de type Open JDK Version 6.

## **6 Annexe : Generic System 2 - Analyse Conceptuelle**