

L'infocentre collaboratif sur mesure Ontomantics

Introduction et Concepts

Contributeurs :

**Philippe BAUMANN,
Hayeme BECHA,
Sébastien ESPEISSE,
Patrick GUERIN,
Christophe TAVERNE,
Laurent TROMEUR
Loïc VILLANNE**

Date de dernière modification : 04/12/2009

Ontomantics S.A.S
Centre d'Innovation
16 rue Léonard de Vinci
45074 Orléans Cedex 2

<http://www.Ontomantics.com>

Société par actions simplifiée au capital de 356 000 euros
RC Orléans - SIRET 489 022 491 00018

Tel 02 38 25 90 29
Fax 02 38 69 80 42

Contact: Philippe Baumann
philippe.baumann@Ontomantics.com

Tel 06 72 35 06 65

Sommaire

1	Introduction et contexte.....	3
1.1	Technologie :.....	3
1.2	Différenciateurs :.....	3
1.3	Bénéfices :.....	3
2	Présentation de la solution Ontomantics.....	4
2.1	Présentation technique simplifiée de la solution Ontomantics.....	4
2.2	Procédé de restitution applicative.....	6
2.2.1	Concept de référentiel applicatif.....	6
2.2.2	Modélisation des règles métier et le concept des expressions.....	7
2.2.3	Notion de bases de données applicatives.....	7
2.2.4	Le Web Engine.....	7
2.3	Gestion des droits et rôles.....	8
2.3.1	Mécanisme	8
2.3.2	Procédé de matricage applicatif des rôles utilisateur.....	8
2.4	Administration et métriques.....	9
2.4.1	Le mécanisme de publication.....	9
2.4.2	Monitoring pour le pilotage et la supervision.....	9
2.4.3	Procédé de déploiement du modèle de données.....	10
2.4.4	Import / export applicatif.....	11
2.5	Pour étendre les possibilités de la plate-forme.....	11
	ANNEXE 1 : Spécifications fonctionnelles d'Ontomantics (V3.0.10)	12
	ANNEXE 2 : Technologies J2EE sur lesquelles s'appuie le produit Ontomantics.....	14

1 Introduction et contexte

1.1 Technologie :

Ontomantics est une jeune entreprise innovante (JEI) du génie logiciel. Soutenue par des acteurs de l'innovation dont Oséo et la Région Centre, la société a développé une solution d'automatisation en rupture avec les outils pré-existants : progiciels standards du commerce ou environnements de développement.

L'infocentre collaboratif Ontomantics adresse les besoins personnalisés de traitement de l'information métier, de dématérialisation et de pilotage des organisations par une automatisation sur mesure en complément des systèmes en place.

Avec seulement deux jours de formation un utilisateur Ontomantics réalise plus vite, l'équivalent de ce que produirait un développeur java chevronné.

1.2 Différenciateurs :

- Conception entièrement web ne nécessitant pas d'installation de postes clients, notamment une ergonomie fluide obtenue avec une large palette de composants d'interface utilisateur.
- Utilisation facile sans écriture de code : cinématique applicative entraînée par un moteur logique, pour automatiser des processus complexes et communiquer avec les systèmes existants.
- Cohérence du système : regroupement des applications dans un référentiel sécurisé centralisant les processus et les profils.

1.3 Bénéfices :

Retour sur investissement immédiat :

- facilité de mise en place sur une plate-forme standard,
- exécution sur mesure sans contraintes techniques,
- efficacité organisationnelle résultante.

Prise en compte des besoins métiers :

- satisfaction des utilisateurs dans le cadre de la gouvernance informatique
- sécurité par un pilotage fin des droits et des rôles,
- traçabilité de l'utilisation.

Pérennité de l'environnement applicatif :

- technologie éprouvée
- isolation du métier et de la technique
- maintenabilité et évolutivité des applications

2 Présentation de la solution Ontomantics

La plate-forme Ontomantics s'inscrit dans le contexte où les utilisateurs ont besoin de mettre en oeuvre rapidement et simplement des outils de productivité, décisionnels et collaboratifs, idéalement des applicatifs web intranets ou extranets, en mode RIA (Rich Internet Application) technologie de prédilection du web 2.0. Les directions informatiques sont d'une part confrontées à une nécessité de réduction des coûts et n'ont pas les moyens de répondre à toutes les demandes de leurs utilisateurs d'autre part. Le produit Ontomantics vise donc à répondre à ce besoin en facilitant notamment les phases de conception, de maintenance et de déploiement applicatif tout en respectant les standards éprouvés du monde de l'entreprise (en particulier le socle Java/J2EE).

Les procédés techniques mis en oeuvre au coeur de la plate-forme applicative Ontomantics, s'appuient sur des composants libres et reconnus. Ils permettent d'établir un socle applicatif qui encapsule les descriptions applicatives réalisées par les utilisateurs "concepteurs" au sein d'un référentiel centralisé sous forme de métadonnées..

Ceci, afin de pouvoir ensuite les "jouer", les gérer, les maintenir, et les exploiter dynamiquement. Le terme « jouer » est employé volontairement puisque les applications ne sont pas exécutées au sens habituel du terme mais interprétées dynamiquement. En effet, le procédé de restitution applicative n'utilise pas de code, ni de compilation pour pouvoir exécuter les applications.

On ne parle donc pas de programmation au sens habituel du terme, mais bien de description applicative, puisque les applications ne sont pas réalisées à partir de l'écriture d'une logique de développement habituelle, mais d'une modélisation entièrement effectuée à la souris à partir des fonctionnalités prédéfinies et mises à disposition dans des interfaces ergonomiques fournies par la plate-forme.

Ainsi plusieurs organisations et usages peuvent être mis en oeuvre autour d'Ontomantics.

2.1 Présentation technique simplifiée de la solution Ontomantics

La plate-forme Ontomantics est une solution Web 2.0. full AJAX (Asynchronous JavaScript and XML) . Tous les modules qui la composent sont entièrement web. La plate-forme est elle même une application J2EE qui tourne au sein d'un serveur d'applications.

Avant de pouvoir accéder à la plate-forme et donc aux applications web2.0, tout utilisateur doit s'identifier au préalable par l'intermédiaire d'une première page d'identification (saisie des logins et mot de passe).

Une fois connecté et identifié, une interface appelée « switcher » permet à l'utilisateur identifié d'accéder, suivant ses droits, à 3 modules distincts qui forment la plate-forme:

- le module DESIGNER qui permet de concevoir et designer des applications,
- le module PLAYER qui permet de jouer les applications mises à disposition,

– le module MANAGER qui permet de réaliser les tâches d'administration et d'affecter les droits utilisateurs.

Ainsi l'utilisateur autorisé peut designer (concevoir et créer) des applications, assigner des droits et publier (mettre à disposition) ces applications mais également jouer (exécuter) les applications avec un simple navigateur internet et donc sans aucune installation de logiciel tiers sur son poste client. Ceci est possible en se connectant simplement au serveur qui héberge la plate-forme Ontomantics.

Le module DESIGNER

Une application (au sens Ontomantics) est constituée de trois composantes principales:

- les écrans,
- les comportements applicatifs métiers,
- et les modèles de données.

On retrouve ainsi la trame Modèle Vue Contrôleur (pattern MVC) . Le module designer met à disposition un ensemble d'outils permettant de décrire les différents aspects de l'application. Différentes interfaces principales permettent donc au travers des modules de la plate-forme :

- de gérer les applications, de les exporter/importer,
- de modéliser les écrans par un simple drag & drop full web à partir d'une bibliothèque de composants (full ajax) disponibles,
- de décrire des comportements applicatifs métiers, comprenant eux même une ou plusieurs règles métiers
- de définir et gérer les jeux de données et variables nécessaires,
- de modéliser un modèle de données en vue de le déployer dans une source de données autorisée,
- de partager son ou ses propres modèles de données,
- d'affecter des comportements applicatifs aux événements utilisateurs directement « cablés » sur les composants des écrans,
- de définir les rôles applicatifs éventuels,
- de définir des styles graphiques applicables pour le rendu graphique des applications,
- de définir des traductions en vue de l'internationalisation de l'application,
- de jouer l'application en cours de réalisation dans un mode dit de débogage.

Le module PLAYER

Le module PLAYER propose à l'utilisateur la liste des applications qui ont été publiées (par un administrateur) et autorisées en mode « exécution ».

Ainsi, l'utilisateur peut lancer l'application de son choix. Ce module exploite donc pleinement les mécanismes de restitution applicative qui sont décrits ci-après.

Le module MANAGER

Le module MANAGER reprend les fonctions proposées par le module DESIGNER et propose en plus les fonctionnalités d'administration.

Différentes interfaces permettent donc :

- de gérer les groupes utilisateurs, les utilisateurs,

- d'affecter des permissions aux groupes utilisateurs sur les applications hébergées (consultation, modification, suppression, publication, exécution)
- d'administrer les sources de données,
- d'affecter des permissions sur les sources de données pour les rendre accessibles aux groupes de designers,
- de gérer les modèles de données,
- de gérer les bibliothèques de fonctions externes,
- de configurer la plate-forme en ayant accès à ses variables de configuration système,
- de sauvegarder le référentiel applicatif,
- de publier les applications,
- de réaliser le monitoring de la plate-forme,
- de réaliser le monitoring du système,
- d'affecter les rôles applicatifs définis par les designers aux groupes utilisateur.

2.2 Procédé de restitution applicative

2.2.1 Concept de référentiel applicatif

Le référentiel applicatif est la clef de voute de la solution Ontomantics. En effet, plutôt que d'être stockée dans un système de fichiers ou bien dans une structure complexe type XML, la description applicative est contenue dans un référentiel applicatif. Pour des questions de performance, ce dernier s'appuie sur un SGBD qui apporte toute la souplesse et les performances nécessaires aux requêtes complexes induits par le fonctionnement du moteur logique décrit ci-après.

Cette particularité des applications décrites au sein d'un même environnement induit un grand nombre d'avantages:

- uniformisation, homogénéité et cohérence du parc applicatif et des techniques de mise en oeuvre,
- centralisation et simplification de l'accès aux applicatifs de l'entreprise,
- mise à disposition de fonctions transverses de gestion et de supervision,
- facilité à maintenir le parc applicatif, à l'administrer à partir d'un point unique,
- possibilité de construire des applications depuis n'importe où sans avoir besoin d'installer de nombreux outils hétérogènes et complexes,
- sécurité garantie par la plate-forme hôte (Les concepteurs bénéficient des retours et corrections engendrées par l'ensemble des utilisateurs),
- les fonctionnalités mises à disposition dans la plate-forme sont éprouvées, testées et maintenues par une société éditrice spécialisée, ce qui garantit la qualité des fonctions proposées. Ainsi, les concepteurs n'ont plus qu'à se focaliser sur la description de la logique métier sans se préoccuper de la technique de mise en oeuvre, de compatibilité des bibliothèques ou des détails de restitution.
- aucun déploiement applicatif n'est nécessaire pour utiliser les applications puisqu'elles sont directement interprétées à la demande.

Le référentiel adopte une structure optimisée pleinement exploitée par le moteur logique Ontomantics.

2.2.2 Modélisation des règles métier et le concept des expressions

Au sens Ontomantics, un comportement qui peut être directement rattaché à un évènement applicatif, possède une ou plusieurs règles. Ces dernières étant constituées de conditions et d'actions.

Un premier groupe d'actions est prévu pour être interprété si l'ensemble des conditions renvoie une valeur ou renvoie « vrai ». Un second groupe de conditions est quant à lui prévu pour être interprété si l'ensemble des conditions renvoie « null » ou « faux ». Partant de ce principe et à partir de tous les types de conditions et d'actions prévues, il est possible de couvrir la totalité des besoins métiers des applications de gestion.

Enfin, toutes les erreurs ou cas non gérés et exceptions sont redirigés vers des comportements spéciaux dits d'erreur qu'il est possible de définir d'une manière similaire.

Un aspect central de la modélisation des règles métiers est le concept d'expressions. En effet, au sein de la plate-forme, les règles sont présentées sous forme d'arborescence. Il est possible de préciser sa description au niveau de chacun des noeuds qui le nécessitent en y greffant des expressions.

Cette modélisation permet ainsi aux concepteurs d'applications de pouvoir décrire tous types de processus du plus simple au plus complexe, la plate-forme fournissant de nombreux types d'opérateurs et d'opérandes.

2.2.3 Notion de bases de données applicatives

En parallèle du référentiel applicatif, la plate-forme a besoin de pouvoir accéder aux bases de données applicatives clientes. Ainsi, un accès indépendant a été implémenté en parallèle de l'accès raditionnel vers le référentiel. Cet accès JDBC, optimisé par le moteur logique, est couplé à un pool de connexions dans un objectif de performance.

Ces différents accès aux données sont règlementés par l'intermédiaire du module d'administration qui permet d'établir des règles précises pour chacun d'eux. (Lecture seule, mise à jour autorisée ou non, déploiement autorisé ou non, etc...).

2.2.4 Le Web Engine

Contrairement à un système traditionnel où l'application est compilée ou pré-compilée, la solution Ontomantics s'appuie sur un moteur logique (WebEngine) dit: «moteur logique d'interprétation dynamique évènementiel et récursif ».

En effet:

- il interprète une description dynamiquement c'est à dire qu'il produit les pages et actions résultantes à la volée.
- Il joue les règles qui ont été décrites à partir du déclenchement d'évènements utilisateurs (clics, double clics, changement de valeur d'un composant, survol à la souris, etc...) sur lesquels ont été « cablés » des comportements applicatifs.
- Enfin, ce mécanisme est récursif, dans la mesure où des règles peuvent s'appeler elles-mêmes.

Pour résumer le fonctionnement du moteur logique, on peut dire qu'il s'appuie sur un algorithme qui lui permet d'extraire les descriptions applicatives stockées dans le référentiel applicatif, de recréer les objets applicatifs nécessaires ainsi que leurs

liaisons afin de pouvoir jouer et servir dynamiquement les applications demandées par les utilisateurs.

En contrepartie, on imagine aisément que les ressources nécessaires pour extraire toute une description applicative sont importantes et à plus forte raison lorsqu'un grand nombre d'utilisateurs simultanés exécutent les applicatifs.

En conséquence, un système de cache à plusieurs niveaux a été conçu pour optimiser la performance des traitements applicatifs. Cette mise en cache est une avancée technologique supplémentaire de la plate-forme.

Les objets qui constituent les applications ne sont donc "demandés" au référentiel qu'une première fois. Après quoi ils sont placés en mémoire afin de libérer le référentiel de requêtes nombreuses et coûteuses en temps. Pour améliorer ce procédé et le rendre plus transparent, un processus se charge de mettre en cache automatiquement et intelligemment les applicatifs. Les applicatifs les plus sollicités sont traités en priorité.

Ainsi, à partir d'un certain temps, la plate-forme ne travaille plus qu'en mémoire pour gérer les descriptions applicatives. Bien entendu, les modifications applicatives effectuées par les designers sont répercutées dans le référentiel et au niveau du système de cache.

2.3 Gestion des droits et rôles

2.3.1 Mécanisme

Le mécanisme de gestion des droits s'appuie la hiérarchie des groupes et utilisateurs. Le principe de base consiste à affecter des permissions aux groupes utilisateurs sur des applications de la plate-forme. Ainsi, les utilisateurs héritent des permissions des groupes auxquels ils appartiennent. Par ailleurs, un utilisateur faisant partie de plusieurs groupes ayant des permissions différentes sur des applications communes hérite des droits les moins restrictifs.

Les permissions applicables sont les suivantes :

- modification,
- suppression,
- exécution,
- consultation,
- publication.

Notons également que le mécanisme de gestion des droits de la plate-forme, également stocké au sein du référentiel applicatif, peut être couplé à un annuaire d'entreprise de type LDAP/Active Directory.

Dans ce cas la hiérarchie des groupes utilisateurs n'est alors plus gérée par la plate-forme mais déléguée à un service d'annuaire d'entreprise.

2.3.2 Procédé de matricage applicatif des rôles utilisateur

Une des problématiques des tâches d'administration est de pouvoir affecter des droits très précis lors de l'exécution d'une application. Par exemple, il peut être nécessaire de permettre à un groupe d'utilisateurs de jouer une application, mais de ne pas avoir accès à certaines données sensibles et à certaines fonctionnalités

alors qu'un autre groupe d'utilisateurs pourra consulter les informations sensibles et aura accès à toutes les fonctionnalités.

Partant du principe que la personne qui administre le parc applicatifs (qui peut être constitué de centaines d'applications) n'a pas la maîtrise ni la connaissance des conceptions applicatives des applications présentes sur la plate-forme, la notion de rôle applicatif a été introduite.

Un rôle applicatif est donc la définition d'une façon de jouer une application. Ainsi, le designer de l'application peut créer autant de rôles qu'il désire et peut ensuite affecter ces rôles à des écrans, des comportements et des composants.

Ainsi, si un rôle est affecté à un écran, ce dernier ne sera consultable que si l'application est jouée dans ce rôle. De même pour les comportements et les règles.

Le moteur logique déduit le rôle dans lequel jouer l'application en fonction du matriçage applicatif. Le rôle de l'administration est donc grandement simplifié puisqu'il ne reste plus qu'à faire une matrice définissant les rôles affectés à un groupe utilisateurs (qui a bien évidemment la permission de jouer l'application).

Comme pour les permissions applicatives, si un utilisateur appartient à plusieurs groupes possédant des affectations de rôles contradictoires, les rôles les moins restrictifs sont appliqués lors de l'exécution de l'application.

2.4 Administration et métriques

2.4.1 Le mécanisme de publication

Indépendamment des permissions utilisateur affectées aux groupes par rapport aux applications hébergées et indépendamment des rôles applicatifs, un mécanisme de publication a été mis en place de manière à faciliter et à accélérer la mise à disposition ou le retrait des applicatifs sur la plate-forme.

Ainsi, une simple action dans l'administration permet de publier ou de dépublier une application sur la plate-forme. En cas de dépublication, ses affectations de permissions et de rôles sont conservés, mais l'application n'est plus disponible.

Par ailleurs, il est également possible d'automatiser la publication des applications souhaitées pendant des plages horaires prédéfinies pour différents besoins.

2.4.2 Monitoring pour le pilotage et la supervision

Toujours grâce au référentiel applicatif, à l'uniformisation des applicatifs et au regroupement sur un même socle de mécanismes communs (ou fonctions transverses), la plate-forme est capable de communiquer à l'utilisateur un grand nombre d'états et variables de fonctionnement en temps réel.

Il est ainsi possible de connaître en temps réel quel utilisateur est connecté sur quel applicatif, de pouvoir interagir en temps réel avec lui grâce aux nouvelles technologies Ajaxpush.

La traçabilité peut être poussée puisque ce socle commun rend possible le pistage de la plupart des actions utilisateur jusqu'aux composants graphiques, constituant les écrans.

En regroupant et retraçant toutes ces mesures, il est ensuite possible de retracer des statistiques très précises.

Il est notamment possible de connaître de manière générale:

- le nombre d'écrans visités,
 - le nombre de connexions,
 - la durée moyenne de connexion,
 - la proportion du nombre de connexions par application,
 - la proportion de la durée moyenne de connexion par application,
 - la proportion du nombre d'écrans visités par application,
- ou bien pour chacune des applications:
- le nombre d'écrans visités,
 - le nombre de connexions,
 - la durée moyenne de connexion.

Ce qui est novateur ce n'est pas tant d'obtenir ces statistiques en mode exécution d'application, mais c'est surtout d'obtenir ces informations décisionnelles dans "le mode développement d'applications" sans l'ajout d'outils supplémentaires et technologies tierces.

En effet, ces statistiques peuvent être obtenues également en phases de design puisque le même socle étant utilisé en mode « player » et en mode « design », il devient donc plus facile de piloter des équipes dans le développement d'applicatifs en visualisant ces indicateurs précis en temps réel.

2.4.3 Procédé de déploiement du modèle de données

Une autre innovation de la plate-forme repose sur la couche du modèle de données Ontomantics. En effet, l'utilisateur Designer a la possibilité de concevoir le(s) modèle(s) de données qui sera(ont) manipulé(s) par l'application à partir d'un module de la plate-forme prévu à cet effet. Ce dernier permet donc de décrire précisément les entités, les attributs ainsi que leur types.

Les règles des comportements applicatifs manipulent directement ces entités et attributs. Or, cette couche « modèle de données » n'est en réalité qu'une abstraction qui permet d'interagir avec les bases applicatives décrites précédemment (cf. 2.2) en apportant un niveau d'indirection.

Il s'agit bien d'abstraction puisque pour être utilisé, ce modèle de données a du être déployé dans une source de données physique (SGBD) et sa structure doit être parfaitement synchronisée.

Ainsi cette fonctionnalité apporte beaucoup de souplesse au designer puisqu'il est notamment possible:

- de resynchroniser le modèle de données avec la source de données en cas de modification de structure,
- à l'inverse, d'importer une source de données afin de déduire le modèle correspondant,
- de déployer un modèle de données dans différentes sources de données pour avoir la possibilité de travailler avec une base de données dite de tests et une base de données dite de production par exemple.

En outre, la plate-forme intègre l'interface de connexion avec la plupart des types de bases de données du marché. Ainsi, pour le designer, il n'est pas nécessaire

d'avoir à gérer les différents dialectes SQL ni de connaître les différentes techniques logicielles de connexion aux bases de données les plus courantes.

2.4.4 Import / export applicatif

Afin de pouvoir échanger les applications avec d'autres référentiels, un système d'import/export a été conçu. Il permet d'extraire une application du référentiel et de l'exporter sous forme de fichier « .xml ».

La fonction d'import, réalise l'opération inverse et faisant les contrôles nécessaires afin de garantir l'intégrité du référentiel, notamment par rapport aux différences éventuelles au niveau de la version de la plate-forme.

2.5 Pour étendre les possibilités de la plate-forme

A ce jour un nombre prédéfini d'actions et de conditions (éléments constitutifs des règles métiers) est mis à disposition des designers de la plate-forme.

Afin de retirer cette limite, un système de fonctions externes qu'il est possible d'appeler au niveau des comportements applicatifs a été mis en place. Ces fonctions sont développables dans le langage Java standard suivant un framework défini par Ontomantics. Les bibliothèques ainsi réalisées sont directement intégrables dans la plate-forme (à chaud).

Ainsi, les possibilités de la plate-forme peuvent être facilement étendues par des "utilisateurs avancés", connaissant le langage de programmation Java.

ANNEXE 1 : Spécifications fonctionnelles d'Ontomantics (V3.0.10)

- plate-forme web 2.0 full Ajax fondée sur un référentiel applicatif (clusterisable). Point d'entrée unique et centralisé aussi bien pour les modes d'utilisation dits: d'administration, de design ou de player,
- découpage fonctionnel de la plate-forme en 3x modules distincts: Manager, Designer et Player,
- système d'identification sécurisé (avec ou sans protocole https),
- moteur de règles événementiel et récursif (Précision des calculs en mode BigDecimal).

Variables d'environnements exploitables,

- ergonomie point&click intuitive et assistée pour la description des règles.
- Possibilité de consulter et tester les requêtes SQL générées,
- applications en cours de développement immédiatement disponibles et exécutables en ligne grâce au moteur logique d'interprétation dynamique intégré,
 - débbugger permettant de faciliter le paramétrage
 - multi bases de données. Connexion possible aux bases jdbc. (Implémentation actuelle:
Oracle, SQL server, MySql, Postgresql). Gestion des schémas et des bases nommées,
 - module de synchronisation assistée permettant de gérer et déployer les modèles de données dans les sources de données autorisées,
 - pool de connexion jdbc,
 - système de librairies de fonctions java externes utilisables au sein de la plate-forme pour étendre ses possibilités (frameworks Ontomantics spécifique),
 - Interface intuitive point&click et Drag&drop,
 - plate-forme Multi-langues (langues Anglais/Français actuellement disponibles),
 - applications internationalisables,
 - encapsulation de la gestion des groupes et des droits utilisateurs,
 - compatibilité LDAP/Active directory/annuaire Ontomantics,
 - système de droits couplé à un procédé de publication applicatives,
 - système avancé de rôles applicatifs pour permettre de définir finement des droits au sein des applications,
 - système de droits au niveau des sources de données et des modèles de données,
 - jeu étendu de composants Ajax disponibles :
 - saisie de texte,
 - label,
 - saisie de mot de passe,
 - zone de texte,
 - zone de texte formaté,
 - groupe,
 - bouton,
 - lien hypertexte,
 - cases à cocher,
 - boutons radio,
 - tableau paramétrable,
 - tableau croisé,
 - tableau éditable,
-

- champs à complétion assistée,
 - saisie de dates (calendrier),
 - graphique,
 - média (.mov, .avi, .swf, ...)
 - image,
 - upload,
 - googleMap,
 - saisie html libre,
 - arbres
 - menus
 - différents types d'évènements rattachables aux composants afin de permettre de déclencher des règles métier,
 - notion d'héritage d'écran et de templates pour faciliter la mise en page,
 - validateurs de convertisseurs de champs paramétrables au niveau présentation et/ou métier,
-
- génération automatique du menu applicatif possible,
 - multiples assistants intégrés pour accélérer la production applicative (Assistant de génération de formulaires à partir d'une source de données ou de modèles, assistant de création de graphique, assistant de publication, etc...),
 - import/export excel, cvs, xml,
 - module de définition de styles CSS intégré et paramétrable permettant de réaliser des templates événementiels. Possibilité de restreindre des applications à certains types de navigateurs,
 - module de publication pdf/word/html intégré à la plate-forme s'appuyant sur JasperReports (Possibilité de cabler directement les champs du rapport à publier sur des règles métier),
 - possibilité de s'affranchir de la plate-forme en rendant les applications directement accessibles par un lien internet,
 - possibilité de transmettre des paramètres aux applications,
 - mode anonyme disponible (Fonctionnalité site web),
 - module de monitoring et de supervision de la plate-forme:
 - monitoring en temps réel des utilisateurs connectés,
 - statistiques de connexions utilisateur,
 - différentes statistiques disponibles aussi bien en mode player qu'en mode design d'applications,
 - GED (Gestion électronique de documents) intégrée,
 - fonctionnalités d'import/export d'applications Ontomantics
 - outil de report de bug vers le support intégré,
 - prototype de "reverse cahier des charge" permettant de produire un document pdf formaté
- qui récapitule l'ensemble des éléments constitutifs de l'application: structure des: données, écrans, règles de l'application.
- système de sauvegarde incrémentale optionnel,
 - compatibilité avec les navigateurs suivants: Firefox 2.x à 3.x, Internet explorer 7 à 8 et Internet explorer 6 en mode player uniquement, safari 3.x et 4.x, Google Chrome 1.x et 2.x, Opera 9.x et 10.x en mode player uniquement.

ANNEXE 2 : Technologies J2EE sur lesquelles s'appuie le produit Ontomantics

- Java/JEE,
- EJB3,
- Hibernate + c3pO,
- Framework JSF,
- Icefaces,
- Jboss,
- Frontal Apache,
- Jdbc (proxool),
- LDAP/Active directory,
- JasperReports

Schéma d'architecture

