RNTL FAROS
Composition de contrats pour la Fiabilité d'Architectures Orientées Services

Livrable F-4.1
Coordonnateur : Mireille Blay-Fornarino

Contrats et compositions de services de l'application SEDUITE (SErvices de Diffusion Ubiquitaire d'InformaTions dans des Etablissements scolaires)

Projet FAROS

Décembre 2006

Projet RNTL FAROS : `http://www.lifl.fr/faros`

Table des matières

1. Introduction

L'objectif de ce document est de donner les premières bases pour la définition de contrats dans l'application SEDUITE. Cette application est un des trois systèmes informatiques qui servent d'éléments de validation pour le projet RNTL FAROS (Composition de Contrats pour la Fiabilité d'Architectures Orientées Services). Ce document constitue le livrable F-4.1 du quatrième lot.

L'application SEDUITE adresse la diffusion des informations dans les établissements d'enseignements. Elle est actuellement déployée dans le département informatique de Polytech'Nice Sophia Antipolis et dans les locaux de l'Institut d'Education Sensorielle Clément Ader pour les adolescents déficients visuels. Dans ce contexte, l'hétérogénéité des supports est indispensable. Les grands écrans diffusent des informations générales dont le contenu varie en fonction du lieu où ils sont situés et de ce fait de la population qui fréquente plus particulièrement cet emplacement : informations sur la vie des associations au niveau de la cafétéria, celles concernant les jurys et les réunions pédagogiques au niveau de la machine à café fréquentée par les enseignants de l'école, les emplois du temps en cours dans le hall, …

Cette application à la fois en exploitation et en évolution constante vise à permettre un partage des informations communes dont la diffusion centrée utilisateur s'adapte en fonction du contexte d'usage. Pour cela, une architecture basée sur une composition de services web a été mise en place (SOA). Elle s'appuie sur une forte modularité des différents services élémentaires (informations, étudiants, ...). Des études sont en cours pour prendre en compte les supports mobiles (PDA, téléphone) et prendre en compte la diffusion d'informations personnelles. En fonction du contexte d'usage, la réception des informations est différente. La notion de contexte inclut ainsi à la fois le profil de l'utilisateur (ses préférences – informations qu'il souhaite recevoir, son handicap éventuel – problèmes visuels, ...) et les conditions sur l'environnement (personne en cours, en pause, réception, etc.).

Dans un premier temps nous présentons brièvement l'application SEDUITE dans sa partie actuelle en exploitation et dans les évolutions à court terme envisagées (cf. 2).
Cette application est définie comme une composition de Web Services pour la partie système d'information et de composants pour la partie support de réception et capture de l'environnement. Nous décrivons ces compositions dans une deuxième partie (cf. 3).
La troisième partie est totalement prospective. Par l'analyse de l'application existante et visée ainsi qu'en prenant en compte différents problèmes que nous avons déjà rencontrés, nous exprimons les exigences sur notre système et établissons les bases pour quelques contrats que nous avons déjà pu diagnostiquer (cf. 4). Nous menons cette troisième partie volontairement comme guidée par les besoins et laissons ouvertes les possibilités de mises en œuvre des contrats dans nos plateformes cibles.

2. L'application SEDUITE : fonctionnalités

Cette application est connue dans sa version actuelle sous le nom de « EPUB », qui correspond à SEDUITE 1.0. Vous pouvez trouver différents éléments concernant la version en exploitation sous http://rainbow.essi.fr/epub.

Elle est développée sur la base d'un assemblage de web services pour la partie système d'informations et d'assemblages de composants Wcomp pour la partie prise en compte du contexte et diffusion adaptée. Cette application est par nature très évolutive car nous sommes amenés à ajouter régulièrement de nouveaux services, voire même déjà à substituer des services existants par de nouveaux services. C'est pourquoi elle se présente aujourd'hui à la fois comme une application en exploitation et comme une plateforme d'intégration de services.

a. Vision générale

Cas d'utilisation

SEDUITE adresse les trois cas d'utilisations présentés par la Figure 1 : Diffusion des informations au public, gestion des informations et administration de l'application. Même s'il n'est pas d'usage d'embarquer l'administration dans les cas d'utilisation, dans notre cas, l'adaptation « dynamique »[1] induit une gestion de la plateforme elle-même. Cette caractéristique est due à un développement orienté services et à une gestion pour l'instant très évolutive du contexte.

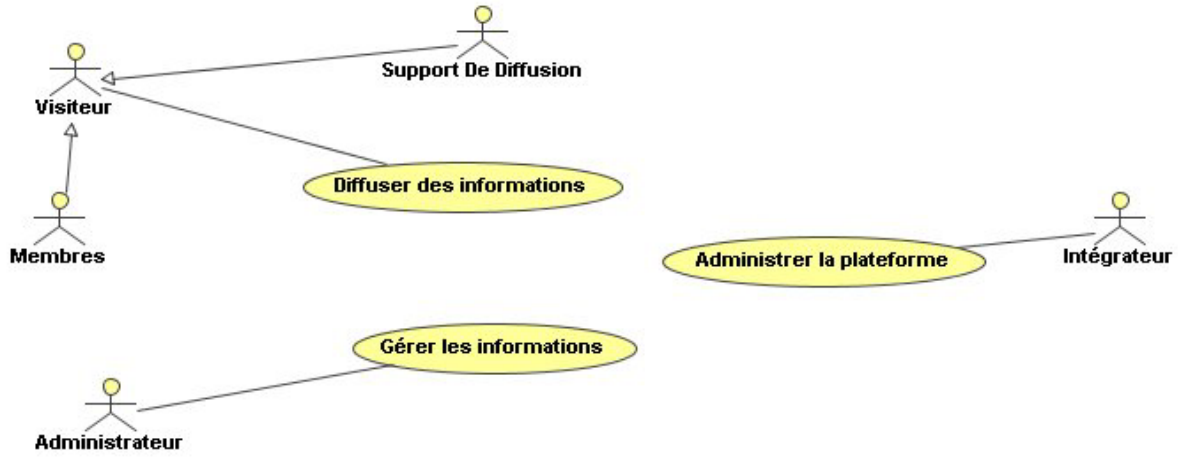


Figure 1 : Contexte de l'application SEDUITE

Le cœur de l'application est constitué par le système de diffusion des informations qui regroupe plusieurs cas d'utilisation (cf. Figure 2). La diffusion personnalisée repose à la fois sur le système d'information et sur l'appréhension du contexte.

[1] Il s'agit d'adapter l'application alors qu'elle est en exploitation pour ajouter de nouveaux services, …

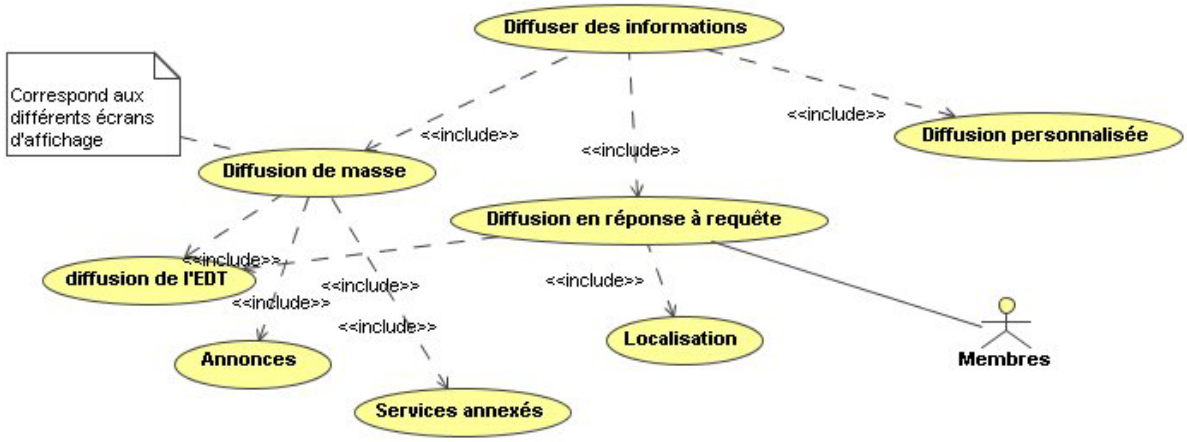


Figure 2 : Cas d'utilisation : diffuser des informations

La gestion des informations est distribuée entre chacun des services en support à la diffusion.

L'administration de la plateforme consiste essentiellement à ajouter de nouveaux services et à faire évoluer les services et clients en leur ajoutant de nouvelles possibilités dont la diffusion de nouveaux types d'informations. La Figure 3 identifie quelques-uns des grands cas d'utilisation que nous avons identifiés pour l'administration de la plateforme.

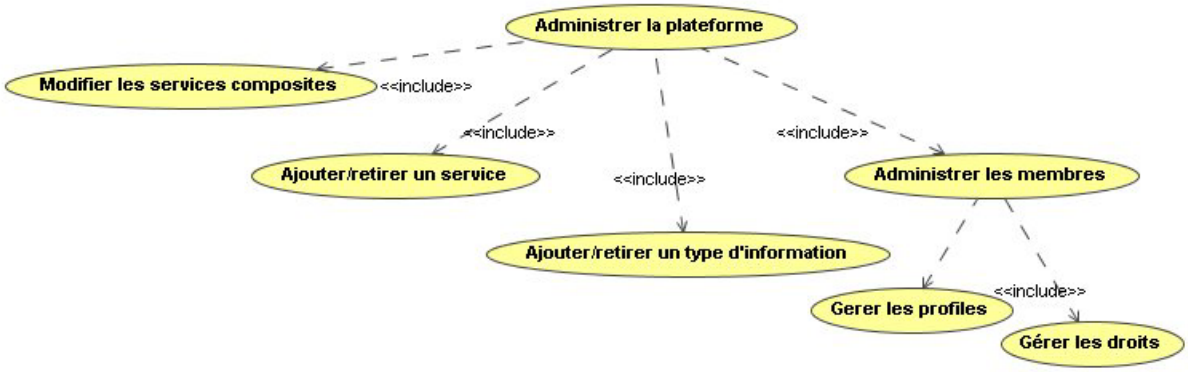


Figure 3 : Cas d'utilisation : administration de la plateforme SEDUITE

Architecture et technologies

Le système d'informations est développé en C# et met en jeu des web services et des clients lourds pour les écrans de diffusion des informations et d'administration.
La figure suivante visualise l'ensemble des services mis en jeu aujourd'hui dans l'application SEDUITE en exploitation.

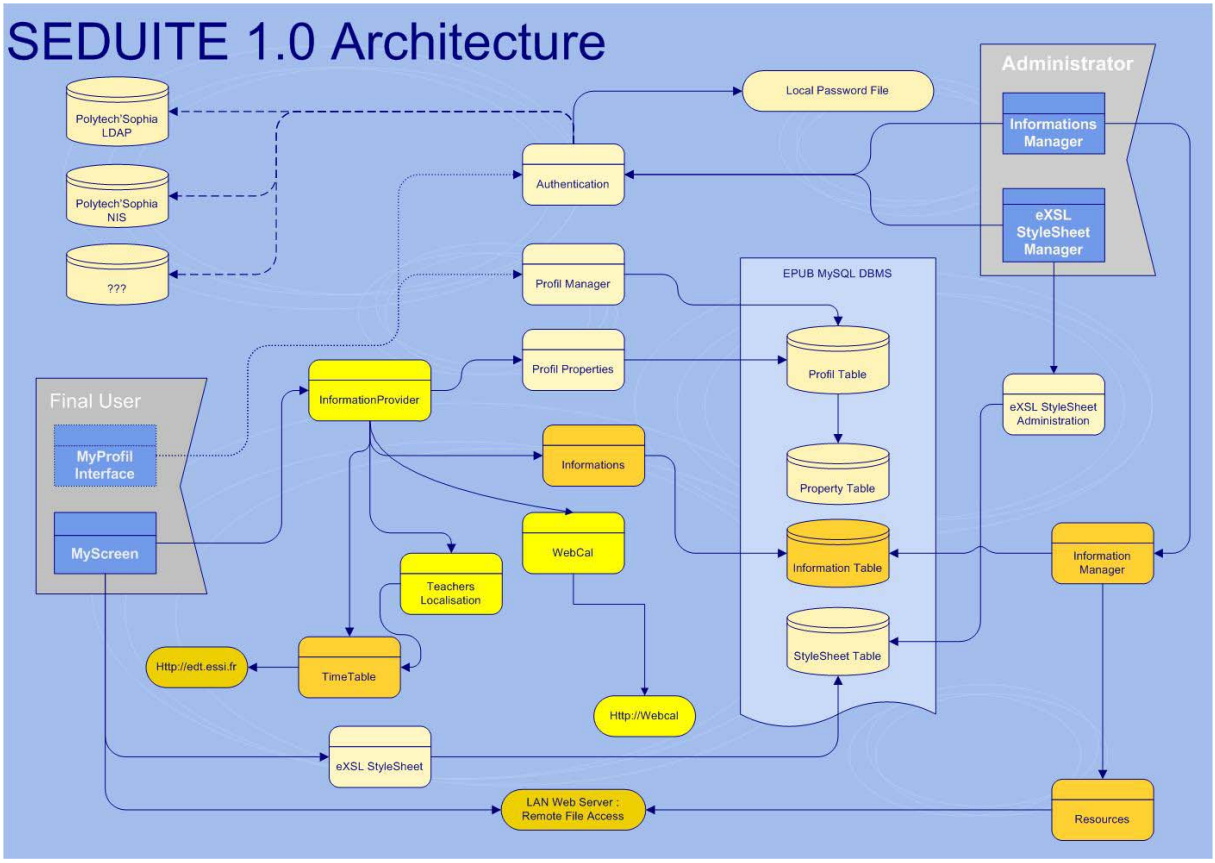
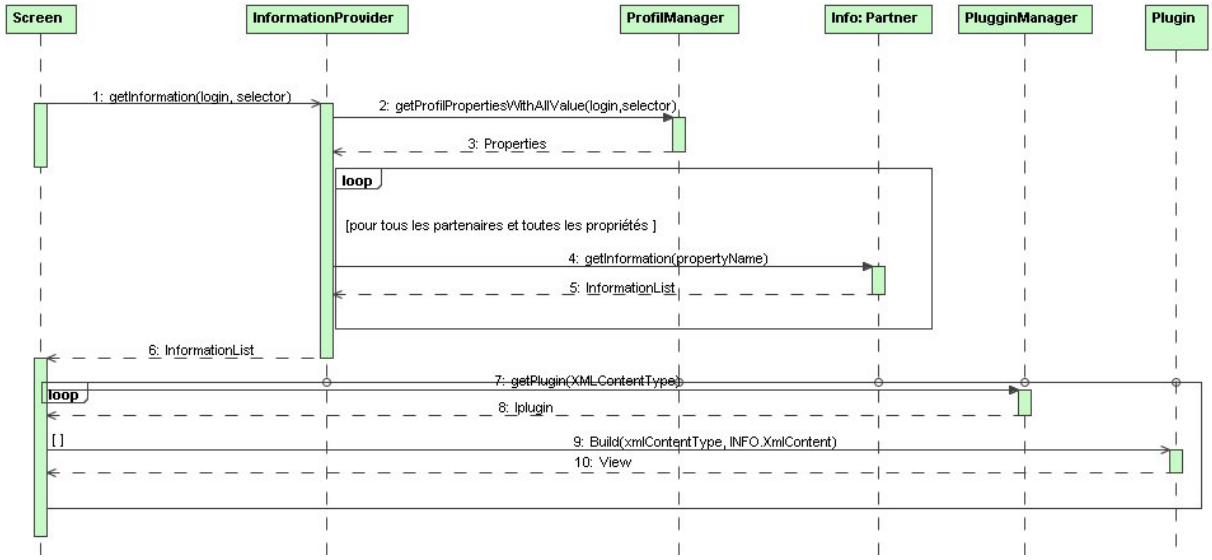


Figure 4 : Architecture de SEDUITE 1.0
(Les éléments orangés sont ceux correspondant à des services métier, les éléments bleus sont des clients, tandis que ceux non colorés correspondent à des éléments de services techniques).

b. Diffusion de masse

Le principe est qu'un « afficheur » reçoit des informations personnalisées en fonction du profil de son utilisateur (le profil est défini au lancement de l'application dans le fichier de configuration). Pour l'instant, nous avons les profils suivants : (screen, hall-pour-tous), (screen, 4ieme-pour-enseignant), (screen, Clement Ader-pour-DV), …
Un profil définit l'ensemble de types d'informations qui intéressent l'utilisateur telles que par exemple : emploi du temps, évènement, localisation des enseignants, agenda, ….
Ces informations sont données par des services différents. Des adaptateurs (dits Partner) nous permettent d'uniformiser les requêtes et les réponses. Celles-ci sont des chaînes de caractères XML au format « information » extensible par des champs xmlContent et xmlContentType.
En fonction des types d'informations, des plugins d'affichage dédiés sont sélectionnés. Ce mécanisme dynamique de sélection permet d'ajouter de nouveaux types d'informations dynamiquement.

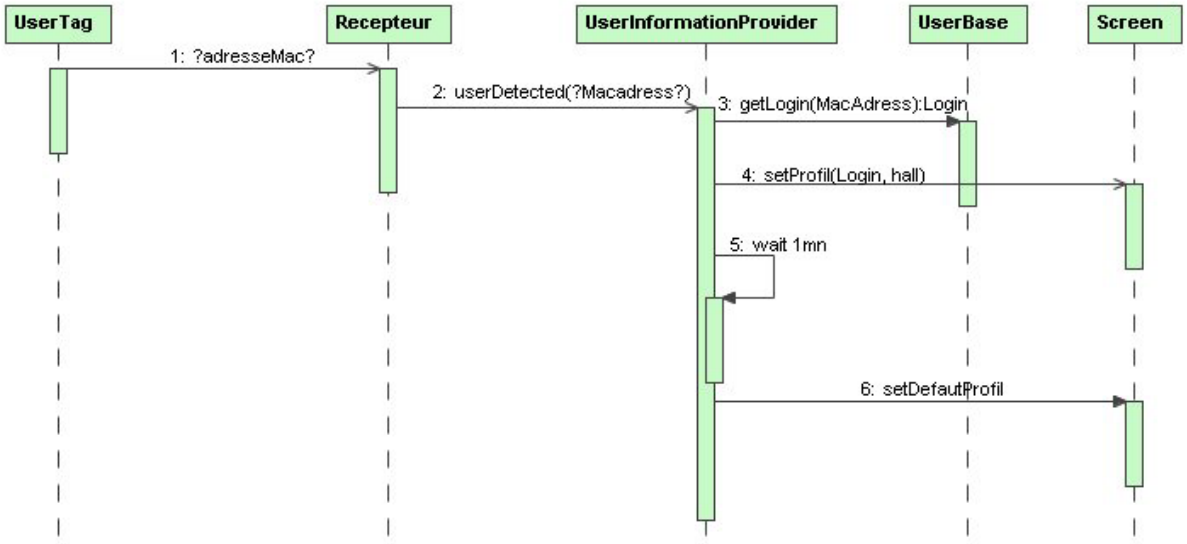
Voici un scénario représentant un afficheur quelconque.



c. Diffusion d'informations en fonction du contexte

Cette partie est plus hétérogène et est particulièrement extensible côté client par l'ajout rapide de nouveaux composants.

A l'écran de l'accueil est associé un serveur UPnP. Un service de gestion des utilisateurs gère l'association entre les tags utilisateurs et leur login. Le serveur UPnP associé à l'écran demande l'affichage à l'écran des informations en fonction de l'utilisateur présent. Cela donne lieu à un scénario simple où l'écran modifie son profil pour afficher les informations qui intéressent cette personne. Ce scénario se complique dès lors qu'on est amené à reconnaître plusieurs utilisateurs ayant des profils différents à proximité de l'écran. Un tel service de composition d'informations contextuelles n'est pas encore déployé.



d. Diffusion d'informations en fonction de requêtes

Nous avons défini un service de localisation qui permet de repérer les personnes qui passent à portée d'un récepteur. Ces informations sont stockées et permettent de repérer les personnes à leur dernière place connue, en mémorisant l'heure où cette information a été obtenue (cf. Figure 5). Le service d'EDT permet aussi la localisation d'une personne dans une salle pour une heure et une durée fixe en fonction de l'emploi du temps.

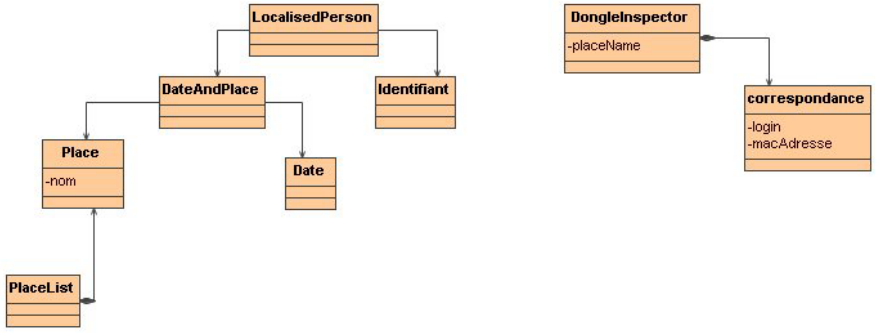


Figure 5 : Eléments de localisation

On peut accéder directement au service InfoProvider pour obtenir des informations en fonction d'une date ou d'un type.

3. Composition de services : Spécification

a. Compositions actuelles

Nous avons une approche en couche de la composition des différents services web, comme le montre la Figure 4 : Architecture de SEDUITE 1.0.

Paradigmes d'évolution

La **notion de partenaires** nous permet aujourd'hui d'ajouter de nouveaux web services. Un partenaire nous permet d'avoir une interface unique pour tous les services (cf. Figure 6). Au chargement de la classe correspondant à un nouveau type de partenaire (InfoPartner, TeacherLocPartner, par exemple), celui-ci crée une instance qu'il enregistre auprès du service InfoProvider.

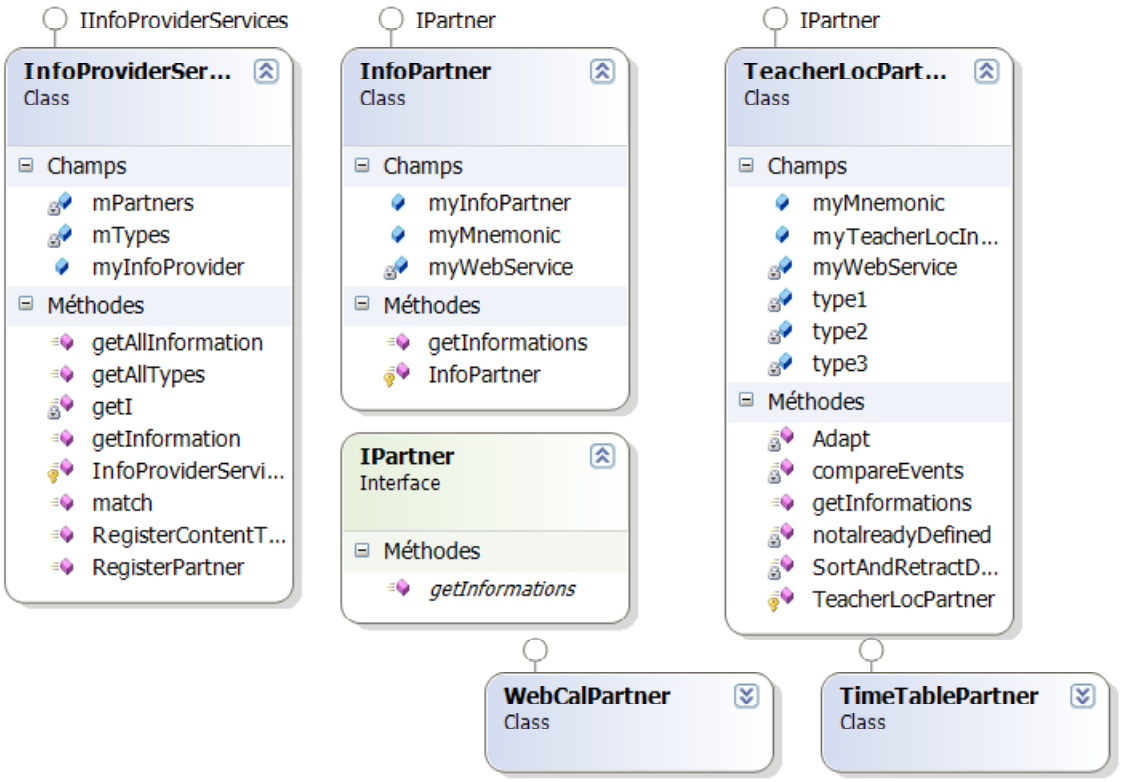


Figure 6 : Le Service InfoProvider et la notion de partenaires.
IPartner est l'interface des partenaires, *InfoPartner, TeacherLocPartner, WebCallParter, TimeTablePartner* sont des classes de Partenaires ;
La méthode *RegisterPartner* de *InfoProviderService* permet au partenaire de s'enregistrer et *RegisterContentType* de préciser les types d'informations fournit par un partenaire.

Les **plug-ins d'affichage** nous permettent d'adapter l'affichage en ajoutant de nouveaux plugins qui sont recherchés par l'application cliente lorsqu'elle reçoit des informations qu'elle ne sait pas afficher.

b. Spécification pour une composition basée sur des orchestrations

L'application SEDUITE est sujette à de nombreuses modifications et ajouts de services tandis que le système de diffusion s'étend et place l'utilisateur au centre en lui fournissant des informations mieux adaptées à ces attentes en fonction du contexte.

La notion de contexte est sujette à de nombreuses interprétations. Nous avons décidé de nous appuyer sur une notion de contexte d'exécution étayée par différentes sondes que nous serons capables de placer. En l'occurrence, actuellement la pertinence des sondes suivantes est en cours d'étude. Elles pourront être de deux types : fonctionnelles ou structurelles. Par exemple il pourra s'agir de l'heure de la journée, de l'emplacement des écrans fixes, du nombre d'informations à afficher, du nombre et de l'identité de personnes qui ont été détectées ou de la présence ou de l'absence de dispositifs ou d'équipements dans l'environnement.

Pour prendre en compte ces différents points nous pensons rajouter les deux services suivants : *ContextEvaluator* et *OrchestrationSelector* (cf. Figure 7).

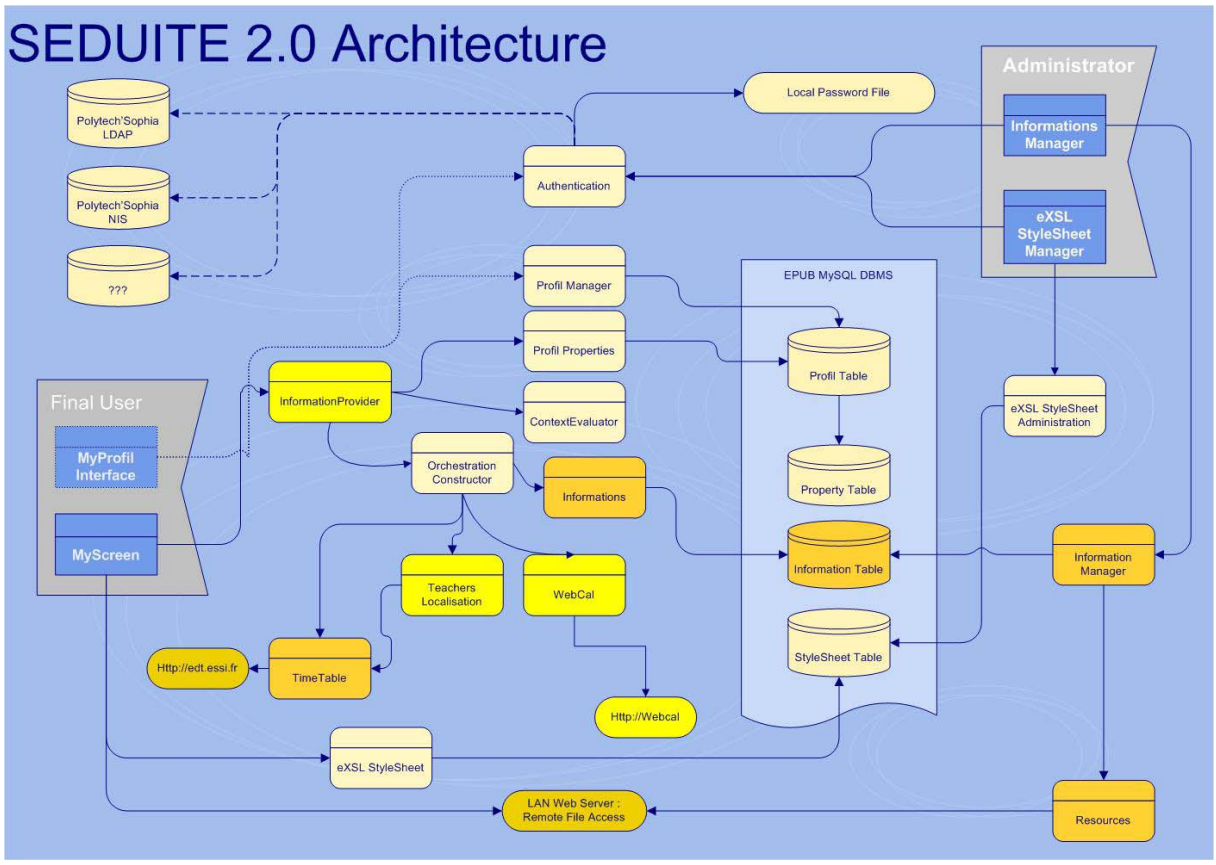


Figure 7 : Architecture de SEDUITE

Le premier nous permet d'envisager la capture du contexte comme un service, ce qui devrait nous permettre de faire évoluer l'application. Ce service pourrait être lui-même défini comme une composition de services d'évaluation du contexte.

Le second, en fonction du profil de l'utilisateur et du contexte, a la charge de déterminer l'orchestration des services à mettre en œuvre. Comme les orchestrations elles-mêmes devront être définies de manière séparée et devront évoluer dynamiquement.

La Figure 8 illustre par deux diagrammes d'activités la partie orchestration pour d'une part le grand écran (hall-pour-tous) et d'autre part l'écran des professeurs (4ieme-pour-enseignant) En prenant en compte les profils variés des utilisateurs, la multiplicité des supports et la gestion du contexte, il s'agira donc de construire dynamiquement ces orchestrations en respectant les contrats associés au récepteur des informations et au sélecteur d'orchestrations.

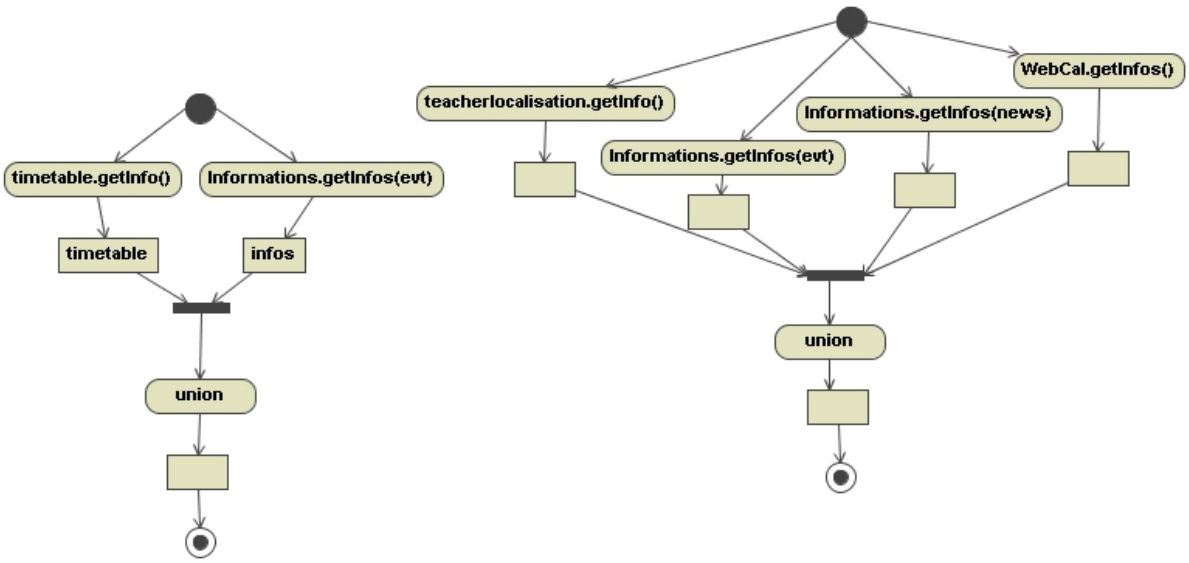


Figure 8 : Vers des orchestrations (gauche: grand écran, droite : écran coin café)

La prise en compte du contexte nous permettra d'étendre ces orchestrations pour prendre en charge la sélection des informations en fonction de leur pertinence. Par exemple, une des exploitations de la composition portera sur le couplage entre des services, donnant la possibilité d'avoir des informations de localisation de différentes qualités (GPS, tags RFID, détection de dispositifs Bluetooth, emploi du temps, etc.). En couplant ces informations aux timestamp*s*, on pourra ordonner partiellement les informations de localisation.

La partie administration des services (saisies des informations) est actuellement totalement séparée dans une démarche SOA. Les bases de données sont séparées. Il est néanmoins nécessaire de prévoir la corrélation de ces différents services à la fois pour masquer l'hétérogénéité des services, gérer les droits d'accès de manière partagée et ainsi éviter de s'authentifier à chaque fois.

Dans la suite, nous parlons de service composite lorsque nous désignons un service résultat d'une composition par orchestration de plusieurs services tel que l'obtention d'informations personnalisées « union » des informations souhaitées.

4. Besoins en contrats pour SEDUITE

L'application SEDUITE ne comprend actuellement aucune vérification de contrats. Des manques ont déjà été ressentis en matière de contrôle de l'évolution de l'application. Des besoins sont pressentis tandis que l'application se développe.

Nous présentons à présent les exigences posées sur le système. Les contrats qui en dérivent pour forcer leur respect sont présentés quand ils apparaissent naturellement. D'autres seront dérivés dans un processus global d'ingénierie des modèles par raffinements corrélés des éléments de l'application et des exigences qui leur sont associées.

a. Gestion des délais d'obtention des informations

Une personne ne doit pas attendre plus de $t_{attenteMax}$ pour voir les informations qui l'intéressent. Cette exigence va présenter différentes déclinaisons en fonction des cas d'utilisation, du point de vue technique ou rendu de services.

Nom	(CT) Respect d'un temps d'attente maximum pour l'affichage des informations
Description	
Expression informelle	Soit t1..tn les types d'infos qui intéressent un utilisateur, Soit $t_{attenteMax}$, le temps maximum d'attente d'une personne devant un écran, Soit $t_{affichage}$ le temps d'attente pour qu'une information de l'un de ces types soit affichée, On veut : $t_{attenteMax} > t_{affichage}$
Contexte temporel et spatial	Ces contrats doivent être valides à l'exécution et peuvent être vérifiés partiellement lors de composition conceptuelle, au déploiement, à l'exécution et lors de la génération des orchestrations.
Parties concernées	Utilisateur / Système SEDUITE
nature (Comportement, QOS, architecture)	Exigence
langage candidat	*OCL ou langage de QoS (QML-like)*
Hypothèses sur l'environnement	Temps resté devant l'écran (pour l'utilisateur) >= $t_{attenteMax}$
Règles de gestion : Cycle de vie	
réactions potentielles (contrôle),	Message d'erreur, filtrage des infos, regroupement des infos,…
Négociation	Envisageable en intégrant au système la notion de pertinence des informations
Parties concernées : ceux que l'on veut prévenir	Console de déploiement et console d'administration
Questions/remarques	A ce niveau l'exigence est très générale. Du fait de la complexité des situations et de la composition des services, elle sera déclinée ci-après en fonction des contextes d'usage différents.
Infos à destination des contrats	
Responsabilités	Exigence utilisateur + hypothèses sur l'environnement
Acteurs Humains confrontés à la définition de ces contrats	Ces contrats pourront être définis par les architectes de l'application, mais des paramètres devront rester apparents pour permettre la modulation du temps maximum d'attente soit par un acteur humain, soit par apprentissage du système lui-même.

Différentes déclinaisons de cette exigence sont envisagées :

- Les personnes présentes devant l'écran acceptent d'attendre un temps donné maximum : $t_{attenteMax}$. Le récepteur met au maximum un temps $t_{reception}$ à détecter un tag et à le renvoyer vers UserInfoProvider. Le temps moyen d'affichage des informations pour une personne est de $t_{affichage.}$ Le temps de détection des personnes doit être intégré dans le temps mis à afficher une information.

- On peut ajouter des intervalles de temps à chaque appel de service et calculer un intervalle de bout en bout par propagation de contraintes sur les intervalles. L'équipe Triskell a déjà mis en œuvre cela pour des composants avec des contraintes de *timeout* [cite]. Ce contrat s'exprime alors comme :

Infos à destination des contrats	
Spécifications contractuelles	Chaque service fournit un intervalle de temps précisant le minimum et le maximum de temps nécessaire à répondre à une requête [tmin, tmax]. Un service composite émet des exigences quant au temps maximal qu'il accorde à chaque service pour lui répondre.
Responsabilités	Chaque service est responsable des temps donnés
Acteurs Humains confrontés à la définition de ces contrats	Les experts techniques peuvent associer les temps aux services. Cependant, nous devons prévoir de pouvoir calculer ces temps dynamiquement par un système de sondes.
Remarques sur la composition de contrats :	Les services composites (orchestrations) calculent leur intervalle de temps en fonction des engagements des services composants.

- Lorsque les informations à afficher sont des images, si elles sont trop volumineuses, leur temps de téléchargement ralentit trop le système. La solution est alors de télécharger les images avant leur affichage… Le problème est un peu le même avec le transfert de vidéo sur un réseau à faible bande passante. Il peut alors être résolu par le streaming vidéo si la bande passante est faible mais constante. Il s'agit donc de proposer une adaptation en réaction à une violation de contrat.

Chaque service doit donc s'engager à fournir une taille maximum d'information et/ou le service composite exige une taille maximum des informations attendues de chacun des participants.

Infos à destination des contrats	(CT Taille des infos)
Spécifications contractuelles	Chaque service fournit la taille maximum et minimale des informations qu'il s'engage à transmettre. Cette information peut être donnée sous la forme d'une fonction dépendante du type d'information demandée. Un service composite émet des exigences quant à la taille des informations qu'il peut « supporter »
Responsabilités	Chaque service est responsable de la taille des informations.
Acteurs Humains confrontés à la définition de ces contrats	Les experts techniques peuvent associer les temps aux services. Cependant, nous devons prévoir de pouvoir calculer ces temps dynamiquement par un système de sondes.

- En situation de mobilité nous pouvons profiter des supports mobiles pour réduire les temps d'affichage et répartir l'information. Voici une nouvelle exigence qui raffine les délais d'obtention dans ce contexte mieux cerné.

Nom	(CTcontexte) Diffusion des informations en un temps minimum dans un contexte spatial donné
Description	Exemple : Les étudiants dans la cafétéria reçoivent sur les grands écrans les informations du BDE et s'ils ont sur eux des supports personnels ils peuvent recevoir des informations qui leur sont plus précisément destinées. L'environnement s'adapte en fonction du contexte pour que le nombre d'informations affichées sur les grands écrans répondent au contrat précédent.
Expression informelle	Soit t1..tn les types d'infos qui intéressent le public. Soit un ensemble d'utilisateurs u1… un qui ont déclaré les types d'information qui les intéressent et auxquels sont associés des supports personnels de diffusion d'informations ([s11,s12], ….[sn1,snn]). Soient G1..Gn les écrans de diffusion généraux. Etant donné un temps maximum d'attente, on désire, lorsque le nombre d'informations est trop important, basculer la diffusion des informations « moins grands public » vers les supports personnels.
Contexte temporel et spatial	A l'exécution, lorsqu'on constate que le nombre d'informations à afficher est trop important, seuls les supports mobiles identifiés dans la proximité des écrans de diffusion sont concernés.
Parties concernées	Identification des usagers, supports de diffusion et service de diffusion d'informations
Nature	QoS (nb d'informations)
langage candidat	*OCL 2*
Hypothèses sur l'environnement	Utilisateurs déclarés et identifiés
Règles de gestion : Cycle de vie	
réactions potentielles (contrôle),	Trace. (peu importe si un utilisateur n'a pas reçu ses informations mais on veut avoir une trace de cette erreur en vue de debug).
négociation	
Parties concernées :	Console de déploiement et console d'administration
Questions/remarques	Ce contrat est surtout l'expression d'une adaptation. En conséquence la réaction s'exprime pour l'instant uniquement sur le fait que le récepteur est sorti de la zone de réception.
Infos à destination des contrats	« Toute information à destination d'un utilisateur identifié lui est communiquée sur un des supports. »
Spécifications contractuelles	C1 : Les supports restent accessibles entre l'identification de l'usager et la diffusion soit un temps t_{min} C2 : le service de diffusion s'engage à diffuser toutes les infos sur un support dans un temps t_{max} après la détection de l'usager
Responsabilités	La responsabilité de l'accessibilité appartient au support.
Remarques sur la composition de contrats :	Ce contrat est une variante de CT. Il doit également être corrélé à celui qui évalue les capacités des supports à diffuser certaines informations.

b. Exigences sur l'administration des informations

Certaines des exigences relatives à l'administration sont induites par les exigences précédentes. Nous les explicitons ici d'un point de vue administration.

Taille maximum des informations

Si on connaît la taille des images, soit parce que celle-ci est obtenue lors de l'ajout de l'information par le service d'administration des informations, soit avant l'exécution (conception, déploiement) par analyse du contenu de la base de données, on peut placer un contrat sur l' « entité » qui gère/stocke les images : en l'occurrence il pourra s'agir du service web de saisie des informations ou de la base de données.
Dans ce cadre nous pouvons distinguer deux réactions à la violation de contrat :
- le téléchargement préalable sur des postes « clients » identifiés est une réaction à la violation du contrat lors de l'ajout de l'information,
- la réduction de la taille des images en réponse à une demande d'informations pour respecter une taille maximum exigée par le client. Cette réaction peut violer d'autres contrats sur la qualité nécessaire pour tel ou tel support de visualisation (problème de cohérence globale).

Nom	(IM) Conséquent de (CT) : Taille maximum d'une image
Description	
Expression informelle	Précondition : tailleImage < Max qui dépend du système.
Contexte temporel et spatial	A l'exécution surtout ou au chargement de l'info
Parties concernées	Ecrans de rendu, service de diffusion d'informations, service de gestion des informations
nature (Comportement, QOS, architecture)	Comportement ou QoS
langage candidat	*OCL ou langage de QoS (QML-like)*
Hypothèses sur l'environnement	Le Max est donné en fonction des connaissances que l'on a du réseau, mais il peut également varier en fonction des supports d'affichage. Le récepteur a la capacité à mémoriser l'image.
Règles de gestion : Cycle de vie	
réactions potentielles (contrôle),	Gestion de cache, réduction de la taille de l'image,
Négociation	A l'exécution, elle peut permettre d'établir un compromis entre la taille de l'image, sa qualité et la taille maximum attendue.
Parties concernées : ceux que l'on veut prévenir	Console de déploiement et console d'administration
Questions/remarques	Cette exigence se décline en plusieurs contrats

Infos à destination des contrats	
Spécifications contractuelles	Obligation du Serveur : getInfo() : Info : taille(Info) < Maximum évalué par le client et le réseau Obligation du client : Il a assez de mémoire pour réceptionner l'information Obligation du réseau : Il peut transférer l'information en un temps < à un temps donné
Responsabilités	Cf. ci-dessus
Remarques sur la composition de contrats :	Ce contrat est partiellement du à CT, et les réactions à la violation de ces contrats peuvent mettre en jeu d'autres contrats en particulier sur la QOS.

- On peut étendre le récepteur ou le service *UserInfoProvider* pour que l'un ou l'autre gère la diffusion des informations en fonction de l'ordre d'arrivée face à l'écran. A partir de quel nombre de personnes présentes le système ne pourra plus répondre à temps ? Peut-on choisir la distance de détection des tags en fonction de ces informations ?

Cela pourrait se faire si on a des formules pour lier des propriétés extra-fonctionnelles entre elles (nombre d'utilisateurs, débit de diffusion, temps de réponse).

c. Contrôle de l'évolution de l'application : administration et exigences

Les différentes exigences formulées ici correspondent aujourd'hui à des informations au mieux éparses dans les documents fonctionnels de l'application. Elles ne correspondent pas à des entités physiques de l'application. Cette étude devrait nous permettre d'identifier les cas où il est possible de poser les contrats pour contrôler l'évolution de la plateforme.

Cohérence entre type d'information et existence des plugins d'affichage

Lorsqu'on ajoute un nouveau type d'information, par exemple des images ou des convocations, il faut avoir créé le ou les plugins d'affichage correspondant. Cette exigence se propage jusqu'aux supports de diffusion auxquels devront être associés ces plugins d'affichage lorsqu'ils requièrent des informations d'un nouveau type. Il s'agit donc de vérifier lors du déploiement de l'application ou à l'exécution si nous prenons en charge la découverte de nouveaux types d'informations la présence des différents composants.

Nom	(PA) Existence des plugins d'affichages
Description	
Expression informelle	- $\forall t \in$ types d'information, $\exists p$ un plugin d'affichage / $t \in$ ensemble des types d'infos supportés par p. - Tout support de diffusion S qui demande des informations de type t, doit connaître un plugin d'affichage qui supporte t.

Contexte temporel et spatial	- A la création d'un nouveau type d'info. Cet aspect n'est pas aujourd'hui capturé par un service. - A la demande d'information.
Parties concernées	- Service d'administration des informations - Supports d'affichage, le système dans son ensemble
nature (Comportement, QOS, architecture)	Architecture
langage candidat	Langage dédié aux contrats architecturaux ou OCL + la capacité d'introspecter l'architecture
Hypothèses sur l'environnement	
Règles de gestion : Cycle de vie	
réactions potentielles (contrôle),	Warning par l'outil d'analyse de la plateforme.
Négociation	Il existe un plugin XSL qui supporte n'importe quelle info. Mais la qualité est alors bien moins bonne.
Parties concernées : ceux que l'on veut prévenir	Le développeur
Questions/remarques	Le même type de contrat est nécessaire pour - spécifier qu'un service de contenu doit se déclarer comme diffusant cette information - spécifier qu'une interface de saisie de ce type d'info est conseillé (accès direct à la BD par exemple, c'est donc un simple warning dans ce cas).
Infos à destination des contrats	
Spécifications contractuelles	Ecrire en OCL sur l'architecture
Responsabilités	Du créateur de type d'informations, car c'est un pré-requis du système.
Remarques sur la composition de contrats :	Le même type de contrat est nécessaire pour - spécifier qu'un service de contenu doit se déclarer diffuser cette information - spécifier qu'une interface de saisie de ce type d'info est conseillé (accès direct à la BD par exemple, c'est donc un simple warning dans ce cas). Cet ensemble de contrats répond à une exigence de plus haut niveau qui est : A tout type d'information sont associés des plug-ins d'affichage, des environnements de saisie et des services de diffusion.

Validation des feuilles de styles XSL vis-à-vis des types d'informations déclarées traitées :

Les infos manipulées par les services et les extensions xmlContent entraînent des structures complexes dont la cohérence peut être spécifiée par des contraintes d'intégrité (Etude du côté de la validation de schéma xsl par Schematron).

Nom	*(VXSL) Validation des feuilles de styles XSL*
Description	
Expression informelle	Tout type d'information (DTD ou XSD XML) déclaré traité par une feuille de style XSL doit effectivement l'être (validité de la feuille)
Contexte temporel et spatial	- A l'ajout d'une feuille de style
Parties concernées	- Service d'administration des feuilles de styles
nature (Comportement, QOS, architecture)	
langage candidat	Appel à un outil extérieur de validation
Règles de gestion : Cycle de vie	
réactions potentielles (contrôle), Négociation	Refus de la nouvelle feuille de style
Parties concernées : ceux que l'on veut prévenir	Le développeur
Infos à destination des contrats	
Spécifications contractuelles	
Responsabilités	Du créateur de la feuille de style Du système de gestion des feuilles de style

Nombre de requêtes supportées par un service

Si on peut savoir le nombre de requêtes que supporte un service/machine, on peut déterminer la validité de certaines compositions de services qui accroissent « trop » le nombre de requêtes à des sous-services.
Ce type d'exigence correspond à la fois à des contrats d'architecture et des contrats sur le comportement.

Nom	*(SMR) Seuil maximum de requêtes*
Description	
Expression informelle	Le nombre de requêtes supportées par un service < MaxRequeteParMinute et le système s'engage à respecter cette contrainte
Contexte temporel et spatial	Au déploiement lorsque le service déclare ce seuil et que les clients précisent le nombre de requêtes envisagé
Parties concernées	Architecte du système, services et orchestrations
nature (Comportement, QOS, architecture)	Comportement et architecture
langage candidat	- si on obtient le nombre de requêtes par une moyenne (valeur mesurée puis stockée et

	réutilisée à la configuration d'une composition de services, donc avant nouvelle exécution), l'utilisation des timeout semble adaptée. - Si on veut surveiller le système en cours d'exécution d'autres techniques doivent être mises en place [Thèse d'Hervé Chang en préparation]).
Hypothèses sur l'environnement	Disponibilité des informations : maximum de requêtes supportées et nombre de requêtes attendues ou moyenne réelle.
Règles de gestion : Cycle de vie	
réactions potentielles (contrôle),	Stopper des invocations de services
Négociation	Non
Parties concernées : ceux que l'on veut prévenir	L'administrateur de la plateforme et le client désobligeant si on sait l'identifier.
Infos à destination des contrats	
Spécifications contractuelles	- Max Requêtes supportées (nbreRequetes/s) - Max Requêtes demandées (nbreRequetes/s)
Responsabilités	Services et Client
Acteurs Humains confrontés à la définition de ces contrats	Les experts techniques associent le nombre maximum de requêtes aux services.
Remarques sur la composition de contrats :	Le client pourra correspondre à un service composite ou orchestration, et ses propres contraintes devront être propagées.

Gestion de la qualité des informations perçues par l'utilisateur

- *Informations redondantes :*

Les informations de localisation sont potentiellement multiples et redondantes, par exemple l'emploi du temps global et l'agenda personnel de quelqu'un, sa localisation par Bluetooth ou autre technique équivalente et par l'emploi du temps. Si nous sommes capables d'expliciter la qualité d'une information et sa pertinence alors nous devrions pouvoir établir des contrats basés sur la redondance et ou l'unicité d'une information.

Nom	*(Unicité) En réponse à une requête une seule information d'un type donné*
Description	Exemple : A une demande de localisation d'une personne nous obtenons une information en provenance de l'edt et du service de localisation. Le respect de ce contrat forcera la diffusion de l'information la plus pertinente
Expression informelle	∃ ! information de type « localisation » ou « edt »
Contexte temporel et spatial	A l'exécution, ce contrat se traduit par une modification des orchestrations.
Parties concernées	Architecte du système, Système
nature (Comportement, QOS,	QoS

architecture)	
langage candidat	OCL dans un premier temps puis des ontologies ?
Hypothèses sur l'environnement	Capacité de notre système à identifier une redondance d'informations
Règles de gestion : Cycle de vie	
réactions potentielles (contrôle),	(voir négociation)
Négociation	Choix de l'information et/ou construction en fonction de la pertinence
Parties concernées : ceux que l'on veut prévenir	
Questions/remarques	

- *Mise en commun d'informations :*

Si plusieurs personnes présentes demandent par leur profil l'affichage d'informations communes ou qui peuvent être rassemblées sur un même écran (contraints par le nombre de personnes) peut-on autoriser une adaptation dynamique ? Est-ce une solution à une rupture de contrat ?

Nom	*(Partage) En réponse à plusieurs requêtes « simultanées » une seule information d'un type donné*
Description	Exemple : plusieurs personnes demandent un emploi du temps identifié par un même diplôme, un seul edt est affiché.
Expression informelle	Soient r1…rn des requêtes, Soient i1…in les informations obtenues, toutes les informations redondantes sont éliminées.
Contexte temporel et spatial	A l'exécution
Parties concernées	
nature (Comportement, QOS, architecture)	QOS et architecture
langage candidat	OCL
Hypothèses sur l'environnement	Capacité de notre système à identifier une redondance d'informations, capacité de notre système à prendre en compte une suite de requêtes en compte
Règles de gestion : Cycle de vie	
réactions potentielles (contrôle),	Regrouper les informations ce qui impose de modifier l'architecture.
Négociation	
Parties concernées : ceux que l'on veut prévenir	Administrateur
Questions/remarques	Cette exigence peut être contradictoire avec le temps de diffusion des informations, car elle suppose un regroupement des requêtes (cf. ci-après). Son contour devra être précisé par l'usage.

Remarques :
Les contrats exprimés ici semblent tous liés à la composition de services ou de requêtes. On pourrait envisager de chorégraphier les différentes actions effectuées, pour former des chorégraphies d'adaptation ou de réparation.

Contraintes subjectives :
Suite à l'article de Christophe Jacquet : http://wwwsi.supelec.fr/~jacquet/thesis/ il semble également important de veiller à éviter le morcellement de l'information ou inversement la surcharge de l'écran. Ces données sont pour l'instant très subjectives, mais elles pourraient correspondre à des contrats en termes de nombre d'informations, de distance entre deux informations ou de stabilité d'une information.

d. Gestion des droits

La gestion des droits intervient à la fois au niveau de la diffusion des informations, la gestion des informations et l'administration.

Accès aux informations personnelles et/ou de groupe

Les droits dans cette application sont motivés par la protection des personnes : qui a le droit de savoir où se trouve une personne ? L'ajout d'informations telles que des convocations d'étudiants, les notes, etc. exige de prendre en compte des notions de droits d'accès aux informations. En nous basant sur l'infrastructure de l'école (Directeur, Enseignants, Administration, Etudiants, Chercheurs), nous envisageons d'intégrer à l'application, des droits individuels et de groupes (par la fonction et par les amitiés).

Un contrat simple pourrait être : l'écran n'affiche plus de données « personnelles » si plus d'un utilisateur est détecté. La définition plus précise de ces contrats reste à faire en nous appuyant sur des modèles métiers dédiés. De plus la définition d'orchestrations induit une notion de propagation des droits qui reste également à étudier en nous basant sur les travaux dédiés.

Gestion des informations

Nous avons également à gérer les droits en matière de gestion des informations. Ces droits sont actuellement distribués aux différents services qui gèrent individuellement leur droit d'accès, éventuellement en adressant un service d'identification commun. La gestion centralisée de différents types d'informations nécessite une propagation des droits entre plusieurs systèmes. Ce point reste à étudier. Le respect d'une démarche SOA nous a pour l'instant conduit à envisager séparément ces droits.

Administration des informations

Le réagencement des web services par modification dynamique des orchestrations via le service d'orchestration ne doit être permis qu'aux personnes autorisées ou dans un cadre reconnu d'adaptation dynamique. Ce point reste également à approfondir.

5. Conclusion

Nous exprimons d'abord dans ce document des exigences sur le système et ses utilisateurs (utilisateurs finaux, fournisseurs de plugin de visualisation, etc.). Nous donnons aussi, de manière informelle, les possibilités de raffinement de ses exigences sous forme de contrats, et donc de responsabilités sur des éléments de l'architecture du système. Ce travail préparatoire met déjà en évidence des besoins d'indépendance vis-à-vis des plateformes et de compétences multi-domaines, à la fois en termes de gestion d'IHM, de documents, de sécurité, de distribution…
Nous soulignons à présent les points qui se révèlent déjà comme nécessitant d'une ingénierie dirigée par les modèles et les besoins de validation entre l'expression, pour l'instant informelle, des exigences et les contrats mis en place pour leur validation.

Des abstractions aux mises en œuvre

Alors que l'application étudiée est en cours d'exploitation et que sa complexité n'est pas encore à l'échelle des grands systèmes d'informations, les exigences et les contrats qui en découlent, se révèlent déjà nombreux et relativement complexes. Il apparaît nécessaire de les exprimer au niveau de l'ensemble du système pour pouvoir les raffiner en fonction des différentes entités mises en jeux effectivement dans l'application. Ses contrats peuvent eux-mêmes être raffinés vers des contrats de plus bas niveau, dans une approche MDA « classique », avec un passage de PIM à PSM. Quel que soit le niveau de raffinement considéré, il est nécessaire d'effectuer des abstractions sur la forme des spécifications contractuelles (langage d'expression), l'architecture (services et/ou orchestrations), etc. Ces abstractions ont directement besoin d'être réalisées par des techniques d'ingénierie des modèles. L'étape suivante va consister à identifier les mécanismes nécessaires à leur mise en œuvre.

Compositions et abstractions

L'application SEDUITE se présente comme une composition de systèmes d'information. Elle est très riche en exigences non-fonctionnelles, à la fois sur l'ensemble de l'application et sur chacune de ses parties. Les contrats globaux peuvent alors être vus comme le résultat de la composition des contrats sur les parties. Inversement, de certaines exigences données sur l'ensemble résulte un ensemble d'exigences sur les parties. Ainsi de nombreux sous-contrats sont mis en jeu lorsque l'on traite par exemple de la seule notion de temps de réponse acceptable pour la réception d'une information.
La complexité du terrain et la diversité des usagers forcent à faire évoluer l'ensemble des contrats en fonction des besoins, sans pour autant relâcher les contraintes globales sur le système.
Les mécanismes de composition des services et des contrats sont au cœur du projet FAROS et trouvent dans SEDUITE une application particulièrement riche pour de telles expérimentations. Il s'agit dans le cadre de cette application de pouvoir faire évoluer les contraintes à prendre en compte, en rajouter ou en relâcher en fonction du contexte d'utilisation et des orchestrations précédemment en place.

Des métiers

Alors que nous avons choisi une démarche informelle, les « mots » nous ont manqué pour exprimer certaines exigences dont les droits d'accès, la QoS ou l'ergonomie des IHMs. Des

langages métiers dédiés permettent d'adresser ces points. Il semble nécessaire de les prendre en compte dans notre définition des contrats. Ainsi les contrats relatifs au temps d'exécution semblent constituer un domaine à part entière de même que ceux relatifs à la qualité des affichages.

Indépendance des plateformes

La mise en œuvre des contrats semble a priori dépendre autant de la nature du contrat, du moment de sa validation, de l'architecture de l'application que de la plateforme ciblée. Ainsi il paraît essentiel lors de la définition des contrats de rester libre des plateformes cibles, ce qui devrait nous permettre de positionner le même contrat à la fois sur des services web par exemple que sur des assemblages de composants Wcomp. Un contrat tel que celui sur la durée du temps de réception des informations met en effet en jeu ces différentes entités simultanément.

Contrats et quantification

L'expression et la validation des contrats reposent sur la quantification de données et l'explicitation de seuils. Ces données sont dans un premier temps difficilement quantifiables et s'expriment actuellement de façon qualitative. Le passage à des contrats exige de pouvoir les quantifier en utilisant différents facteurs qui restent à identifier plus précisément. Afin d'établir les seuils, nous envisageons la mise en place de systèmes de « traces » permettant de mesurer les durées d'affichage et de chargement des informations diffusées, les demandes d'avance rapide ou inversement les temps de pause, …. Nous envisageons également une étude centrée utilisateurs basée sur l'observation et l'évaluation ergonomique du système actuel permettant, en fonction des réactions des usagers d'obtenir des données statistiques permettant d'établir des seuils adaptés aux usages et d'ajuster alors les stratégies de diffusion.

Processus de raffinement et « complétude » des contrats vis-à-vis des exigences

Le raffinement et la complétude des contrats au fur et à mesure des raffinements constituent une grande partie du postulat de base du projet RNTL FAROS. Les modèles et outils conçus et produits dans les autres lots du projet devront permettre d'établir précisément les responsabilités des différentes entités logicielles tout au long du cycle de vie (conception, implémentation, déploiement, reconfiguration) pour dégager les bénéfices attendus : pas de double vérification des contraintes exprimées, approche modulaire, facilité d'intégration et de remplacement, etc. Par ailleurs, l'exigence a tendance à être exprimée à un haut niveau, puis à être analysée et traduites en des contrats de plus bas niveau.