



## OR2 - Accesso e composizione user-oriented dei servizi

### RI2.1 - STUDIO E ANALISI DI METODI E STRUMENTI PER LA COMPOSIZIONE DI SERVIZI IN AMBITO ENTERPRISE

Obiettivo dell'attività è stato da una parte, illustrare le tecniche presenti in letteratura per il Mashup, ed in particolare quelle rivolte al contesto enterprise evidenziandone linee guida e architetture di riferimento; dall'altra studiare e l'analizzare i framework per l'Enterprise Portal intesi come ambienti per l'integrazione d'informazioni, persone e processi aziendali.

Partendo dallo studio degli Enterprise Mashup, nel linguaggio Web, un Mashup è una pagina Web o un'applicazione che attraverso un utilizzo combinato di dati, componenti grafici o funzionalità, provenienti da fonti eterogenee, è in grado di comporre nuovi servizi.

L'aspetto innovativo introdotto dalle tecniche di Mashup nel panorama della composizione è, diversamente dagli approcci d'integrazione a livello di dati o di logica applicativa, la capacità d'integrazione anche di elementi di presentazione, possibile grazie esclusivamente all'uso di tecnologie web standardizzate. A tal proposito, durante l'attività è stata valutata la portabilità di tale metafora all'interno di contenuti Enterprise. Si è arrivati, dunque, agli Enterprise Mashup: applicazioni che combinano le risorse aziendali, le applicazioni e i dati esterni, permettendo di concentrare tutti i dati di interesse in una singola presentazione, agevolando le azioni di collaborazione tra le imprese e gli sviluppatori e di interazione degli utenti con le applicazioni pre-esistenti. L'introduzione delle tecniche di Mashup all'interno di architetture Enterprise deve tener conto delle preesistenti tecnologie, metodologie, paradigmi architetture che le compongono ed integrarsi, sicuramente, all'interno di un modello architetture basato sul Service-oriented architecture (SOA) ed estenderlo attraverso i modelli e tecnologie proprie del Web 2.0.

L'adozione di tali principi fondamentali, è confluito nella definizione di un **modello architetture** di riferimento che sottende la costruzione degli Enterprise Mashup. Esso introduce meccanismi sia per la gestione che la manipolazione delle entità Mashup, definendo un framework per la creazione di un ambiente di Enterprise Mashup. I livelli architetture, in esso definiti, evidenziano le diverse necessità che un sistema di Enterprise Mashup deve supportare. La formalizzazione di tali necessità parte dall'individuazione delle sorgenti enterprise da coinvolgere nel processo combinativo e dalla definizione di strategie e meccanismi atti alla loro trasformazione, necessaria per un corretto svolgimento dei processi di Mashup.

Procedendo con l'analisi degli Enterprise Mashup, la natura diversa degli elementi che possono essere "mashed" (mescolati) comporta una definizione di tre categorie di alto



livello, ognuna delle quali presenta linguaggi, metodologie e tecniche di programmazione diverse. E' possibile dunque individuare tre distinti domini: **presentation-oriented** (orientato alla presentazione), **data-oriented** (orientato ai dati) e **process-oriented** (orientato ai processi).

L'analisi del Mashup si conclude con la presentazione dei **Pattern** per il Mashup. I modelli sono organizzati in quattro categorie principali: **Harvest, Enhance, Assembly, Manage** e dividono il loro focus tra creazione, riduzione e fruizione di contenuti e funzionalità applicative. Nonostante l'eterogeneità dei gruppi di pattern, tra alto e basso livello di astrazione, nulla preclude ai diversi pattern di lavorare insieme per trovare una soluzione a un problema noto. I modelli inseriti nel gruppo Harvest sono una classe generale di soluzioni che si prestano a ottenere dati provenienti da fonti al di fuori della portata degli strumenti tradizionali. Seguono i Pattern della categoria di Enhance che mostrano come i sistemi possono essere ampliati e migliorati grazie al Mashup.

La sezione Assembly, invece, mostra come le diverse soluzioni possono essere costruite dalla combinazione di dati ed elementi di presentazione da fonti multiple.

L'attività effettuata ha interessato anche lo studio e l'analisi del modello degli Enterprise Portal. È possibile dividere tale attività di studio e analisi in due fasi; nella prima, è stata prodotta una descrizione di questi strumenti, illustrando le caratteristiche principali e fornendo una panoramica del mercato, nella seconda, la più importante, sono stati analizzati i prodotti più indicativi.

Nella prima fase sono stati messi in luce gli aspetti salienti che caratterizzano un Enterprise Portal. Questo strumento è in continua evoluzione e si sta orientando verso un approccio user-centric. Un Enterprise Portal è un'infrastruttura tecnologica che abilita gli interlocutori di un'azienda (dipendenti, clienti, partner) all'utilizzo di una serie di servizi e informazioni erogati dall'azienda stessa.

Gli Enterprise Portal sono costruiti sopra i Portal Framework, conosciuti come prodotti di Horizontal Portal, che sono forniti da vendor (commerciali o open source) o da provider di servizi cloud. Il mercato a tal riguardo è ampio: da un lato ci sono i grandi fornitori (come Microsoft, IBM, Oracle, Cisco), orientati verso le grandi suite, le User Experience Platform (UXP) che sono una raccolta integrata di tecnologie e metodologie che offrono la possibilità di progettare e fornire una gamma completa di funzionalità tutte inserite in un'unica piattaforma, dall'altro ci sono i Lean Portal (come Liferay, Backbase), che sono invece prodotti più "leggeri" che offrono unicamente un punto di accesso personalizzato alle informazioni, ai processi e alle persone e che sono "spinti" da soluzioni open source o hanno una parte core open source e dal cloud.

A valle dello studio effettuato, la seconda fase dell'attività ha determinato come output finale una razionalizzazione delle caratteristiche che presentano i più importanti framework di Horizontal Portal presenti nel mercato, allo scopo di fornire uno strumento per indirizzare la scelta verso un tipo di soluzione piuttosto che verso un altro nella fase del progetto riguardante la realizzazione del modello architeturale, che si prefigge di



realizzare MODERN, e nel caso in cui si prenda la decisione di inserire nel modello stesso un framework per Enterprise Portal.

Per ogni framework è stata fornita una presentazione sintetica, una vista ad alto livello dell'architettura del prodotto (dove possibile), un elenco minimale delle macro categorie di feature e di strumenti offerti e una serie di punti di forza e limiti. Nei casi in cui si è ritenuto opportuno, è stata proposta nell'Appendice un'analisi più approfondita delle feature e degli strumenti. L'analisi, inoltre, è stata condotta seguendo un approccio orientato alla vision di MODERN, ovvero sono state analizzate le feature dei framework ritenute in linea con il progetto MODERN ed il confronto finale tra questi è stato realizzato sulla base di questo set di feature individuato.

#### **RI2.2 - DEFINIZIONE DI UN MODELLO PER LA COMPOSIZIONE DI SERVIZI IN AMBITO ENTERPRISE**

Obiettivo dell'attività è stato definire una proposta di modello architetturale a supporto della composizione user-oriented di servizi, capace di introdurre soluzioni e componenti in grado di alimentare la collective knowledge aziendale, aggregando conoscenza interna (acquisita a partire sia dai sistemi legacy aziendali sia dalle attività del worker – conoscenza tacita) e conoscenza esterna (sistemi RSS, procedure, ecc.).

Il modello architetturale proposto vuole introdurre in esso il concetto di Personal Environment come ambiente personale integrato con il Sistema Aziendale in grado di abilitare l'utente alla composizione di informazioni e servizi in base alle proprie necessità ed esigenze.

In quest'ottica, il modello vuole estendere il concetto di composizione intesa come semplice capacità di definire servizi come aggregazione opportunamente orchestrata di provider di dati e servizi di base. In particolare si vuole supportare la composizione user-oriented dei servizi, intesa come caratterizzazione del processo di aggregazione di dati e servizi che, composti a "tempo di esecuzione" in base alle specifiche esigenze professionali degli utenti, preveda un supporto adattivo che tenga in conto il profilo operativo dell'utente, favorendo e semplificando così l'utilizzo del sistema per una più ampia classe di persone (non solo sviluppatori esperti).

L'attività svolta, come previsto in fase di proposizione progettuale, è stata divisa in due fasi. La prima fase prevedeva una prima definizione del modello di composizione dei servizi, output principale dell'attività, mentre nella seconda fase, definita di ripresa, sono state integrate le risultanze delle attività di ricerca di cui agli OR 1, 3 e 4, oltre che con le risultanze delle altre linee di ricerca dell'OR2.

Le principali problematiche affrontate sono state relative alla definizione di un modello adattivo dinamico che incontrasse e soddisfacesse le necessità dei futuri utenti del sistema, attraverso l'utilizzo di strumenti del web 2.0 per fornire un supporto alla composizione dinamica e *user-centered* dei servizi all'interno di un contesto aziendale esteso (Enterprise 2.0).



La necessità di integrare servizi e sistemi legacy interni all'azienda con servizi e strumenti disponibili all'esterno della stessa, è stata un'ulteriore sfida per la realizzazione del modello finale proposto. Il risultato di tale lavoro è stato la definizione dei due modelli seguenti:

- modello logico/funzionale – in cui viene descritta una vista ad alto livello del sistema esplicitando le componenti concettuali fondamentali. Da questo modello viene fuori una struttura “a stella” in cui sono mostrate le interazioni e le interdipendenze dei principali “domini applicativi d’interesse” (e delle componenti rappresentanti) per il soddisfacimento dei requisiti funzionali del sistema;
- modello architetturale – in cui viene presentata la descrizione delle componenti ad un livello più basso rispetto a quello presentato nel modello precedente. Il risultato è la definizione di un'architettura a 6 componenti e gravitante intorno a 2 componenti principali, l'interfaccia grafica e il motore di elaborazione centrale.

A livello di benefit, questa attività di ricerca ha consentito di gettare solide basi per il prosieguo del progetto fornendo informazioni e contenuti rilevanti per attività successive dello stesso OR e degli OR seguenti.

La seconda fase dell'attività ha visto effettuata una revisione, soprattutto in ambito tecnologico, alla luce dei feedback emersi dai risultati degli altri OR di MODERN, in particolare quelli provenienti da OR3-OR4-OR5.

In particolare, nel modello di composizione dei servizi è stato affrontato a livello architetturale il supporto alla costruzione della dashboard personale nel personal environment. La necessità di aggiornare ed estendere tale modello deriva dalla definizione di un modello semantico per la cooperazione applicativa (D.3.2) che favorisce l'integrazione e la fruizione di sistemi eterogenei al fine di migliorare i metodi di lavoro e l'accessibilità alle informazioni.

La cooperazione applicativa definisce infatti un modello semantico basato su ontologie per l'integrazione e la collaborazione tra app interne, esterne e legacy di un sistema.

Il raffinamento di questo modello in relazione al modello semantico di collaborazione applicativa agisce a livello architetturale inserendo l'ambiente di cooperazione applicativa come gestore ed orchestratore dei servizi accessibili dal Personal Environment.

L'estensione del modello di composizione dei servizi con la cooperazione applicativa abilita in definitiva un nuovo modello di fruizione delle app e dei servizi del Personal Environment, dove le app e i servizi del sistema cooperano tra loro, agendo anche su dati condivisi da più business process. Inoltre è stata migliorata anche la fase di ricerca ed istanziazione delle app nella Dashboard personale grazie ad una migliore aderenza dei risultati restituiti dal motore di ricerca alle richieste effettuate, sfruttando il potenziale semantico del modello ontologico adottato.



In seguito si è focalizzata l'attenzione sullo strato di "Presentazione" dell'architettura estesa, caratterizzando il concetto di Personal Environment come ambiente personale integrato con il Sistema Aziendale in grado di abilitare l'utente alla composizione di informazioni e servizi in base alle proprie necessità ed esigenze.

A tal scopo è stata condotta un'analisi dei processi, del portafoglio applicativo e dei sistemi, seguendo un approccio metodologico teso a evidenziare diversi livelli di astrazione e prospettive di analisi. Quindi sono stati definiti uno o più scenari di riferimento, astratti dal contesto aziendale di riferimento (nel caso in esame Poste Italiane), che tengono conto dei risultati delle attività di ricerca previste in OR3.

Infine è stato evidenziato come la scelta di Outlook come PE, visto come diretta conseguenza delle analisi svolte, evidenzia tutte le caratteristiche esposte in precedenza.

### **RI2.3 - STUDIO E DEFINIZIONE DI MODELLI E TECNOLOGIE DEL WEB 2.0 A SUPPORTO DELL'ACCESSO AI SERVIZI**

L'attività condotta, in una prima fase, ha visto l'analisi di tecnologie e modelli volta alla razionalizzazione delle soluzioni e metodologie per la definizione di uno strato di accesso ai servizi che possa supportare l'applicazione delle tecniche di composizione di servizi in accordo ai nuovi modelli d'interazione in grado di favorire ed arricchire la user experience degli utenti di un sistema sviluppato in accordo ai principi dell'EAEA.

L'attività si è focalizzata su due ambiti di applicazione: la prima rivolta all'identificazione e analisi critica del panorama tecnologico esistente, la seconda alla razionalizzazione e sistematizzazione di queste in un modello integrato.

Si è osservato come l'emergere di tecnologie e standard a supporto dell'orientamento ai servizi abbia consentito di progredire significativamente verso una maggiore automazione e integrazione dei sistemi enterprise, fino alla definizione di architetture orientate ai servizi (SOA).

È stata effettuata innanzitutto un'analisi dei requisiti dei servizi infrastrutturali, con l'identificazione delle caratteristiche di base per l'implementazione dei servizi di accesso partendo dallo studio di **Thomas Erl** che individua otto principi di progettazione per le architetture SOA, e proponendone una rielaborazione focalizzando l'attenzione sui principi legati alla sicurezza. Secondo questa chiave di lettura, le linee guida individuate da Erl si sintetizzano in:

- **condivisione di un contratto formale:** ciascun servizio condivide un contratto formale, che descrive lo scambio di informazioni con il servizio;
- **basso accoppiamento:** i servizi devono essere progettati in modo di ridurre al massimo la dipendenza da un altro servizio;
- **composizione e riusabilità:** i servizi possono comporre altri servizi.



- **discoverability:** meccanismo mediante il quale è possibile ricercare, “scoprire” e recuperare un servizio;
- **interoperabilità:** i servizi devono poter operare su piattaforme eterogenee.
- **livelli di sicurezza:** i principi di sicurezza per una SOA non differiscono molto da quelli di un sistema tradizionale (**authentication, authorization, confidentiality, integrity e non-repudiation**)

Una successiva attività di studio ha riguardato la composizione dinamica dei servizi d'interfaccia. Sono, quindi, stati analizzati i tool di mashup in ambito enterprise (Jackbe Presto, Convertigo, WSO2 Mashup Server, IBM Mashup center) nell'ottica dei cinque principi fondamentali (composizione di applicazioni, integrazione lightweight, WOA, repository e community), individuati nel RI 2.1, alla base di ogni tool di Enterprise Mashup.

Proseguendo con l'attività, in merito alle problematiche legate alla multicanalità, sono state analizzate le tecnologie per la creazione di una user interface moderna, compliant con le esigenze del Web 2.0 nel contesto del livello di front-end multicanale dell'architettura EAEA. In particolare sono state analizzate le principali tecnologie per l'implementazione di RIA (Light RIA, Heavy RIA, HTML5) con successivi confronti in base al tipo di distribuzione dell'applicazione, risorse hardware utilizzate, sicurezza, ecc. Un'analisi comparativa delle tecnologie RIA ha permesso poi di evidenziare come la scelta di una tecnologia rispetto ad un'altra sia fortemente dipendente dai requisiti operativi (requisiti di sistema, multiplatforma, cross-browser, ecc.).

Sono stati anche analizzati i modelli e le tecnologie Web 2.0 a supporto dell'accesso ai servizi, considerando sia i modelli di accesso sincroni/asincroni che i modelli di rappresentazione dei dati.

Particolare attenzione è stata dedicata al **REpresentational State Transfer (REST)**, un paradigma architetturale per la realizzazione di applicazioni Web. Al centro del paradigma REST vi è il concetto di risorsa, alla quale possibile accedere tramite un identificatore globale oppure attraverso componenti distribuite sulla rete (client e server). Queste ultime comunicano attraverso interfacce standard, scambiandosi rappresentazioni di queste risorse. È possibile definire un web service RESTful (anche chiamato RESTful web API) utilizzando il protocollo HTTP e rispettando i principi di REST che ne determinano l'implementazione: identificatore di risorse globale tramite URI, utilizzo preciso dei metodi http, risposta tramite gli standard XML, JSON, API hypertext-driven.

Per quanto concerne invece i formati di rappresentazione lo studio si è soffermato su **JSON** (JavaScript Object Notation), un formato, semplice e leggero, per lo scambio di dati che si basa su un sottoinsieme del Linguaggio di Programmazione JavaScript, ma completamente indipendente dal linguaggio di programmazione.



A valle degli studi è stata predisposta una matrice di copertura che ha permesso di illustrare quali requisiti dei servizi infrastrutturali (accoppiamento, composizione, ecc.) sono soddisfatti dai modelli di accesso ai servizi analizzati.

L'attività si è conclusa con lo studio di modelli e tecniche per favorire la User Experience. Il risultato ottenuto evidenzia che non esiste ancora una definizione condivisa e comunemente accettata della tematica e di ciò che questa coinvolge. La descrizione più consolidata definisce la UX come *"una conseguenza dello stato interno dell'utente (predisposizione, bisogni, motivazioni, umore, ecc.) delle caratteristiche del sistema progettato (complessità, scopo, usabilità, funzionalità, ecc.) e del contesto (ambiente) all'interno del quale le interazioni si realizzano."*

Le analisi condotte sulla UXD hanno permesso di individuare cinque dimensioni chiave interdipendenti: *utilità, usabilità, desiderabilità, sostenibilità e social*. Per sfruttare i vantaggi della UXD, è necessario valutare il comportamento di un sistema per ciascuna dimensione. Sorge, pertanto, l'esigenza di definire delle metriche, ovvero delle misure sia oggettive che soggettive, che possono fornire indicazioni sulla qualità della UX di un determinato sistema.

Una selezione delle metriche individuate ha permesso, quindi, di scegliere quelle più significative nei confronti della UX. Considerando le metriche selezionate, è possibile svolgere un'analisi dei requisiti del modello d'interazione. Sono stati definiti i principi di progettazione da prendere in considerazione per l'ottenimento di una buona UX.

Infine, volendo identificare un modello d'interazione efficace, in grado di rispettare i requisiti descritti, l'attività di analisi ha permesso di asserire che un modello di accesso ai servizi basato su un'interfaccia a componenti e sul concetto di application store è, a nostra considerazione, quello che permette di risolvere la maggior parte dei problemi connessi alla UX. Mediante l'interfaccia a componenti, infatti, è possibile conseguire l'adattamento dell'ambiente in base alle competenze e preferenze dell'utente. In questo modo, è possibile arricchire l'ambiente di lavoro con strumenti sociali (chat, wiki, blog, ecc.) e tanti altri applicativi a seconda dello specifico processo aziendale. Nel modello razionalizzato, quindi, per ciascun processo aziendale l'utente potrà creare più dashboard, ciascuna con una particolare distribuzione di widget. Mediante l'utilizzo di tale modello sarà possibile effettuare la condivisione tra gli utenti di conoscenza e strumenti applicativi, sfruttando le funzionalità di social networking, ranking e feedback che favoriscono l'emersione del capitale intellettuale in accordo all'obiettivo principale di MODERN. La seconda fase dell'attività ha posto il focus sui vantaggi nell'utilizzo di JSON-LD come formato di interscambio dei dati tra gli strati applicativi di MODERN con quelli di memorizzazione della conoscenza condivisa tramite database semanticizzati.

Dallo studio dei documenti D3.2 D3.3 si è evinto che l'utilizzo di database semanticizzati incide necessariamente sul formato di interscambio per la cooperazione applicativa, introducendo di conseguenza il problema dell'identificazione di un ulteriore formato per il trasferimento di informazioni dallo strato semantico a quello applicativo e viceversa.



L'utilizzo di JSON come linguaggio per lo scambio di dati tra i sistemi legacy e la presenza di un database semanticizzato per quanto riguarda la conoscenza condivisa rende naturale l'utilizzo dell'estensione al linked data di JSON, ossia JSON-LD, come linguaggio per la serializzazione e il trasferimento delle informazioni tra i vari strati del sistema.

JSON-LD estende JSON attraverso alcuni costrutti per il linked data, che sono sintatticamente identificati dal simbolo '@', consente attraverso l'esecuzione di precisi algoritmi la conversione da e verso il formato RDF e ha una serie di librerie dedicate implementate nei più diffusi linguaggi di programmazione. JSON-LD si presenta come un ottimo formato per la realizzazione di servizi RESTful, si presta alla modellazione di informazioni legate al social web e ha anche un supporto automatico per la modellazione dei messaggi di posta elettronica.

#### **RI2.4 - REALIZZAZIONE DI UNO O PIÙ DIMOSTRATORI**

L'attività è consistita nella messa in essere della metafora applicativa individuata nelle attività di ricerca e rappresentabile nella forma di piattaforma di Workspace Management. La piattaforma ricalca un modello di Personal Environment direttamente legato ai processi di business aziendale, secondo un approccio orientato alla consumerization, e costruito ricorrendo alle più recenti versioni dei sistemi operativi presenti oggi sul mercato. La piattaforma consente la composizione dell'ambiente di lavoro nella forma di dashboard e workspace personalizzabili, ed è in grado di integrare i paradigmi di accesso e composizione dei servizi analizzati nel progetto di ricerca. La principale entità logico/funzionale dell'artefatto realizzato è rappresentata dal Workspace, che fa riferimento a un unico processo di business ed è accessibile soltanto da determinate famiglie di utenti; si presenta nella forma di una o più Dashboard, in pratica un cruscotto gestionale mediante il quale l'utente entra in contatto con applicazioni, strumenti e servizi che gli permettono di interagire con gli altri utenti e accedere alle informazioni presenti nel sistema informativo. Ogni dashboard può ospitare un insieme di applicazioni (App) associabili a strumenti sia del Web 2.0 e sia di natura strettamente operativa, direttamente collegati a specifiche funzioni aziendali. Le App sono scelte tra quelle presenti in un repository centralizzato (App Repository), al quale gli utenti accedono in base ai privilegi a loro assegnati.

La piattaforma è stata modellata secondo una logica Multi-Layer dove ogni strato è auto consistente; tutto ciò con l'obiettivo di migliorare la scalabilità e stabilità complessiva. La scelta tecnologica applicata durante la fase di sviluppo (relativa sia alla sezione di front-end multicanale e sia allo strato contenente la logica di business) è legata, a concetti quali Rich Internet Application e RestFull WebServices che di fatto garantiscono un ottimo livello dei parametri suddetti.

Sono stati, inoltre, implementati una serie di strumenti di valutazione social-driven delle performance del Personal Environment. I suddetti strumenti si focalizzano, principalmente, sul modello di valutazione della workspace e widget reputation, definito



in RI2.2. In particolare, è stata progettata e realizzata una console grafica per la rappresentazione della variazione delle misure delle suindicate grandezze prestazionali, in funzione della variabilità di scenari di simulazione, generati automaticamente dal sistema.