Ecole Doctorale EDITE

Thèse présentée pour l'obtention du diplôme de Docteur de Télécom SudParis

Doctorat conjoint Télécom SudParis et Université Pierre et Marie Curie

Spécialité : Informatique et Télécommunications

Par Zhenzhen ZHAO

User-Generated Service: A User-Centered Design Approach in Web Mashups through Social Experience

Soutenue le 2 mai 2012 devant le jury composé de :

Benedicte LE GRAND	**Professeur à Paris 6**	Président
Pieter BALLON	**Professeur à Vrije Universiteit Brussel**	Rapporteur
Xiaodi HUANG	**Professeur à l'université de Charles Sturt**	Rapporteur
Christine BALAGUE	**Professeur à Management SudParis**	Examinatrice
Yacine GHAMRI-DOUDANE	**Professeur à ENSIIE**	Examinateur
Romain DURAND	**Co-Founder et CTO à Transatel**	Examinateur
Noel CRESPI	**Professeur à Télécom SudParis**	Directeur de thèse

Thèse n° 2012TELE0010

ACKNOWLEDGEMENTS

The research work reported in this thesis has been carried out at the Service Architecture group, Wireless Networks and Multimedia Services department, Institut Mines-Télécom, Télécom SudParis, France.

I would like to especially thank the following persons:

Prof. Noel Crespi, for supervising this thesis, guiding me through problems, and providing me valuable opportunities. I cannot thank you enough for your confidence and belief in me. Thank you for all your generous support and for your kindness.

Prof. Xiaodi Huang, Prof. Pieter Ballon, Prof. Benedicte Le grand, Prof. Christine Balagué, Prof. Yacine Ghamri-Doudane and Dr. Romain Durand, for being the official reviewers, for your valuable comments and constructive criticisms.

All the colleagues at Télécom SudParis, for your valuable advice given to the earlier versions of this thesis, and for the wonderful and inspiring lunch discussions.

Known and unknown people who took part in the end-user survey, for providing valuable information in this research. I am really grateful to each of you.

My parents, for always encouraging me to do my best and to strive for excellence. Your encouragement and support enabled me to ultimately achieve this milestone.

… and all friends for your everlasting love and support, and for valuables helps on this work.

Zhenzhen Zhao

Evry, France, Jan 2012

RÉSUMÉ FRANÇAIS

Le concept de développement utilisateur final (EUD), se réfère à des techniques qui permettent aux non-professionnels aux développeurs de créer ou de modifier le logiciel. Une telle pratique est observée pour être dans l'existence depuis les années 1980 au cours de laquelle la programmation par l'exemple [Halbert, 1984] et de la programmation par démonstration [Nevill-Manning, 1993] avait eu quelques approches populaires, ce qui a permis aux utilisateurs finaux de créer des applications sans programmation classique.

Au cours de dernières années, l'émergence du Web 2.0 et Service Oriented Architecture [Natis, 2003] ont abouti à l'évolution des riches, des applications Web interactives de plus de fonctionnalités, ciblées, non seulement pour les développeurs, mais aussi pour les utilisateurs finaux des entreprises commerciales , et de l'intention de domaine du consommateur dans l'utilisation des applications Web les plus récentes pour une utilisation personnelle ou de la situation. La plate-forme Web 2.0 a été caractérisée par une plus grande interaction entre les utilisateurs, l'intelligence collective et les utilisateurs de confiance en tant que co-développeurs pour créer des services rentables [O 'Reilly, 2005].

Service de l'utilisateur généré (UGS) est le phénomène de la création d'applications Web par les utilisateurs finaux, généralement en combinant contenus web existants ou des applications. Il se réfère à des utilisateurs finaux participants "dans la génération, la combinaison et l'adaptation des services pour créer des fonctionnalités et résoudre les problèmes dans leur travail". Cette création d'applications web composites est communément appelé aussi «mashups». Les mashups sont développés à des fins diverses, dont certaines sont centrées sur le développeur et certains sont centrés sur l'utilisateur. Les mashups qui sont orientées vers les utilisateurs finaux contribuer à la création d'UGS, et la composition de service de bout-axée sur l'utilisateur. L'utilisateur final conduit la recombinaison de données basées sur Internet et la fonctionnalité peut être définie comme les mashups [Grammel et Storey, 2008].

RÉSUMÉ FRANÇAIS

L'existence de mashups réside dans la SOA comme un élément essentiel, tout en ajoutant plus de valeur à cette architecture. SOA est un modèle architectural qui expose des services Web pour l'utilisation et la réutilisation. En fin de compte, si nous ne faisons que l'exposition de services, il n'offre pas vraiment une solution. Mais si nous sommes capables de faire abstraction de ces services dans des applications ou des mash-ups, ce qui démontre la valeur réelle de l'architecture. Technologies Mashup peut aider normales des utilisateurs finaux de combiner les services d'une manière visuelle, et de rendre une des solutions visuelles et applications qui ont une grande quantité de valeur [Ogrinz, 2009].

La création de mashups et d'UGS est également un moyen de franchir une nouvelle étape de la création de contenu généré par l'utilisateur (UGC), qui s'est rapidement développée pour devenir une source majeure d'information et de communication mondiale dans le Web [Wunsch-Vincent-et- Vickery, 2007]. Pratiques ont permis à UGC aux utilisateurs finaux de partager du contenu ou des données (photos, vidéo, texte, etc) dans différentes plates-formes de réseaux sociaux (Facebook, Myspace, etc), vidéo-partage (YouTube), les wikis et les blogs. Après la création de contenu et de partage, un nombre croissant d'utilisateurs sont considérés comme intéressés à combiner le contenu et les services existants afin de créer leurs propres applications Web [Braga et al., 2008], qui est dénommé UGS ou orientés vers l'utilisateur mashups. Enquêtes auprès des utilisateurs ont montré que les utilisateurs finals ne voir les avantages de l'utilisation des technologies de mashup pour la recherche, l'intégration et le partage de l'information [Zang et Rosson, 2008].

Applications d'UGS peut être conçu à des fins diverses, allant de la composition ad hoc pour une utilisation personnelle et la situation, à de plus grandes applications conçues pour les affaires ou une utilisation en entreprise. La méthode de développement d'un mashup peut être manuel ou un outil assistée [Yu et al., 2008]. Les méthodes manuelles de développement mashup se référer à l'utilisation de technologies conventionnelles de programmation sur le Web pour la création de services. Cette méthode nécessite bonnes compétences en programmation et la connaissance intime sur les régimes et la sémantique des sources de données ou les conventions d'affaires du protocole d'échange de messages. Les technologies telles que Ajax, services RESTful et les API, RSS et Atom microformats ont fait de ce processus plus simple. Néanmoins, manuel mashup développement est une tâche difficile même pour les développeurs qualifiés [Yu et al., 2008]. Le développement d'un

«moteur de recherche», en combinant plusieurs moteurs de recherche existants en utilisant leur API ouverte peut être un exemple de développement mashup manuel.

Pour faire du développement mashup plus simple et de permettre, même non expérimenté des utilisateurs finals à un bouillon de leurs propres applications Web, un certain nombre d'outils de développement et les cadres ont vu le jour et un tel procédé est connu comme outil de développement assisté par mashup. Par exemple: un utilisateur peut utiliser l'outil de Yahoo! Pipes pour aller chercher de flux favoris à partir de sites différents sports (ESPN, BBC Sports, etc), les trier et les filtrer (par exemple en fonction de 'soccer') et obtenir un résultat agrégé dans une seule page. Permettre aux utilisateurs finaux de faire leurs propres vitesses mashup le processus de création de service globale et résout des situations-problèmes pour les utilisateurs tout en présentant leur créativité. Ces outils ont des caractéristiques différentes et des approches différentes de la composition.

Dans le cadre de notre travail, la référence à UGS et 'mashup' le terme sera plus généralement impliquent mashups fin créés par l'utilisateur qui ne nécessitent pas n'importe quel type de programmation, et dont le développement est normalement pris en charge par une plate-forme.

Mashups peuvent être classés selon différentes bases conceptuelles. Le premier est "Mashup client" et "Mashup d'entreprise". Mashups clients sont généralement créés à des fins personnelles pour résoudre le problème de la situation. Deux exemples de mashups clients sont les suivants: 1) Un service qui récupère plusieurs des amis de Facebook et des parcelles de leur image dans une carte Google Map, sur la base de leur emplacement, et 2) Un service qui récupère les flux RSS de plusieurs sites, comme BBC Sports, CNN Sports et ESPN, les combine, les filtre en fonction de certains mots clés (par exemple "tennis"), les trie selon la date (le dernier exemple en premier), et affiche le résultat filtré sur une seule page. Les mashups d'entreprise, d'autre part, sont mis au point pour résoudre les problèmes dans le domaine des affaires, et nécessitent une plus grande collaboration entre un certain nombre de personnes pour mener à bien les processus d'affaires d'une manière coordonnée. Un système d'approbation des rapports de dépenses dans une entreprise serait un exemple d'un mashup d'entreprise. Ce système peut être un mashup de services additionnels ou des ressources comme en temps réel des graphiques et des rapports sur l'intelligence d'affaires (BI) qui indiquent combien le budget est laissé dans la société, en aidant à prendre des décisions

comme si d'approuver le montant indiqué serait de créer un plus situation budgétaire-, ou tout autre problème. Le analogue termes "consommateurs Mashup" et "Business Mashup" sont souvent utilisés pour décrire les mashups à la clientèle et de l'entreprise, respectivement.

Une autre approche pour les classifications mashup est la suivante: "Mashup de personnalisation de la page Web" et "Mashup Processus. Mashups" Web page de personnalisation sont utilisées pour changer sites en supprimant les éléments, ajouter des widgets supplémentaires et de changer leurs interfaces utilisateur). Processus mashups permettent l'agrégation des données, du contenu et des services, et les inclure dans les processus automatisés séquentiels [Grammel et Storey, 2008].

Une manière similaire pour classer les mashups comme "front-end" et "back-end" mashups. Front-end mashups aider à construire Web frontaux, comme des tableaux de bord, l'utilisation de widgets ou gadgets et peu ou pas de programmation (iGoogle, Netvibes, Pageflakes). Back-end mashups se combinent accessibles sur le Web de données et services en plus de services web utiles qui peuvent être appelées facilement en utilisant une interface REST (Kapow, Yahoo Pipes).

Une autre terminologie similaire pour la classification mashup est "Mashup horizontale" et "Mashup verticale". Mashups horizontales peut être vu comme un processus de regroupement des ensembles de services généralement similaires ou complémentaires à agréger leurs sorties, dans lequel il n'ya pas d'interaction entre les modules de service, mais où l'personnalisable frontal offre plus de valeur à des utilisateurs finaux qui souhaitent résoudre une tâche particulière. Un mashup verticale, d'autre part, est un processus d'orchestrer les sorties de certains services dans les entrées des autres, où les modules de services sont reliés entre eux et les paramètres sont passés entre les modules afin de produire un nouveau service amélioré.

Il a été un grand intérêt au cours des dernières années dans les technologies de mashup, et des décideurs mashup de nombreuses plates-formes de développement de mashups, outils et [Grammel et Storey, 2008] ont vu le jour à un rythme rapide. Malgré cet intérêt rapide mashups, des outils complets de développement de mashups sont loin d'être adéquate, ce qui, pour la plupart des cas, la création d'une application mashup nécessite encore un effort de programmation important [Yu et al., 2008]. Une telle méthode de développement mashup

nécessite la connaissance de langages de programmation avancés à intégrer les API Web et se nourrit, ce qui implique une expertise considérable de programmation [Zang et Rosson, 2008]. Les utilisateurs finaux qui n'ont pas de compétences en programmation ne sont pas en mesure de tirer pleinement parti de la valeur et les avantages de mashups qui nécessitent l'utilisation d'API, des services RESTful, Atom et RSS.

Il ya les plates-formes de développement qui sont actuellement disponibles dans le Web, comme Yahoo! Pipes, Intel Mashmaker, IBM Mashup Center, etc, qui ne nécessitent pas de connaissances en programmation. Toutefois, pas tous ces outils sont faciles à comprendre et à utiliser, surtout pour les normales des utilisateurs finaux qui n'ont pas l'art de la pensée computationnelle [Zang, 2008]. Bien que des études ont été faites pour le développement d'outils de mashup, peu d'outils tels que Intel et Mashmaker Marmite [Ennals et Garofalakis, 2007, Wong et Hong, 2007] et les efforts de recherche ont examiné les petits besoins de la programmation de moins avertis des utilisateurs finaux [Zang, 2008].

Plusieurs recherches des utilisateurs ont été menées dans des mashups.

Zang, 2008 dans l'article "*Mashups for the Web-active user*", soulignent le fait que les outils mashup actuellement disponibles ne soutiennent pas suffisamment le savoir-faire technique des utilisateurs finaux. L'auteur présente le terme Web-active de définir les utilisateurs qui sont considérés comme ligne active. Ils sont les utilisateurs qui utilisent Internet sur une base quotidienne, et qui tentent de trouver de nouvelles façons d'intégrer leurs activités en ligne. Comme tels, ils sont les prêts à faire de nouvelles initiatives avec la technologie, bien qu'ils ne possèdent pas l'expertise de programmation pour créer des mashups. Une enquête a été menée avec plus de 200 étudiants qui ont été considérés comme des web-actif. Lorsque expliqué à propos des mashups, les utilisateurs ont donné la rétroaction que l'utilité est plus important pour eux que la difficulté perçue de l'action. Cela montre que ces utilisateurs sont prêts à apprendre de nouvelles technologies si elles le jugent utile pour eux.

Le document donne un aperçu de plusieurs activités en ligne de l'utilisateur et les préférences. La plupart des utilisateurs ont été connus pour créer du contenu en ligne «juste pour le fun» ou «de socialiser». Une analyse plus poussée a révélé que un fort prédicteur de leur volonté de créer des mashups serait pour le partage en ligne passe-temps. On a vu que les

personnes ayant moins de technologie initiative préfèrent mashups qui impliquent des personnes et de l'activité sociale et ceux de l'initiative de plus de technologie préfèrent plus complexes, les médias mashups connexes.

Zang et Rosson, 2008 dans leur article "*What's in a mashup? And why? Studying the perceptions of web-active end-users*" L'étude des perceptions de web-actifs les utilisateurs finaux "donne un aperçu sur ce type de contenus et services pourrait utilisateurs finaux veulent réellement mash up. 116 potentiels (novice), les développeurs de mashups ont été invités à donner des exemples de mashups qu'ils souhaitent créer. Les résultats montrent que les utilisateurs sont plus intéressés à développer des mashups pour le contenu des nouvelles et des cartes. Les auteurs justifient cependant que cela pourrait être parce que les deux exemples de mashups fournies aux utilisateurs avant l'enquête illustré ces deux idées. Un grand nombre d'exemples ont été donnés pour le sport, puis à titre de référence (dictionnaire, encyclopédie, etc), suivie par sociale (par exemple Facebook) et les activités personnelles, que les utilisateurs sont de plus en plus l'adoption de l'Internet.

Dans un autre article, "*Mashups on the Web: End User Programming Opportunities and Challenges*" [Zang, 2009], l'auteur affirme que, parmi les trois processus de mashups composent, à savoir. la collecte de données, manipulation de données et la présentation des données, les utilisateurs finaux trouvent habituellement au stade de manipulation de données le plus déroutant. Pour examiner la manipulation des données plus en détail, 12 participants (web-actifs les utilisateurs finaux) ont été interviewés, et a également demandé de créer un mashup exemple en utilisant l'outil Yahoo! Pipes. Yahoo! Pipes est un outil ayant pour accent sur l'intégration des données et est essentiellement un langage de programmation visuelle. Les résultats montrent que notamment le nom de n'importe quel module était le seul moyen pour l'interprétation de ses fonctionnalités, et dans cette plate-forme, les utilisateurs ont été souvent confondu entre les noms et les fonctionnalités de deux ou plusieurs modules tels que le module de boucle et le module de filtre. L'étude suggère que, pour maîtriser chaque module, les utilisateurs doivent passer un temps considérable. En substance, l'auteur conclut que Yahoo! Pipes n'est pas un outil pour les non-programmeurs. L'organisation globale de l'outil et les noms des modules donnent à penser que cet outil simplifie essentiellement d'agrégation de données pour les développeurs. Pour les moins avertis de programmation des utilisateurs finaux, l'outil leur impose de définir des actions d'une manière programmatique, qui peuvent souvent pas être possible, en raison de la différence sous-jacente entre les façons dont les développeurs et les utilisateurs finaux normales peuvent résoudre les problèmes. Les

programmeurs ont généralement développé la capacité de ce qu'on appelle la «pensée de calcul». Dans le cadre de mashups, la pensée de calcul se réfère à la manière dont les programmeurs peuvent se décomposer problèmes dans une manière qui soit acceptable par les systèmes informatiques. Les utilisateurs finaux, ont généralement des capacités de la pensée computationnelle. Toutefois, dans le long terme, il a été recommandé qu'au lieu de construire des outils faciles à utiliser, il pourrait être bénéfique pour encourager les utilisateurs finaux à accroître leurs compétences en calcul.

Grammel et Storey, 2008, dans le document "*An End-user Perspective on Mashup Makers*" ont étudié plusieurs outils mashup existants, comme Yahoo! Pipes, Microsoft Popfly, IBM Mashup Center, Google Mashup Editor et comparé leurs caractéristiques sur six aspects différents, à savoir. Niveaux d'abstraction, aide à l'apprentissage, les caractéristiques de la communauté, la recherche, conception de l'interface et des techniques de génie logiciel.

Trois niveaux d'abstraction sont définis, que les différences dans le niveau de la programmation et des connaissances informatiques requises, à savoir: haut niveau d'abstraction, l'abstraction de niveau intermédiaire et niveau d'abstraction faible. Le haut niveau d'abstraction, de connaissances en programmation n'est pas nécessaire, mais la souplesse obtenue est faible, par exemple la réutilisation des mashups déjà construits, mashups et le paramétrage et la programmation widget de par l'exemple. Niveau d'abstraction intermédiaire encapsule les détails technologiques dans le système-spécifiques notations, mais nécessite généralement aux utilisateurs d'avoir des connaissances sur les flux de données, types de données et des widgets d'interface utilisateur. Niveau d'abstraction faible nécessite des connaissances en programmation, mais la plus grande flexibilité peut être acquise. La majorité des outils de commentaire présenter soutien à tous les niveaux d'abstraction, à l'exception Yahoo! Pipes et Google Mashup Editor. Pour une abstraction de haut, tous les outils permettent de réutilisation complète des mashups, la programmation par l'exemple est pris en charge par IBM Mashup Center et Intel Mash Maker. Pour un niveau d'abstraction intermédiaire, Microsoft Popfly, Yahoo! Pipes et IBM Mashup Center ont langages flot de données visuelles, tandis que le Serena Mashup Center dispose d'un langage d'orchestration visuelle processus / workflow. Dans une abstraction bas niveau, textuelles Domain Specific Language (DSL) des éditeurs pour des langues telles que HTML, JavaScript, etc sont utilisés pour représenter des éléments de mashup.

Soutien à l'apprentissage comprend de base aide à l'apprentissage comme des tutoriels, des screencasts, etc documentation de l'API, les communauté de soutien basé sur l'apprentissage, comme des forums de discussion, le partage des ressources, et spécifiques au contexte des suggestions qui fournissent des conseils aux utilisateurs en fonction de leur activité actuelle dans la construction d'un mashup. En général, tous les décideurs mashup ont été trouvés à avoir un bon support pour l'apprentissage dans les aspects mentionnés ci-dessus, cependant, les suggestions spécifiques au contexte fourni par Microsoft Popfly seulement.

Caractéristiques communautaires comprennent des installations pour les mashups de partage, la catégorisation collaborative des mashups par des techniques telles que le marquage et de tarification, la fourniture de forums de discussion, et l'extension du partage des ressources et des mashups créés au cours de réseaux sociaux comme Facebook. Tous les mashups commentaire disposent d'installations pour les mashups de partage, les caractéristiques de marquage sont fournis par Yahoo! Pipes, Microsoft Popfly et Intel Mashup Center, et les caractéristiques de notation sont fournis par Intel Mashmaker, en plus de ces trois outils. Les forums de discussion sont prévus dans chacun d'eux, à l'exception Intel Mashmaker, et le soutien des réseaux sociaux est fourni par Microsoft Popfly seulement.

Consultables réfère à chercher et trouver des mashups et des éléments de mashup et comprend de recherche textuelle, la navigation mashups par les propriétés structurales, tandis que spécifiques au contexte des suggestions d'éliminer la nécessité d'effectuer une recherche réelle. La plupart des outils ont été vus commentaire pour avoir un bon support pour la recherche en mode texte seulement, et moins pour la navigation par des propriétés structurelles, et des suggestions spécifiques au contexte.

Conception de l'interface utilisateur est également un aspect important de la détermination de la facilité d'utilisation de mashups. Interfaces peuvent être générées automatiquement en fonction de la sortie de l'opération mashup, cependant, manquent de souplesse. Composition d'assurance-chômage, d'une manière visuelle ou textuelle est considéré comme l'approche la plus flexible, mais aussi assez compliqué. Visuel UI composition se trouve être la plus commune parmi les outils de commentaire, soutenue par Intel Mashup Center, Serena Mashup Composer et Intel MashMaker.

Techniques de génie logiciel se référer à des techniques comme la gestion de la demande de débogage, de contrôle de version, les tests et le changement, qui ont été

considérées comme importantes pour le développement de l'utilisateur final. Parmi les outils examinés, le débogage, la sortie est vu pour être pris en charge par Microsoft Popfly, Yahoo! Pipes et Intel Mashup Center, tandis que le contrôle de version est supporté par Intel Mashup Center et Serena Mashup Composer.

En raison de l'exigence de compétences en programmation, et les plates-formes de mashup qui présentent de nombreux détails techniques pour le montage et la création de services, une web-savvy encore un utilisateur final non-programmeur aura *peu de motivation* pour s'engager dans des mashups qui, autrement, créer une opportunité pour la génération des services à valeur ajoutée pour le même utilisateur. Pour apprendre à utiliser les plates-formes de mashup, les utilisateurs doivent avoir une motivation suffisante et assez perçu-utilité.

Ainsi, la question de recherche générale dans cette thèse est "? Comment amener les utilisateurs finaux plus vers la création et l'utilisation de mashups" Du point de vue conception de l'utilisateur-galop, cette question est décomposé dans les questions suivantes:

- Comment les utilisateurs finaux perçoivent mashups?
- Comment l'utilisateur final perspectives aider dans la conception d'un «utile» et «facile à utiliser" mashups de consommateurs?
- Comment les utilisateurs finaux perçoivent notre conception? et leur acceptation des mashups à la consommation en général?
- Comment améliorer notre conception basée sur les utilisateurs finaux des commentaires?

Cette thèse contribue à la recherche et le développement de services générés par l'utilisateur à partir des quatre aspects suivants.

Tout d'abord, puisque le service généré par l'utilisateur (UGS) est un nouveau sujet, la première contribution de la thèse est de fournir un aperçu complet de la mise au point l'état de l'art dans ce domaine. Par comparaison avec le concept de contenu généré par les utilisateurs, nous donnons la description précise d'UGS, puis passer par différentes technologies afin d'analyser leurs problèmes. Les nouvelles tendances de UGS avec de nouvelles fonctionnalités importantes sont illustrés, à partir de perspectives à la fois techniques et commerciales. Les technologies sous-jacentes clés, ainsi que les plates-formes actuelles et les outils sont également détaillées.

Notre deuxième contribution réside dans la conception et la mise en œuvre une plate-forme de mashup simples qui fournit la plus grande motivation pour l'utilisation des mashups. Suivie par la revue de la littérature des études globales sur la perspective de l'utilisateur final sur les mashups de services, nous présentons un cadre mashup le consommateur fondé sur les activités quotidiennes, en rapportant les événements de calendrier de tous les jours à des gadgets utiles. Le système, appelé EBSP (Prestataire de services basée sur des événements), offre un haut niveau d'abstraction pour les utilisateurs finaux, des expositions contexte d'orientation et de fonctionnalités de personnalisation, et se concentre sur l'intégration des fonctionnalités plutôt que d'intégration de données. Le système a été évalué par 131 utilisateurs finaux pour tester sa facilité d'utilisation. En outre, comme un exemple représentatif, le système est utilisé dans un projet de l'utilisateur l'acceptation de modèle pour les mashups à la consommation.

Notre troisième contribution réside dans la recherche de suivi de la recommandation de gadget, qui est l'un de l'écoulement important dans la conception EBSP. Nous nous concentrons sur le problème de l'utilisation de l'annotation sociale pour soutenir l'expérience de service Web à découvrir. Un gadget amélioré web découverte modèle est proposé sur la base de l'étiquetage social (Folksonomie), d'une part, en explorant les relations tag, puis à l'aide de trois différents régimes de découverte gadget. Utilisation de la Widgetbox.com service en ligne à titre d'exemple, nous démontrons l'exactitude et l'efficacité de notre modèle proposé.

Quatrièmement, à partir du modèle d'acceptation des utilisateurs du système EBSP nous avons appris que le partage et la collaboration dans les mashups de consommateurs ont un effet positif sur l'espérance de la performance et l'effort. Ce qui nous a inspirés pour développer le système EBSP aux réseaux sociaux, qui permettent de partager des événements et des mashups. Suivie d'une enquête sur les recherches récentes sur les applications de réseaux sociaux, nous avons conçu un nouveau système pour l'activité orientée vers les réseaux sociaux en ligne nommé Dig-Event (Do-it-ensemble de l'événement). Dig-Event est un système ouvert, l'espace social pour les utilisateurs de partager des événements et de découvrir les activités et les intérêts communs de placement parmi les contacts sociaux comme camarades de classe, les familles, amis et collègues. Il permet aux utilisateurs de partager leurs activités, afin de personnaliser leurs cercles sociaux, à organiser des

événements (en sélectionnant l'activité à base de gadgets), et de socialiser autour d'eux. Les caractéristiques de recommandation événement et l'intégration avec les réseaux sociaux existants renforcer davantage l'expérience de socialisation événement.

Un résumé détaillé des contributions à la recherche de cette thèse est répertorié.

La première contribution de cette thèse est associée à la notion de user-generated service. Artical I, intitulé "The Incoming Trends of End-user Driven Service Creation", illustre le fait que la création de services est le prochain grand utilisateur final battage mené dans le domaine de la technologie, et cette tendance va conduire à un vaste ensemble de nouvelles fonctionnalités et de services. L'auteur a eu la responsabilité principale pour le contenu technique et travail d'écriture papier. Le document soulève trois questions générales à cette nouvelle tendance:

- Pourquoi devrais-utilisateurs finaux de créer des services?
- Quels sont les travaux en cours sur l'utilisateur final la création de services axée sur?
- Quels sont les défis de permettre aux utilisateurs finaux de faire la création de services future?

En ce qui concerne la première question, l'auteur montre que de l'utilisateur final perspective, la meilleure façon d'y parvenir est de donner personnalisation de l'utilisateur les moyens de les utilisateurs finaux mêmes pour développer les services répondant à leurs besoins spécifiques, tandis que du point de vue des affaires, l'idée principale derrière la fin - création d'un utilisateur du service est de raccourcir l'analyse / le développement de la chaîne de commercialisation, exploiter l'intelligence collective et de promouvoir l'innovation dans le développement de logiciels.

L'auteur illustre ensuite les tendances entrants de l'utilisateur final la création de services d'entraînement avec de nouvelles fonctionnalités importantes, et fournit des cinq principaux thèmes de recherche à partir de perspectives à la fois techniques et commerciales.

Artical II, intitulé "A Survey of User Generated Service", les points que la recherche actuelle vise à favoriser l'évolution de l'utilisateur absolument réussi Generated Content (UGC) à généré par les utilisateurs du service (UGS). L'auteur de cette thèse était l'auteur principal du

document. L'auteur clarifie la notion d'UGS en le comparant avec UGC. Les deux d'entre eux peuvent être considérés comme un ensemble d'activités qui consistent dans les interactions entre l'utilisateur et le système. Ces activités ont un résultat à valeur ajoutée pour l'utilisateur. Cela implique qu'une certaine quantité de l'effort de l'utilisateur créatif est impliqué dans le contenu ou le processus de création de service. En outre, à la fois UGC et UGS l'accent sur le contenu / service créé par l'utilisateur final, et non pas des professionnels. En ce qui concerne la différence, la première chose est liée au chiffre d'affaires. Dans le web actuel, le processus de création de contenu est libre. Le système est libre d'utilisation, et pas de recettes seront ajoutées sur le contenu que l'utilisateur créé. Cependant, dans d'UGS, les utilisateurs peuvent avoir la possibilité de partager les revenus entre les multi-acteurs, et le développement de la plate-forme de création de services tend à être chargée. La seconde différence est relative à l'intention des utilisateurs et la cible. A la différence de l'UGC, qui se concentre sur la "création pour le plaisir", la socialisation et les aspects de partage des connaissances, généré par les utilisateurs du service accorde plus de poids sur la personnalisation du service et la réutilisation des services.

Certaines caractéristiques essentielles de UGS et UGS des plates-formes suivantes:

- Ordinaire des utilisateurs finaux pour le service / application de création, sans programmation effective
- Base de données de service accessible des services existants et le contenu
- Un environnement riche et visuelle pour le service / contenu de sélection, l'interaction, le regroupement, filtrage, etc avec une flexibilité suffisante et la personnalisation
- Génération de nouveaux services à valeur ajoutée des applications composites (service), ce qui nécessite un certain effort créatif, pour une utilisation personnelle, sociale, de la situation (ad hoc), ou en collaboration dans le domaine du consommateur ou de domaine d'entreprise.

L'auteur continue de détailler les technologies sous-jacentes clés de généré par les utilisateurs du service, les plates-formes actuelles et des outils, ainsi qu'un bref aperçu des tendances entrants générés par les utilisateurs de service.

La deuxième contribution est relative à l'analyse de conception, la mise en œuvre et l'utilisation d'un système basé sur les événements de service de mashup.

Artical Court III, intitulé "An Event-Based Functionality Integration Framework", présente un cadre basé sur les événements de service de mashup d'aborder la question de la personnalisation du service. Le cadre proposé aborde la question de mashup à partir d'une nouvelle perspective en extrayant et le raisonnement le contexte par l'événement généré par l'utilisateur, tout en recommandant et en agrégeant les services contextuels dynamiquement en réponse aux exigences fonctionnelles de l'utilisateur.

Artical IV, intitulé "Mashup Services to Daily Activities – End-user Perspective in Designing a Consumer Mashups", a introduit un système mashup centrée sur l'utilisateur, le nom du prestataire de services basé sur les événements (EBSP). L'auteur de cette thèse a pris la responsabilité principale pour le contenu technique dans ce travail d'écriture et papier. L'idée vise à concevoir et mettre en œuvre un système de mashup centrée sur l'utilisateur qui fournit une plus grande motivation pour l'utilisation des mashups, en rapportant les événements de calendrier de tous les jours à des gadgets utiles. Le document souligne que, pour permettre aux non-experts aux utilisateurs d'être engagés dans cette pratique innovante, l'utilité perçue et la facilité perçue d'utilisation sont les principaux facteurs à prendre en compte car ils ont un impact direct et positif sur les utilisateurs à accepter et à utiliser n'importe quelle technologie.

Suivie d'une revue de la littérature des études globales sur la perspective de l'utilisateur final sur les mashups de services, ce document conclut que: d'une part, mashup processus est encore considéré comme assez compliqué et décourageant pour non-experts les utilisateurs finaux. D'autre part, les systèmes de web page de personnalisation mashup sont vu fournir un moyen pratique pour les utilisateurs d'agréger des services sélectionnés, appelés widgets ou gadgets dans leur tableau de bord personnel pour la création d'un environnement personnalisé. Ces méthodes, cependant, manquent de souplesse étant donné que les gadgets ne peuvent pas communiquer entre eux ou avec tout autre service web. En outre, ces systèmes présentent des bases de données de services importants et souvent de permettre l'accès à la tierce partie pour la fonctionnalité accrue du système, qui ne fournissent pas nécessairement une meilleure solution et la qualité de l'expérience pour l'utilisateur.

Artical V, intitulé "Consumer Mashups: End-User Perspectives and Acceptance Model", présente un modèle d'acceptation de l'utilisateur pour les mashups de consommateurs, d'identifier précisément les facteurs qui entraînent à leur adoption et dans quelle mesure. L'auteur de cette thèse a été co-auteur de l'article et a été principalement responsable des méthodologies de l'acceptation par les utilisateurs du modèle proposé. Ce modèle est basé sur la théorie unifiée de l'utilisation et l'acceptation de la technologie (UTAUT) modèle.

L'étude est menée avec un sondage en ligne de 131 utilisateurs de recueillir des données afin de vérifier le modèle. Les utilisateurs sont invités à améliorer les informations sur le système EBSP. En plus des constructions internes de UTAUT, le modèle de recherche proposé utilise plusieurs constructions extérieures spécifique pour la consommation des systèmes de mashup: l'intégration des connaissances, le partage et la collaboration, Do-it-yourself et la jouissance perçue. Un modérateur externe, l'innovation grâce à la technologie, est présenté à un degré variable d'effet sur l'intention de personne d'utiliser une technologie. Neuf hypothèses sont formulées.

Un sondage en ligne a été développé sur la base du modèle de recherche. En particulier, nous sommes intéressés à savoir ce que cela prend-il pour les utilisateurs d'exprimer leur volonté en utilisant la technologie de mashup. Nous demandons aux utilisateurs de donner leurs expériences dans l'utilisation du système EBSP, avant de le faire, un didacticiel vidéo sur les outils de mashup brève commerciales est prévue pour les familiariser avec la technologie de mashup. Ensuite, nous fournir un didacticiel vidéo brève sur le système EBSP, avec divers exemples. Un total de 423 demandes sont envoyées en ligne, dont nous recevons 233 réponses partielles. Seulement 131 sont complètes et valides pour notre étude.

Les données obtenues à partir de l'enquête a été analysées statistiquement par le biais de Cronbach Alpha test, personne de corrélation et les régressions multiples. Plus de résultats sont présentés dans le Artical V. L'analyse des relations dans le modèle de recherche est présentée comme suit:

Hypothèse 1: L'intégration des connaissances a été trouvé pour avoir un impact modérément positif sur l'espérance de rendement. Cela signifie que si les utilisateurs peuvent

intégrer diverses sources d'information et de différentes applications en une seule plateforme de mashup, qu'ils considèrent comme la plate-forme pour être utile.

Hypothèse 2: le partage et la collaboration ont été trouvés à avoir un effet modérément positif sur l'espérance de rendement. Cela signifie que si les utilisateurs peuvent partager des informations et des applications avec l'autre, par exemple, voir des autres applications, de réutiliser leurs applications, et travailler en collaboration pour développer une application, puis les utilisateurs perçoivent le système pour être utile pour eux.

Hypothèse 3: le partage et la collaboration ont été trouvés à avoir un effet positif mais faible sur l'espérance de l'effort. Cela indique que les installations pour les applications de partage et de travail collaboratif ont un rôle positif, mais faible pour réduire l'effort nécessaire à la construction d'une application pour les participants de l'enquête.

Hypothèse 4: Organisation d'activités de la vie quotidienne et les processus a un impact modérément positif sur l'espérance de rendement. Parmi toutes les constructions concernées, OAP est observé que la plus forte par rapport à l'espérance de rendement. Cela indique que les utilisateurs de l'enquête envisager une plate-forme pour être le plus utile si elle permet à l'organisation de leurs activités au jour le jour les activités.

Hypothèse 5: Do-it-Yourself attribuer se trouve à avoir un effet modérément positif sur l'intention comportementale. Modérateurs comme innovation avec la technologie, les sexes et les âges ont été vus à affecter cette relation. La tendance à créer des applications web, en cas de besoin on observe augmenter à mesure que les augmentations de l'esprit d'innovation technologiques de moyenne, haute à très haute. Les mâles ont été vus pour avoir une plus grande initiative pour faire leurs propres applications. Il a été observé que les personnes au-dessus de 30 ans étaient très intéressés par le développement de leurs applications. Cela peut être dû au fait qu'ils ont assez d'expérience et la connaissance de la technologie pour essayer de nouvelles choses pour mettre les valeurs dans leur vie quotidienne.

Hypothèse 6: Do-it-Yourself attribuer a été trouvé pour avoir un effet positif mais faible sur le plaisir perçu. Il indique que peu de gens aiment travailler avec le système parce qu'ils sont en mesure de développer leurs propres applications ad-hoc. Il peut y avoir un autre

facteur qui affecte la jouissance perçue plus de bricolage fait. Cependant, on voit que les gens avec le plus grand d'innovation technologique du plaisir à travailler avec le système de plus, car ils peuvent faire des demandes de leur propre manière. Aussi, les personnes ci-dessus de 30 ans a montré un niveau élevé de jouissance de travailler avec le système, parce qu'ils pouvaient faire eux-mêmes et d'assembler des services que par la nécessité. Ceci est cohérent avec l'effet de l'âge dans l'hypothèse 5.

Hypothèse 7: l'espérance de rendement a été trouvé pour avoir une forte incidence positive sur l'intention d'utiliser les systèmes de mashup. L'espérance de rendement est observé que la plus forte par rapport à l'intention comportementale, par rapport à toutes les autres constructions concernées. Ceci est cohérent avec notre conviction que les utilisateurs ont besoin d'avoir un système qui est très utile pour eux s'ils décident de l'utiliser. La relation a été animée par le sexe et l'âge tels que les mâles ont une plus grande intention d'utiliser le système quand ils le jugent utile. Ce résultat est cohérent avec l'hypothèse similaire dans le modèle UTAUT (Venkatesh et al., 2003). Les gens de plus de 30 ans ont une forte intention d'utiliser le système, si elles le jugent utile.

Hypothèse 8: Espérance effort a été trouvé pour avoir un effet positif mais plus faible sur l'intention comportementale. Il signifie que les utilisateurs choisissent d'utiliser un système de mashup s'ils pensent que cela ne nécessite pas beaucoup d'efforts, mais ce n'est pas aussi important que le système devant être «utile» à eux de décider d'utiliser le système. Les mâles ont montré un plus grand intérêt à utiliser le système si elles le jugent à prendre moins d'effort, alors que les femelles ont montré moins d'intérêt pour cette question. Ceci est en contraste avec le modèle UTAUT, qui suppose que les femmes ont une plus forte relation entre l'espérance de l'effort et l'intention comportementale. Les gens au-dessus de 30 ans a montré un grand intérêt dans l'utilisation du système si elle ne nécessite pas beaucoup d'effort.

Hypothèse 9: L'influence sociale n'a pas été trouvé pour avoir un effet significatif sur l'intention comportementale. Toutefois, lorsque l'on analyse la relation fondée sur le sexe, les femelles ont été observés à être affectée par l'influence sociale dans l'utilisation du système de mashup. Ceci est cohérent avec les études précédentes (Venkatesh et al., 2003). Le groupe d'âge supérieur à 30 ans avaient une plus grande intention d'utiliser le système de mashup où l'influence sociale sur eux a été pris en considération.

Arctical XI, intitulé "The Design, Implementation and Usage Analysis of an Event-Based Consumer Mashups System", est une version complète et globale de la III papiers, IV et V. Ce document présente le cadre mis à jour, conception et évaluation d'un basé sur les événements système de mashup de service. L'auteur de cette thèse a surtout contribué à le contenu technique et travail d'écriture de papier XI.

Le système et les méthodes présentées ci-dessus ont été déposées dans la Partie I-Unis en matière de brevets d'État, intitulé "System and Methods for Service Provision Based on Events".

La troisième contribution est liée à améliorer l'adéquation entre l'utilisateur créé l'événement et gadgets système recommandées dans le système EBSP.

Artical XIII, intitulé "A System for Web Widget Discovery Using Semantic Distance between User Intent and Social Tags", décrit l'état de l'art sur la recherche sociale liée à l'annotation et présente une méthode de découverte de services par le biais de l'étiquetage social. L'auteur de cette thèse a pris la responsabilité principale pour le contenu technique dans ce travail d'écriture et papier. Le système d'expérimentation est réalisée en utilisant les services de widget WidgetBox.com, où chaque widget est annoté par les balises. L'objectif principal est un système automatique qui peut découvrir des widgets (gadgets) qui correspondent à la requête des utilisateurs. Dans le premier ouvrage sur les mashups de services basés sur les événements, l'entrée d'utilisateur dans les détails des événements peut être considéré comme une telle requête. La méthode de service détaillée découverte est classé en deux étapes: découverte tag et le widget de découverte. Tout d'abord, la découverte tag utilise de Kullback-Leibler (KL) Divergence et recuit déterministe (DA) algorithme pour trouver la relation de chaque étiquette et chaque événement. Pour chaque mot clé d'entrée, les dix premiers tags sont extraites. Ensuite, le widget de découverte est réalisée en utilisant les dix premiers mots-clés: trois régimes sont pris en compte dans ce processus pour obtenir une liste définitive des widgets.

Nous avons en outre mis en place un système pour tester les méthodes. Débit du système est illustré. Ensemble de données du système, les résultats de découverte tag et les résultats de découverte de widget sont présentés en détail dans le Artical XIII. Enfin, nous

avons effectué un test d'utilisabilité comparant nos résultats avec mots-clés correspondant à des résultats. Notre système a prouvé son exactitude et l'efficacité.

La quatrième contribution mettre le système EBSP une étape supplémentaire dans le contexte de l'interaction sociale par la conception d'un réseau social en ligne autour d'activités.

Artical VI, intitulé "The Design of Activity-Oriented Social Networking: Dig-Event", présente la conception du système et rationnelle pour un site de réseautage social une nouvelle activité axée sur appelée Do-it-ensemble d'événements (Dig-Event). Ce système a été inspirée par des recherches antérieures sur les agendas et les applications populaires de réseaux sociaux comme Facebook et Google +. Dig-Event est un système ouvert, l'espace social pour les utilisateurs de partager des événements et de découvrir les activités d'intérêt mutuel entre les contacts sociaux comme camarades de classe, les familles, amis et collègues. Il permet aux utilisateurs de partager leurs activités dans le cercle personnalisé sociale, organiser des événements en sélectionnant l'activité à base de gadgets, et de socialiser autour d'eux. Par rapport à courants existants basés sur les événements des réseaux sociaux, les six principes de conception ont été proposées pour concevoir Dig-Event:

Principe de conception 1: Les réseaux sociaux entre pairs, et non des étrangers. Parmi les nombreuses raisons qui attirent les utilisateurs de réseaux sociaux, dont le principal semble être la capacité de communiquer avec vos amis, votre famille et collègues. Alors que Going.com, Upcoming.yahoo.com, Eventful.com et RU-In? se concentrer sur la mise en réseau géo-social et relier les gens ayant des intérêts similaires, nous croyons que le partage des activités et de découverte parmi les contacts existants est toujours le premier choix comparant à d'autres contacts inconnus. La plupart des gens ont tendance à découvrir l'intérêt mutuel de leur social existant (source très fiable).

Principe de conception 2: Partager et découvrir au jour le jour les activités, plutôt que des événements officiels.

Principe de conception 3: la gestion de l'événement, non seulement «4W» - Quoi, Quand, Où, Qui, mais aussi «H» - Comment. Calendriers faire un ordonnancement de soutien, c'est à dire décider de la «quand», «où» et «qui» d'un événement. Il ya un support limité pour aider les utilisateurs à réaliser l'événement. Différent de l'activité existants basés sur des

approches de réseautage social, Dig-Event organise également le «comment», en aidant les utilisateurs à organiser les événements à travers leurs propres mashups gadgets préférés.

Principe de conception 4: le libre accès à personnalisés milieux sociaux, et non pas «tout ou rien". Notre conception a été largement inspirée par Google + cercle social. Nous avons choisi le modèle de partage asymétrique de Google + au lieu du modèle de l'amitié réciproque, comme Facebook et modèle d'abonnement asymétrique comme Twitter, de sorte que les utilisateurs puissent partager un cours avec les gens, mais ils n'ont pas à partager le dos, réduisant ainsi le coût de la possibilité de découvrir l'événement.

Principe de conception 5: recommandation de l'événement sur la découverte événement. Un site avec le partage de manifestation publique a de nombreuses possibilités pour faire des recommandations personnalisées pour la découverte. Nous avions prévu que, en plus de vues de liste d'événements de navigation, les utilisateurs peuvent passer beaucoup de leur temps d'écoute des événements individuels. Par conséquent, nous avons décidé d'ajouter des recommandations liées à des événements dans les pages d'événements individuels. Moteur de recommandation de l'événement estime chaque activité dans les trois dimensions: (i) titre de l'événement / Type / description (ii) le temps (iii) l'emplacement.

Principe de conception 6: Intégration dans les réseaux sociaux et des calendriers, au lieu de les remplacer. Dans Dig-Event, les événements sont intégrées aux sites de réseaux sociaux existants comme Facebook et Twitter. Cette intégration permet aux utilisateurs de poster des événements à ces services extérieurs. Ces messages contiennent des pointeurs de retour à la page de l'événement dans Dig-Event. Notez que les mashups gadget haut de l'agenda peut également être considéré comme léger intégration.

En outre, sur la base des principes de conception, le système a été mis en œuvre. Cadre du système et un scénario d'utilisation sont présentés en détail dans le Article VI.

Article VII, intitulé "Dig-Event: Let’s Socialize around Events" est un document de démonstration. Il présente les principes de conception et les caractéristiques principales du système, qui ont été inclus dans le Article VI.

Le système et les méthodes présentées ci-dessus, a été déposé dans la partie II-Unis en matière de brevets d'État, intitulée " Systems and Methods for Social-Event Based Sharing".

Le système de Dig-Event peut être consulté via le lien suivant:
http://myebsp.com/

Une vidéo de démonstration du système de Dig-Event peut être téléchargé via le lien suivant:
http://www.box.com/s/o4lrkzz6zffz4pzdvqt5

ABSTRACT

Web 2.0 is envisioned as an open garden for services and it puts people into the centre of the vision. The success of the user generated content raises the new idea of enabling end-users to create their own services. Enabling end-users to do service creation can reduce the time on marketing new services, harness the collective intelligence, and promote innovations in software development. This thesis contributes to the research and development of user-generated services from the following four aspects.

Firstly, since the user-generated service (UGS) is a new topic, the first contribution of the thesis is to provide a comprehensive survey of the state–of-the-art development in this area. By comparison with the concept of user generated content, we give the specific description of UGS, and then go through different technologies to analyse their challenges. The emerging trends of UGS with extensive new features are illustrated, from both technical and business perspectives. The underlying key technologies, as well as the current platforms/tools are also detailed.

Web mashup, is an integrated web application that combines data from multiple sources into a single interface, has been gaining wide popularity over the past few years. Within the web 2.0 scope, UGS is referred to as user-oriented mashups. Due to inherent programming difficulties required in integrating data and services from multiple sources, mashups have largely been a programmer's affair. Our second contribution lies in designing and implementing a simple mashup platform which provides the greater motivation for mashups usage. Followed by the literature review of overall studies on end-user's perspective on service mashups, we present a consumer mashup framework based on daily activities, by relating every-day calendar events to useful gadgets. The system, named EBSP (Event-Based Service Provider), offers a high level of abstraction to end users, exhibits context-orientation and personalization features, and focuses on functionality integration rather than data integration. The system was evaluated by 131 end-users to test for its usability. Also, as a representative example, the system is used in a proposed user-acceptance model for consumer mashups.

Our third contribution resides in the follow-up research of web widget discovery, which is one of the important flows in the EBSP design. We focus on the problem of using social annotation to support web widget discovering experience. An enhanced web gadget discovery model is proposed based on social tagging (Folksonomy), firstly by exploring the tag relations, and then by using three different gadget discovery schemes. Using the online service Widgetbox.com as an example, we demonstrate the accuracy and efficiency of our proposed model.

Fourthly, from the user acceptance model of the EBSP system we learned that sharing and collaboration in consumer mashups have a positive effect on performance expectancy. This inspired us to expand the EBSP system to social networks, which enable to share events and mashups. Followed by a survey of recent research on social network applications, we designed a novel system for activity-oriented online social networking named Dig-Event (Do-it-together Event). Dig-Event provides an open, social space for users to share events and discover the activities of mutual interests among social contacts like schoolmates, families, friends, and colleagues. It allows users to share their activities, to customize their social circles, to conduct events (by selecting activity-based gadgets), and to socialize around them. The features of event recommendation and integration with existing social networks further boost the event socializing experience.

Keywords: Web 2.0, Social Web, User-generated Service, Web mashups, Social annotation, Social networks.

LIST OF ORINGINAL ARTICLES

This thesis is based on the following original articles, which are referred to in the text by their Roman numerals (I–X):

Published Articles:

I. (2009) **Z. Zhao**, N. Laga and N. Crespi, 'The Incoming Trends of End-user Driven Service Creation', *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, Volume 21, pp. 98-108, 2009, Springer-Verlag.

II. (2009) **Z. Zhao**, N. Laga, N. Crespi, 'A Survey of User Generated Services', *IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2009)*, Beijing, China, Nov 6-8, 2009. pp. 241-246.

III. (2011) **Z. Zhao**, S. Bhattarai and N. Crespi, 'An Event-Based Functionality Integration Framework', *9th International Conference on Web Services (ICWS '11)*, Washington DC, USA. July 4-9, 2011. pp. 720-721.

IV. (2011) **Z. Zhao**, S. Bhattarai, J. Liu and N. Crespi, 'Mashup Services to Daily Activities – End-user Perspective in Designing a Consumer Mashups', *13th International Conference on Information Integration and Web-based Applications and Services (iiWAS '11)*, 5-7 December, 2011, Ho Chi Minh City, Vietnam.

V. (2010) S. Bhattarai, **Z. Zhao** and N. Crespi, 'Consumer Mashups: End-User Perspectives and Acceptance Model', *12th International Conference on Information Integration and Web-based Applications and Services (iiWAS' 10)*, Paris, France, Nov 8-10, 2010. pp 930-933.

VI. (2011) **Z. Zhao**, J. Liu and N. Crespi, 'The Design of Activity-Oriented Social Networking: Dig-Event', *13th International Conference on Information Integration and Web-based Applications and Services (iiWAS '11)*, 5-7 December, 2011, Ho Chi Minh City, Vietnam.

VII. (2012) **Z. Zhao**, J. Liu and N. Crespi, 'Dig-Event: Let's Socialize around Events', *2012 ACM Conference on Computer Supported Cooperative Work (CSCW 2012)*, 11-15 February, 2012, Seattle, Washington.

VIII. (2010) H. Zhang, **Z. Zhao**, S.Sivasothy, C. Huang and N. Crespi, 'Quality-Assured and Sociality-Enriched Multimedia Mobile Mashup', *EURASIP Journal on Wireless Communications and Networking*.

IX. (2011) **Z. Zhao**, N. Laga and N. Crespi, 'User-Centric Service Selection, Integration and Management through Daily Events', *2011 IEEE International Conference on Pervasive Computing and Communications (PERCOM '11 workshop)*, Seattle, USA, March 21-25, 2011. pp. 94-99.

X. (2012) P. Yao, Z. Hu, **Z. Zhao** and N. Crespi, 'A Study of Social Behavior in Collaborative User Generated Services', *ACM Conf on Ubiquitous Information Management and Communication (ICUIMC 2012)*, 20-22February, 2012, Kuala Lumpur, Malaysia.

Accepted Articles:

XI. (2012) **Z. Zhao**, N. Crespi and S. Bhattarai, 'The Design, Implementation and Usage Analysis of an Event-Based Consumer Mashups System', *International Journal of Web and Grid Services (IJWGS)*.

Submitted Articles:

XII. (2012) N. Laga, E. Berlin, I. Bedini, N. Crespi, B. Molina and **Z. Zhao**, 'A Flexible Service Selection for Executing Virtual Services', Second revision on *World Wide Web Journal (WWW)*.

XIII. (2012) (2012) **Z. Zhao** and N. Crespi, 'A System for Web Widget Discovery Using Semantic Distance between User Intent and Social Tags', *9th International Conference on Service Computing (SCC '12)*, Honolulu, Hawaii, USA, June 24-29, 2012.

LIST OF PATENTS

I. (2010) Z. Zhao and N. Crespi, Systems and Methods for Service Provision Based on Events. United States Patent Application: 12,909,898. Filed October 22, 2010.

II. (2011) Z. Zhao and N. Crespi, Systems and Methods for Social-Event Based Sharing. United States Patent Application: 13,310,877. Filed December 05, 2011

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background

The concept of end user development (EUD), refers to techniques that allow non-professional developers to create or modify software. Such a practice is observed to be in existence since the 1980's during which Programming by example [Halbert, 1984] and Programming by demonstration [Nevill-Manning, 1993] had been some popular approaches, which allowed end-users to create applications without conventional programming.

Over last several years, the emergence of Web2.0 and Service Oriented Architecture [Natis, 2003] have resulted in the evolution of richer, interactive web applications of greater functionalities, targeted not only for developers, but also for end-users of business enterprises, and of consumer domain intent in using the latest web applications for personal or situational use. The Web 2.0 platform has been characterized by greater interaction between users, harnessing collective intelligence, and trusting users as co-developers for creating cost effective services [O' Reilly, 2005].

User-generated service (UGS) is the phenomenon of creation of web applications by end-users, usually by combining existing web contents or applications. It refers to end-users participating "in the generation, combination and adaptation of services to create functionality and solve problems in their work". This creation of composite web applications is commonly also referred to as 'mashups'. Mashups are developed for various purposes; some are developer-centric and some are user-centric. The mashups that are oriented towards end-users contribute to generation of UGS, and end-user-driven service composition. End user driven recombination of web based data and functionality can be defined as mashups [Grammel and Storey, 2008].

The existence of mashups lies in SOA as a core component, while adding more value to this architecture. SOA is an architectural pattern that exposes web services for use and reuse. Ultimately, if we are just exposing services, it does not really offer a solution. But if we are able to abstract these services into applications or mashups, this demonstrates the real value of the architecture. Mashup technologies can help normal end-users combine services in a visual manner, and render a visual solutions/ applications which have a great amount of value [Ogrinz, 2009].

The creation of mashups and UGS is also a way to go one step further from the creation of User Generated Content (UGC), which has rapidly grown to become a major source of information and world-wide communication in the Web [Wunsch-Vincent and Vickery, 2007]. UGC practices have enabled end-users to share content or data (pictures, video, text etc.) in various platforms from social networks (Facebook, Myspace, etc.), video-sharing (YouTube), to wikis and blogs. After content-creation and sharing, a growing number of users are seen to be interested in combining existing content and services to create their own web applications [Braga et al., 2008], which is referred to as UGS or user-driven mashups. User surveys have shown that end-users do see the advantages in using mashup technologies for searching, integrating and sharing information [Zang and Rosson, 2008].

UGS applications can be designed for various purposes, ranging from ad hoc composition for a personal and situational use, to larger applications designed for business or enterprise use. The development method of a mashup may be manual or tool-assisted [Yu et al., 2008]. Manual methods of mashup development refer to the use of conventional Web programming technologies for service creation. This method requires good programming skills and intimate knowledge about the schemes and semantics of data sources or the business protocol conventions for message exchange. Technologies like Ajax, RESTful services and APIs, RSS and Atom microformats have made this process simpler. Nevertheless, manual mashup development is a challenging task even for skilled developers [Yu et al., 2008]. The development of a ‘Search engine’, by combining multiple existing search engines using their open APIs can be an example of manual mashup development.

To make mashup development simpler and to enable even non-experienced end-users to mash up their own web applications, a number of development tools and frameworks have emerged and such a method is known as tool-assisted mashup development. For example: a

user can use the Yahoo! Pipes tool to fetch favourite feeds from various sports websites (ESPN, BBC sports etc), sort and filter them (e.g. according to 'soccer') and get an aggregated result in a single page. Enabling end-users to do their own mashup speeds up the overall service creation process and solves situational problems for the users while exhibiting their creativity. These tools have different features and different composition approaches.

In the context of our work, the reference to UGS and the term 'mashup' will most generally imply end-user-created mashups that do not require any kind of programming, and whose development is normally supported by a platform.

1.2. Motivation and research question

There has been a large interest over the past few years in mashup technologies, and many mashup development platforms/tools/'mashup makers' [Grammel and Storey, 2008] have emerged at a fast pace. Despite this rapid interest in mashups, comprehensive mashup development tools are less than adequate, which, for most cases, creating a mashup application still requires a significant programming effort [Yu et al., 2008]. Such a method of mashup development requires the knowledge of advanced programming languages to integrate web APIs and feeds, implying considerable programming expertise [Zang and Rosson, 2008]. End-users who lack programming skills are not able to fully leverage the value and benefits of mashups which require the use of APIs, RESTful services, Atom and RSS feeds.

There are development platforms which are currently available in the web, like Yahoo! Pipes, Intel Mashmaker, IBM Mashup Center etc. which do not necessitate programming knowledge. However, not all of these tools are easy to understand and use, especially for normal end-users who lack the art of computational thinking [Zang, 2008]. Although studies have been made for development of mashup tools, few tools like Intel Mashmaker and Marmite [Ennals and Garofalakis, 2007, Wong and Hong, 2007] and little research endeavors have examined the needs of the less programming savvy end-users [Zang, 2008].

Due to the requirement of programming skills, and the mashup platforms which present many technicalities for assembling and creating services, a web-savvy yet a non-programmer end user will have *little motivation* for engaging in mashups which would

otherwise create an opportunity for the generation of value added services for the same user. To learn using mashup platforms, users must have enough motivation and enough perceived-usefulness.

Thus, the general research question in this thesis is “How to bring end-users closer to creating and using mashups?” From a user-cantered design perspective, this question is decomposed into the following questions:

- How do end-users perceive web mashups?
- How end-user perspectives help in designing a “useful” and “easy-to-use” consumer mashups?
- How end-users perceive our design? and their acceptance of consumer mashups in general?
- How to improve our design based on end-users’ feedback?

1.3. Scope and methodology

In order to solve the problems presented in Section 1.2, methodologies of design, implementation, and evaluation are utilized in this thesis:

Firstly, followed by a literature review of overall studies on end-user’s perspective on service mashups, we learn that there is the need for existing mashup platforms to offer solutions by bringing enough value to their existing activities and to meet their daily life needs [Hoyer et al, 2008], while offering greater simplicity and usability features [Yu et al., 2008], in order to achieve a greater user motivation for mashups use. There is a need to leverage users’ hobbies and interests [Zang, 2008] to engage them in service composition, which have reportedly [Zang and Rosson, 2008] been a major motivation for Internet usage. Based on these studies, we design and implement an easy-to-use, calendar-based, information mashup platform that motivates end-users for service composition, having a direct and positive influence on organizing better their personal and social life events. The system, named EBSP (Event-Based Service Provider), is evaluated by 131 end-users to test for its usability. A user acceptance model for consumer mashups is designed based on the UTAUT (Unified Theory of Acceptance and Use of Technology) [Venkatesh et al., 2003] which

proposes hypothesis for user acceptance analysis of the proposed and similar systems. Statistical analysis is performed though the data received from end-user survey.

Secondly, this thesis deals with the problem of using social annotation to support gadget/widget discovery experience, as a follow-up research of event-gadget matching in the EBSP design. An enhanced web widget discovery model is proposed based on social tagging (Folksonomy), by firstly exploring the tag relations using Kullback-Leibler (KL) Divergence metric, followed by three different widget discovery schemes. Using the online service Widgetbox.com as an example, we demonstrate the accuracy and efficiency of our proposed model.

Thirdly, from the user acceptance analysis of EBSP system we learned that sharing and collaboration in consumer mashups has a positive effect on performance expectancy. This inspires us to expand the EBSP system to social networks, to enable event and mashup sharing and collaboration. Followed by a survey of recent research on social network applications, we designed a novel activity-oriented online social networking system named Dig-Event (Do-it-together Event). This system has been inspired by previous research on calendaring and popular social network applications like Facebook and Google+. Comparing with current existing event-based social networks, six design principles have been proposed to design Dig-Event, and based on the design principles, the system has been implemented.

1.4. Contributions of the thesis

Regarding the aforementioned research problems, this thesis contributes from four different aspects. A brief overview of the contributions is provided below, while the detailed contributions with respect to each paper are presented in Chapter 3 where a summary of the research contributions of this thesis is listed.

The first contribution of this thesis is associated with the concept of user-generated service. In Paper I, the author illustrates the incoming trends of end-user service creation, and provides five main research topics from both technical and business perspective. Paper II clarifies the concept of user-generated service by comparing it with user-generated content,

and details the underlying key technologies of user-generated service as well as current platforms/tools.

The second contribution is regarding the design, implementation and usage analysis of an event-based service mashup system. Paper III proposes an event-based service mashup framework to approach the issue of service personalization. Paper IV introduces a user-centered mashup system, named event-based service provider (EBSP), the design principles, usage scenario, and implementation details are presented in detail. In Paper V, the usability test is performed on the EBSP system, and a user acceptance model for consumer mashups is presented, to precisely identify what factors lead to their adoption and to what extent. Journal Paper XI, is a complete and comprehensive version of the papers III, IV and V. The system and methods presented above, has been filed in the United State Patent I: "System and Methods for Service Provision Based on Events".

The third contribution is related to enhance the matching between user-created event and system recommended gadgets in the EBSP system. Paper XIII presents a method of widget/gadget discovery through social tagging. The detailed widget discovery method is classified into two stages: tag discovery and widget discovery. We further set up a system to test the methods. Finally, a usability test is performed by comparing our results with keyword matching results to prove the accuracy and efficiency.

The fourth contribution put the EBSP system a step further in the context of social interaction by designing an online social network around activities. Paper VI presents the extension version of EBSP: an event-based online social network, named Do-it-together Event (Dig-Event), which is designed to meet the requirements of users' collaboration requirement in mashups. Comparing with current existing event-based social networks, six design principles have been proposed to design Dig-Event. The design principles, usage scenario and implementation details are presented in detail. Furthermore, the system and methods presented above, has been filed in the United State Patent II, named "Systems and Methods for Social-Event Based Sharing".

The contributions of the thesis including research questions and papers are summarized in Figure 1.1.

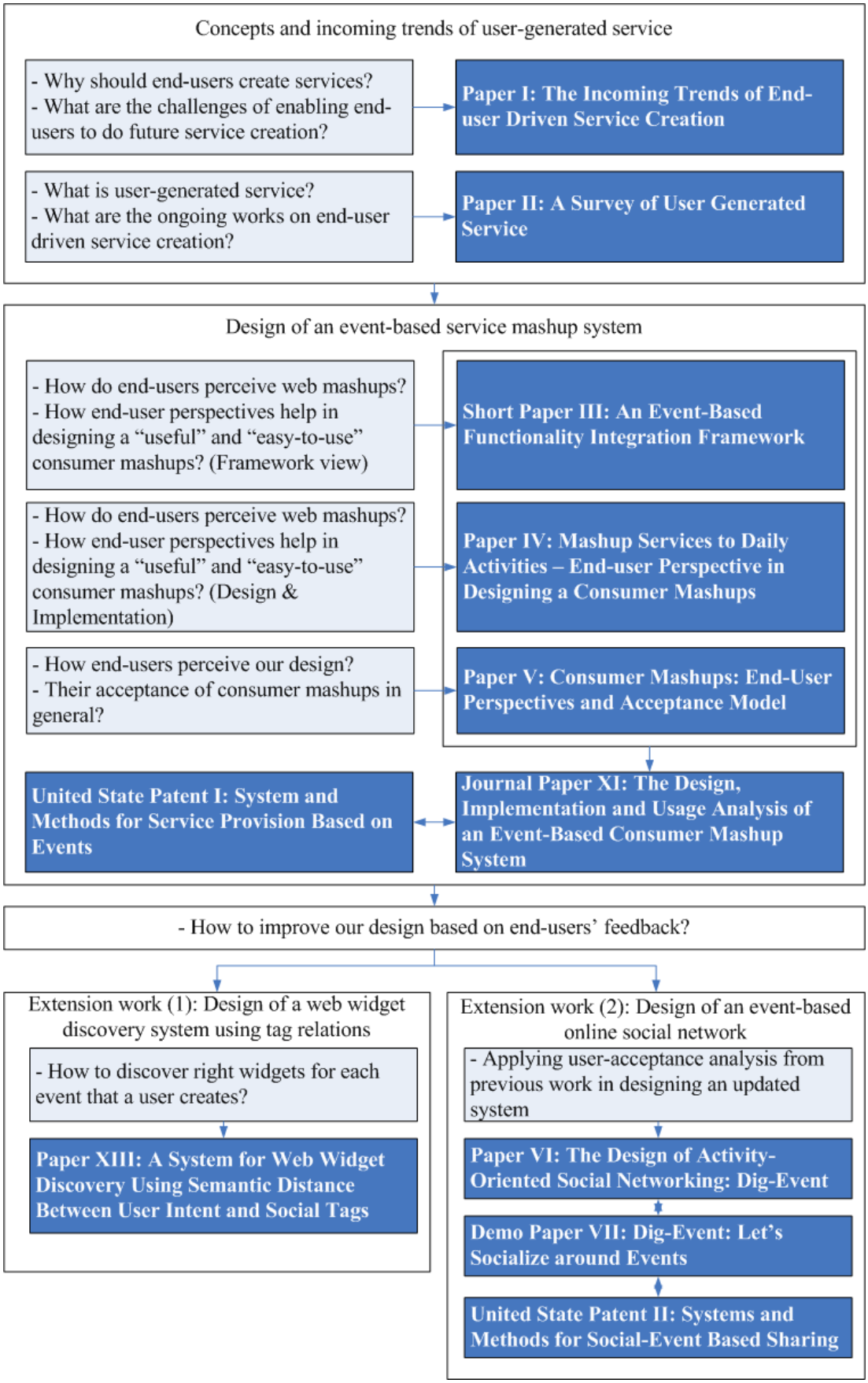

Figure 1.1 Contributions of the thesis

1.5. Organization of the thesis

The organization of the thesis is as follows:

In this chapter, the background of the research topics, motivation and research problems, scope and methodology, as well as a brief overview of the contributions of this thesis are shortly discussed.

Chapter 2 presents a literature overview of the related research topics of this thesis, including introduction of web mashups (definition, classification, examples, architecture), end-users' perspective on mashups, user acceptance model, and related works on social annotation.

Chapter 3 summarizes the main contributions of the original papers which are made up of four parts: concept of user-generated service and its incoming trends; a design, implementation and usage analysis of an event-based service mashup system; a design, implementation and evaluation of a web widget discovery system through social tagging experience; and finally, a design and implementation of an event-based online social network.

Chapter 4 draws a conclusion of the thesis and presents the further work.

The Appendix section contains the questionnaire used in the EBSP user survey.

2. LITERATURE REVIEW

2.1. Introduction to web mashups

The article "*Services Mashups - The New Generation of Web Applications*" [Benslimane et al., 2008] have introduced the recently emerged concept of mashup in the context of Web applications. Also, a 'mashup' has been defined to be a way to create new web applications by combining existing web resources, utilizing data and web APIs. The purpose of developing a mashup is to share information, aggregate information and publish this content as a new generation of Web applications. The paper focuses on Service mashups, which aim at designing and developing novel and modern Web applications based on easy-to-accomplish end-user service compositions. Various mashup tools and platforms have been developed by researchers, which let developers and end-users access and compose various data/services from Web applications. Tools like IBM's QEDWiki, Yahoo Pipes, Google Mashup Editor, and Microsoft's Popfly are some well-known examples of mashup platforms that users can adopt.

The article "*Mashing up Search Services*" [Braga et al., 2008] have shed light on the concept of user created contents and user created services. During the past few years, end-users have been increasingly involved in content-creation process of modern Web applications (e.g. Wikipedia) and in the emergence of social Web applications, such as MySpace or YouTube. End-users are playing an important role for the application providers, since these applications are entirely run by collective user generated content. Now, a growing user community is being interested to harness the benefits of existing content and services on the Web, by developing its own web applications. This phenomenon is known as "Web mashups" and it is adopted largely by skilled users because reusing third party content is a non-trivial task. Service mashups, is termed to be user-driven Web service integration approach, and it is gaining importance because it is targeting even the less skilled users. The burden to compose the service entirely has been alleviated by this idea, but end-users will still need to manage the logical connections and service flows in the mashup environment.

The article "*Understanding Mashup Development*" [Yu et al., 2008], have given an introduction of what mashups are, and have contrasted them to tradition integration efforts. Web mashups are defined to be Web applications, generated by combining content, presentation, or application functionality from disparate sources. The article gives novice readers a clear understanding about mashup development, by clearly categorizing two different approaches to mashup development, namely Manual Mashup Development and Tool-Assisted Mashup Development. Manual mashup development is not a trivial task; it requires good knowledge of programming and is still considered to be a prerogative of skilled developers. In order to enable even inexperienced end-users to mash up their own Web applications, there are available a number of mashup-specific development tools. The ones presented on this article are, Yahoo Pipes, Google Mashup Editor (GME), Microsoft Popfly, Intel Mash Maker, and Quick and Easily Done Wiki (QedWiki).

The authors point out the need to bridge user interface integration (which is user oriented and event driven) with the more traditional forms of integration (which are enterprise oriented and more orchestrational). The article is seen to stress on the need of having simple and easy-to-use mashups for all users. "Mashups are about simplicity, usability, and ease of access. This simplicity has the upper hand over feature completeness or full extensibility".

1.1.1. Mashup classification

Mashups can be classified according to different conceptual bases (see Figure 2.1). The first one is "Client Mashup" and "Enterprise Mashup". Client Mashups are usually created for personal use for situational problem solving. Two examples of client mashups are: 1) A service that retrieves several 'friends' from Facebook and plots their picture in a Google Map, based on their location; and 2) A service that fetches RSS feeds from several websites, like BBC Sports, CNN Sports and ESPN, combines them, filters them according to some keyword (e.g. 'tennis'), sorts them according to date (e.g. latest first), and displays the filtered result on a single page. Enterprise Mashups, on the other hand, are developed for problem solving in the business domain, and require greater collaboration among a number of people to carry out business processes in a coordinated manner. A system for approving expense reports in a company would be one example of an enterprise mashup. This system can be a mashup of additional services/resources like real-time charts and reports on business

intelligence (BI) which indicate how much budget is left in the company, helping to make decisions such as if approving the indicated amount would create an over-budget situation, or any other problem. The analogous terms “Consumer Mashup” and “Business Mashup” are often used to describe Client and Enterprise mashups, respectively.

Another approach for mashup classifications is: “Web Page Customization Mashup” and “Process Mashup”. Web page customization mashups are used to change websites by removing elements, adding additional widgets and changing their user interfaces (UIs). Process mashups allow for the aggregation of data, content and services, and include them in automated sequential processes [Grammel and Storey, 2008].

A similar way to categorize mashups is as “Front-end” and “Back-end” mashups. Front-end mashups help to build web front ends, such as dashboards, using widgets/gadgets and little to no programming (iGoogle, Netvibes, PageFlakes). Back-end mashups combine web-accessible data and services into more useful web services that can be called easily using a RESTful interface (Kapow, Yahoo Pipes).

Another similar terminology for mashup classification is “Horizontal Mashup” and “Vertical Mashup”. Horizontal mashups can be seen as a process of grouping sets of typically similar or complementary services to aggregate their outputs, in which there is no interaction between service modules, but where the customizable front-end offers more value to end-users who wish to solve a particular task. A vertical mashup, on the other hand, is a process of orchestrating the outputs from some services into the inputs of others, where service modules are connected together and the parameters are passed between the modules to produce a new enhanced service.

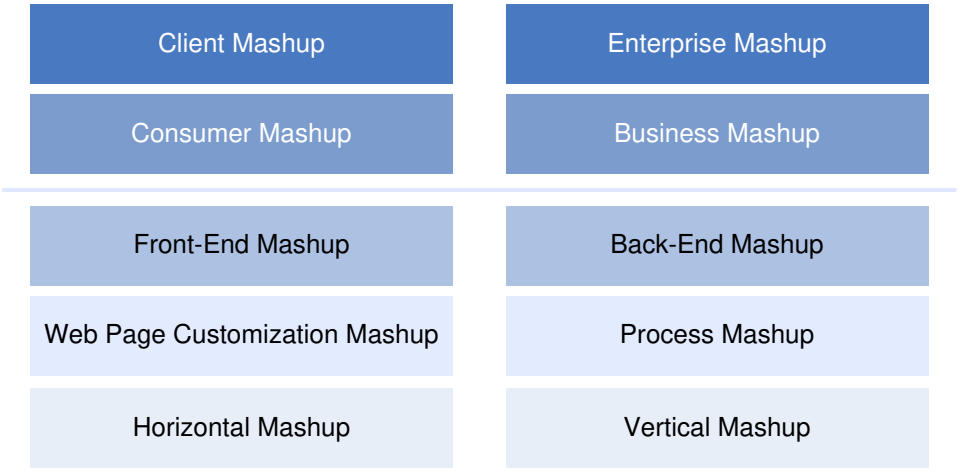

Figure 2.1 Mashup classification - similarities and differences

1.1.2. Examples of mashups and mashup platforms

In the article "MashMaker: Mashups for the Masses", [Ennals and Garofalakis, 2007], readers are given an introduction of the MashMaker Tool. MashMaker is basically a browser plug-in and an interactive tool for editing, querying, manipulating and visualizing "live" semi-structured data. MashMaker draws on the design principles of more "natural" software tools (like Web browsers and spreadsheets) and simple scripting languages, rather than formal database models, schemas and queries. The goal of MashMaker, as described in the paper, is to allow non-expert users to easily create their own mashups based on data and queries produced by other users and by remote sites. Non expert users are often interested in finding a lot of information, which they would also like to manipulate.

This paper presents an example of mashup development in the MashMaker Platform with a scenario. A user is planning to rent a house, and wants to know which of the houses available has the best restaurants nearby. First, the user goes to a housing Website and clicks the "Add to MashMaker" icon on their browser bookmark bar. Then MashMaker starts up and lists the houses mentioned in the Website in a tree-structure, with each house as a node. When the user clicks at any house node, the MashMaker automatically suggests the user to apply the 'Things nearby' widget. When user accepts and clicks on this widget, its property can be set to 'food'. So food which is nearby the location of the house is searched (i.e. restaurants). Further, the user will be suggested the 'filter' widget, to find restaurants within 0.5 miles of the house. By a similar automatic recommendation process from the system, the user will find specific information of his/her interest. The user might decide that this composition is very

useful, so he/she can save it as a widget, which can be used in future. If the user decides to publish it, the entire widget will be available to others using the same platform.

An article entitled "*Doing the Mobile Mash*" [Trevor, 2008], is about Yahoo Pipes going mobile on the iPhone. It provides readers an introduction of Yahoo Pipes, which is a service platform for processing well-structured data formats (XML, RSS etc). Developers of all capabilities can use Pipes' Web-based visual programming environment to combine data and create mashups, without having to write any code. Yahoo Pipes has the provision of feed aggregation and filtering, allowing users to get content and filter it with keywords. Also, it provides data transformation and geocoding the contents, and using REST APIs for designing complex mashups. With Apple's iPhone, Yahoo! Developers have found an ideal platform to move the Pipes mashup from the desktop to the mobile domain. The iPhone needs an interface (that can be downloaded from the Pipes Website) that is consistent with the local applications running in the device's browser.

The article "*Providing Personalized Mashups Within the Context of Existing Web Applications*" [Diaz et al, 2007] have coined the term "mashup personalization" to describe the approach of using mashup techniques for the end-user to enrich the content of existing web applications. A proof-of-concept framework is introduced through the tool MARGMASH (MARGinal MASHups). Personalization is defined as the process of tailoring pages to individual users' characteristics or preferences that will be meaningful to their goals. However, it is also true that no design can provide information for every situation, and no designer can include personalized information for every user [Rhodes, 2000, Diaz et al., 2007]. Hence it is suggested that traditional approaches be complemented by mechanisms that allow end-users to add their own content once the application is deployed. This concept is related and complemented by the "do-it-yourself" (DiY) principle brought by mashups by allowing the layman to combine existing data from disparate sources in innovative ways.

MARGMASH is a tool which works as a component in the Web browser which is the window to Web applications. The tool is used to add more value and functionality to the page that we view in the browser, by mashing it up. The working of MARGMASH is now explained with an example.

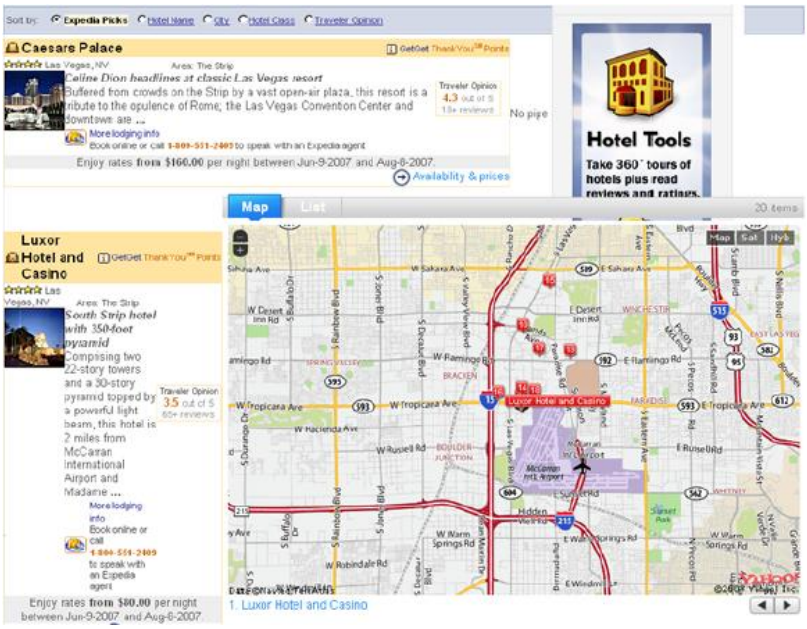

Figure 2.2 Original website with its "margmashed" counterpart

First the user has to log into the MARGMASH framework and give the URL of an existing Web application or Website, for example "www.expedia.com", which offers a trip-planning service. After opening the Website, first, the user needs to provide visual clues on the selected page, i.e. to make annotations on the selected page that will help to identify and classify that page. To do this, the eligible markups on the page will be highlighted by MARGMASH, and the user has to select them (done through dialog boxes). In the next step, the user identifies which markup fragments play the role of the mashup anchors. Mashup anchors are used for placing the mashup output. Also, they provide some feed data for the associated mashup. In this manner, framed by MARGMASH, the user browses along Expedia till a page is reached where a mashup needs to be posted (see Figure 2.2). This is indicated by clicking on the "Margmash It!" button.

The paper entitled "*Making Mashups with Marmite: Towards end-User Programming for the Web*" [Wong and Hong, 2007], has also focused on the need to have tools that simplify mashup creation for end-users. To address the need of allowing end-users to mashup services easily, the authors have developed an end-user programming tool called Marmite, which lets end-users create mashups that combine existing Web content and services. The paper presents the design, implementation and evaluation of the tool Marmite.

Marmite is presented as a tool which lets everyday users create mashups easily. It is currently implemented as a Firefox plug-in. In Marmite, data is processed in a flow, by a series of operators. An example scenario is presented, that lets users find all addresses in a Web page or pages, keep only the addresses of Pennsylvania, and then plot these addresses into Yahoo Maps (see Figure 2.3).

In the Marmite platform, on the left panel, there are a set of operators that users can choose to extract and process the data/contents from Web pages. On the middle, there is the flow of data, by using several operators, and on the right side, the current values of data (intermediate, or final result) are presented.

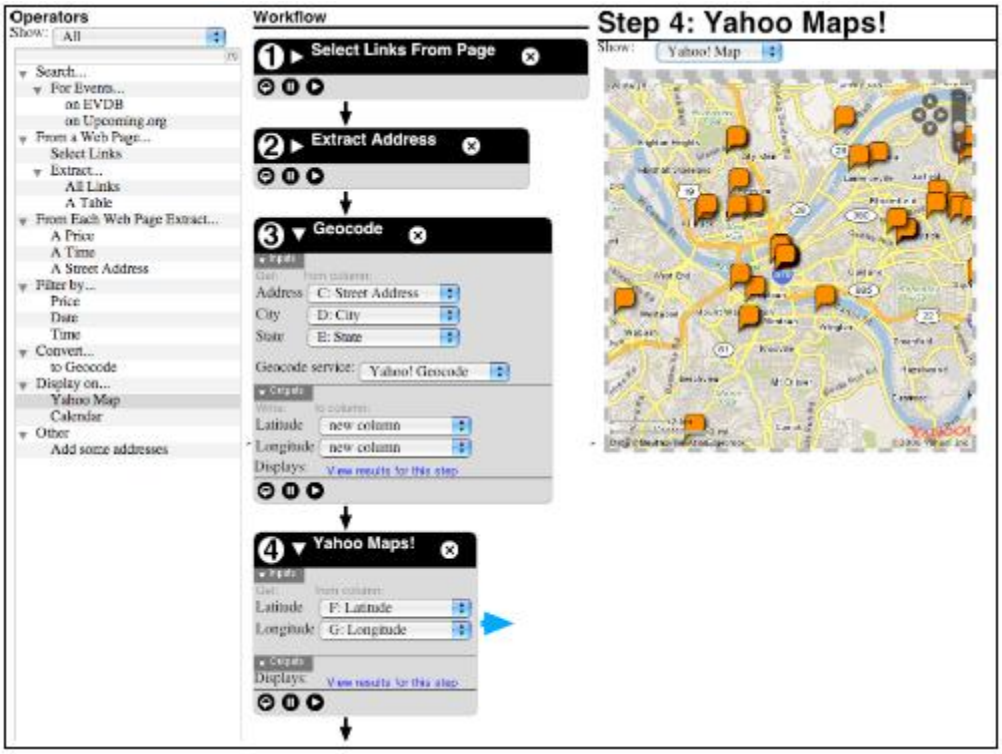

Figure 2.3 The Marmite environment.

The tool Marmite is valuable, in the sense that it has been developed keeping the ultimate end-users in mind. The design has been motivated after usability studies of current tools in the Web. Usability tests of Marmite have also been done among a limited user group, and 50% of them were able to do a given task with ease, 66% of them belonging to the programmers group. Marmite developers are now looking for ways to make the environment simpler for non-programmers, to expand the set of operators and add more functionality and usability in the system.

In the article “Professional and End-User-Driven Service Creation in the SPICE platform” [Droegehorn et al., 2008], authors describe the SPICE platform. The platform is said to provide an environment for future converged services. The platform addresses the needs of professional as well as end-users. Using this rich service platform, professional developers can build services combining the Internet and Telecom service enablers (Web and Telco convergence). End-users can use the ‘end-user studio’ to express their creativity, and their needs of the real-world situation. The End User Studio has a graphical interface to build composed services. To create a service, users need to drag and drop the required building blocks, and use the mouse to draw connections between the blocks. This doesn’t require any kind of knowledge about programming, and hence encourages end-users towards their own service creation.

In the article “*A User-Centric Service Creation Approach for Next Generation Networks*”, [Yelmo et al., 2008], authors have introduced the OPUCE platform. The purpose of the paper is to address the relatively unclear service provisioning approach to the architecture and technologies of the Next Generation Networks which are now considered well known. In doing so, the main focus in on providing an environment for user-centric service creation, which would improve the service offering of NGNs with profitable and value added services, faster and cheaper. This approach is directed towards the convergence of Web and telecom networks, in a user-centric manner. OPUCE platform enables a highly intuitive service provisioning environment, from the point of view of end user, with no special technical knowledge. In this platform, a palette of several building blocks, also known as base services are present, which expose basic functionalities, for example, send_SMS or initiate_call. Users can select the appropriate blocks, and link them in order to generate a workflow of event-actions. For example, “When call is terminated, send SMS”. This is an example of an event-based mashup, in which one event can take place only after a predefined event has taken place.

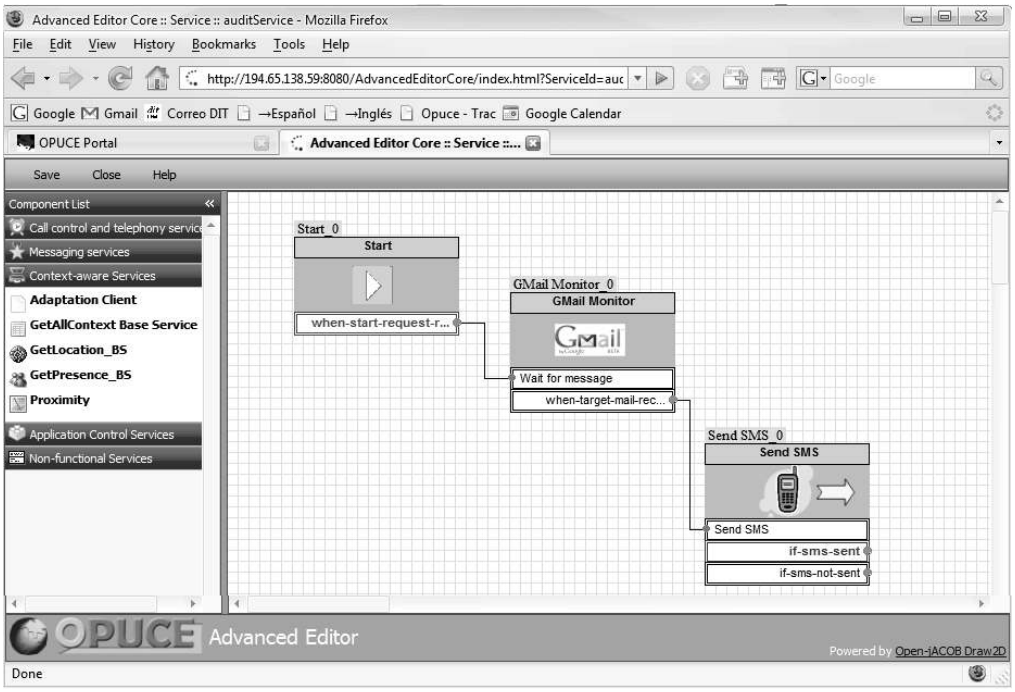

Figure 2.4 Messaging reachability service in the OPUCE environment.

One example of composite service creation in the OPUCE environment, named Messaging Reachability service is presented. Using this service, users can receive their emails by traditional means (using an email service, e.g. Gmail), but when they are unavailable to read it, (e.g. while driving a car), they can configure the system to automatically convert the text (email) into speech, and have it delivered by a voice call to their mobile phone. Alternatively, users can have the text (email) delivered to their phone via SMS. Figure 2.4 presents the OPUCE environment for creation of this service.

The paper "*Mashing-up integrated location, social networks and recommendations: the POI Radar*" [Valla et al., 2009] has illustrated the POI (Point of Interest) Radar Service, which is a mobile mashup of integrated location, social network and user's recommendation service. The purpose of the mashup is to provide notifications to the mobile phone user, about nearby points of interest (e.g. museums in a city), as voted by the user's contacts (social community), and based on the user's context, while he/she is on the move. This concept is part of a more general eTourism platform that the researchers are building to provide Web2.0 services to tourists on their mobile phone.

Whenever the user is in a new city, using this service, the user can set his/her preferred POIs and can authorize a trusted group of people within his/her social community to give

recommendations about the preferred POIs. Additionally, this mechanism is context based, with the location and social network being the context providers. Such a system helps the mobile user/tourist to navigate the city more easily, giving more time to enjoy his/her preferences, e.g. restaurants, museum, churches etc. Such POIs are actually provided to the system by a POI component (eg. the local tourist bureau). This is a powerful mashup that has been devised by the developers. To use this service, the end user needs to have this service in his/her mobile phone and vote for a trusted group of friends within his/her social community to get recommendations.

1.1.3. Architecture of mashup platforms

1.1.3.1 General architecture

The article "*Towards Service Composition Based on Mashup*" [Liu et al., 2007], has expressed that generally, mashup is usually done at the Web browser, by drag and drop applications from different sources together. There must, however be some backend infrastructure to support mashup. Mashup application is architecturally comprised of three different participants: API/content providers, the mashup hosting site, and the consumer's Web browser [Merill, 2006] [Liu et al., 2007]. This is similar to the popular three-tier architecture shown in Figure 2.5.

The API/content providers are the providers of the content being mashed up. To facilitate data retrieval, providers often expose their content through Web protocols such as REST, Web services (SOAP, WSDL, UDDI) and RSS/Atom. However, several data sources, which do have the potential of being excellent data sources do not conveniently expose their APIs yet. Mashups that extract content from sites like Wikipedia, TV Guide, and virtually all public domain Web sites do so by a technique known as screen scraping [Merill, 2006, cited in Liu et al., 2007].

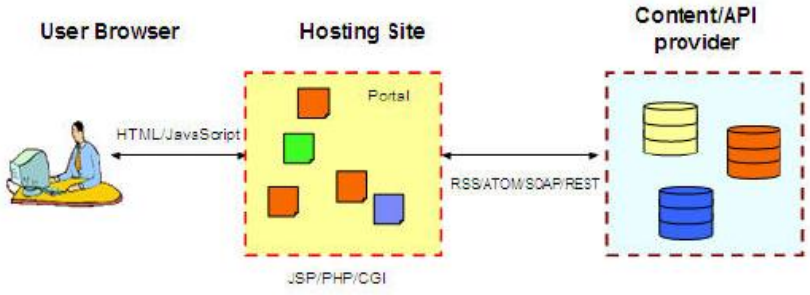

Figure 2.5 Mashup: Basic Architecture.

On the second tier, we have the mashup hosting site. It is where the mashup logic resides, but not necessarily where it is executed. Mashups can be implemented either in the server side, similar to traditional Web applications, or on the client side. Server-side dynamic content generation technologies include Java servlets, PHP or ASP. The mashed content can also be generated directly within the client's browser through client-side scripting (e.g. using JavaScript) or applets.

On the third tier, there is the consumer's Web browser. The browser is where the application is rendered graphically and where user interaction takes place. Mashups often use client-side logic to assemble and compose the mashed content.

A mashup component architecture for enterprise use [Lizcano et al., 2008], have been presented and have been called the Enterprise 2.0 Mashup Stack (see Figure 2.6). Hoyer et al., 2008 have also referred to this stack in describing about design principles of enterprise mashups architecture to cover long tail of user's needs. Although this architecture has been defined for enterprise mashups, it provides the functional component architecture valid for consumer information mashups as well.

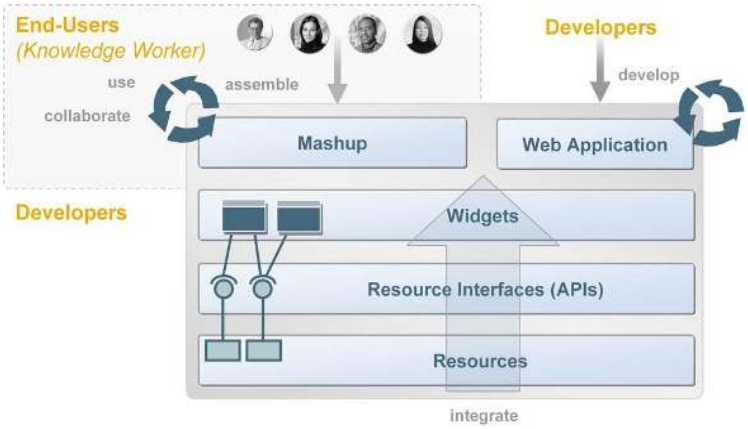

Figure 2.6 Enterprise Mashup Stack.

The bottom layer of the stack (see Figure 2.6) consists of the actual Web resources (content, data or application functionality), which are the core building blocks of the mashups. Each resource is identified by a Universal Resource Identifier (URI) in the Representational State Transfer (REST) architecture style, giving easy accessibility to browsers, mobile devices etc. To source or access the resources, well defined interfaces known as Application

Programming Interfaces (APIs) are used. APIs encapsulate actual implementation separate from the specification and allow loose coupling of existing web based resources. The REST architectural style represents the CRUD operations (Create, Read, Update and Delete) by the HTTP verbs Put, Get, Post and Delete. The Atom Publishing Protocol (APP) allows for publishing and editing web resources following the REST architectural style. A Google driven initiative, GData (which has also been used in the current work, to interface with Google calendar functionalities) uses the APP extension mechanism and also provides queries and authentication functionalities.

Gadgets, also known as widgets, are mini applications which provide a graphical, simple and efficient means of user interaction with the web resources, abstracting the technical description from the functionality. Gadgets provide application domain functions or information-specific functions. They can be visual (rendering visual content) or non-visual (back end functionality, e.g., accessing a web based resource), and are extremely flexible, customizable, and reusable. Until now, there is not a widely used model for gadgets, although the W3C has published a draft specification.

The aggregation, assembling, or grouping of several gadgets stored in the gadget repository will allow end-users to create a composite application such as a mashup. The creation of a mashup is done in a visually intuitive manner, aggregating and linking component from different resources. The actual application (mashup) will be a workspace created by the users themselves to suit their individual need, without skills being required for programming.

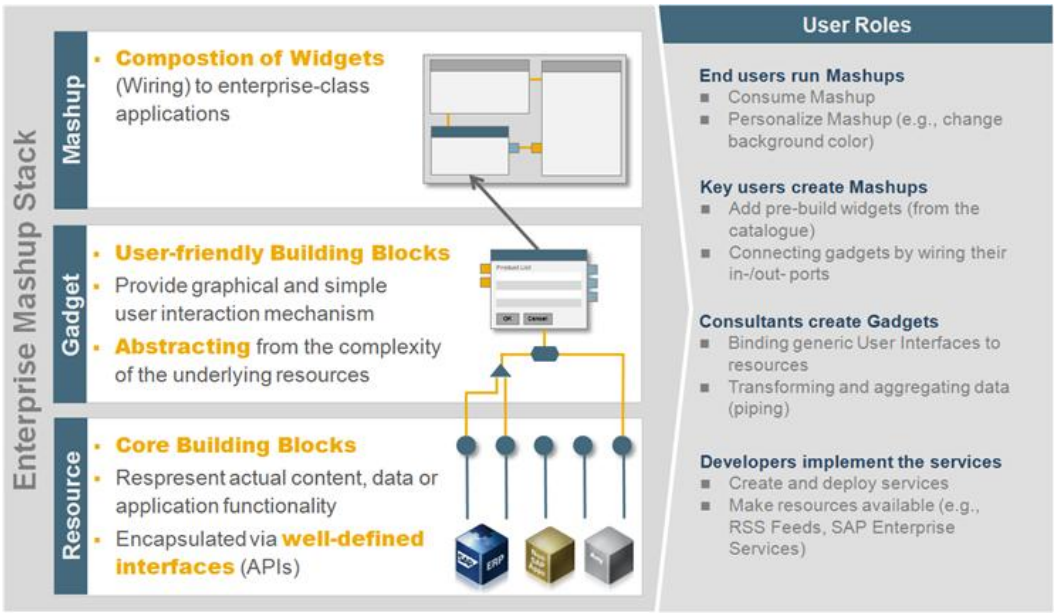

Figure 2.7 Enterprise Mashup Stack and User Roles.

Figure 2.7 shows the enterprise mashup stack along with user roles, such as developers, consultants, key users and end-users. Developers create and deploy services, and are the ones who actually implement the services. Consultants create gadgets, by binding generic user interface to resources. Key-users create mashups, while end-users are believed to consume created mashups. Mashup platforms should be intuitive and simple enough to use, such that the differences in the roles between key users and end-users are made narrower.

Hoyer et al., 2008 regard such user-centric approach for rapid development and maintenance of applications as an attempt to reach the phenomenon termed as 'Web2.0 Long Tail', which refers to the need of Web2.0 applications to change easily and quickly to meet daily needs, and the changing needs of people, requiring individualized, ad hoc applications. Addressing The Long Tail of user needs becomes possible only when the resource composition style is lightweight [Lizcano et al., 2008]. This style reuses building blocks from different contexts to build individual personalized applications. The composition will take place both in the resource layer (which is called 'piping') and in the gadget layer (which is called 'wiring'), as shown in Figure 2.8.

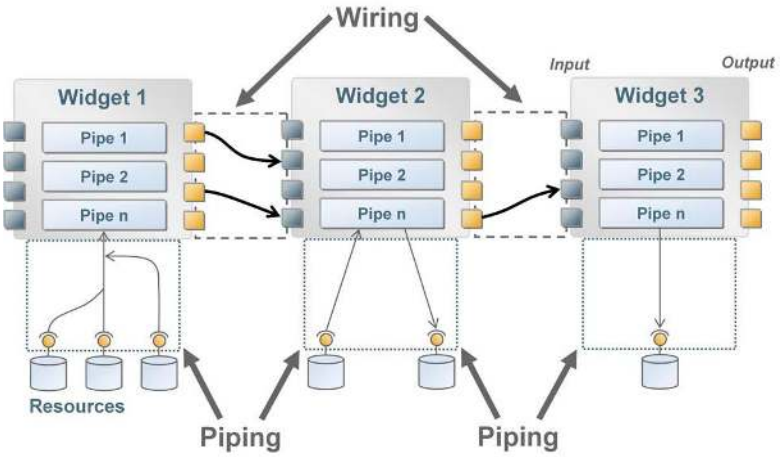

Figure 2.8 Piping Vs. Wiring.

The piping composition integrates a number of heterogeneous web based resources by defining data chains/graphs, concatenating the resources. In piping composition, the output of each process will feed as direct input to the next one. The processes involve aggregating, transforming, filtering, sorting, mixing and manipulating the content, data and application functionality of the resources.

Wiring is the process of an end user visually connecting the input and output parameters of existing gadgets, and enabling intercommunication between them. Wiring is done in the gadget layer, or presentation layer. This top-down approach of software development enables end-users to identify a need, find gadgets to meet that need, and wire the gadgets together to create a composite web application on their own.

1.1.3.2 Platform specific architectures

Braga et al., 2008, have an article about "*Mashing up Search Services*" .The article relates to creating a powerful search service, which can handle multidomain queries, and highlights the issues which make the composition of 'search' services different from other services Multidomain queries, (for example: finding a particular conference to be held within a particular time in locations having a particular weather, and reachable via a low-cost flight from a particular place) can rarely provide accurate results when using general-purpose search engines like Google or Yahoo, within a single usage. This article presents a graphical user-interface for the design of a powerful search service and also gives the functional architecture of the platform (see Figure 2.9).

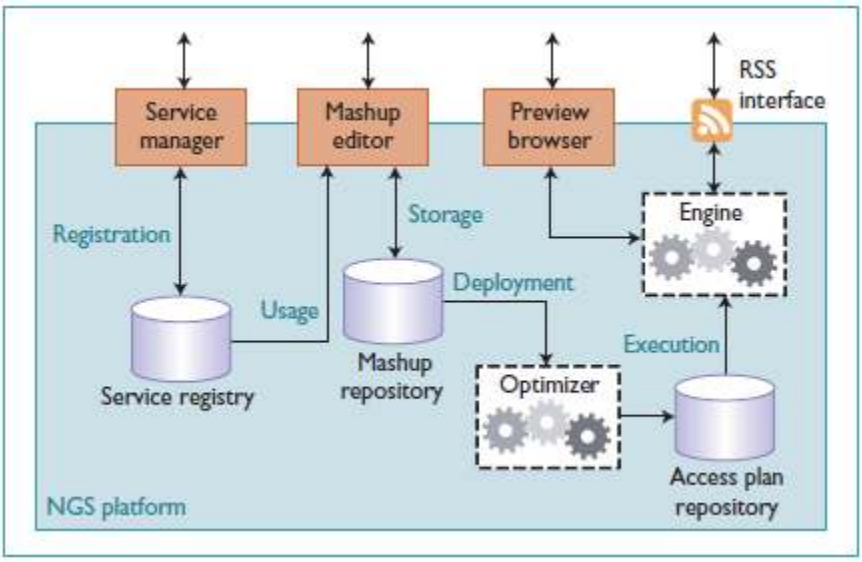

Figure 2.9 Functional platform architecture of the search service

The architecture of Figure 2.9 presents several functional blocks like service registry, service manager, mashup repository, engine etc. Service registry is a database of services registered in the platform, and where users can choose from, to compose any other service. The Ajax-based mashup editor lets user mash up the services and the result is stored in the mashup repository. The optimizer translates the mashed up query into an executable plan; the engine executes the plan by invoking the respective services. The user previews the result or consumes it via the RSS interface.

In the article "A User-Centric Service Creation Approach for Next Generation Networks", [Yelmo et al., 2008], the authors have presented a platform, referred to as the OPUCE. This is a research project that has applied the user-centric approach to the creative combination of Web and network services over Next Generation Networks. The architecture of the platform is presented, which consists of several dedicated modules (Figure 2.10).

Referring to Figure 2.10 the 'Portal' is an integration of the GUIs of different modules through which end-users and administrators can perform management and service creation tasks. Based on the role of different users, the portal can be roughly divided into Service Portal (for users who are service creators), User Portal (consumers of the OPUCE services), and Admin Portal (administrators). 'User Information Manager' is in charge of storing information about the user (profile, context etc). 'Service Lifecycle Management' controls the entire lifecycle of all services, and must deal with a great amount of services that are

continuously being deployed and withdrawn in this environment. 'Service Advertising' recommends services to end-users based on two ways: the first is through explicit user subscription to service categories or keywords, and secondly through intelligent matching of user profiles with service descriptions. 'Service Repository' stores the description of services already present in the platform, plus the services created by end-users. 'Basic Service Editor' enables service composition in handheld devices like PDAs as well. For end-users (consumers), 'Service Execution Environment' can be regarded as an environment where services are actually composed and executed. For service creators, it can be regarded as an environment which is designed to offer a common model and a consistent runtime environment for developing and deploying network services. The network resources are exposed using Web services interfaces, which allows for a loose coupling and easy composition. 'Context Awareness' allows the dynamic adaptation of services according to the information retrieved from the profiles within the 'User Information Manager'.

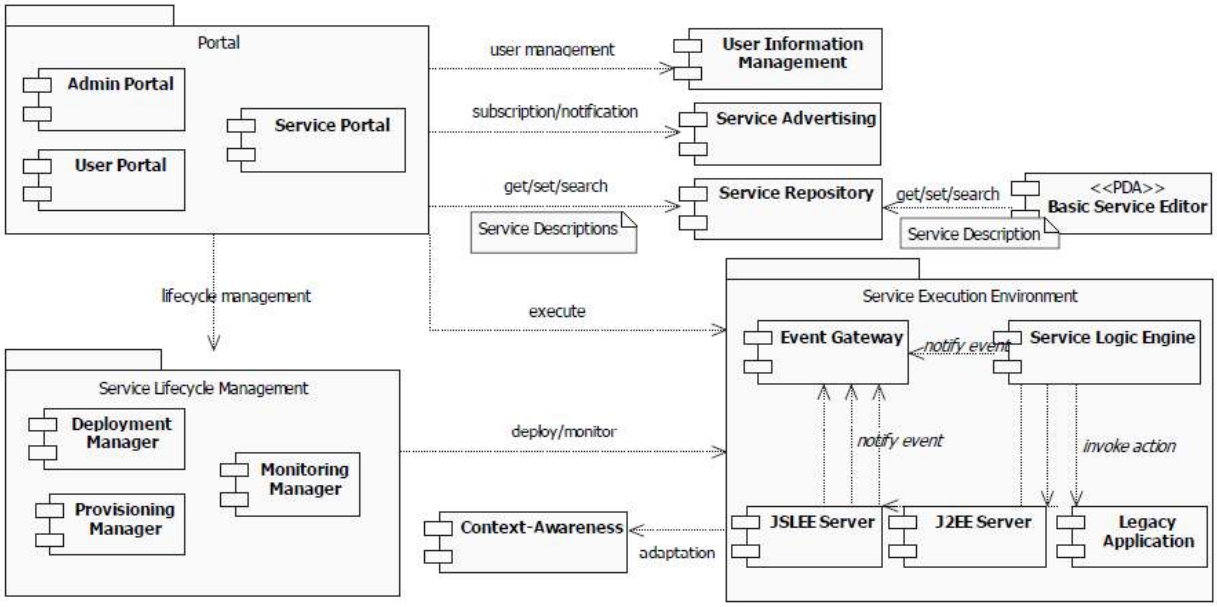

Figure 2.10 Architecture of the OPUCE platform

The OPUCE is a powerful platform having many functional modules and value added features like context awareness and intelligence. Though this has been presented as a scenario of NGN networks, the architecture can be a useful reference for any UGS platform in the Web2.0 environment.

In the article “Service Creation in the SPICE Service Platform”, [Almeida et al., 2008], the focus is on the effective design aspects of a Service Creation Environment (SCE) for easy development of services over heterogeneous platforms. To describe SPICE services, a specialized description language named SPATEL has been designed. The platform is designed for developers, as well as end-users, who make use of the developer studio, and end-user studio respectively, to compose services.

In the End-user studio (see Figure 2.11), the end user will be creator, who can specify his/her needs and create new services by a process which will be much system-assisted. The service will be the result of the assembly of pre-existing components in the system.

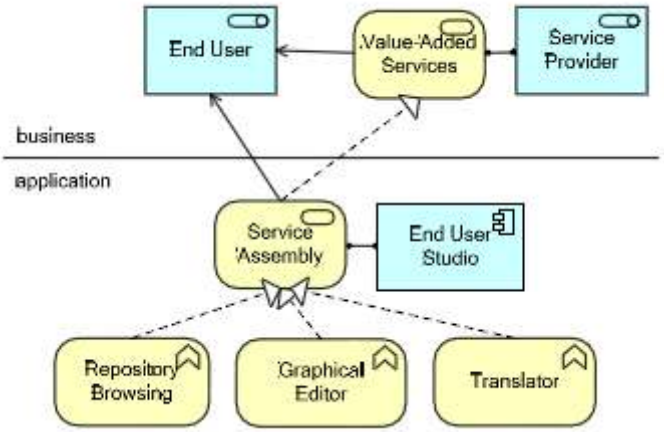

Figure 2.11 End User Studio core components and their relations

The end-user studio targets service composition through component reuse, and this process is supported by different tools with different degrees of friendliness, ranging from natural language interpreters, wizard based rule editors and graphical component assembly. Also, the studio is provided with translators to allow service definitions to be transformed to the SPATEL developer language, to allow reuse of functionalities, and to focus the purpose of the studio on service assembly by end-users.

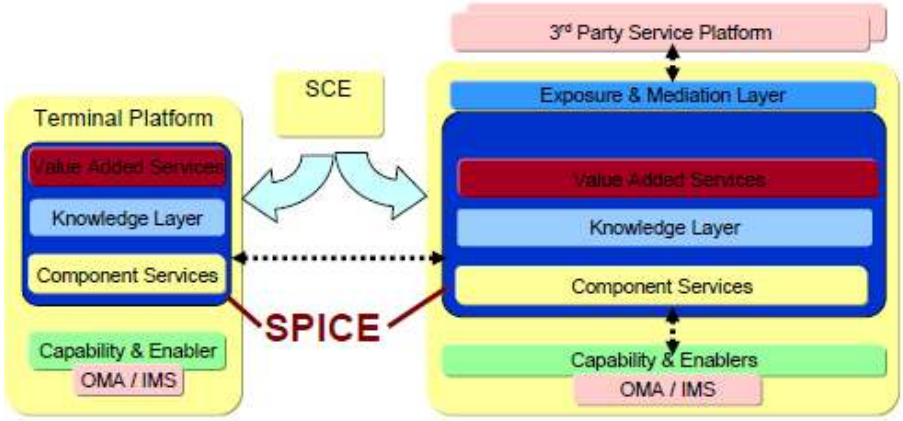

Figure 2.12 Layers in the SPICE Architecture

Also, the article presents overall, layered platform architecture of the SPICE platform (see Figure 2.12). The components are logically divided into "Component Services", the "Knowledge Layer" and the "Value-Added Services" layer, based on their offered interfaces. Service compositions are usually deployed by the service creation environment in the VAS layer. The basic components offer access to the underlying capabilities and enablers of the network, like access to IMS, or OMA enablers like presence, IM etc, the SPICE platform having been developed for the convergence of Telco resources with the web world, and service creation over heterogeneous platforms.

2.2. End-users' perspective on mashups

Mashups combine information and application functionality from different sources, and have been getting considerable attention in the past years. However, the creation of a mashup may not always be so easy from the point of view of end-users. Several, but not abundant research have been made for end user requirements in mashups, and the acceptance of mashups.

Zang, 2008 in the article "*Mashups for the Web-active user*", point out the fact that the currently available mashup tools do not adequately support the technical knowhow of end-users. The author presents the term "Web-active user" to define users who are considered to be active online. They are users who use internet on a daily basis, and who try to find out new ways to integrate their online activities. As such, they are the ones willing to make new

initiatives with technology, although they don't have the programming expertise to create mashups. A survey was conducted with over 200 students who were considered to be web-active. When explained about mashups, the users gave the feedback that usefulness is more important to them than the perceived difficulty of action. This shows that these users are willing to learn new technologies if they consider it useful for them.

The paper gives several insights to user's online activities and preferences. Most of the users were known to create content online 'just for fun' or 'to socialize'. Further analysis revealed that a strong predictor of their willingness to create mashups would be for sharing hobbies online. It was seen that people with less technology initiative prefer mashups that involve people and social activity and those with more technology initiative prefer more complex, media related mashups.

Zang and Rosson, 2008 in their article "*What's in a mashup? And why? Studying the perceptions of web-active end-users*" gives an insight on what kind of contents and services might end-users actually want to mash up. 116 potential (novice) mashup developers were asked to give examples of mashups that they would like to create. The following figure (see Figure 2.13) shows the frequency of ideas generated.

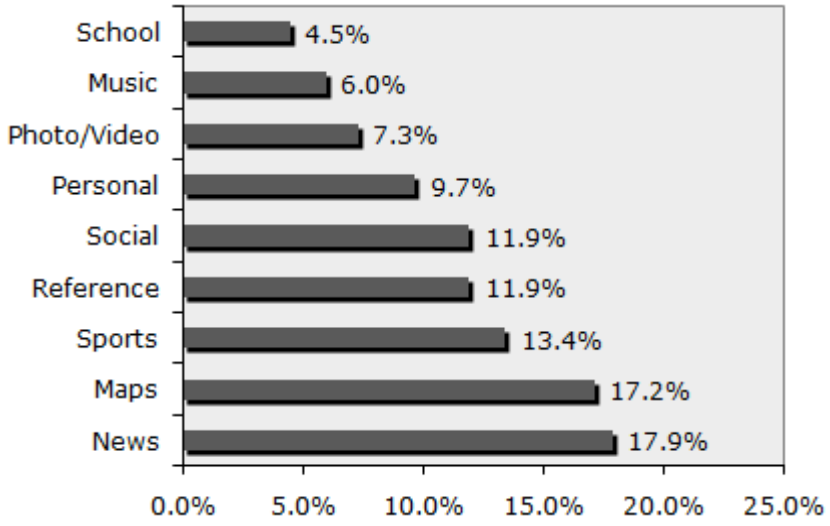

Figure 2.13 End-users ideas for creating mashups

The figure shows that users are most interested to develop mashups for news contents and maps. The authors however, justify that this could be so because the two mashup examples provided to users before the survey illustrated both these ideas. A high number of

examples were given for sports, and then for reference (dictionary, encyclopedia etc), followed by social (e.g. Facebook) and personal activities, which users are increasingly adopting in the Internet.

In another paper, "*Mashups on the Web: End User Programming Opportunities and Challenges*" [Zang, 2009], the author states that among the three processes of composing mashups, viz. data gathering, data manipulation and data presentation, end-users usually find the data manipulation stage most confusing. To examine data manipulation in more detail, 12 participants (web-active end-users) were interviewed, and also asked to create an example mashup using the Yahoo! Pipes tool. Yahoo! Pipes is a tool having as strong focus on data integration and is essentially a visual programming language. The results show that specifically the name of any module was the only means for interpreting its functionality, and in this particular platform, users were often confused between the names and functionalities of two or more modules like Loop module and Filter module. The study suggests that to master each module, users would need to spend considerable time. In essence, the author concludes that Yahoo! Pipes is not a tool for non-programmers. The overall organization of the tool and the names of the modules suggest that this tool mainly simplifies data aggregation for developers. For the less programming savvy end-users, the tool requires them to specify actions in a programmatic manner, which may often not be possible, because of the underlying difference between the ways in which developers and normal end-users can solve problems. Programmers will have generally developed the ability of what is called 'computational thinking'. In the context of mashups, computational thinking refers to the way in which programmers can decompose problems into a way that is acceptable by computer systems. End-users, generally have lower abilities for computational thinking. However, in the long run, it has been recommended that instead of building easy to use tools, it might be beneficial to encourage end-users to increase their computational skills.

Grammel and Storey, 2008, in the paper "*An End-user Perspective on Mashup Makers*" have studied several existing mashup tools, like Yahoo! Pipes, Microsoft Popfly, IBM Mashup center, Google Mashup Editor and compared their features over six different aspects, viz. Levels of abstraction, learning support, community features, searchability, UI design and software engineering techniques.

Three levels of abstraction are defined, that differ in the level of programming and computer knowledge required, viz: high level abstraction, intermediate level abstraction and low level abstraction. On high level abstraction, programming knowledge is not required, but the achieved flexibility is low, e.g. reusing already built mashups, mashups and widget parameterization and programming by example. Intermediate level abstraction encapsulates the technological details into system-specific notations, but generally requires users to have knowledge about data flows, data types and UI widgets. Low level abstraction requires programming knowledge, but the highest flexibility can be acquired. Majority of the reviewed tools exhibit support for all levels of abstraction, except Yahoo! Pipes and Google Mashup Editor. For a high abstraction, all of the tools allow for complete reuse of mashups; programming by example is supported by IBM Mashup Center and Intel Mash Maker. For an intermediate level abstraction, Microsoft Popfly, Yahoo! Pipes and IBM Mashup center have visual dataflow languages, while the Serena Mashup Center has a visual process/workflow orchestration language. In a low level abstraction, textual Domain Specific Language (DSL) editors for languages such as HTML, JavaScript etc. are used to represent mashup elements.

Learning support includes basic learning support like tutorials, screencasts, API documentation etc, community based learning support like discussion forums, sharing of resources, and context-specific suggestions that provide guidance to users according to their current activity in building a mashup. In general, all the mashup makers have been found to have good support for learning in the above mentioned aspects, however, the context-specific suggestions being provided by Microsoft Popfly only.

Community features include facilities for sharing mashups, collaborative categorization of mashups by techniques such as tagging and rating, provision of discussion forums, and extension of sharing resources and created mashups over to social networks like Facebook. All the reviewed mashups have facilities for sharing mashups, tagging features are provided by Yahoo! Pipes, Microsoft Popfly and Intel Mashup Center, and rating features are provided by Intel Mashmaker, in addition to this three tools. Discussion forums are provided in all of them, except Intel Mashmaker, and support for social networks is provided by Microsoft Popfly only.

Searchability refers to searching and finding mashups and mashup elements and includes text based search, browsing mashups by structural properties, while context-specific

suggestions eliminate the need to perform actual search. Most of the reviewed tools were seen to have good support for text based search only, and lesser for browsing by structural properties, and context specific suggestions.

User Interface design is also an important aspect of determining the usability of mashups. UIs can be automatically generated based on output of the mashup operation, however, lacking flexibility. UI composition, in a visual or textual manner is considered the most flexible approach, but also quite complicated. Visual UI composition is found to be the most common among the reviewed tools, being supported by Intel Mashup Center, Serena Mashup Composer and Intel Mash Maker.

Software Engineering techniques refer to techniques like debugging, version control, testing and change request management, which have been considered important for End user development. Among the reviewed tools, debugging the output is seen to be supported by Microsoft Popfly, Yahoo! Pipes and Intel Mashup Center, while version control is supported by Intel Mashup Center and Serena Mashup Composer.

2.3. User acceptance model

1.1.4. The technology acceptance model (TAM)

The technology acceptance model [Davis, 1989] is used in information systems domain to model how do individual users accept and adopt a technology. The model identifies two major variables, perceived usefulness and perceived ease-of-use to determine user's intention in using any information technology related system.

In the article "*Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology*" [Davis, 1989] has defined two specific variables - perceived usefulness and perceived ease-of-use, which are hypothesized to be actual determinants of user acceptance of the system. These two variables determine an individual's intention to use a system with intention to use serving as a mediator of actual system use. Perceived usefulness has been defined as "the degree to which a person believes that using a particular system would enhance his or her job performance." Perceived ease-of-use has been defined as "the degree to which a person believes that using a particular system would be free of effort."

These specific variables were described by multi-item measurement scales, which were pretested and then validated (convergent, discriminant and factorial) by means of empirical studies between end-users of the system. Perceived usefulness was significantly correlated with current usage (r = .63) and future usage (r = .85) of the system. Also, perceived ease-of-use was found to be significantly correlated with current usage (r = .45) and future usage (r = .59). It was seen that usefulness had a significantly greater correlation with usage behavior than ease-of-use. The TAM model is represented in Figure 2.14.

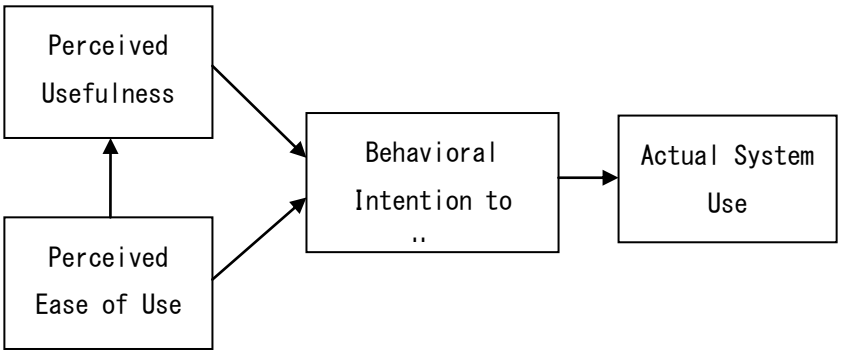

Figure 2.14 The technology acceptance model

1.1.5. Unified theory of acceptance and use of technology (UTAUT)

The UTAUT model formulated by Venkatesh et al. in 2003, is presented in the article “User acceptance of information technology: Toward a unified view". This model is a unified model, formed after reviewing and combining the elements of eight similar models, viz. the theory of reasoned action, the technology acceptance model, the motivational model, the theory of planned behavior, a model combining the technology acceptance model and the theory of planned behavior, the model of PC utilization, the innovation diffusion theory, and the social cognitive theory [Venkatesh et al., 2003]. This model has four core constructs/determinants of intention and usage, which are performance expectancy, effort expectancy, social influence and facilitating conditions. Also, it has four moderators, gender, age, experience and voluntariness of use which are stated to mediate the effects of the four constructs on intention and behavior (usage). The UTAUT model has been found to outperform all the eight models when tested with the original data, with an adjusted R2 of 69%. The individual models, could otherwise explain between 17% to 53% of the variance in user intentions to use information technology. Testing the model with two new sets of

organizational data gave similar results (adjusted R2 of 70%). The proposed model is shown in Figure 2.15.

In forming the model, seven constructs have been identified to be significant direct determinants of intention or usage in one or several of the individual models: performance expectancy, effort expectancy, social influence, facilitating conditions, attitude toward using technology, self-efficacy and anxiety. Of these, the model theorizes attitude toward using technology, self-efficacy, and anxiety not to be direct determinants of intention.

The four key constructs defined by the model are as follows [Venkatesh et al., 2003]:

- "Performance expectancy is defined as the degree to which an individual believes that using the system will help him or her to attain gains in job performance".
- "Effort expectancy is defined as the degree of ease associated with the use of the system".
- "Social influence is defined as the degree to which an individual perceives that important others believe he or she should use the new system".
- "Facilitating conditions are defined as the degree to which an individual believes that an organizational and technical infrastructure exists to support use of the system".

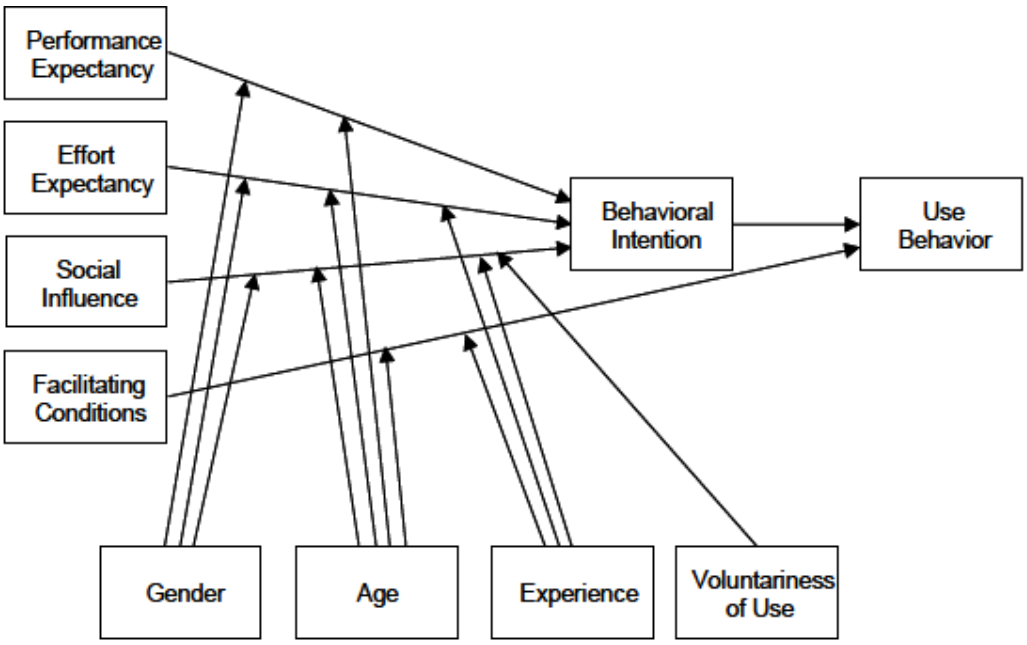

Figure 2.15 The UTAUT model

There are four moderators, gender, age, experience and voluntariness, which are infact user characteristics that are posited to have a mediating effect on the relationship of the four constructs to intention and usage. Of these, experience refers to experience (familiarity) in using the system and voluntariness accounts for whether the user is obliged or free to use the system at own will. There are several hypotheses that are postulated and statistically proven to show relations between the four constructs, intention and usage, as stated in Table 2.1.

Hypothesis 1	"The influence of performance expectancy on behavioral intention will be moderated by gender and age, such that the effect will be stronger for men and particularly for younger men".
Hypothesis 2	"The influence of effort expectancy on behavioral intention will be moderated by gender, age, and experience, such that the effect will be stronger for women, particularly younger women, and particularly at early stages of experience".
Hypothesis 3	"The influence of effort expectancy on behavioral intention will be moderated by gender, age, and experience, such that the effect will be stronger for women, particularly younger women, and particularly at early stages of experience".
Hypothesis 4a	"Facilitating conditions will not have a significant influence on behavioral intention".
Hypothesis 4b	"The influence of facilitating conditions on usage will be moderated by age and experience, such that the effect will be stronger for older workers, particularly with increasing experience".
Hypothesis 5a	"Computer self-efficacy will not have a significant influence on behavioral intention"
Hypothesis 5b	"Compute anxiety will not have a significant influence on behavioral intention".
Hypothesis 5c	"Attitude toward using technology will not have a significant influence on

	behavioral intention".
Hypothesis 6	"Behavioral intention will have a significant positive influence on usage".

Table 2.1 Hypotheses proposed by the UTAUT model

In this thesis, a user-acceptance model for service mashups is proposed based on the Unified Theory of Use and Acceptance of Technology (UTAUT). Various hypotheses are proposed to show relations between the variables proposed in the model, which ultimately are used to model "intention to use".

2.4. Social tagging (folksonomy)

1.1.6. Definition of social tagging

Lumere, 2008, mentions in his article "Social Tagging and Music Information Retrieval" that social tagging is an unstructured word assign to an item. Unlike keyword, the words could be from single word to short phrase, the word assigned is unstructured and does not have vocabulary control, and there is no restriction in placing a mark-up tag as well as the number of tags with the item. Mostly, tags are assigned by the normal users, which are non-expert, for their personal use and future preference. These tags given to an item become useful when the resources put together into one pool of information, which is referred as folksonomy.

Wal, 2004 gives the definition of folksonomy as the combination of the word folk, meaning people, and the word taxonomy, meaning an arranged in hierarchical structure. He defines it as folksonomy the result when people tag information to an item freely in the objective of one's own retrieval. Tagging is done in the social environment where many people can freely tag into one distinctive item.

Moreover, Shirky, 2005, describes the action of folksonomy as clustering and creating the ontology of words to create a better category of words and information. This information is used to create service discovery and recommendation for the users. He defines that to create an effective clustering method, ontology is require to be considered.

1.1.7. Classification of social tagging

Research done by Marlow et al., 2006, state that the folksonomy of social tagging can be divided into two categories: one is system design and attributes, and the other is user's incentive. System design and attributes is divided into 7 points as follow:

Tagging right, the restriction of system allowing the users to tag resources, they extend into two more: self-tagging, allowing only the owner to tag resources; and free-for-all tagging, allowing any users to tag any resources.

Tagging Support, mechanism of supporting the tagging process, there are three types: blind-tagging, not allow users to view the other tag of the resource; viewable tagging, other tag is viewable by the other; and suggestive tagging, the tag is recommended to the owner of resource.

Aggregation, the tag aggregating around the source, divided into two types: bag-model, allow the occurrence of the same tag in the resource; set-model, does not allow the occurrence of tag in the resource.

Type of Object, refers to type of resources, any viewable elements are considered as able for tagging, nowadays there are resources such as images, websites, blogs, widget, etc.

Source of Material, resources of the system can be supplied by the users, the website itself.

Resource Connectivity, the resources in the system can be linked to each other. They can be categorized into three groups: linked, grouped, or none. Example of linked resource connectivity would be Del.icio.us, where the web pages are linked to the system, while in Flickr resource can be assign to group.

Social Connectivity, some service also allow the users of the system to link to each other as if using social networking experience, they can also be categorized to linked, grouped, or none. Users may connect to each other by direct link to each other or they are connected through group.

The paper states that the user's incentive plays a significant role in any social tagging system. There are 6 motivations and expressions that could be considered as the users incentives:

Future Retrieval, an intention of marking a resource to retrieve such resource in the future.

Contribution and Sharing, adding tag to create value to either known or unknown audience.

Attract Attention, tagging resource to gain attention from other people, could be spam tag in some cases.

Play and Competition, some may use the system as to compete with other in some certain rules and environment.

Self Presentation, for user to mark their identity on the resource.

Opinion Expression, expression of opinion to the element to the other.

Continuing from the previous research, Chukmol et al., 2008, state that in service discovery there are only three aspects from the system architecture that seems to be interesting, there are: "Tagging Right", "Tagging Support, and "Aggregation".

1.1.8. Problem statement on social tagging

Zhou et al., 2007, mention that even though social tagging folksonomy is becoming popular in the Semantic Web, there are two main area of problem to investigate:

Uncontrolled vocabulary. The vocabulary defined by user is uncontrolled and is not authoritatively determined. This results in two following limitations. Firstly, ambiguity, people may use same word for different meaning, for example, word 'golf' could either refer

to a car, or a sport. Secondly, word synonym, different word may have the same meaning. With such limitation user may result in missing the valuable information and receiving irrelevant information.

Non-hierarchical structure. Folksonomy provides a flat annotation space, there are no hierarchy presented, make it difficult to browse into system, bridging folksonomy and the traditional hierarchical ontology.

Meanwhile, Problems of Social Bookmarking Today - Part One, 2007, an article from RubyRailays.com, suggest that there are many problems that exist in social tagging. In this article, it states that:

There are no common tagging conventions, where people can tag something like "WebDev", or "Web-Dev".

There is no tagging relation. Nowadays there are so many tags, and most of them have relationships between each other, but the relation structure does not exist.

Micro tagging, in some resources like blog and documents. With existence of micro tagging users can search for minor resources in the whole element. For example, even a document is tagged with 'Ruby', but in the document there could exist onyx a specific area with content related to ruby.

Ontology, synonyms, and typo correction, these are the features that are missing causing the searching to be difficult. Ontology and words synonym would help in searching for required resources easier. While typo correction would resolve the error of people miss typing the tag information.

Aurnhammer et al., 2006, state that each tagging data have their own relation with other tagging, while the relation between each tags are varies and different. Whereas in the current system there are no services that consider the relation between each tag, and discover the tag similarity from user's preferences. In their paper they investigate on retrieving such relations of tags from the user preference.

While in Dubinsky et al., 2006, the authors believe that no exist of tag evolution in the resource should be concern. For example, today the resource may be tag as ‘web development’ while in two-month time, the resource may not only be consider as other thing else. He states that the resources may have an evolution, while the user preferences to that element may also change in time.

1.1.9. Overview of research work on social tagging

1.1.9.1 Tag relationship discovery

Many web services such as Del.icio.us and Flickr.com allows users to tag their desire keyword to an element provide in the web site. As the service grown bigger, the number of users increases and the number of tags in the system also increases, this comes to a point where people realise that tag can have relationship with each other, maybe in term of synonym [Chukmol et al., 2008], or through the resource they are notate with [Wu et al., 2006] [Dubinko et al., 2006], or even through word ontology [Li et al., 2007] [Zhou et al., 2007].

Many researchers have investigated and try to implement on many method in discovering tag relation. Most of the researchers we found tend to use the information from the existed service like Flickr [Dubinko et al., 2006], Del.icio.us [Zhou et al.,2007,], this could be a result from implementing an existed information is better than creating new one, and as well as tag relation is more efficient using large scale of information data [Li et al., 2007]. Many research papers provide a great perspective displaying the possibility of discovering tag relationship using different kind of algorithm, model and method, which are from many different field of study such as the semantic network, and information retrieval.

These researches are very interested in the term of idea, different type of implementation method, and how each of them looks at the problem differently. For example, some paper believe in creating tag ontology [Li et al., 2007], some focus tag clustering [Wu et al., 2006], whereas some consider both perspective [Zhou et al.,2007], while some concern on the evolution of tag relation in a time scale [Dubinko et al., 2006].

1.1.9.2 Web service discovery through social tagging

As social tagging is becoming popular, there are various web services such as Del.icio.us, and Flickr using tags to discover services and resources for users. In web services, they uses the approach of keyword matching to find resources for users, for example, if users search for “travel” the system will return resources with tag “travel” and neglect the tag such as “travels” and “travelling” which literally the same.

Following from the problem stating above, number of researchers investigate the approach of discovering resources using social tagging. There are various techniques that are used, Aurnhammer et al., 2006, uses users’ resource preferences to recommend them more resources, while Chukmol et al., 2008, implement another web services; WordNet, finding word synonym and finding resource containing tag’s synonym, whereas paper like Ding et al., 2010, introduce their own technique of QEBT and QPBT for service discovery.

1.1.10. Detailed description of research work

This section we investigate on the research work that is interesting to our research, they are divide into different section according to their focus area of research. At the end of each section we will also have a comparison table to conclude the main idea of each paper.

1.1.10.1 Tag relationship discovery

Li et al., 2007, focus their research on large-scale social tagging services. They proposed a system call Effective Large Scale Annotation Browser (ELSABer). The system focus on three areas:

Semantic Browsing, this helps in browsing the annotation of tags, trying to solve the problem like synonym.

Hierarchical Browsing, Create top down notation of information, when the information data set is too large, it becomes easier for the user to search for relevant element with such notation.

Efficient Browsing, faster browsing when the number of elements is large.

Creating semantic browsing, the authors use the tags similarity in the elements. The elements with most tag similarity are the selected elements. Hierarchical browsing uses the concept of coverage and interception rating to find the tag value. Each tags are define as either; father tag, sub-tag, or other tag. Each tag would also have an assigned value; this value would determine the position of the tags in the hierarchical tree. Creating efficient browsing, the authors' follows the observations that the popular tags are often use to notate the URLs, while popular URLs are a resource of tag semantic discovery. With such observation, the authors use only the popular tags and URLs to create hierarchical tree and semantic search, causing the output to be faster and more relevant.

Wu et al., 2006, introduce the concept of implementing social annotation into social tagging. They compare the effectiveness of annotation and usual services with no annotation. The experiment uses the log-likelihood on the data set. They applied the model of Separable Mixture Model into the system, and extend it to apply with the probabilistic model to find the semantic relationship in the social annotation data. They create conceptual dimension of information, each dimension act as if they are the clusters of tags, in each dimension they concede of 10 different tags that are related to each other.

Moreover, the research also conducts on a semantic search system. They use probability equations to find such result. In this algorithm, there are four phases:

Basic search model, the equation returning the rank of the resources.

Resource discovery, a developed basic model which takes the keyword matching technique into account.

Personalized search model, finding the tag of which will be related to each different user interest.

Complicated query support, model develop from the basic model, helping the system to handle the searching of more than one keyword, and also the existence of word such as 'and' or 'or'.

As the results from their experiments, the researchers use an approach of bottom-up to create the folksonomy in social tagging environment. The models they proposing have prove that it resolves the ambiguity of tag synonyms, and tag grouping resolves the tag similarity problem.

In research conducted by Zhou et al., 2007, the researchers implement the concept from the previous two researches. They introduce the concept of implementing both hierarchical architecture and clustering algorithm together. The researchers start they work on having a full set of tags in hand, then they categorize each tag into each cluster according to the relationship of each tags, figure 2.16 shows the clustering process.

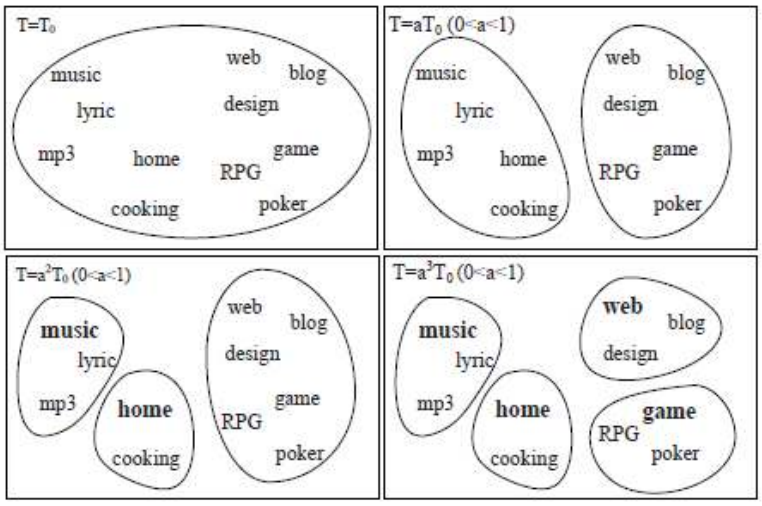

Figure 2.16 Clustering process

In the figure 2.16 we can see that the researchers start with list of tags, they implement an algorithm of sorting tags into each cluster, finding their related tags and the leading tag of the cluster, which is shown in bold. The researchers based their equation on KL-Divergence equation, equation for finding the relation between two probabilities.

Using the clustering results and the top node of each cluster, the researchers implement another algorithm to sort the clusters into hierarchical manners; they call this algorithm, Deterministic Annealing Algorithm (DA). Using DA they can successfully create a tree of tag clustering relation. For example, the first cluster with leading tag ‘travel’ will have the tags of which similar to travel, this may be something like ‘travelling’ or ‘trip’, below the

first node they are cluster of ‘hotel’ and ‘transport’, in each cluster they would also contain the number of the tags of which will be similar and relate to the leading tag.

Dubinko et al., 2006, research on the tag evolution in the time scale finding the relationship of tags concerning the time frame. For example, last month, a relation between the word ‘ruby’ and ‘web dev’ could be high, but this month, it does not show a good relationship anymore. Authors represent their work using flash as visualization; figure 2.17 displays the screenshot from the system.

Figure 2.17 Screenshot of the system with QUERY WATERFALL

In this system, the time will be changing depending on the users input, which is day, week, or month scale. On the left side of the screen are tags that had a strong relation to the input query, on the right on the image with the tag. As the time changes the tag on the left and the images on the right will change. Note that sometimes tag and image may not change as the time goes by as they might still represent a strong relationship during other month as well. The size of the word is in proportion to the relationship to the query tag.

As we have reviewed the on-going research on tag relationship discovery, we found each paper have different focusing area on discovering relationship, while they also have

different implementation technique based on their focusing area. Table 2.2 sum up the information we found, “authors” and “paper” column is the author and the research paper, “goal” column is the area they are focus in their research, while “implementation technique” column explain the technique the researchers implement to achieved their goal.

Author(s)	Paper	Goal	Methods
Li et al.	Towards Effective Browsing Large Scale Social Annotation	– Tag semantic – Hierarchy creation	– Tag concept similarity using the term frequency and inverse document frequency rules of Information Retrieval field – Find father-tag by coverage rule, and sub-tag by intersect rule
Wu et al.	Exploring Social Annotations for Semantic Web	– Social semantic	– Create conceptual dimension of related tag, user, and resource using bigram Separable Mixture Model – Rank tag relation within the dimension space using the value from log-likelihood Expectation-Maximum method
Zhou et al.	An Unsupervised Model for Exploring Hierarchical Semantics from Social Annotation	– Hierarchical cluster	– KL-Divergence for finding tag relationship creating cluster of tags – DA Algorithm to create hierarchical structure
Dubinko et al.	Visualizing Tags over Time	– Tag relation evolution	– Finding the tag relation, using term frequency and inverse document frequency, in accordance to defined time frame

Table 2.2 Comparison table on tag relationship discovery

1.1.10.2 Web service discovery through social tagging

Aurnhammer et al., 2006, comes up with a navigation map for images discovery using social tagging annotation. The researchers use website Flick.com as their data resource. The system allows the user to input a keyword into the system, while it will generate a list of images for the users.

To explain the system, we will follow the layout of figure 2.18. Figure 2.18 shows the screenshot of the interface from the system. This screenshot divide into four zones. First, the suggestion display, after the user input the keyword into the system, he will be provided with a set of images related to his search. This set of images is found using common tag and similarity search. Second, user collection area, user can put the images that they like into this area for further usage. Tag notate to these images are considered in the next area, tag visualization. Tag visualization concerns the tags that are notate to the image selected. In this area the system finds the tag that relate to existed tag. The inner of the circle is the existed tags, while the outer ring is the tag associate to the one inside. Size of each circle indicates the amount of images notates by the tag, while the distance between the white circle and red circle indicate the similarity between the selected tags, and the suggested tags. Last area is the tag menu, where all of the tag in the previous area is shown.

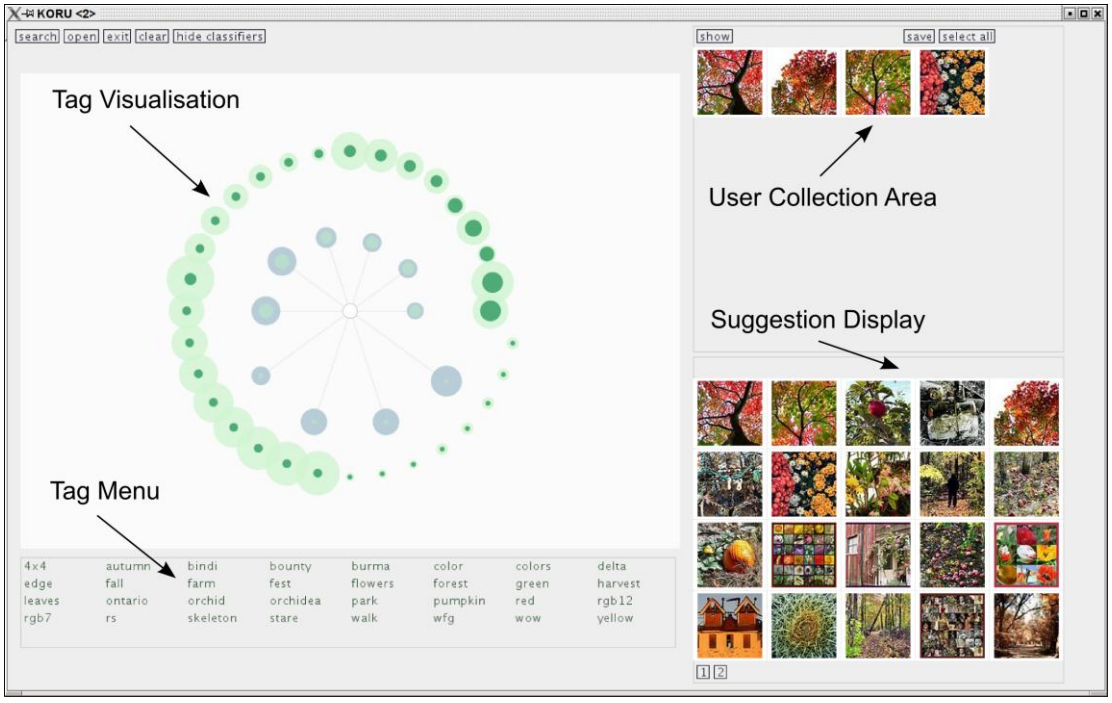

Figure 2.18 Screenshot interface of navigation map

The researchers do not provide any evaluation and comparative of this method and the others. However, they have proved that we can uses visualization effects to support users in navigating and searching in social tagging services.

Chukmol et al., 2008, implement the uses of words synonym together with weighting to discover the service. The researchers use the synonym from WordNet, Miller, 1995, to compute the similarity between each word. The authors' algorithm is to find the number of synonym that is the same to each other to find the number vector of such word. The system firstly find the resource that have the exact same tag, however, if they cannot find such resource, it goes to step 2. In step 2, the system looks for the synonym of query tag from WordNet, and then finds the resources that have the same tag matching with the synonym of query tag. With list of results researcher uses 'AND', 'OR' approach to get the number of resources. 'AND' considered the resources retrieve from two query tags.

Ding et al., 2010, this paper compare the results of discovering web service based on tag using QEBT, Query Expansion Based on Tag, and QPBT, Query Pruning Based on Tag, algorithm. As the result, the researchers state that using QPBT algorithm is more effective than QEBT. QEBT algorithm follow a simple process of finding the set of resources that contain the query tag, and then follow the same step with the tag that have the highest similarity with the query tag, while the process continue with the new tag until the similarity of the tag is more than threshold that is predefined. While QPBT use a more complicate algorithm, it develop a concept of pruning, if there are other resources that would contain the same tag as the resource in no pruning set, that resources will be considered as relevant, then the resource will be add to no-pruning set as well as the result set.

The researchers construct, as experiment using the two algorithms, they find out that QPBT results are less than QEBT, and the resource is also more relevant. However, the researchers also state that in the future work it would be interesting to construct an experiment which will find the number of recall and precision of the two algorithms, of which it could be useful in improving the algorithm.

Table 2.3 is the summary of papers reviewed in this chapter.

Author(s)	Paper	Goal	Methods
Aurnhammer et al.	Augmenting Navigation for Collaborative Tagging with Emergent Semantics	– Navigation Map – Combine Image Properties and User's Queries	– Oriented Gaussian Derivative and Euclidean Distance for image distance – Uses nearest neighbour classifier to find the nearest related image
Chukmol et al.	Enhancing Web Service Discovery by using Collaborative Tagging System	– Service discovery through notated tags	– Word's synonym comparing using WordNet finding word's synonym
Ding et al.	A Web Service Discovery Method Based on Tag	– Discovering service using user's query	– QEBT and QPBT algorithm

Table 2.3 Comparison table on content/service discovery

In this thesis, an enhanced web gadget discovery model is proposed based on social tagging, to match relevant gadgets to the user enquiry. We follow the approach in Zhou et al., 2007 to build a mathematical model, by exploring the tag relations using Kullback-Leibler (KL) Divergence and Deterministic Annealing (DA) methods.

3. SUMMARY OF RESEARCH CONTRIBUTIONS

In this chapter, the contributions of the original publications are presented in detail. The contributions consist of four parts: the concept and incoming trends of user-generated service, a design and usage analysis of an event-based consumer mashup system, the design and usage analysis of a web widget discovery method through social tagging experience, and finally, the design and implementation of an event-based online social networking system. The third and fourth contributions are follow-up research of the second contribution, from service discovery and socializing perspectives respectively.

Papers I and II clarify the concept of user-generated service, analyze the challenges, and illustrate the incoming trends of user-generated service. Papers III, IV and V present the framework, design, and evaluation of an event-based service mashup system, respectively. Journal Paper XI is a complete and comprehensive version of papers III, IV and V. Paper VIII addresses state of the art and solutions of a social annotation based widget discovery system. Finally, Paper VI and demo paper VII present the design and implementation of an event-based online social networking system. Papers VIII, IX, X, and XII are collaborative work conducted during PhD studies, however, these contributions will not be summarized in this thesis.

3.1. Concepts and incoming trends of user-generated service

This section gives the definition of user-generated service, analyzes the challenges, and illustrates the incoming trends of end-user driven service creation. Paper I and Paper II contribute to this aspect.

Paper I, entitled “The Incoming Trends of End-user Driven Service Creation”, illustrates that service creation is the next big end-user driven hype in the technology arena, and this trend will lead to a huge set of new features and services. The author had the main

responsibility for the technical content and paper writing work. The paper raised three general questions towards this new trend:

- Why should end-users create services?
- What are the ongoing works on end-user driven service creation?
- What are the challenges of enabling end-users to do future service creation?

As for the first question, the author illustrates that from end-user perspective, the best way to achieve user personalization is giving ways to end-users selves to develop services satisfying their specific needs; while from the business perspective, the main idea behind end-user service creation is to shorten analysis/development of marketing chain, harness collective intelligence and promote innovation in software development.

The author then illustrates the incoming trends of end-user driven service creation with extensive new features, and provides five main research topics from both technical and business perspectives. Five incoming trends are shown in Figure 3.1 with more detailed explanations shown in Figure 3.2.

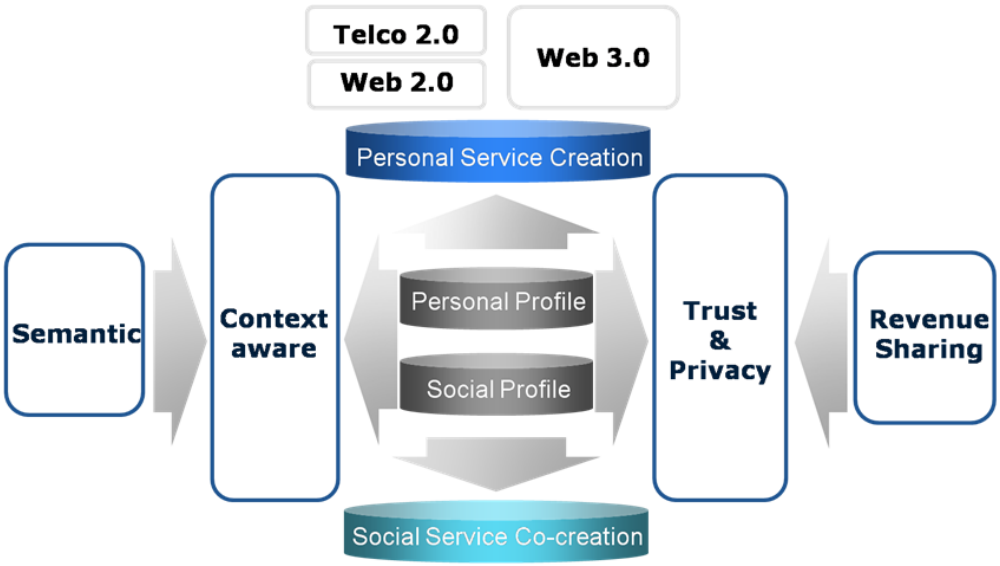

Figure 3.1 Five trends of end-user driven service creation (advanced version)

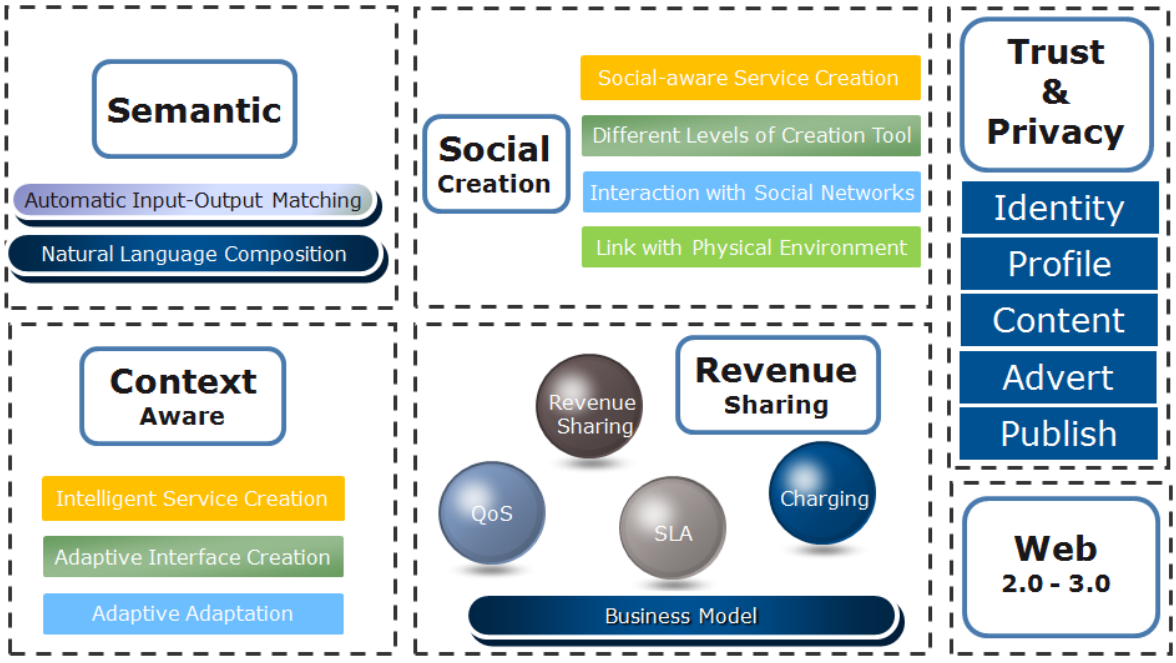

Figure 3.2 Five trends of end-user driven service creation (details)

Paper II, entitled "A Survey of User Generated Service", points that the current research aims to foster the evolution from the absolutely successful user generated content (UGC) to user-generated service (UGS). The author of this thesis was the main author of the paper. The author clarifies the concept of UGS by comparing it with UGC. Figure 3.3 shows the similarities and differences between two terms. Both of them can be seen as a set of activities that consist in interactions between user and the system. These activities have a value-added result for the user. This implies that a certain amount of user creative effort is involved in the content or service creation process. Moreover, both UGC and UGS emphasize on the content/service created by the end-user, not the professionals. Concerning the difference, the first thing is related to the revenue. In the current web, the content creation process is free. The system is free of use, and no revenue will be added on the contents that the user created. However, in UGS, users may have the possibility to share the revenue among multi-actors, and the development of service creation platform tends to be charged. The second difference is concerning the user intention and target. Differing from UGC, which focuses on the "creation for fun", socialization and knowledge sharing aspects, user-generated service puts weight on the service personalization and service reuse.

	User Generated Content	User Generated Service
Similarities	✧ interactions between user and the system	
	✧ certain amount of user creative effort is involved	
	✧ emphasis on the ordinary users, not the professionals	
Differences	**Revenue Concern**	
	✧ No revenue added content	✧ Possible revenue added service
	✧ System is free of charge to use	✧ Platform tends to be charged
	User Intention and Target Concern	
	✧ Content creation and sharing	✧ Service personalization and reuse

Figure 3.3 UGC v.s. UGS

Some key characteristics of UGS and UGS-platforms include:

- Ordinary end-users for service/ application creation, without actual programming
- Accessible service database of existing services and content
- A rich, visual environment for service/content selection, interaction, grouping, filtering etc. with adequate flexibility and customization
- Generation of new, value-added composite applications (service), requiring some creative effort, for personal, social, situational (ad hoc), or collaborative use in consumer domain or enterprise domain.

The author continues to detail the underlying key technologies of user-generated service, current platforms/tools, as well as a brief overview of incoming trends of user-generated service.

3.2. Design of an event-based service mashup system

This section deals with the research issues related to design, implementation and usage analysis of an event-based service mashup system. Four publications, namely Paper III, Paper IV, Paper V, and Paper XI, contribute to this from their own perspectives.

Short Paper III, entitled “An Event-Based Functionality Integration Framework”, presents an event-based service mashup framework to address the issue of service personalization. The

proposed framework addresses the mashup issue from a new perspective by extracting and reasoning the context through user generated event, while recommending and aggregating the contextual services dynamically in response to the user's functional requirements. The three layer system framework, including knowledge Layer, Logic Layer, and Presentation Layer, is shown in Figure 3.4.

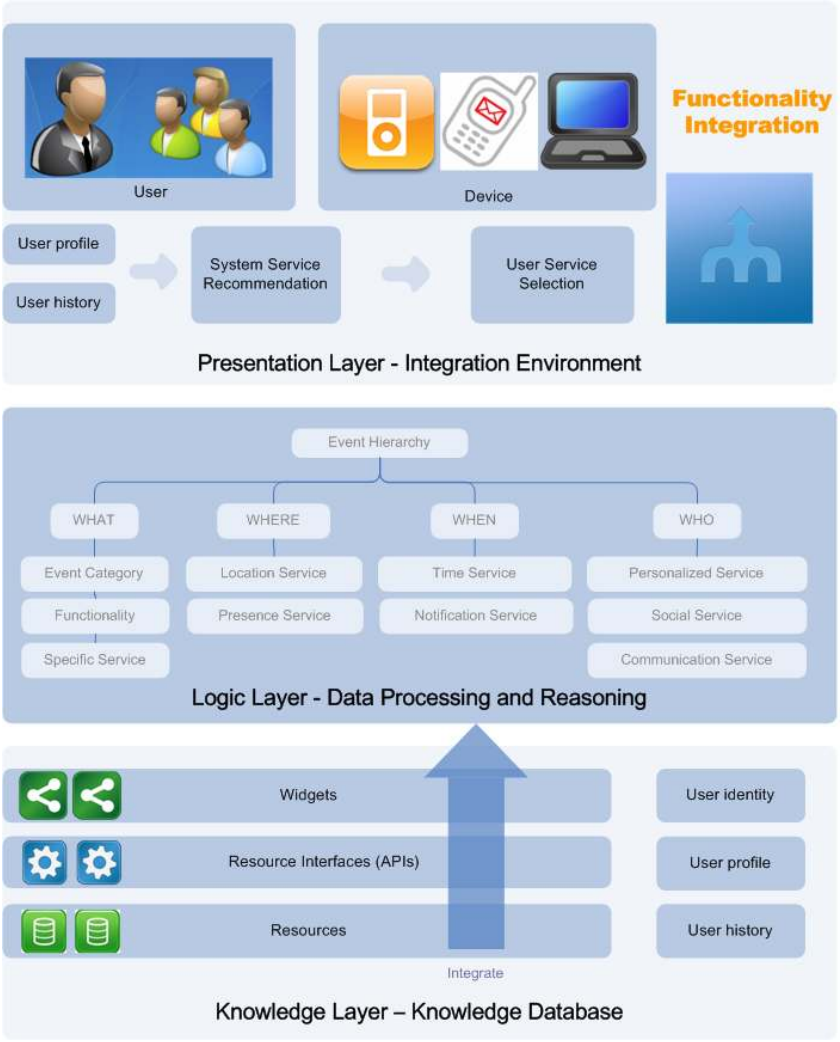

Figure 3.4 Event-based functionality integration framework

Service integration is done in the presentation layer. A user uses device to enter details for an event like meeting, art exhibition etc, and then the user input data and user identity are passed to the logic layer, where the data is processed, and reasoning is performed. Resources are looked up from the bottom layer, which is a layer for aggregation of resources.

Paper IV, entitled “Mashup Services to Daily Activities – End-user Perspective in Designing a Consumer Mashups”, introduced a user-centered mashup system, named event-based service provider (EBSP). The author of this thesis took the main responsibility for the technical content in this paper and paper writing work. The idea aims to design and implement a user-centered mashup system which provides greater motivation for mashups usage, by relating every-day calendar events to useful gadgets. The paper points out that in order to allow non-expert users to be engaged in this innovative practice, perceived usefulness and perceived ease-of-use are the primary factors to consider since they have a direct and positive impact on users to accept and use any technology.

Followed by a literature review of overall studies on end-user’s perspective on service mashups, this paper conclude that: on the one hand, process mashup is still considered quite complicated and discouraging for non-expert end-users. On the other hand, web page customization mashup systems are seen to provide a convenient way for users to aggregate selected services, referred to as widgets or gadgets in their personal dashboard for the creation of a personalized environment. These methods, however, lack flexibility since the gadgets cannot communicate with each other or with any other web service. Moreover, such systems exhibit large service databases and often permit access to the third party for increased system functionality, which does not necessarily provide a better solution and quality of experience for the user.

Thus, the author presents a consumer mashup framework based on daily activities, to firstly acquire the context information through user generated daily event, followed by the recommendation and aggregation of precisely relevant contextual services.

In Paper IV, the design principles, design framework, a usage scenario and design methods are described in detail. Figure 3.5 illustrates the flowchart of the system process.

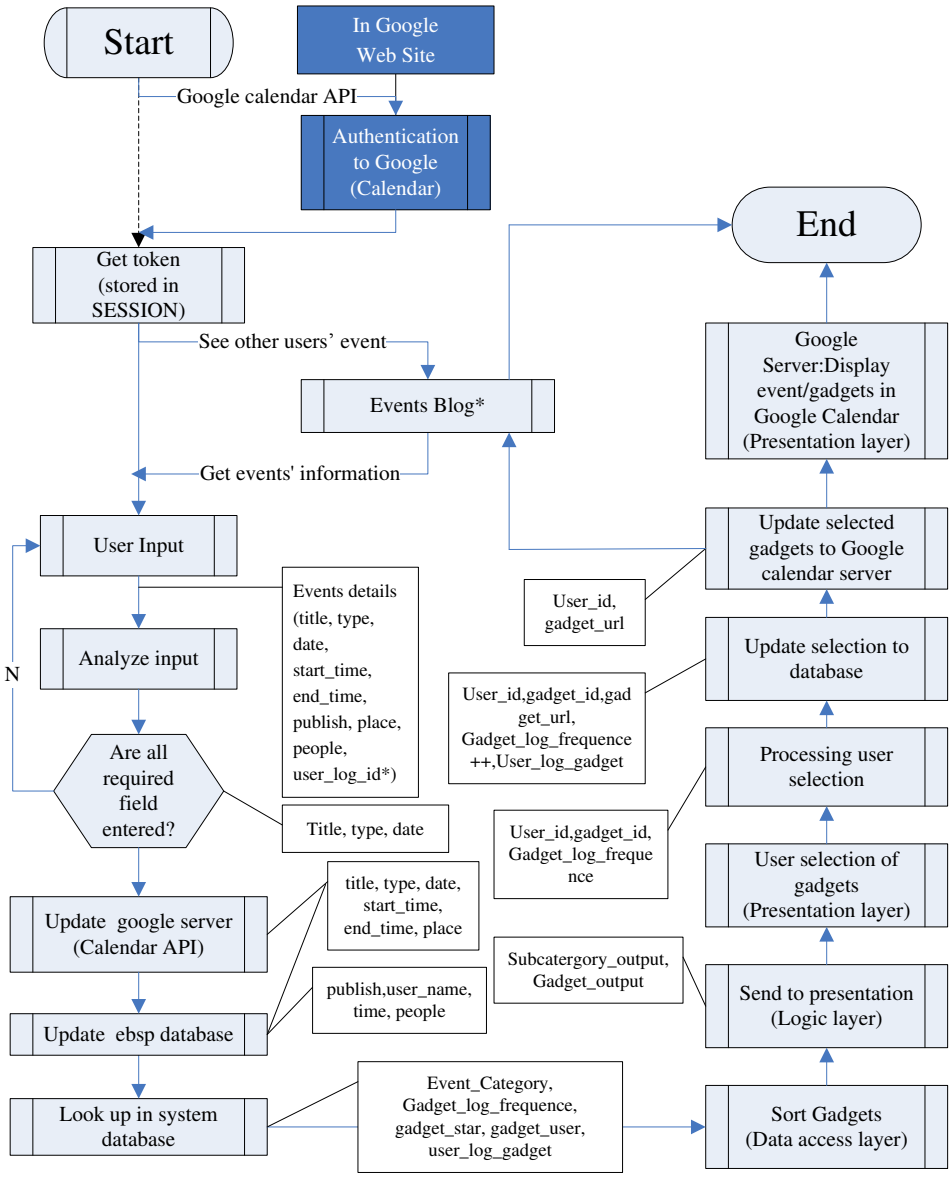

Figure 3.5 Flowchart of the system process

In the proposed system, the matching between events and gadgets is essential. In paper IV, to look up the exact gadgets of interest in the system database (service discovery), a hierarchical, directory-based search has been implemented, based on the input for event hierarchy. The implementation of the event hierarchy is shown in Figure 3.6.

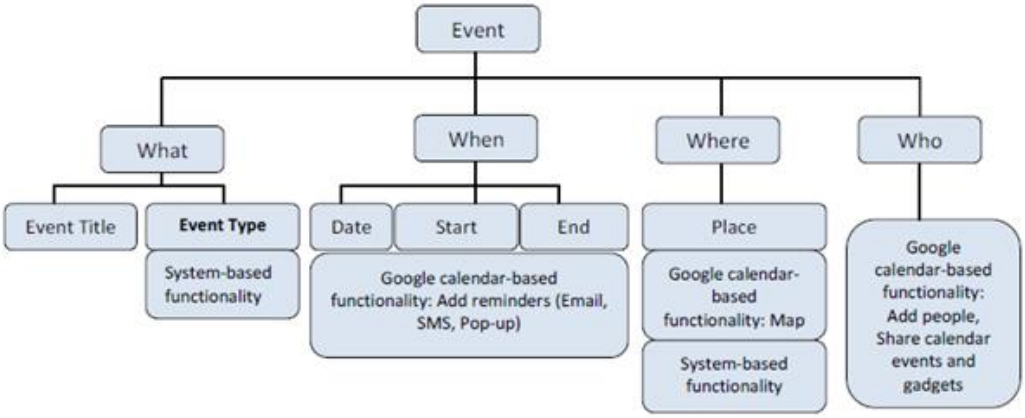

Figure 3.6 The implementation of event hierarchy

After the gadget discovery, the gadget recommendation (i.e. the order of the gadgets displayed in the webpage) is reasoned according to three rules: 1) based on user's gadget selection history; 2) based on gadget rating and gadget user number; and 3) based on the gadgets selected in the inherited event*. The flowchart of this process is shown in Figure 3.7.

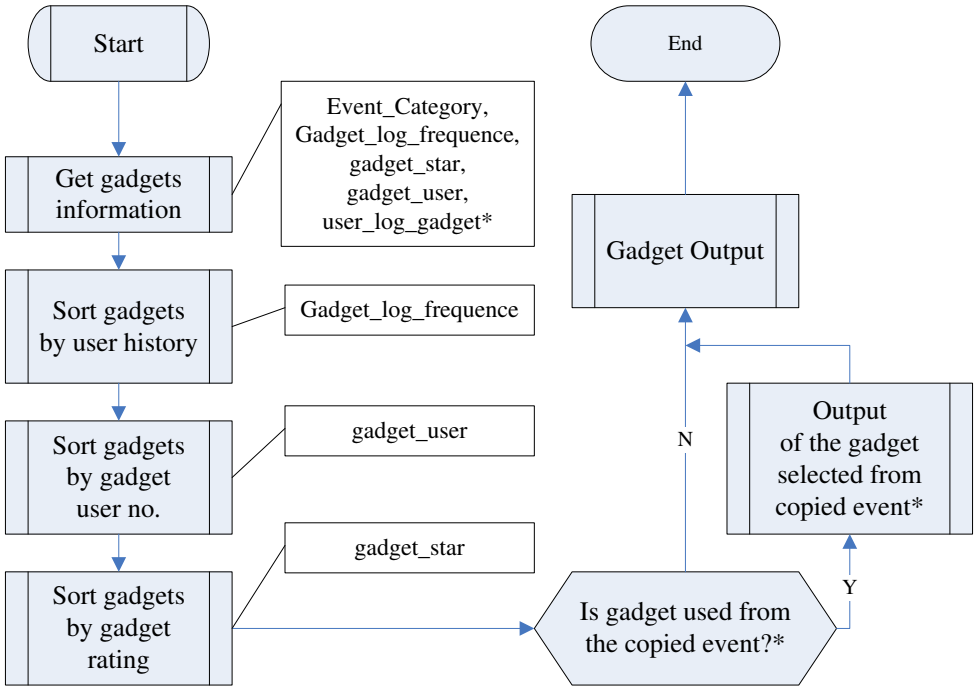

Figure 3.7 Flowchart of gadget recommendation process

Paper V, entitled "Consumer Mashups: End-User Perspectives and Acceptance Model", presents a user acceptance model for consumer mashups, to precisely identify what factors lead to their adoption and to what extent. The author of this thesis was a co-author of the article and was mainly responsible for the methodologies of the proposed user acceptance

model. This model (see Figure 3.8) is based on the Unified Theory of Use and Acceptance of Technology (UTAUT) model.

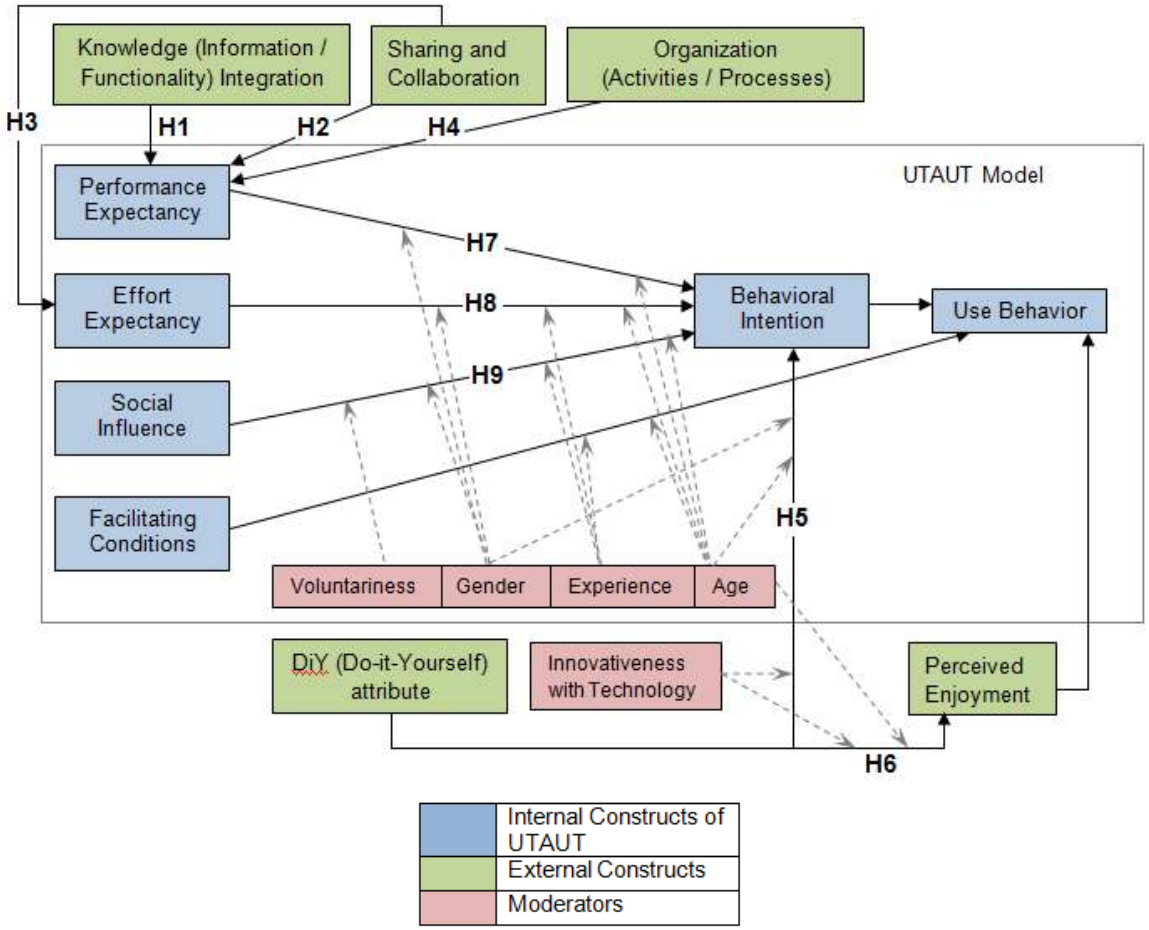

Figure 3.8 Flowchart of gadget recommendation process

The study is conducted with an online survey of 131 users to gather data to verify the model. Users are asked to give feedback to EBSP system. In addition to the internal constructs of UTAUT, the proposed research model uses several external constructs specific for consumer mashup systems: Knowledge Integration, Sharing and collaboration, Do-it-yourself and Perceived enjoyment. An external moderator, Innovativeness with Technology, is introduced to have a varying degree of effect on individual's intention to use a technology. Nine hypotheses are formulated.

An online survey is developed based on the research model. Particularly, we are interested to find out what does it take for users to express their willingness in using the mashup technology. We ask users to give their experiences in using EBSP system, before

doing it, a brief video tutorial about commercial mashup tools is provided to make them familiar with mashup technology. Then we provide a brief video tutorial on the EBSP system, with various examples. A total of 423 requests are sent out online, of which we receive 233 partial responses. Only 131 are complete and valid for our study.

Data obtained from the survey has been statistically analyzed through Cronbach's Alpha Test, Person's correlation and multiple regressions. More results are shown in Paper V. The analysis of the relationships in the research model is presented as follows:

Hypothesis 1: Knowledge integration was found to have a moderately positive impact on the performance expectancy. This means that if users can integrate various sources of information and different applications into one mashup platform, they consider the platform to be useful.

Hypothesis 2: Sharing and collaboration were found to have a moderately positive effect on performance expectancy. This means that if users can share information and applications with each other, e.g. see other's applications, reuse their applications, and work collaboratively to develop an application, then the users perceive the system to be useful for them.

Hypothesis 3: Sharing and collaboration were found to have a positive but weak effect on effort expectancy. This indicates that facilities for sharing applications and collaborative working have a positive, but small role to reduce the effort needed in building an application for the participants of the survey.

Hypothesis 4: Organization of daily life activities and processes has a moderately positive impact on the performance expectancy. Among all the relevant constructs, OAP is observed to have the strongest relation to performance expectancy. This indicates that the users of the survey consider a platform to be most useful when it allows the organization of their day-to-day activities.

Hypothesis 5: Do-it-Yourself attribute is found to have a moderately positive effect on behavioural intention. Moderators like innovativeness with technology, genders, and ages were seen to affect this relation. The tendency to create web applications, as and when

required is observed to increase as the technological innovativeness increases from average, high to extremely high. Males were seen to have a greater initiative for making their own applications. It was observed that the people above 30 years of age were highly interested in developing their applications. This can be due to the fact that they have enough experience and acquaintance with technology to try new things to bring values to their daily lives.

Hypothesis 6: Do-it-Yourself attribute was found to have a positive but weak effect on perceived enjoyment. It indicates that few people enjoy working with the system because they are able to develop their own ad-hoc applications. There may be some other factor which affects perceived enjoyment more than DiY does. However, it is seen that people with the greatest technological innovativeness enjoy working with the system more, as they can make applications their own way. Also, people above 30 years showed a high level of enjoyment working with the system, because they could make and assemble services themselves as per the need. This is consistent with the effect of ages in hypothesis 5.

Hypothesis 7: Performance expectancy was found to have a stronger positive effect on the intention to use the mashup systems. Performance expectancy is observed to have the strongest relation to behavioral intention, as compared to all other relevant constructs. This is consistent with our belief that users need to have a system which is very useful for them if they decide on using it. The relationship has been moderated by the gender and age such that males have a greater intention to use the system when they consider it useful. This result is consistent with similar hypothesis in the UTAUT model (Venkatesh et al., 2003). People above 30 years have a strong intention to use the system, if they consider it useful.

Hypothesis 8: Effort Expectancy was found to have a positive but weaker effect on behavioral intention. It means users choose to use a mashup system if they think it doesn't require much effort, however this is not as important as the system having to be 'useful' to them to decide to be using the system. Males showed a greater interest in using the system if they consider it to be taking less effort, whereas females showed lower interest in this matter. This is in contrast to the UTAUT model, which assumes women have a stronger relation between effort expectancy and behavioral intention. People above 30 years of age showed a high interest in using the system if it doesn't require much effort.

Hypothesis 9: Social influence was not found to have a significant effect on behavioral intention. However, when analyzing the relation based on gender, females were seen to be affected by social influence in using the mashup system. This is consistent with previous studies (Venkatesh et al., 2003). The age group above 30 years had a greater intention to use the mashup system when social influence on them was considered.

Journal Paper XI, entitled "The Design, Implementation and Usage Analysis of an Event-Based Consumer Mashups System", is a complete and comprehensive version of the papers III, IV and V. This paper presents the updated framework, design and evaluation of an event-based service mashup system. The author of this thesis has mainly contributed to the technical content and writing work of paper XI.

The system and methods presented above have been filed in the United State Patent I, entitled "System and Methods for Service Provision Based on Events".

The EBSP system can be accessed through the following link:
http://myebsp.com/myebsp/

A demo video of the EBSP system can be downloaded through the following link:
http://www.box.com/s/9i73c4149776r7b8iyt2

3.3. Extension work (1): Design of a web widget discovery system through social annotation

One of the main challenging issues in Section 3.2 is the matching between user-created event and system recommended gadgets. Based on the research question of "how to discover right gadgets for each event that a user creates", this section presents a web widget discovery method through social annotation.

Paper XIII, entitled "A System for Web Widget Discovery Using Semantic Distance between User Intent and Social Tags", describes the state of the art on social annotation related research and presents a method of service discovery through social tagging. The author of this thesis took the main responsibility for the technical content in this paper and

paper writing work. The experiment system is performed using widget services from WidgetBox.com, where each widget is annotated through tags. The main objective is an automatic system that can discover widgets (gadgets) that would match users' query. In the former work on event-based service mashups, user input in event details can be seen as such a query. The detailed service discovery method is classified into two stages: tag discovery and widget discovery. Firstly, tag discovery uses Kullback-Leibler (KL) Divergence and Deterministic Annealing (DA) algorithm to find the relation of each tag and each event. For each input keyword, top ten tags are extracted. Then, widget discovery is performed using the top ten tags: three schemes are considered in this process (see Figure 3.9) to get a final list of widgets.

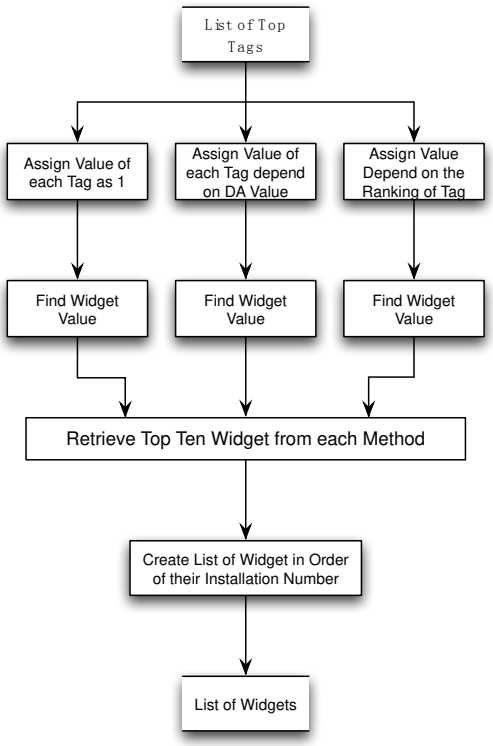

Figure 3.9 Widget value calculation method

We further set up a system to test the methods. System flow is illustrated in Figure 3.10. System dataset, tag discovery results and widget discovery results are presented in detail in Paper XIII. Finally, we have performed a usability test comparing our results with keyword matching results. Our system has proved its accuracy and efficiency.

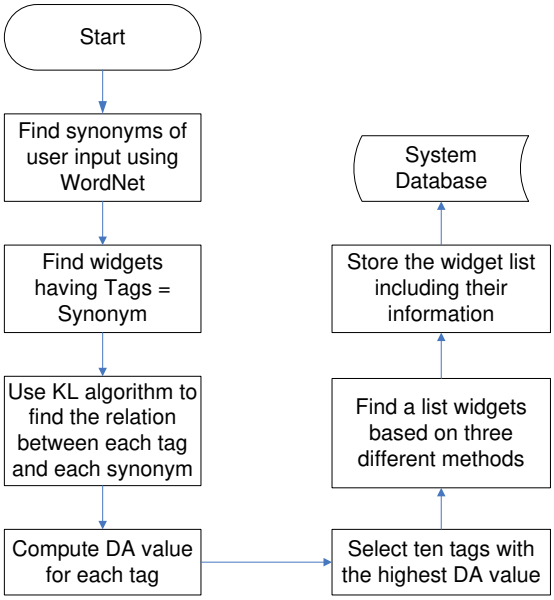

Figure 3.10 Flow diagram of back-end system

3.4. Extension work (2): Design of an event-based online social network

We have learned from user-acceptance analysis in Section 3.2 that sharing and collaboration is found to have a moderately positive effect on performance expectancy in service mashups. This means that if users can share information and applications with each other, e.g. see other's applications, reuse their applications, and work collaboratively to develop an application, then the users perceive the system to be useful for them. In this section, the extension version of EBSP: an event-based online social network is designed to meet the requirements of users' collaboration requirement in mashups.

Paper VI, entitled "The Design of Activity-Oriented Social Networking: Dig-Event", presents the system design and rational for a novel activity-oriented social networking site called Do-it-together Event (Dig-Event). This system has been inspired by previous research on calendaring and popular social network applications like Facebook and Google+. Dig-Event provides an open, social space for users to share events and to discover the activities of mutual interest among social contacts like schoolmates, families, friends, and colleagues. It allows users to share their activities in the customized social circle, conduct events by selecting activity-based gadgets, and socialize around them. Compared with current existing event-based social networks, six design principles have been proposed to design Dig-Event:

Design Principle 1: Social networking among peers, not strangers.

Design Principle 2: Share and discover day-to-day activities, rather than official events.

Design Principle 3: Event management in not only "4W" - What, When, Where, Who, but also "H"- How.

Design Principle 4: Open Access to customized social circles, not "all-or-none".

Design Principle 5: Event recommendation over event discovery.

Design Principle 6: Integration into social networks and calendars, instead of replacing them.

Furthermore, based on the design principles, the system has been implemented. System framework and usage scenario are presented in detail in Paper VI.

Paper VII, entitled "Dig-Event: Let's Socialize around Events" is a demo paper. It presents the design principles and the main characteristics of the system, which have been included in Paper VI.

The system and methods presented above, has been filed in the United State Patent II, entitled "Systems and Methods for Social-Event Based Sharing".

The Dig-Event system can be accessed through the following link:
http://myebsp.com/

A demo video of the Dig-event system can be downloaded through the following link:
http://www.box.com/s/o4lrkzz6zffz4pzdvqt5

These original papers are reprinted at the end of the printed version of this thesis, with the permissions from the original publishers. In the electronic version, they are not included,

but are available via the original publishers. In the next chapter, conclusions are drawn from the overall work.

4. CONCLUSIONS AND FUTURE WORK

Web 2.0 is envisioned as an open garden for services. The success of the user-generated content raises the new idea of enabling end-users to create their own services. A growing number of users are interested in combining existing contents and services to create their own web applications. This trend is referred to as UGS or user-driven mashups.

Emergence of social web shed light on the concept of collective intelligence. The social involvement within Web 2.0 paradigm can be classified into three categories: user-generated content for socializing and knowledge sharing; user-enhanced social relationships through social networks; user-enriched knowledge representations through social annotation like social tagging, rating and commenting.

The objective of this thesis is to design a use-friendly mashup that leverages the evolution of social web.

This thesis firstly illustrates the concept and incoming trends of user-generated service and web mashups. With the objective of developing a simple mashup platform that provides the greater motivation for mashup usage, an event-based mashup system has been designed and implemented. After the development of the EBSP system, an evaluation of the system was performed to gather initial feedback by means of an online survey. A research model for evaluating user-acceptance of service mashups has been proposed, based on the UTAUT model. Data collected from the survey were used to test the relationships between the variables of the model.

Then, to enhance the accuracy of the event and gadget matching, a web widget discovery method using social annotation has been proposed. An automatic system has been set up to execute the results by firstly discovering the tag relations and then using such relations for gadget retrieving. By comparing with current keyword matching algorithm, the system has been validated for its accuracy and efficiency.

4.CONCLUSIONS AND FUTURE WORK

The last work of this thesis was focused on the design and rational of a novel activity-oriented social network system named Dig-Event within an open, social calendar. Dig-Event aims to push the surge of online collaboration from user-generated-content to user-generated-event, by sharing and discovering the activity of mutual interest among social contacts. Using Dig-Event, one can share activities in customized social circles. The system recommends activity-based gadgets from which users can choose gadgets of interest to organize certain events. The system further recommends relevant events to the users.

Future work includes two parts: As for the widget discovery using social tagging, we plan to conduct user researches to further test the quality of the tags that attached to the gadgets in order to perform better recommendation to the user. Regarding the event-based online social network, we plan to test the system usability and see what lessons we can learn: from both user and system sides.

The second phase of the evolution of the online world is combined with 3 different terms - Web 2.0, mashup and social networking, but all intertwined in the brave new Internet. Now that user-generated service is well under way, new trends are beginning to emerge among social web looking to improve the collective intelligence and will add great value in the current market.

5. REFERENCES

Almeida, J. P., Baravaglio, A., Belaunde, M., Falcarin, P. & Kovacs, E. (2008). Service Creation in the SPICE Service Platform. In International Symposium on a World of Wireless, Mobile and Multimedia Networks.

Aurnhammer, M., Hanappe, P., & Steels, L. (2006). Augmenting Navigation for Collaborative Tagging with Emergent Semantics. ISWC 2006. LNCS, vol. 4273.

Benslimane, D., DustDar, S. & Sheth, A. (2008). Services Mashups - The New Generation of Web Applications. IEEE Computer Society, 13-15

Bouillet, E., Feblowitz, M., Feng, H., Liu, Z., Ranganathan, A., & Riabov, A. (2008). A Folksonomy-Based Model of Web Services for discovery and automatic composition. 2008 IEEE International Conference on Services Computing.

Braga, D., Ceri, S., Martinenghi, D. & Daniel, F. (2008). Mashing Up Search Services. IEEE Computer Society, Internet Computing, IEEE Press, 16-23.

Buchmann, A., Bornhovd, C., Cilia, M., Fiege, L., Gartner, F., & Liebig, C. (2004). DREAM: Distributed Reliable Event-based Application Management.

Calisir, F. & Calisir, F. (2004). The relation of interface usability characteristics, perceived usefulness, and perceived ease of use to end-user satisfaction with enterprise resource planning (ERP) systems. Computers in Human Behavior

Chukmol, U., Benharkat, A.-N., & Amghar, Y. (2008). Enhancing Web Service Discovery by using Collaborative Tagging System. 4th International Conference on Next Generation Web Services Practices.

REFERENCES

Cilia, M., Bornhovd, C., & Buchmann, A. P. (2003). CREAM: An Infrastructure for Distributed, Heterogeneous Event-based Applications.

Cowan, J. (2005). RESTful Web Services. http://home.ccil.org/~cowan/restws.pdf

Curbera et al. (2003). The Next step in Web Services. Communications of the ACM, 86(10), 29-34

Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information. MIS Quarterly.

Diaz, O., Perez, S. & Paz I.(2007). Providing Personalized Mashups Within the Context of Existing Web Applications. WISE , 493-502

Ding, Z., Lei, D., Yan, J., Bin, Z., & Lun, A. (2010). A Web Service Discovery Method Based on Tag. International Conference on Complex, Intelligent and Software Intensive Systems.

Droegehorn, O., Konig, I., Belaunde, M., Le-Jeune, G., Cupillard, J. & Kovacs, E. (2008). Professional and End-User-Driven Service Creation in the SPICE platform. In Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, CA, USA.

Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., & Tomkins, A. (2006.). Visualizing tags over time. In 15th Int. WWW Conference (pp. 193–202). Edinburgh, Scotland: ACM Press. TY - CONF.

Ennals, R., & Garofalakis, M. (2007). Mashmaker: Mashups for the Masses. In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, China, 1116-118.

Gallego, M.D., Luna, P. & Bueno, S. (2008). User acceptance model of open source software. Computers in Human Behavior 24, 2199–2216

REFERENCES

Grammel, L. & Storey, M. (2008). An End User Perspective on Mashup Makers. University of Victoria Technical Report DCS-324-IR.

Halbert, D.C. (1984). Programming by Example.

Hoyer, V., Stanoesvka-Slabeva, K., Janner, T. & Scroth C. (2008). Enterprise Mashups: Design Principles towards the Long Tail of User Needs. IEEE International Conference on Services Computing.

Keene, C. (2009). Five Free Mashup Tools You Should Know About.

Laga, N., Berting, E., & Crespi, N. (2008). User-centric Services and Service Composition, a Survey. Software Engineering Workshop , pp 3 - 9.

Li, R., Bao, S., Fei, B., Su, Z., & Yu, Y. (2007). Towards Effective Browsing of Large Scale Social Annotations. International World Wide Web Conference.

Lim, J.-H., & Tian, Q. (2003). Home Photo Content Modeling for Personalized Event-Based Retrieval.

Liu, H., & Singh, P. (2004). ConceptNet: a practical commonsense reasoning toolkit. BT Technology Journal 22 , 221-226.

Liu, X., Hui, Y., Sun, W. & Liang, H. (2007). Towards Service Compostion Based on Mashup. In IEEE Congress on Services.

Lizcano, D., Soriano, J., Reyes, M. & Hierro, J. J. (2008). The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. IEEE Computer Society.

Lumere, P. (2008). Social Tagging and Music Information Retrieval. Sun Micro System Inc.

REFERENCES

Marchewka, J. T., Liu, C. & Kostiwa Kurt. (2007). An Application of the UTAUT Model for Understanding Student Perceptions Using Course Management Software. Communications of the IIMA, 7(2), 93-104.

Marlow, C., Naaman, M., Boyd, D., & Davis, M. (2006). HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, ToRead. In the proceedings of the 7th Conference on Hypertext and Hypermedia, (pp. 31 - 40). Denmark.

Merill, D. (2006). Mashups: The new breed of Web app - An Introduction to Mashups.

Mianwei Zhou, S. B. (2007). An Unsupervised Model for Exploring Hierarchical Semantics from Social Annotations. ISWC/ASWC 2007, LNCS 4825 , 680-693.

Milanovic, N., & Malek, M. (2004). Current solutions for Web service composition. IEEE Internet Computing, 8 (6), 51-59.

Miller, G. A. (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11 , 39-41.

Namoune, A., Wajid, U. & Mahendjiev, N. (2009). In proceedings of UGS 2009.

Natis, Y.V., (2003). Service-Oriented Architecture Scenario.

Nevill-Manning, C. (1993). Programming by Demonstration. New Zealand Journal of Computing.

Nunally, J., & Bernstein, I. (1995). Teorı´a psicome´trica. Mexico DF: McGraw-Hill.

O' Reilly (2005). What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software . O'Reilly Network.

Ogrinz, M. (2009). Mashup Patterns: Designs and Examples for the Modern Enterprise.

Pahlke I. & Beck, R. (2009). Assimilation of Enterprise Mashup Systems − The Impact on Changes in Work Processes. In proceedings of DIGIT 2009.

Pautasso, C., Zimmermann, O. & Leyman F. (2008). RESTful Web Services vs. “Big” Web Services: Making the Right Architectural Decision. WWW 2008 / Refereed Track: Web Engineering - Web Service Deployment.

Rhodes, B.J. (2000). Margin Notes: Building a Contextually Aware Associative Memory. In International Conference on Intelligent User Interfaces.

Sheth, A.P. & Gomadam, K. (2007). SA-REST: Semantically Interoperable and Easier-to-use Services and Mashups. IEEE Internet Computing, 84-87.

Shirky, C. (2005). Retrieved 2010 9-12 from Ontology is Overrated: Categories, Links, and Tags.

Trevor, J. (2008). Doing the Mobile Mash. Computer, 41(2), 104-106

Valla, M., Dolfini, D., Fra, C., Ricciardi & C. A. (2009). Mashing-up integrated location, social networks and recommendations: the POI Radar. In Proceedings of 13th International ICIN Conference- Beyond the Bit Pipes, Bordeaux, France.

Venkatesh, V., Morris, M., Davis, G. B., & Davis, F. (2003). User Acceptance of Information Technology: Toward a unified view. MIS Quarterly, 27, 425-478.

Wal, T. V. (2005). Folksonomy. Online Information 2005. London, UK.

Wong, J., Hong, J. I. (2007). Making Mashups with Marmite: Towards end-User Programming for the Web. In Proceedings of the SIGCHI conference on Human factors in computing system, California, 1435-1444.

Wu, X., Zhang, L., & Yu, Y. (2006). Exploring Social Annotations for the Semantic Web. International World Wide Web Conference Committee .

REFERENCES

Wu, Y., Tao, Y.& Yang, P. (2007). Using UTAUT to explore the behavior of 3G mobile communication users. The International Conference on Industrial Engineering and Engineering Management (IEEM)

Wunsch-Vincent, S. & Vickery, G. (2007). Participative Web: User-Created Content.

Yelmo, J. C., del Alamo, J. M., Trapero, R., Falcarin, P., Yu, J., Carro, B. & Baladron, C. (2008). In Proceedings of First ITU-T Kaleidoscope Academic Conference -Innovations in NGN: Future Network and Services, K-INGN, Geneva, 211-218.

Xiao, H., Zou, Y., Tang, R., Ng, J., & Nigul, L. (2009). An Automatic Approach for ontology-Driven Service Composition.

Yu, J., Benatallah, B., Casati, F. & Daniel, F. (2008). Understanding Mashup Development. IEEE Internet Computing, 12(5), 44-52

Zang, N. & Rosson, M.B. (2008). What's in a mashup? And why? Studying the perceptions of web-active end users. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)

Zang, N. (2008). Mashups for the Web-active user. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).

Zang, N. (2009). Mashups on the Web: End User Programming Opportunities and Challenges.

Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., & Chang, H. (2004). QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering, 30 (5), 311 - 327.

6. APPENDICES

Appendix A: Survey Questionnaire

Hello,You are invited to participate in our survey End-user driven Mashups. There are three parts in this survey and it will take approximately 35 minutes to complete the questionnaire. Your survey responses will be strictly confidential and data from this research will be reported only in the aggregate. Thank you very much for your time and support. Please start with the survey now by clicking on the Continue button below.

Part I: User Information

1. Your Age

2. Gender

1. Male
2. Female

3. Education Level

1. Primary School
2. High School
3. Bachelors
4. Masters
5. Doctorate
6. Other

4. Current Situation

1. Student
2. Employed
3. Other

5. Functional Work Area

1. Accounting
2. Finance

3. Human Resources
4. Information Systems
5. Telecommunications
6. Marketing
7. Earth Sciences
8. Production
9. Purchasing
10. Sales
11. Other

6. Country

7.

	Not at All	Low	Average	High	Extremely High
How much of a technology-savvy person are you? (based on internet usage, experience with new applications, and knowledge about latest technological developments). Please rate yourself.	❑	❑	❑	❑	❑

8.

	Not at All	Low	Average	High	Extremely High
How keen are you in experimenting with new applications and technologies to bring more value to your daily life experiences? Please rate yourself.	❑	❑	❑	❑	❑

Part II: Introduction to Mashups and Testing of Prototype

9. Have you heard of the term ‘mashups’, ‘web mashups’ or ‘mashup technologies’ in the internet?

1. Yes
2. No

10.

	Not at All	Low	Medium	High	Extremely High
Please indicate your level of familiarity with the purpose and use of web mashup tools like Yahoo! Pipes, Microsoft PopFly, Intel Mashmaker, Intel Mashup Center, iGoogle, Netvibes, PageFlakes, Kapow, QedWiki etc.	❑	❑	❑	❑	❑

11.

Please indicate which ones you are familiar with (if applicable)

Mashups are web applications that can be created by a normal web user by combining existing content and applications on the internet, to produce something more useful and valuable. Mashups combine information/applications in a visually intuitive way, which would otherwise require programming expertise. Please watch carefully the following tutorial video of Mashup tools for End-users.Four tools are described: iGoogle, Microsoft Popfly, Yahoo! Pipes and Intel Mashmaker.

Video 1

Now you probably have an idea of what mashups are and how they could be useful to you. If you want to learn more, you can try visiting these sites, when you have time.For now, you will be asked to use a simple mashup tool, a calendar-based mashup prototype, currently in beta version. This system will let you mashup (aggregate) iGoogle gadgets/ widgets in your Google Calendar. You can make use of the additional benefits of calendar like sharing event, adding reminders etc. Please watch carefully the following tutorial video about this calendar mashup called epspdemo (EBSP).

Video 2

The video showed you how to use the system ebspdemo (EBSP)for several events, using several useful widgets.Now, please to go to http://www.ebspdemo.info/ and try using the system yourself, by setting an event e.g: ‘French Language Class, and choose your applications/ gadgets of interest. Use the individual applications to see what they can do. Also try to share your event with others. Note how much time it takes for you.Based on your experience with the system ebspdemo.info, please answer the following questions.

12. Could you complete the given task?

1. Yes
2. No

13. How long did it take for you to complete the given task?

1. Less than 5 minutes
2. 5-10 minutes
3. 10-15 minutes
4. Greater than 15 minutes
5. Not Applicable

14.Did you encounter any problems and errors before getting the expected output?

1. Yes
2. No
3. If yes, indicate what type of problem

Thank you for evaluating the EBSP system. We shall now move on to Part III, which is the final section.

Part III: User Acceptance of Consumer Mashups

The ultimate goal of Mashup systems is to achieve three things: Simplicity, Usability and Ease of Access.In this section, we gather perceptions from normal web users on various attributes of consumer mashups, which play a decisive role on the intention to create mashups and use mashups platforms.Based on your experience with the EBSP system, and an understanding of mashup platfoms (as seen in the videos), please give a rating for the following sets.

15. Integration of Information/ Functionality

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
EBSP will let me gather different sources of information into a single interface/ platform.	❑	❑	❑	❑	❑
EBSP will let me aggregate different applications into a single interface/ platform.	❑	❑	❑	❑	❑
EBSP provides an integrated way for searching and tracking information and services.	❑	❑	❑	❑	❑
EBSP provides an integrated platform for sharing	❑	❑	❑	❑	❑

information and services with others.					

16. Organization of Activities and Processes

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
EBSP provides me a way to organize my activities and processes related to work/studies/daily life.	❑	❑	❑	❑	❑
EBSP will assist me to organize my work/study/daily life related activities in a new way.	❑	❑	❑	❑	❑
EBSP will let me have greater control over organizing my activities and processes related to work/study/daily life.	❑	❑	❑	❑	❑

17. Perceived Usefulness

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
EBSP addresses my daily life/study/work related issues and needs.	❑	❑	❑	❑	❑
I would find EBSP useful in my daily life/study/work.	❑	❑	❑	❑	❑
Using EBSP would enable me to accomplish tasks more quickly.	❑	❑	❑	❑	❑
Using EBSP would allow me to perform tasks in a better way.	❑	❑	❑	❑	❑

18. Sharing and Collaboration

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
EBSP provides me a way to share information and applications with others (friends/ colleagues/ family).	❑	❑	❑	❑	❑
EBSP provides me a means to see what applications others have created (aggregated) and	❑	❑	❑	❑	❑

use those applications.					
EBSP enables me to collaborate with others in organization of my work/study/daily life activities.	❑	❑	❑	❑	❑

19. Ease-of-Use

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
My interaction with the system is clear and understandable.	❑	❑	❑	❑	❑
It is easy for me to become skillful at using the system.	❑	❑	❑	❑	❑
I find the system easy to use.	❑	❑	❑	❑	❑
Learning to operate the system is easy for me.	❑	❑	❑	❑	❑

20. Doing-it-Yourself (DiY)

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
I can choose applications by myself in EBSP.	❑	❑	❑	❑	❑
I can combine (aggregate) applications in a single interface by myself in EBSP.	❑	❑	❑	❑	❑
I can combine (aggregate) applications into one platform without the need for programming in EBSP.	❑	❑	❑	❑	❑
I can create a customized and personalized application interface by myself in EBSP.	❑	❑	❑	❑	❑

21. Facilitating conditions

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
I have the resources necessary to use the system (what do I need?).	❑	❑	❑	❑	❑
I have the knowledge necessary to use the system (how do I do it?).	❑	❑	❑	❑	❑
Specialized instruction (text/audio/video etc.) concerning the system is available to me.	❑	❑	❑	❑	❑

22. Perceived Enjoyment

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
I would find it enjoyable to create and use mashups in EBSP system.	❑	❑	❑	❑	❑
It would be fun to create and use mashups in EBSP system.	❑	❑	❑	❑	❑

23. Social influence

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
I have friends/boss/co-workers who are using EBSP.	❑	❑	❑	❑	❑
My friends/ boss/co-workers suggest that I should use EBSP.	❑	❑	❑	❑	❑
I have friends/boss/co-workers have been helpful to me in the use of EBSP.	❑	❑	❑	❑	❑

24. Intention to use the System

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
I anticipate that I will use the EBSP mashup system for my work/study/daily life activities.	❑	❑	❑	❑	❑
I predict that I will use the EBSP mashup system for my work/study/daily life activities.	❑	❑	❑	❑	❑
I intend to use the EBSP mashup system for my work/study/daily life activities.	❑	❑	❑	❑	❑

We have reached the end of the survey. For any queries, comments and suggestions please fill in below.