UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE 269 MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR (MSII)

Laboratoire des sciences de l'ingénieur, de l'informatique et de l´imagerie (ICube) UMR7357

THÈSE présentée par :

Razia SULTANA

soutenue le : **30 septembre 2014**

pour obtenir le grade de: **Docteur de l'université de Strasbourg**

Discipline/ Spécialité: Instrumentation et Procédés Photoniques

---

TITRE de la thèse :

RESEARCH ON SYSTEM ARCHITECTURE FOR DEVICE AND CONTENT INDEPENDENT APPLICATIONS INCLUDING 3D IMAGING AND VIRTUAL REALITY AS CONTENT

THÈSE dirigée par :

M. MEYRUEIS Patrick	Prof. Dr., Université de Strasbourg
M. CHRIST Andreas	Prof. Dr., Offenburg University of Applied Sciences

RAPPORTEURS :

M. ALAYLI Yasser	Prof. Dr., Université Versailles St Quentin
M. KETATA Mohamed	Prof. Dr., Université CNAM Paris

AUTRES MEMBRES DU JURY :

M. DELLEY Antonie	Prof. Dr., Ecole d'ingénieurs et d'architectes de Fribourg
M. BLANDET Thierry	Ingénieur, Université de Strasbourg

To my parents and lovely sisters

There were moments when I asked myself "was it a right decision to start with a doctoral thesis? "
Days were passing by without any progress at all, independent of how hard I tried. Sometimes it
was frustrating to feel I am not moving forward with my work, actually moving backward.

Suddenly, through positive comments from thesis supervisor or after presenting my work and re-
sults in international conferences when plenty of interesting questions were coming, I was appre-
ciated by many researchers; I found the answer of my question "yes it was a right decision to start
with a doctoral thesis".

My exciting doctoral thesis journey

Razia SULTANA

TITRE de la thèse : RESEARCH ON SYSTEM ARCHITECTURE FOR DEVICE AND CONTENT INDEPENDENT APPLICATIONS INCLUDING 3D IMAGING AND VIRTUAL REALITY AS CONTENT

Résumé

Les réseaux traitant l'information, intègrent différentes technologies incluant une large gamme de dispositifs aux capacités et puissances très variées et capables de gérer une immense quantité d'informations sous forme de données d'une grande diversité de formats. La recherche sur l'imagerie 3D, la réalité virtuelle et autres techniques holographiques permettront d'offrir de nouvelles techniques d'interactions homme/machine (IHM), notamment pour les appareils mobiles. Cela enrichira encore la variété et la diversité de ces dispositifs. Beaucoup d'efforts ont été faits, notamment ces dernières années, afin d'établir une intégration ouverte, évolutive et transparente des différentes technologies permettant l'affichage de contenus, notamment d'images 2D et 3D et ce pour tout type de système, y compris mobile en tenant compte, notamment et de plus en plus, de la situation géographique individuelle de l'utilisateur final. Sur ces différents points, la tâche est loin d'être terminée. Cette thèse représente, dans ce contexte, une contribution, vers la généralisation de la transportabilité des informations pour des objets communicants de natures différentes, notamment portables, particulièrement dans le cas de l'imagerie en 3 dimensions, de plus en plus exploitée.

L'objectif de ce travail de recherche est de trouver un moyen de résoudre le problème énoncé ci-dessus en proposant une architecture pour fournir un accès sans contrainte, continu et personnalisé pour le contenu et les applications interactives partout et à tout moment avec la plus grande quantité possible d'appareils communicants. Un exemple d'exploitation utile et particulièrement adaptée d'un tel système est l'apprentissage sur mobile en raison de la grande diversité des appareils communicants disponibles, dotés de caractéristiques et de fonctionnalités très différentes. Comme solution au problème considéré, une nouvelle architecture nommée "Smoothie" est proposée, un prototype a été développé et évalué en tant que composant important d'un jeu d'apprentissage collaboratif de langues étrangères nommé LLG (*Language Learning Game*).

Résumé en anglais

Today's network landscape consists of many different network technologies, a wide range of end-devices with large scale of capabilities and power, an immense quantity of information and data represented in different formats. Research on 3D imaging, virtual reality and holographic techniques will result in new user interfaces (UI) for mobile devices and will increase their diversity and variety. A lot of effort is being made in order to establish open, scalable and seamless integration of various technologies and content presentation for different devices including those that are mobile, considering the individual situation of the end user. Till today the research is going on in different parts of the world but the task is not completed yet.

The goal of this research work was to find a way to solve the above stated problems by investigating system architectures to provide unconstrained, continuous and personalized access to the content and interactive applications everywhere and at anytime with different devices. As a Solution of the problem considered, a new architecture named "Smoothie" is proposed, developed as prototype and evaluated as an important feature of a collaborative foreign language learning game named LLG.

Acknowledgements

Contents

List of Figures

List of Tables

List of Appendices

Résumé long

1. Introduction

Les réseaux traitant l'information, intègrent différentes technologies incluant une large gamme de dispositifs aux capacités et puissances très variées et capables de gérer une immense quantité d'informations et de données d'une grande diversité de formats.

La recherche sur l'imagerie 3D, la réalité virtuelle et autres techniques holographiques permettront d'offrir de nouvelles techniques d'interactions homme/machine (IHM), notamment pour les appareils mobiles. Cela enrichira encore la variété et la diversité de ces dispositifs. Beaucoup d'efforts ont été faits, notamment ces dernières années, afin d'établir une intégration ouverte, évolutive et transparente des différentes technologies permettant l'affichage de contenus, notamment d'images 2D et 3D et ce pour tout type de système, y compris mobile en tenant compte, notamment et de plus en plus, de la situation géographique individuelle de l'utilisateur final. Sur ces différents points, la tâche est loin d'être terminée. Cette thèse représente, dans ce contexte, une contribution, vers la généralisation de la transportabilité des informations pour des objets communicants de natures différentes, notamment portables, particulièrement dans le cas de l'imagerie en 3 dimensions, de plus en plus exploitée.

2. Problèmes identifiés concernant l'interopérablité des dispositifs communicants

2.1   Bilan de l'état de l'art

Les différents types de dispositifs communicants utilisés en réseau par différents utilisateurs, à des moments différents, ou en parallèle par un même utilisateur, le sont généralement de façon non prévisible et doivent être reconnus par le réseau et le système associé, en temps réel ou quasi temps réel, pour déterminer précisément leurs spécificités.

2.2   Gestion de la dualité utilisateur - dispositifs utilisé, mobile ou fixe

Dans un endroit comme une maison ou un bureau, il est généralement plus « pratique » d'utiliser un PC pour communiquer. En déplacement, l'utilisateur souhaite pouvoir accéder au même contenu avec le même confort et la même richesse mais en utilisant un appareil mobile communicant. Ainsi, si un système est forcément dépendant du dispositif utilisé, du point de vue des fonctionnalités de communication, d'interactivité, de capacité à gérer des données 3D ou de l'affichage des données. Dans le même temps, ce système « matériel et logiciel » doit être le plus possible indépendant du type de dispositif communicant à la fois pour l'accès à l'information et pour les possibilités de communication synchrone et asynchrone. En règle générale, les systèmes de gestion de contenu sont incapables, pour le moment, de gérer toutes ces contraintes avec un stockage de données commun. Mentionnons cependant le Site Web Adaptatif (RWD pour Responsive Web Design) conçu pour offrir à l'utilisateur une expérience de lecture et de navigation sans dégradation sur différents dispositifs. RWD est basé sur une conception de sites Web visant à offrir un accès optimal à l'information notamment, de visualisation sur un ensemble de dispositifs différents. Cela concerne, la lecture et la navigation avec un minimum de redimensionnement et de recadrage et un mode de défilement identique pour un large spectre d'objets communicants, des téléphones mobiles aux écrans d'ordinateur de bureau. Un site conçu avec RWD adapte la mise en page du contenu à l'environnement de visualisation grâce à l'utilisation de trames de mise page, d'images de tailles variables et de feuilles de style CSS3. RWD prend en compte la présentation d'un même contenu de façon similaire sur des dispositifs différents, il ne traite pas le contenu lui-même. Certains fournisseurs de contenu offrent différentes versions de présentation du contenu adapté pour chaque dispositif d'affichage. La gestion de ce contenu nécessite donc des moyens humains et du temps de travail extrêmement coûteux.

Un autre problème, pour élaborer les différentes versions de ces contenus est que les auteurs ont besoin de connaissances détaillées concernant les différents appareils communicants et leur comportement, afin d'être en mesure de générer un contenu adapté spécifiquement aux capacités effectives de ces dispositifs. Mais la plupart des auteurs et développeurs de contenu n'ont pas cette connaissance technique, particulièrement dans le domaine de l'imagerie en 3 dimensions. D'autre part, les auteurs de contenu bien qu'ayant les connaissances techniques n'ont pas nécessairement les connaissances nécessaires pour préparer le contenu d'une manière appropriée. Il y a donc là un problème à résoudre, que nous avons considéré, une solution dynamique est donc nécessaire.

2.3   Justification de l'étude

Dans le cas où le nombre total d'utilisateurs de contenus spécialisés ou d'applications interactives est trop faible pour justifier l'adaptation d'applications ou de contenus à tous les périphériques possibles de façon exploitable et rentable, de nouvelles méthodes doivent être envisagées. C'est ce que nous avons considéré.

2.4   Contexte de la technologie que nous proposons

Le contenu et l'interface utilisateur peuvent inclure différents types de données comme du texte, des images 2D, du son, de la vidéo, des images de type réalité virtuelle en 3D et d'autres formats à venir. Les dispositifs communicants du futur devraient tous intégrer des fonctionnalités d'imagerie 3D. En d'autres termes le système de communication répondant au cahier de charges présenté ci-dessus devra être en mesure de traiter automatiquement tous les formats de données existants ou à venir sans nécessiter de lourdes adaptations et de les rendre exploitables sur une diversité de dispositifs correspondant aux besoins et à la demande des utilisateurs.

3. But et concept de la thèse, solution proposée

L'objectif de ce travail de recherche est de trouver un moyen de résoudre le problème énoncé ci-dessus en proposant une architecture pour fournir un accès sans contrainte, continu et personnalisé pour le contenu et les applications interactives partout et à tout moment avec la plus grande quantité possible d'appareils communicants.

Un exemple d'exploitation utile et particulièrement adaptée d'un tel système est l'apprentissage sur mobile en raison de la grande diversité des appareils communicants disponibles, dotés de caractéristiques et fonctionnalités très différentes. Cela est vrai non seulement pour satisfaire les apprenants par exemple dans le cas simple où tous les apprenants appartiennent à une même communauté, mais également dans le cas où un même apprenant utilise des équipements différents en parallèle ou à des moments différents. Ces applications exploitées peuvent alors être considérablement améliorées en incluant entre autres la présentation de contenu de type réalité virtuelle. Quels que soient les objectifs, il est impossible de développer et d'adapter les contenus pour chaque type d'appareil mobile en raison des différentes capacités des appareils, des problèmes de coût ou des objectifs pédagogiques des auteurs.

Une solution doit donc être trouvée pour permettre l'automatisation du processus d'adaptation des contenus à tout type de dispositif communicant. Pour réaliser un tel système, ces trois principales fonctions doivent être mise en œuvre :

- L'identification de l'appareil connecté

- La création, la structuration et le stockage de contenu généralisé

- Un processus de transformation du contenu généralisé, en un contenu optimisé dépendant de l'appareil

4. Solution au problème considéré

4.1 Originalité fonctionnelle de la solution proposée

L'originalité de notre proposition peut se résumer ainsi :

- Nous proposons une architecture permettant une communication indépendante des appareils et du format du contenu

- L'architecture proposée ne vise pas un service, un type d'application ou un groupe spécifique d'utilisateurs. Cette architecture peut être utilisée à des fins académiques, à des fins commerciales, et même pour un usage individuel privé

- L'architecture proposée utilise des outils qui sont soit « open source » soit libres d'utilisation, de sorte que le prototype peut être ouverte à tout le monde sans restriction

4.2 Champ d'application de notre proposition

Le champ d'application de notre proposition d'architecture peut être représenté par la figure 1. La création du contenu, la gestion du contenu une fois affiché sur le dispositif communicant de l'utilisateur final et la manière dont cet utilisateur aborde le contenu ne rentrent pas dans le cadre de notre proposition.
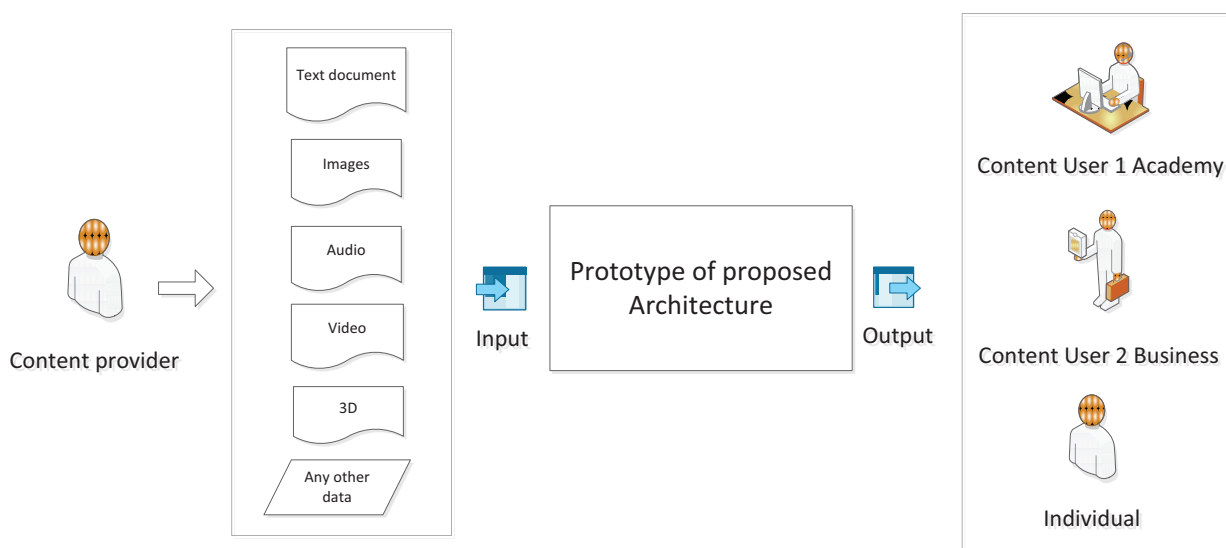
Figure 1. Présentation schématique du champ de la solution proposée

4.3   Présentation de la solution proposée : une nouvelle architecture appelée "Smoothie"

Le mot Smoothie est un nom consacré pour une boisson. Un smoothie est une émulsion de différents produits par exemple des fruits, de la glace pilée, du miel, du sirop, du yaourt, de la crème glacée, du lait de soja, du thé vert, des produits à base de plantes, etc. et dans certains cas il peut aussi contenir du chocolat ou du beurre d'arachide. Nous pouvons donc dire qu'un smoothie peut contenir n'importe quel ingrédient sans limite ni prévisibilité, ne dépendant que de la recette fournie par un individu ou un groupe comme le conseil d'administration d'une entreprise de l'industrie alimentaire. La nouvelle architecture proposée comme une solution au problème mentionné plus haut a des propriétés similaires au smoothie. Il s'agit en quelque sorte d'une émulsion de différentes applications et formats de données. En terme de formats de données tout peut être utilisé en entrée de l'architecture proposée qui n'est ni limitée ni prévisible. En particulier, tout type de données 3D peut être fourni en tant que contenu dans cette architecture. En raison des similitudes entre les propriétés de la boisson et de l'architecture proposée, cette architecture sera appelée "Smoothie".

Tout d'abord, Smoothie détecte si l'utilisateur est connecté au système via un dispositif mobile, ou via un dispositif de bureau. Cette détection se fait par l'analyse de l'en-tête de la requête HTTP en provenance de l'appareil de l'utilisateur. La description des caractéristiques des appareils mobiles et des navigateurs est obtenue grâce à WURFL (Wireless Universal Resource File) dans une version non commerciale. WURFL consiste en un fichier de configuration XML qui contient des informations sur les fonctionnalités et les capacités de la quasi-totalité des appareils mobiles. De plus, la liste des dispositifs référencés dans WURFL est mise à jour quotidiennement par de nombreux contributeurs à travers le monde. Il s'agit donc d'informations à jour qui procurent une bonne fiabilité dans la manipulation des données à destination du terminal utilisé. L'architecture proposée fonctionne avec une combinaison de WURFL et d'une base de données locale enregistrée sur le serveur mis en œuvre.

En ce qui concerne les contenus, les fournisseurs de contenu n'auront plus à se soucier des spécificités des différents dispositifs mobiles et donc de fournir une version optimisée de leur fichier pour chaque appareil. Un contenu sera sauvegardé, dans un premier temps dans son format d'origine, pour des raisons de sécurité, puis il sera converti dans un format généralisé et stocké dans la base de données. Le format généralisé que nous avons retenu est XML ou XML étendu tel que DAE. Une fois connues les capacités de l'appareil communicant identifié, le contenu sera transformé en fonction de l'appareil puis affiché. La conversion et l'individualisation des données sont

réalisées à la volée sur le serveur. Pour le prototypage, un navigateur Web est utilisé suivant le concept RWD (Responsive Web Design), comme porteur du contenu à afficher côté client. Il est possible d'étendre l'individualisation par la mise en œuvre d'autres pipelines pour traiter d'autres formats de sortie tel PDF, WML, etc.
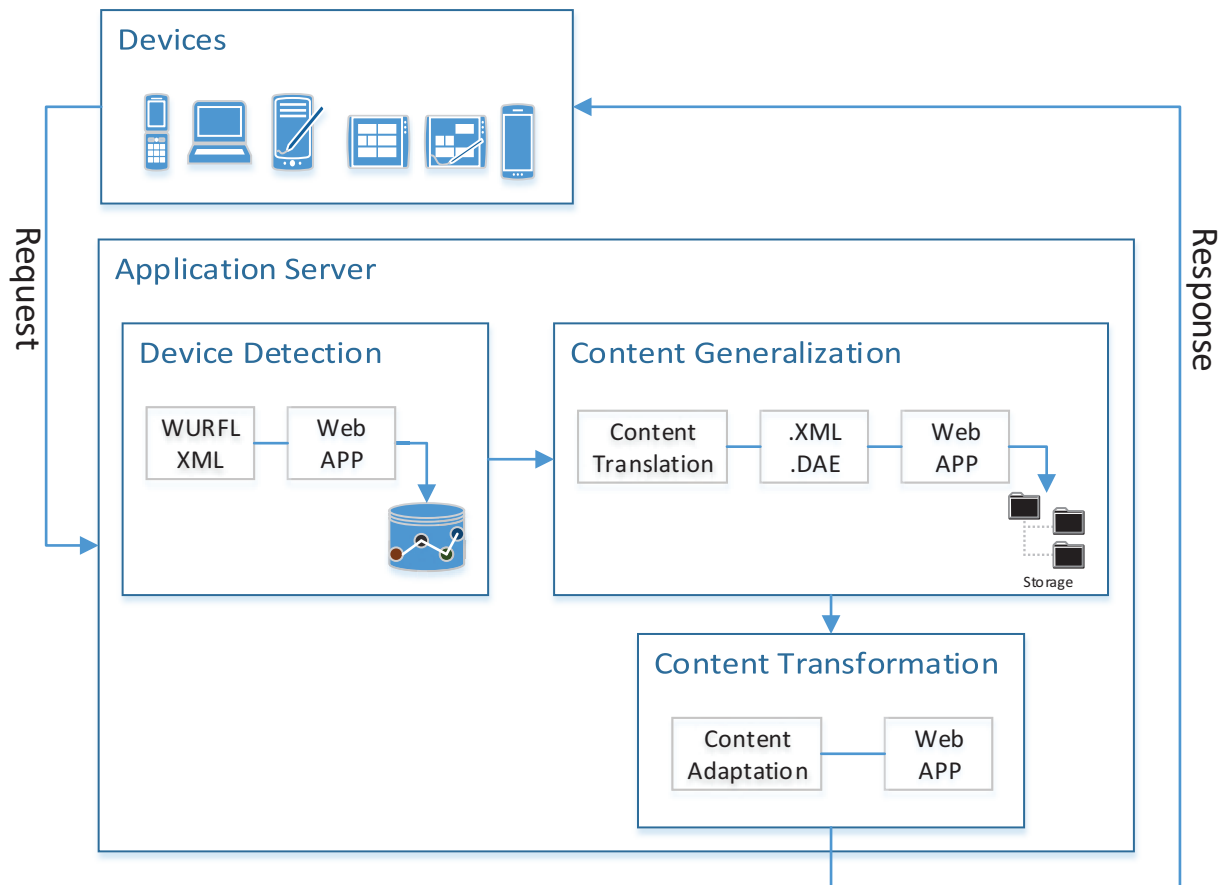
Figure 2. Présentation simplifiée de l'architecture proposée

4.4 Évaluation de Smoothie

Pour sa validation expérimentale, Smoothie, a été mis en œuvre et testé comme un composant très important de Language Learning Game (LLG). LLG est une application développée à l'Université des Sciences Appliquées d'Offenbourg, Allemagne, dans le cadre de son master par Razia Sultana sous la supervision du Professeur Dr. Andreas Christ et du Dr Markus Feisst. Des fonctions complémentaires ont été développées pour permettre la coopération avec l'architecture proposée. La Figure 3 ci-dessous montre une version simplifiée du processus du jeu incluant la détection des appareils communicants, la généralisation de contenu et la conversion de ce contenu. L'objectif du jeu est d'aider un apprenant adulte à acquérir une langue étrangère par le biais d'un outil accessible à tout moment et depuis n'importe où. LLG est proposé aux étudiants du programme de

master international par le département de langue de l'Université des Sciences Appliquées de Offenbourg.

LLG permet aux étudiants de pratiquer la langue à apprendre en utilisant un appareil mobile. LLG nécessite un superviseur qui a une très bonne maitrise de la langue pour effectuer un contrôle final. L'apprentissage d'une nouvelle langue nécessite un grand nombre d'interactions avec d'autres personnes. LLG peut répondre à l'ensemble de ces exigences par la mise en œuvre du système proposé sur un téléphone mobile sans demander une présence physique à un certain endroit et sans que le participant ait à attendre une quelconque réaction. Le formateur ou superviseur va créer le jeu et charger des consignes et du contenu didactique dans un format de données quelconque; Il décide du nombre total de membres pour chaque groupe, ainsi que les conditions de fin du jeu. Les participants écrivent des phrases et envoient du contenu associé, le but est de construire une histoire courte dans la langue étudiée. Quand un membre du groupe écrit et envoie une phrase, les autres membres du groupe peuvent la contester en proposant une autre version de la phrase ou ils peuvent simplement être d'accord avec la phrase proposée. Ils ne sont autorisés par la règle du jeu qu'à corriger des fautes d'orthographe ou de grammaire dans leur propre proposition. Ensuite tous les membres du groupe recevront la phrase proposée et les versions éventuellement corrigées. A ce stade, chaque membre du groupe doit voter pour la version qu'il préfère. La phrase ayant reçu le plus de suffrages sera sélectionnée. En cas d'égalité des voix la première soumission sera gagnante. Chaque membre du groupe doit proposer une phrase à chaque tour. Généralement, le jeu se termine après 15 à 20 tours. A l'issue du jeu, l'histoire courte finale est envoyée à tous les membres du groupe ainsi qu'au superviseur qui en fera la correction. Après cela, tous les participants recevront les corrections du formateur de toutes les erreurs qui se sont produites pendant le jeu. Cette correction d'ensemble est nécessaire, car il pourrait se produire une situation où la majorité des membres du groupe serait d'accord avec une phrase incorrecte. A la fin, les corrections apportées par le superviseur aideront les participants à apprendre et à améliorer leur niveau dans la langue.

Les raisons pour lesquelles LLG a été choisi sont les suivantes :

1) Un formateur peut utiliser n'importe quel type de contenu y compris des données 3D

2) Les étudiants internationaux ont à leur disposition différents types de dispositifs qu'il n'est pas possible de prévoir

La mise en œuvre de Smoothie a permis à LLG de devenir une application réellement indépendante des appareils communicants et a montré que cette liberté apparait effectivement comme une caractéristique essentielle. Il offre une liberté nouvelle au fournisseur de contenu à qui il permet d'utiliser n'importe quel type de format de données sans avoir à se préoccuper des dispositifs utilisés par les utilisateurs. Dans le même temps, il offre à l'utilisateur final la possibilité d'utiliser n'importe quel type d'appareil quand et où il veut en fonction de sa disponibilité ou autres convenances personnelles.

Figure 3. Organigramme simplifié du jeu (LLG) processus

4.5   Résultats expérimentaux

Les tests pour la validation expérimentale ont été réalisés avec

- 3 systèmes d'exploitation différents, à savoir Windows 8, Android 4.1 et 4.4 et LinuxMint 16

- 4 navigateurs différents, à savoir Mozilla Firefox 27, Google Chrome 33, Safari 5.1.7, et Internet Explorer (IE) 10

- 4 types différents de dispositifs à savoir : ordinateur de bureau (Linux), ordinateur portable (Windows), tablette (Android 4.4) et téléphone mobile (Android 4.1)

- Les formats de données 3D tels que : 3ds, obj, dae, stl, x3d, nappe, lwo, ma, blend, FBX, et wrl

- Des formats de documents et des formats d'image tels que doc, pdf, txt, jpg, docx, odt, et rtf

Pour les différentes configurations, la plupart des tests ont été menés avec succès sauf dans les cas d'utilisation de IE et Safari avec une machine fonctionnant sous Windows, en raison de l'absence de WebGL par défaut. Il est certainement possible d'obtenir un résultat positif pour ces cas également, mais il serait nécessaire de disposer de plus de temps et de ressources.

Bien que l'architecture proposée permette une communication indépendante du dispositif utilisé, les appareils mobiles sont dépendants de l'évolution des systèmes de communications, plus particulièrement depuis la 3ème génération. Cependant nous les considérons comme entrant dans le champ d'application de notre proposition, en raison de leurs caractéristiques permettant des vitesses de transfert des données acceptables.

5. Bilan, conclusion et perspectives

L'architecture proposée peut être utilisée comme un composant de n'importe quel système dans lequel la variété des formats de données, des contenus et des dispositifs d'affichage doivent être pris en charge indépendamment de l'objectif de l'application. L'architecture que nous proposons est à la fois applicable à des fins pédagogiques et à des fins commerciales, pour répondre à des besoins spécifiques individuels ou de groupes. Aujourd'hui un énorme volume de contenu est créé tous les jours et il y a un très grand nombre de dispositifs de communication disponibles. Ce contenu est accessible à une part croissante de la population mondiale. Cet accès à l'information devient « de facto » un droit, indépendamment du lieu d'exploitation ou du statut social. L'accès à l'information a aussi tendance à devenir plus libre (même si des entraves existent). L'architecture

proposée est donc une première étape vers l'amélioration de l'accès généralisé à l'information. Il contribue à l'ouverture d'une « porte » pour que tout le monde puisse communiquer avec tout le monde à travers une multitude de dispositifs. L'Education Pour Tous (EPT) peut devenir ainsi une réalité.

Cette thèse présente un prototype qui démontre un concept, une méthode, un système, avec une validation expérimentale satisfaisante, mais elle ne propose pas un produit. Des projets opérationnels pourraient devenir une réalité après de futures recherches et développements complémentaires, qui pourraient consister à:

- Mener à bien des tests qui pourraient être effectués en utilisant d'autres navigateurs disponibles, d'autres systèmes d'exploitation, d'autres formats de données et d'autres types d'appareils, le plus grand nombre possible.

- Elaborer une technique permettant de générer un format de données généralisé comportant le moins de perte de compression de données, quand il est nécessaire de transmettre ces données par l'Internet.

- Concevoir une utilisation plus efficace des capacités des serveurs de sorte que le système soit moins dépendant des fonctionnalités de l'appareil du client final. A ce stade, des installations de type cloud computing pourraient être introduites et mises en œuvre pour accueillir l'architecture proposée.

- Les fonctions d'importation et d'exportation d'un logiciel open source renommé à savoir Blender est utilisé pour préparer le format de données 3D généralisé. Ainsi, l'architecture proposée est dépendante de certaines fonctionnalités de Blender. Il conviendrait de concevoir une méthode pour étendre les fonctionnalités de Blender de sorte que tous les formats de données 3D puissent être importés et exploités, avec un minimum de contraintes.

etc.

Extended Summary

1. Introduction

Today's network landscape consists of many different network technologies, a wide range of end-devices with large scale of capabilities and power, an immense quantity of information and data represented in different formats. Research on 3D imaging, virtual reality and holographic techniques will result in new user interfaces (UI) for mobile devices and will increase their diversity and variety. A lot of effort is being made in order to establish open, scalable and seamless integration of various technologies and content presentation for different devices including those that are mobile, considering the individual situation of the end user. Till today the research is going on in different parts of the world but the task is not completed yet.

2. Problems identified based on interoperability of communicating devices

2.1 Assessment of state of the art

The various kinds of devices used by different users or in different times or in parallel by the same user which is not predictable and have to be recognized by the system in order to know the device capabilities.

2.2 Management of fixed-mobile duality of user and used devices

In a stable place such as at home or in the office it is more convenient to use a PC. While on the move it is very obvious that a user would like to access the same content with the same outlook and feel by using his mobile device. So a system is necessary which is device dependent from the point of communication functionality, interactivity, 3D capabilities, and information presentation and depth. But at the same time it must be device independent from the point of information access, and (a/) synchronous communication possibilities. Typically, content management systems are unable to handle these two opposing facts with one common data storage for the content. At this point it is necessary to mention Responsive Web Design (RWD) to avoid misunderstanding.

RWD is a web design approach aimed at crafting sites to provide an optimal viewing experience like easy reading and navigation with a minimum of resizing, panning, and scrolling across a wide range of devices such as from mobile phones to desktop computer monitors. A site designed with RWD adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images and CSS3 media queries. So RWD is taking care of the content presentation in different devices but not the content itself.

Some content providers are offering different versions of content to support different devices, where the maintenance of the content is highly labor intensive, cost and time consuming. Another problem is, to make these different versions of content the authors and content developers need to have knowledge on different devices and their behavior in order to be able to generate content focusing on the abilities of these devices for example 3D display. But most authors and content developers do not have this precise technical knowledge. On the other hand, people with the technical knowledge do not have necessarily the knowledge to prepare the content in a proper way. The situation seems like a deadlock. So there is a problem where a suitable and dynamic solution is necessary

2.3 Justification of the study

The overall number of users of specialized content or interactive applications is too low to adapt the application/content to all possible devices manually. This is what we have seen.

2.4 Technology we offer in this context

Content and User Interfaces could include different kinds of data format such as text, image, audio, video, 3D Virtual Reality data and other upcoming formats. Also, devices may include 3D imaging functionalities. The system should be able to deal with all the existing and upcoming formats of data without requiring any huge enhancement. In other words, the communication system meeting the specifications outlined above should be able to automatically process all existing and future formats of data, to make them usable on a variety of devices that meet user demands.

3. Purpose and concept of the thesis along with a proposed solution

The goal of this research work is to find a way to solve the above stated problems by investigating system architectures to provide unconstrained, continuous and personalized access to the content and interactive applications everywhere and at anytime with different devices.

A very suitable and useful example of the use of such a system is mobile learning because of the large amount of varying devices with significantly different features and functionalities. This is true not only to support different learners, e.g. all learners within one learning community, but also to support the same learner using different equipment in parallel or at different times. Those applications may be significantly enhanced by including virtual reality content presentation. Whatever the purposes are, it is impossible to develop and adapt content for all mobile devices individually due to different capabilities of the devices, cost issues and author's requirement.

A solution should be found to enable the automation of the content adaptation process. In order to realize such a system three major requirements have to be fulfilled:

- Identification of the connected device

- Generation, structuring and storage of generalized content

- Transformation process from general content to optimized and device dependent content

4. Solution of the problem considered

4.1 Originality of the proposed solution: An architecture

The originality of this thesis could be summarized as

- The proposed architecture aims to provide device and content format independent communication

- The proposed architecture does not focus any specific service or group. It could be used for academic purposes as well as business purposes, even for individual private use

- The proposed architecture aims to use tools those are either open source or at least free to use so that the prototype could be open for everybody

4.2 Scope of the architecture

The scope of the thesis could be depicted as figure 1. The creation of content and the processed content after being displayed at the end user device, how user is dealing with the content does not lie under the scope of this thesis.
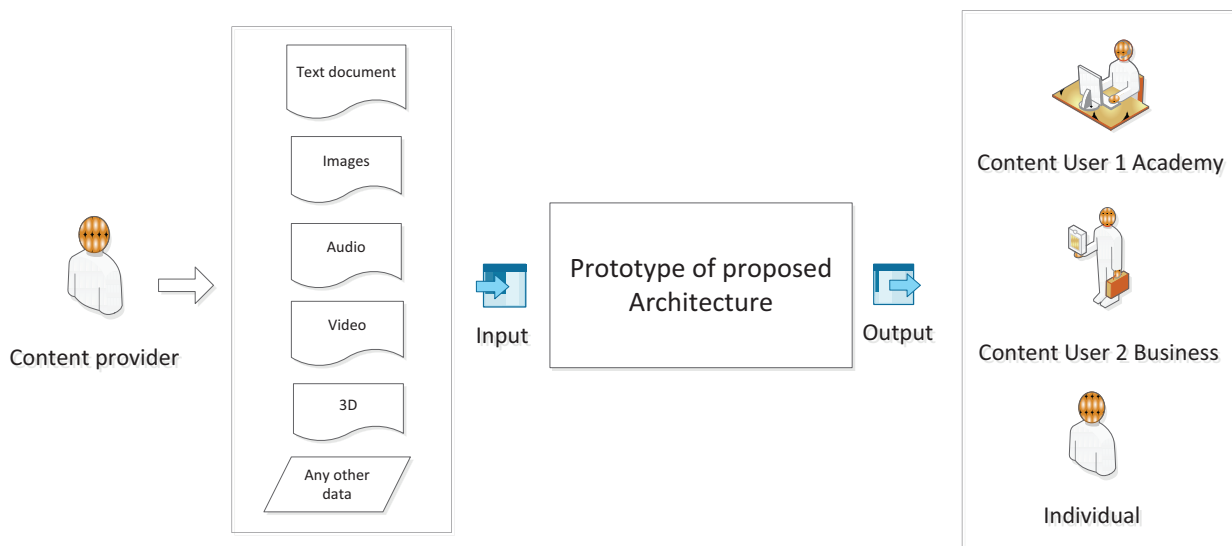
Figure 1. Pictorial representation of the scope of the thesis

4.3　　Presentation of the proposed solution: A new architecture named "Smoothie"

The word Smoothie is a noun, dedicated for a beverage. A smoothie is a blended emulsion of different products for example fruits; crushed ice, honey, syrup, yogurt, ice cream ingredients, soy milk, green tea, herbal supplements etc. and special cases can contain chocolate or peanut butter. So we can derive that a smoothie can contain any ingredient which is neither limited nor predictable, depends upon the recipe of an individual or a group such as board of director of a business institute who deals with food and beverages. The new architecture proposed as a solution of above mentioned problem has similar properties. It is an emulsion of different application and data formats. In terms of data formats anything could be used as an input to the proposed architecture which is neither limited nor predictable. As special case any 3D data could also be provided as content in to the architecture. Due to the similarity of properties in between the beverage smoothie and the proposed architecture, the architecture is named as "Smoothie".

Figure 2 shows the most simplified version of the system overview of Smoothie. First, it is detected whether the user is connecting to the system via mobile device or by desktop device by analyzing the HTTP-request header coming from the end client´s device. Wireless Universal Resource File (WURFL) non commercial version is selected for the description of the features of mobile devices and browsers because it is an XML configuration file which contains information about capabilities and features of many mobile devices in the wireless world. Also, the repository of device in WURFL is updated every day by contributors in the world. So it is an up to date specification that brings reliability in device data manipulation. Proposed architecture works with a combination of WURFL and a local database saved in the server. Regarding content, providers do not have to care about

the different mobile devices and to provide an optimized version of their file for every device. Content will be saved at first in its original format for safety reason then will be translated in a generalized format and stored in the database. A decided generalized format is XML or XML extended format such as DAE. Based on the identified device capabilities generalized content format will be transformed into device dependent manner and displayed to the end client. So on the fly data conversion and serialization is happening in server side. For the prototype web browser is used along with the concept RWD (Responsive Web Design), as a container of the content to be presented at end client´s side. Any time it is possible to extend the serialization by implementing other available pipeline to process other output format such as PDF, WML etc.
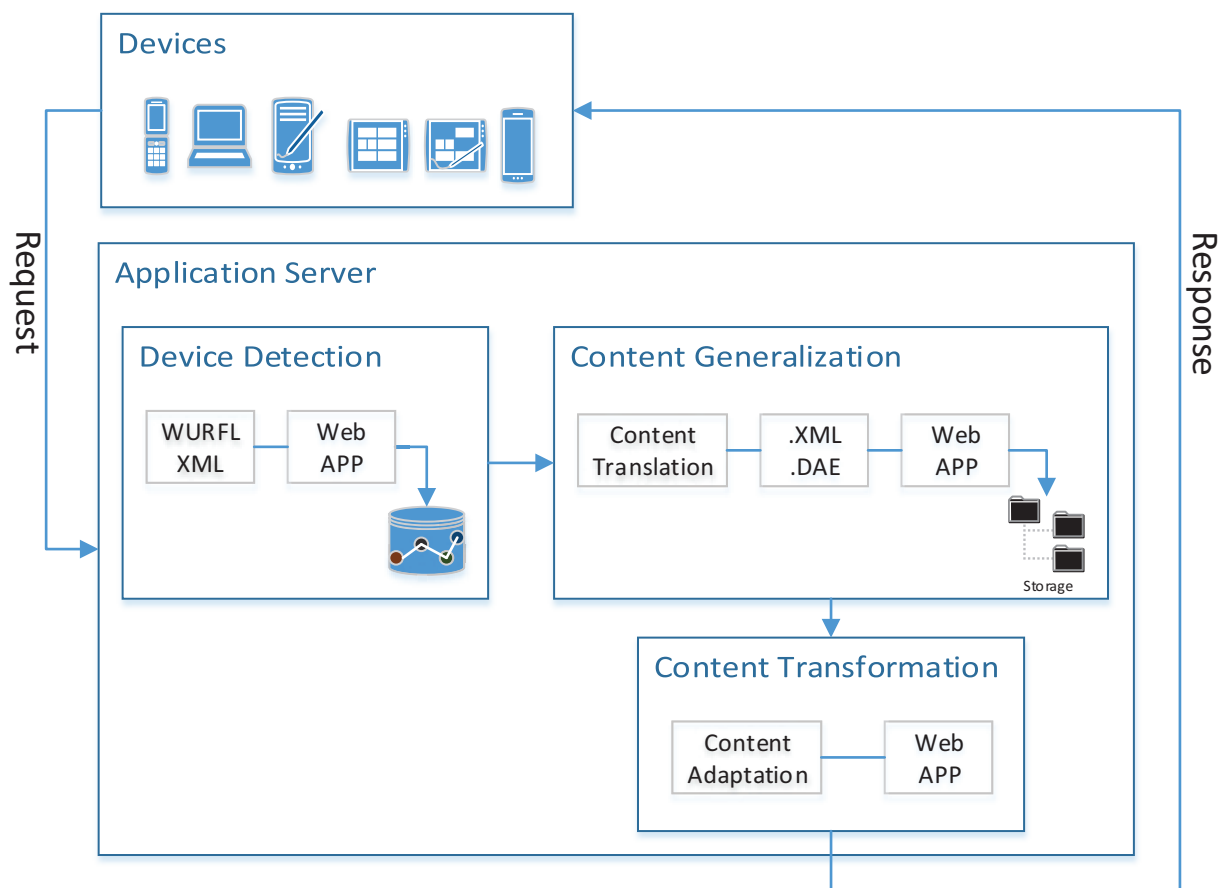
Figure 2. System overview - simplified version

4.4   Evaluation of Smoothie

For experimental validation of Smoothie, it is implemented and tested as a very important feature of Language Learning Game (LLG). It is an application developed in Offenburg University of applied sciences Germany as a master thesis by Razia Sultana under supervision of Professor Dr. Andreas Christ and Dr. Markus Feisst. Enhancement has been made to make it cooperate with the proposed architecture. Figure 3 below shows a simplified version of the game process along with device detection, content generalization and content transformation, in a flowchart.

The focus of the game is on adult learner, to support them by providing a tool which is anytime and anywhere accessible, for foreign language acquisition. LLG is offered by Language department of Offenburg University of Applied Sciences Germany for international master program students.

The LLG allows language learners to practice the target language using a mobile device and requires a supervisor who has very good knowledge of the foreign language to perform an overall check at the very end. New language learning needs a lot of interaction with other people. LLG can meet this entire requirement by implementing the proposed system on a mobile phone without asking for a physical presence on a certain place and without keeping the participant waiting for an action actively. The instructor or supervisor will create the game and upload instructions as learning content having any data format; decide about the total number of group members in each group, and the game end condition. Participants will write sentences and upload supporting content and the aim is to build a short story in the desired language. When a group member writes and sends a sentence the other members of the group will have a chance to dispute by proposing another version of the sentence or they can simply agree with it. They are only allowed by the rule of the game, to change spelling or grammatical mistakes in their proposed version. Afterwards all group members will receive the proposed sentence and possible corrected versions. At this stage, every group member has to vote for the version the individual prefers. The sentence with the most votes gets selected. In case of equal votes the first submitted one wins. Each group member has to create one sentence per round. Typically the game is finished after 15-20 rounds. The final short story is sent to all group members as well as to a supervisor who will perform an overall correction. After that, all the participants will receive the supervisor's corrections of all the mistakes that occurred during the game. This overall correction is necessary, because there might be a situation where the majority of group members agree with an incorrect sentence. At the end, the corrections made by the supervisor will help them to learn and improve their level in desired language.

As a method of evaluation LLG was carefully chosen because of below stated reasons

3) As a learning content language teachers may use any kind of data including 3D

4) All the international students having different kinds of devices which is not possible to predict

The implementation of Smoothie made LLG a real device independent application and added freedom as a very necessary feature. It has provided a freedom to the content provider to be able to

use any kind of data format without having any thought in mind regarding the end client device. At the same time it has also given a freedom to the end client to use any kind of device whenever and wherever they want based on the availability and their personal preference.

Figure 3. Simplified flowchart of the game (LLG) process

4.5   Experimental results

Test has been conducted so far by using

- 3 different operating systems namely Windows 8, Android 4.1 & 4.4 and LinuxMint 16
- 4 different browsers namely Mozilla Firefox 27, Google Chrome 33, Safari 5.1.7, and Internet Explorer (IE) 10.
- 4 different types of devices namely desktop (Linux), laptop (windows), tablet (android 4.4) and mobile phone (Android 4.1)
- 3D data formats such as 3ds, obj, dae,stl,x3d,ply, lwo, ma, blend, fbx, wrl,
- document and image formats such as doc, pdf, txt, jpeg, docx, odt, rtf

Among above stated test parameters most of them were successful except IE and Safari in Windows machine due to lack of default WebGL support. Work around is possible to make the test successful for both the cases but it requires more time and resources.

Even though the proposed architecture is allowing a device independent communication, the mobile devices lie under the lifetime of telecommunication system from 3rd Generation on, are considered under the scope of this thesis due to necessary speed of data transfer capability.

5. Conclusion and future work

The proposed architecture could be used as a part of any system where varieties of devices along with varieties of data or content formats have to be supported independent of purpose. It is equally applicable for academic purposes as well as business purposes to meet requirement either for a group or for individual interest. Nowadays enormous content is being created everyday and there are plenty of communicative devices available out there. As a part of the world population it is a right of everybody, independent of place and social status, to have access to information. This right is being violated by creating intentional barrier. The proposed architecture is the first step towards the betterment of the world by opening a door to be able to communicate with everybody and anybody so that Education For All (EFA) scheme can literally come true.

This thesis provides a prototype to prove the concept but not a product. There are open issues and scope for further research and development such as

- Test could be made by using other available browsers, operating systems, different data formats and different types of devices as many as possible.

- How to make loss less compression of generalized format of data, when it is necessary to be send through internet

- How to make the efficient use of the server capabilities so that the system is less dependent upon the capabilities of end client´s device. At this point facilities of cloud services could be introduced and implemented along with the proposed architecture.

- Import and Export functions of renowned open source software namely Blender is used to prepare generalized 3D data format. So, the proposed architecture is dependent upon the given functionalities of Blender. How to extend given functionalities of Blender so that any 3D data formats along with supporting files could be imported

  and so on.

Summary of the Content in Each Chapter

CHAPTER 1 : Introduction

This chapter describes the need of a system that provides the usual functionalities of communication and also supports the user with all information he needs independent of his used device. This system combines the two aspects: uniform communication and user support in a modular, scalable, and user-centered communication platform.

CHAPTER 2 : Literature Review

This chapter provides the most recent stage in the development of device and content independent communication including 3D imaging as content, incorporating the newest ideas and the most up-to-date features. Throw the state of the art it is shown that there is no project existing so far which literally supports heterogeneity of devices along with heterogeneity of data to support end users independent of their purpose of use.

CHAPTER 3 : Proposed Solution

This chapter explains the aim, concept, scope, originality and goal of the thesis. It also proposes a solution of the problem considered in former chapters, a new architecture called Smoothie. Prototype of Smoothie is using Browser as a container of data, requested by the user, to demonstrate how to establish a device and content independent communication in one hand to provide freedom to the content author to upload any format of content without having the stress if the target user device is able to display the content, on the other hand to provide a freedom to the end users to be able to use any communicative device they have available at hand. Smoothie is easily extendible to be able to use other container for data such as PDF, WML, etc.

CHAPTER 4 : Theoritical Concepts Behind Smoothie

This chapter provides abridged explanation of all theoretical concepts involved in Smoothie. It also introduces three dominant parts of this architecture namely identification of device, preparing generalized content and transformation of content. In identification of device for collection of de-

vice capabilities WURFL API is being used as device description repository. To save collected information about a device and to make the device identification process faster in case of device recognition instead of identification, all information is saved in an object oriented database Db4o. In preparing generalized content a scientifically proven approach behind selecting XML and XML extended DAE for 3D data as format of generalized content is described. In transformation of content it is declared that under the scope of this thesis transformation of content is used as a synonym of content adaptation. Later abridge discussion on initial recommendation for content adaptation from W3C and existing technologies in use today to serve the same purpose is provided. The discussion leads to the decision for Smoothie to use any one or a combination of several approaches for content adaptation based on device capability, connectivity and requested content.

CHAPTER 5 : Experimental Validation of Smoothie

This chapter is divided into two parts namely implementation and evaluation. The implementation part started with introduction of all the used tools to implement Smoothie followed by an assortment of it´s technological contrivances. Implementation is divided into three parts exactly in the same order described in chapter 4 as three dominant parts namely device identification, preparing generalized content and transformation of generalized content. Device identification works closely together with user authentication due to security issues. Since Smoothie will be integrated with other application, user authentication conditions are extensible based on the necessity or design of the other application. In preparing generalized content first all the data is saved in a file system with their original formats to avoid difficulties in case of lossy or corrupted data through any phase of the whole application. Apache Tika and Blender are used as tool to prepare generalized content. In transformation of content, mainly a server side content adaptation is taking place which is easily extensible to use client side and hybrid technique. Apache cocoon and ThreeJS are used as tool to serialize the data at end client´s device. Evaluation part started with introduction of a collaborative language learning game (LLG) followed by advantages of integrating Smoothie with it. LLG is an example of Smoothie serving academic purpose. Later part of this chapter provides sample of test results along with an analysis of achieved results.

CHAPTER 6 : Conclusion

This chapter provides a general conclusion mentioning how Smoothie could be useful for betterment of world population. Proposed architecture Smoothie is a prototype to prove the concept. It is stand alone architecture to be integrated with any other application where different devices and

different data formats have to be supported simultaneously. Theoritically according to design Smoothie has the ability to deal with any format of data but in practice due to the limitation of used tools and dependency upon those used tools makes smoothie limited regarding dealing with data formats. In terms of data visibility as well Smoothie behaves different based on device capabilities, connectivity, and installed application on target device. This chapter also mentions achievements of smoothie and concludes with the possibilities of future improvements and open questions.

Chapter 1   Introduction

Since the early attempts to support human communications by technological media such as the telegraph or telephone, a long time has passed. Many new technologies of communication have been developed and people became accustomed to them. Nowadays, we have so many ways to pass messages to each other that it becomes a complex task to maintain all these different systems. Additionally new methods to communicate not only with human beings but also with machines arise. This includes a range of applications from simple Web-based software up to the completely voice-controlled household. The speed of development brings benefits together with problems.

As first problem the growth of communicative device use could be mentioned. The use of communicative device is growing every day. According to International Data Corporation (IDC) Worldwide Quarterly Smart Connected Device Tracker, the worldwide smart connected device space will continue to surge, with overall shipments surpassing 2 billion units by the end of 2015. In terms of device mix, total PC shipments accounted for 28.7% of the smart connected device market in 2012 while tablets accounted for 11.8% and Smartphone for 59.5%. By 2017, total PCs are expected to drop to 13%, while tablets and Smartphone will contribute 16.5% and 70.5% respectively to the overall market. The demand is going to be shifted from the more expensive PC category to more reasonably priced Smartphone and tablets. The table 1.1 below shows an estimated statistics of smart connected device shipments in 2013 and 2017. In 2017 the expected growth of smart connected device shipment is 58.1% compare to 2013 (shipments in millions) [1].

Cisco says today's 341 million Internet-of-Things (or machine-to-machine) connections are poised to grow to two billion in the coming years as wearables and smart infrastructure come online. By 2018, connected mobile devices will monthly flood the network with 15 times more data than all Internet traffic in 2000 [2]. Cisco IBSG predicts there will be 25 billion devices connected to the Internet by 2015 and 50 billion by 2020. That means 6.58 connected devices per person. Those devices will or may have different operating systems along with variety of capabilities. According

to International Data Corporation press release on 4th September 2013, only for Smartphone there will be at least four leading operating systems (Figure 1.1) [3].

Product Category	2013 Unit Shipments	2013 Market Share	2017 Unit Shipments	2017 Market Share	2013—2017 Growth
Desktop PC	134.4	8.6%	123.11	5%	-8.4%
Portable PC	180.9	11.6%	196.6	8%	8.7%
Tablet	227.3	14.6%	406.8	16.5%	78.9%
Smartphone	1,013.2	65.1%	1,733.9	70.5%	71.1%
Total	**1,556**	100%	**2,460.5**	100%	**58.1%**

Table 1.1 Smart Connected Device Market in 2013 and 2017

It is important to note that these estimates do not take into account rapid advances in Internet or device technology; the numbers presented are based on what is known to be true today [4]. Table 1.2 below shows key global telecom indicators for the world telecommunication service sector [5]. Here all the figures are provided based on estimation.

	Global	Developed nations	Developing nations	Africa	Arab States	Asia & Pacific	CIS	Europe	The Americas
Mobile cellular subscriptions (millions)	6,915m	1,515m	5,400m	629m	410m	3,604m	397m	780m	1,059m
Per 100 people	95.5%	120.8%	90.2%	69.3%	109.9%	89.2%	140.6%	124.7%	108.5%

Table 1.2 Estimated Key Global Telecom Indicators in 2014

It is a necessity that all those connected devices having different platform and capabilities are able to communicate with each other. That means all of those devices have to be able to exchange data through a channel or network and those data have to be understandable by the destination device. As an optimization of this necessity it could also be added that all devices are synchronized in terms of data, with each other. Moreover the user are able to access their resources not only independent of device and data type but also independent of internet connectivity.
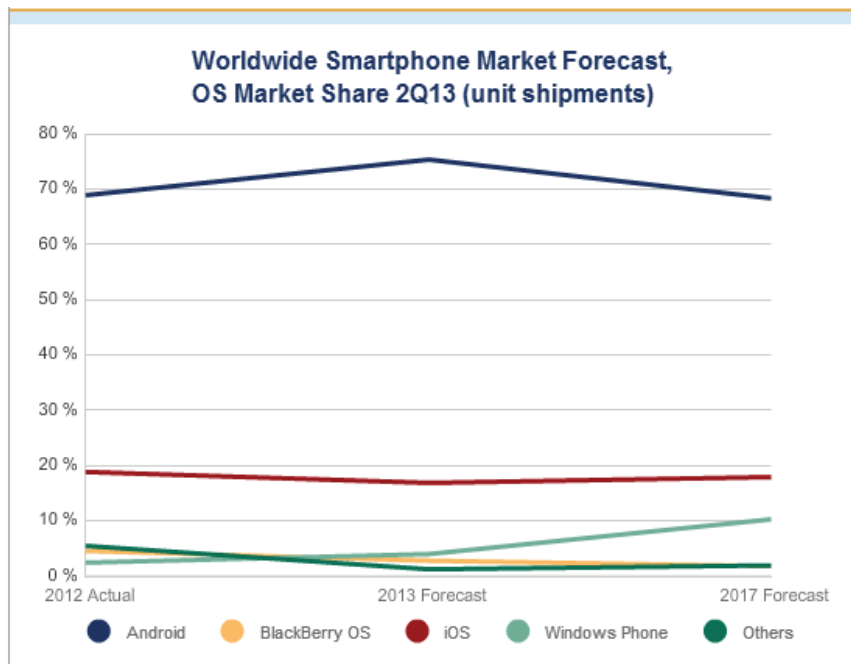
Figure 1.1 Prediction of Smartphone statistics by International Data Corporation

Now, is the situation imaginable that your own devices are not able to communicate with each other? Communicate not in terms of data exchange, but understanding the data and to be able to work with by the destination device.

Before answering this question it is necessary to be introduced with the second problem namely different formats of data. According to Internet Assigned Numbers Authority (IANA) on 11th June 2014 there are round about 1500 registered media types divided in 9 different groups namely

- Application
- Audio
- Example (so far no entry available)
- Image
- Message
- Model
- Multipart
- Text and
- video

Among those 1500 media types 69 are dedicated to text, 77 are for video, 21 for models (for example 3D models), 47 for images, and 142 for audio [6]. Approximately, 2500 file extensions are being supported by 1500 media types where 300 are 3D graphics formats, more than 200 CAD file

formats, more than 500 audio file formats, around 600 image file formats, approximately 400 video file formats and around 300 text file formats [7]. In one hand so far there is no application or device found that is able to deal with all or at least most used data formats. On the other hand use and need of use of those varieties of data is raising everyday. Aaccording to CISCO forecast overall mobile data traffic is expected to grow to 15.9 exabytes per month by 2018, nearly an 11-fold increase over 2013. Mobile data traffic will grow at a Compound Annual Growth Rate (CAGR) of 61 percent from 2013 to 2018 (Figure 1.1) [2].
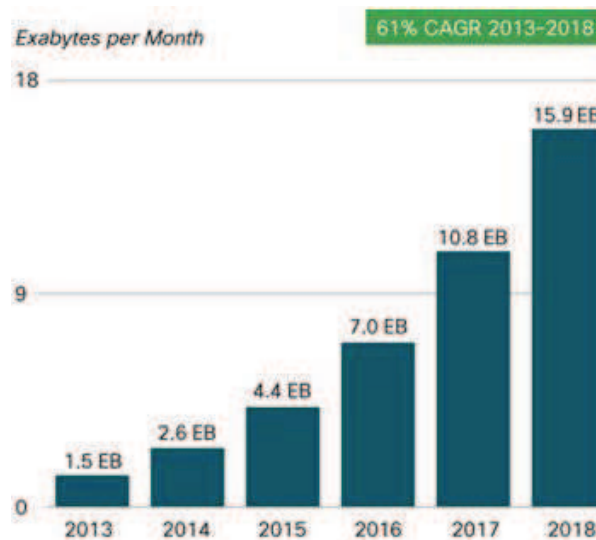
Figure 1.2 Forecast of mobile Data Traffic by 2018

Below a list is given to show mobile data traffic milestones within the next five years [2].

- Monthly global mobile data traffic will surpass 15 exabytes by 2018.

- The number of mobile-connected devices will exceed the world's population by 2014.

- The average mobile connection speed will surpass 2 Mbps by 2016.

- Due to increased usage on smartphones, smartphones will reach 66 percent of mobile data traffic by 2018.

- Monthly mobile tablet traffic will surpass 2.5 exabyte per month by 2018.

- Tablets will exceed 15 percent of global mobile data traffic by 2016.

- 4G traffic will be more than half of the total mobile traffic by 2018.

- There will be more traffic offloaded from cellular networks (on to Wi-Fi) than remain on cellular networks by 2018.

A ubiquitous data presentation method that covers a major protion of overall data traffic is 3D. For a better understanding of existing systems that are able to deal with 3D data, knowledge regard-

ing 3D data itself and it´s usefulness are pre-requisites. The following list shows different sectors where 3D data is used for most useful applications:

- Edutainment (Education along with entertainment)
- Games
- Product presentation
- Navigation
- Surgery and maintenance
- Search and rescue

Use of 3D data as image, virtual reality scene and augmented reality is increasing due to the acceptance and popularity among the world population. 3D representation is a long step ahead compare to 2D representation in performance, cost etc. Today for developers plenty of 3D libraries are available to be used for free of cost. Those libraries are either hardware optimized or software optimized. As it is known that hardware optimized library increases the operational speed but limit the portability. All of the devices currently available in the market those supports 3D data are hardware optimized. This is the reason behind lack of resource sharing in terms of content among 3D data supported devices. The scenario is equally true for Applications dedicated to 3D data, for example 3D modeling applications. As a real life example two most popular 3D modeling applications could be mentioned named Blender and Maya. Blender is a well known open source 3D modeling application; it´s native data format is .blend. Autodesk Maya is another well known 3D modeling application having native data format .ma. Autodesk Maya does not support .blend and Blender does not support .ma. Right now more than 300 different 3D data formats are available in the market [7]. See appendix A for a list showing most used 3D computer graphics formats.

It is impossible for any 3D enabled communicative device to deal with all of most used 3D data formats. Initially .obj later .step and .iges was designed and used for 3D asset exchange purposes. Now .dae is a ubiquitous digital asset exchange format that makes interchange of file format for interactive 3D applications easier. Still the requirement of displaying 3D content independent of device remains unresolved.

The above mentioned problem becomes severe when 3D on mobile devices including wearable computers comes under consideration. At the turn of the millennium, interactive mobile graphics sounded like a contradiction in terms. Mobile devices, such as cellular phones, had very small monochrome displays, slow CPUs, and little memory. Today, several APIs exist both for 2D and 3D

graphics, there are hardware designs to accelerate their execution, and the features of mobile graphics engines begin to rival those of desktop systems. The change has been very quick, promising and has a lot of potential.

There are several reasons behind limited resources of mobile devices compare to desktop. The most fundamental reason is that mobile devices have to obtain the power from batteries. As the amount of power is limited, also the performance of the system must be limited otherwise the battery runs out too quickly. The preferable small size of the devices also limits how much power can be consumed, even if better batteries were available. Liquid cooling or fans definitely are not practical, so great care must be taken in thermal design to avoid hot spots that might burn the whole chip. Moreover, most mobile devices are mass-market consumer products and their business life cycle is very short in respect to time, for such a device customers are typically not ready to pay as much as they might for a high-performance engineering or even home entertainment workstation.

Still during the course of time the resources of a mobile device have improved. The CPU development has followed Moore's law1 yielding smaller and higher performance CPUs. Similarly, the amount of available memory has increased. Graphics, and display has also been improved a lot and all those improvements are still a continuous process. That development was first influenced by the demand from digital cameras, though now the greatest demand probably comes from mobile phones itself. A typical mobile phone around year 2000 had a 84 × 48-pixel 1-bit monochrome display, refreshed a few times per second, but a mobile phone from the year 2013 for example Nokia Lumia 920 has 1280 × 768 resolution with a refresh rate of 60hz.

All modern smart phones include a GPU. The first generation of mobile GPUs implemented the fixed-functionality graphics pipeline of OpenGL ES 1.0 and 1.1. Those GPUs were designed for 3D graphics. The current mobile GPUs are much more flexible and support OpenGL ES 2.0 API, allowing programmers to run fairly complicated shaders at each pixel [8]. Technological development in terms of hardware and software of mobile devices made it possible to have 3D as virtual reality, augmented reality or mixed reality being displayed on our mobile device at hand, which is any time any where present and ready to use for business, learning, and entertainment or for a mixed purpose.

Now the next questions to be answered are- do we need dedicated devices to be able to use application from different sectors; for example is it possible to use very same device for 3D games,

reading text, view image, listen to audio, to see a 3D model of a product and see health report as 3D data for medical purpose? Are different devices along with 3D dedicated hardware able to communicate with each other in an understandable way; for example is it possible to send a 3D model from a handheld device to be printed by a 3D printer?

Right now within one word it is possible to answer all the above mentioned questions and the answer is "No". This barrier is created intentionally or no big step was taken so far globally to establish a device and content independent communication due to business purposes. We have many standards, protocols and specifications such as Zigbee, NFC, and Bluetooth etc to make communicative devices able to be connected and exchange data. But we do not have any standard or specification to make it sure that the sent data will be understood by the destination device independent of device type and capability.

Hence, the problems namely 1) heterogeneity of devices 2) heterogeneity of data formats including 3D data and 3) ever increasing use of both (1 & 2) in an unpredictable combination by the world population is introduced.

1.1 Identified Problems Based on Interoperability of Communicating Devices

To tackle above mentioned issues a system that eases the management of communication is needed. This system should provide an easy way to deal with different formats of data as well as the functionality to translate those between potentially incompatible media. Furthermore, the system should be easy to use and should offer various services to its users with a very low cost, preferably for free. Before starting to look for or design such a system, it is important to have a closer look to identify what are the specific problems those, the systems has to solve.

1.1.1 Fixed-mobile duality of user and their preferred device types

In a stable place such as at home or in the office it is more convenient to use a PC. While on the move it is very obvious that a user would like to access the same content with the same outlook and feel by using his mobile device. So a system is necessary which is device dependent from the point of communication functionality, interactivity, 3D capabilities, and information presentation and depth. But at the same time it must be device independent from the point of information access, and synchronous or asynchronous communication possibilities.

1.1.2   Limitation of typical content management systems

Typically, content management systems are unable to handle above mentioned two opposing facts with one common data storage for the content. At this point it is necessary to mention Responsive Web Design (RWD) to avoid misunderstanding [9]. RWD is a web design approach aimed at crafting sites to provide an optimal viewing experience like easy reading and navigation with a minimum of resizing, panning, and scrolling across a wide range of devices such as from mobile phones to desktop computer monitors. A site designed with RWD adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images and CSS3 media queries. So RWD is taking care of the content presentation in different devices but not the content itself.

1.1.3   Assessment of required appropriate knowledge

Some content providers are offering different versions of content to support different devices, where the maintenance of the content is highly labor intensive, cost and time consuming. Another problem is, to make these different versions of content the authors and content developers need to have knowledge on different devices and their behavior in order to be able to generate content focusing on the abilities of these devices for example 3D display. But most authors and content developers do not have this precise technical knowledge. On the other hand, people with the technical knowledge do not have necessarily the knowledge to prepare the content in a proper way. The situation seems like a deadlock. So there is a problem where a suitable and dynamic solution is necessary.

1.1.4   Assessment of predictability of device to be used

The various kinds of devices used by different users or in different times or in parallel by the same user which is not predictable and have to be recognized by the system in order to know the device capabilities.

1.1.5   Low number of members in any target group

The overall number of users or members of specialized content or interactive applications is too low to adapt the application/content to all possible devices manually. This is what we have seen.

As example of a target group, a distant mobile learning student group can be mentioned because of the large amount of varying devices with significantly different features and functionalities. In this example the system has to support not only different learners, e.g. all learners within one learning community, but also to support the same learner using different equipment in parallel or at different times. Used applications by distant mobile learners may be significantly enhanced by

including virtual reality content presentation. Whatever the purposes are, it is impossible to develop and adapt content for all mobile devices individually due to different capabilities of the devices, cost issues and author's requirement.

1.1.6   Compatibility with existing and future data formats including 3D

Content and User Interfaces could include different kinds of data format such as text, image, audio, video, 3D Virtual Reality data and other upcoming formats. Also, devices may include 3D imaging functionalities. The system should be able to deal with all the existing and upcoming formats of data without requiring any huge enhancement. In other words, the communication system meeting the specifications outlined above should be able to automatically process all existing and future (XML compatible) formats of data, to make them usable on a variety of devices that meet user demands.

1.1.7   Assessment of cost and purpose of use

The system has to support any data, any device and any user so that access to required resource in terms of data is available any time any where independent of the age, sex, localization, social status and used device of the user. In this regard it is neither predictable for which purpose user is going to use the system nor possible to assume if the system is going to be used by a single user or by a group of users. The ideal case for such a system would be

- if it is available for free of cost
- If It is possible to use independent of purpose such as academic, business or private
- If it is integrable to any other application to serve user defined requirements

To meet those goals, very carefully selected software package has to be used to design the system. All software packages used as tool in implementation have to be either open source or at least free to use according to their licensing policy.

There already are systems that are capable of meeting some of those above mentioned functionalities (see chapter 2). But in respect of supporting the user with the given data in a fully satisfactory manner further extensive investigations are demanded.

Chapter 2   Literature Review

2.1   Research on Underlying Media Independent Communication

As mentioned in the Introduction, there already are systems that enable a communication independent of the underlying media. The first of its kind was the mobile people architecture (MPA) [10] developed by the Mosquito Group of the Stanford University. It is typical for device independent communication systems. The main goal of all such systems is to provide access to all its users independent of the actual communication devices used. In the mobile people architecture, this is mainly achieved by extending the traditional ISO-OSI reference model by a personal layer. This layer represents an individual person by managing all the owner's devices and their reach-ability. Thus communication requests no longer go directly to the devices but to the user itself, that is, to his representative, the so-called personal proxy. This proxy decides how to proceed with the incoming request, routes it to the appropriate device, and manages the communication. This so-called personal level routing is the main achievement of the MPA, but it brings one bottle-neck namely the personal proxy. All the communication has to go through the proxy although there is alternative, faster way. This problem is solved in the iceberg architecture [11]. Even though the system is based on the MPA , it has one major difference: it no longer has decentralized proxies for every user but concentrates many proxies in centralized units called iceberg points of presence (IPoPs). These IPoPs have interfaces to many access networks (e.g., telephony, cellular, and Internet) and are interconnected by fast network connections. This ensures that all communication can be routed in a fast and direct way. There is also a billing unit in the system to charge the users for certain services. The iceberg architecture is therefore a highly developed system that enables device independent communication but it does not support the user beyond this functionality and is limited to communication services.

Later on the integrated personal mobility architecture (IPMoA) [12] developed in the university of New south Wales, Sydney, Australia and department of Electrical and Computer Engineering, INSA, Toulouse, France (2003) takes a slightly different approach to ensure the device independency and the mobility of the user. It does not primarily focus on the reachability but on the mobility of the

users. The users are able to access all their data and applications from every remote location and with every available device. By including communication applications, a device independent communication is possible. The whole system is based on agents that commute between the home and the foreign network and exchange the data between those networks. A high level of personalization can thereby be reached, but since nearly all data must be fetched from the home network, there may be problems for instance with time-sensitive and synchronous applications.

All systems presented above have different kinds of information about their users, for example, reach-ability, different addresses and so forth. This information is used to provide the functionalities of the communication systems. There are other newer communication systems available as well. For instance localization by determining the positions of all kind of mobile communication devices (e.g., GSM phones or WIFI devices) has been proposed by Youssef, Agrawala and Shankar (2003) [13] and Zimmermann (2001) [14]. Later from University of Plymouth, UK Wang and Abu-Rgheff (2002) [15]proposed an integrated IP and SIP (Session Initiation Protocol) approach for advanced location management. Another research direction is to use proprietary short range radio techniques based on Bluetooth or infrared to locate the users of the system as proposed by Gonzalez-Castano and Garcia-Reinoso (2002) [16]. In this regard two most discussed communication standards should also be mentioned namely Zigbee [17] and Near Field Communication [18].

2.2   Research on Content Independent Communication

So far none of those systems above considered about the varieties of data that could be and should be sent after establishing a successful communication system. In this regard the very first attempt was taken by Offenburg University of Applied Sciences, Germany. In 2010 they developed ICAT (Identification, Classification, Adaptation and Tagged XML) as the first architecture which theoretically supports device and content independent communication [19]. The bottle-neck of the system is, it depends solely upon UAProf (User Agent Profile). UAProf standard produced by the Open Mobile Alliance defines a CC/PP(Composite Capabilities/Preference Profiles) vocabulary for mobile devices. A UAProf device profile has five components 1) hardware platform, 2) software platform, 3) browser user agent, 4) network characteristic, and 5) WAP characteristics – each with a predefined collection of possible attributes. UAProf also describes how profiles cross the bridge between WAP networks, used for mobile devices, and HTTP networks used by the Internet. One of the key considerations for mobile devices was the limited bandwidth available on the network, which is blurring day by day. As a solution of the limited bandwidth problem the UAProf and CC/PP standards provide a mechanism to send profiles as a URI link to a default profile, and optionally a

collection of "profile diffs" that override the defaults. The profiles themselves reside in a profile repository provided by the manufacturer of the device. In addition, CC/PP profiles can be built incrementally extended by the elements of the network, or extended dynamically by the server application, using information from a user database. In UAProf a small additional vocabulary is used to control conflict resolution. The net result is a composite profile with priority given to the user agent´s preferences. According to Glover and Davies (2005) [20] even though CC/PP and UAProf standards defined a framework for capability and preference information, in practice there are a number of difficulties:

- Manufacturers do not always supply accurate profile information in a consistant way.
- New capabilities require new vocabularies outside that UAProf standard.
- URIs of profile repositories are not always maintained.
- Resource Description Framework (RDF) uses URLs as its named resources, and these re-names are not always updated consistently.
- Accessing a URI to obtain a profile introduces a delay in the response,
- Local repositories avoid the delay but rapidly go out of date.

So ICAT is not able to provide reliable device and content format independent application. Furthermore it also did not focus content format independency that much. For example ICAT is not able to deal with 3D data.

2.3   Research on 3D Enabled Projects

To name some of the example of such projects we have to look back on 1968 when Ivan Sutherland described a head-mounted display with half-silvered mirrors that let the wearer see a virtual world superimposed on reality [21] [22]. His work, as well as subsequent work by others, [23] entailed a serious limitation: Because the wearer was tethered to a workstation, generally powered from an ac outlet, the apparatus was confined to a lab or some other fixed location. In 1997 Steve Mann came up with a solution of this problem [24]. His experiments in attaching a computer, radio equipment, and other devices to his own body culminated in a tetherless system that allows him roam about the city. Simultaneously he can receive e-mail and enjoy various other capabilities exceeding those available on a desktop multimedia computer.

Figure 2.1 Steve Mann wearing his tetherless computer

This new approach to computing made him interested in the visual arts—particularly still-life and landscape imaging in which multiple exposures of a static scene could be combined and illuminated by a variety of light sources. This was the starting of Augmented Reality (AR) and Virtual Reality (VR) in mobile devices.

A short list is provided below to state some of the known virtual reality projects for mobile devices

Name of the Project	THE INVISIBLE TRAIN
Category	Augmented reality game
Description of the Project	First real multi-user Augmented Reality application for handheld devices (PDAs) because like other projects it does not use wearable devices as thin clients only, the software itself runs independently on off-the-shelf PDAs. It is a mobile, collaborative multi-user AR game, where players control virtual trains on a real wooden miniature railroad track. These virtual trains are only visible to players through their PDA's video see-through display as they don't exist in the physical world.
URL	http://studierstube.icg.tugraz.at/invisible_train/
Information retrieved on	23rd January 2014

Name of the Project	AUGMENTED MAPS
Category	Navigation
Description of the Project	MapLens is an Augmented Reality application for Nokia camera phones (Symbian OS S60) with GPS. The phone camera and display are used as a viewer in combination with a paper map, which is augmented with location-based data. When a paper

	map is viewed through the camera the system analyses the video frame and determines the GPS coordinates of the visible portion of the map. MapLens is then able toaccurately overlay location-based media on the paper map. The paper maps used with MapLens are unmodified images from Google Maps.
URL	http://studierstube.icg.tugraz.at/handheld_ar/navigation.php
Information retrieved on	23rd January 2014

Name of the Project	AUGMENTED REALITY FOR WATER BASED ENTERTAINMENT, EDUCATION AND FUN (AREEF)
Category	Edutainment
Description of the Project	AREEF is an international joint R&D project that is targeted towards the development of an affordable and advanced underwater AR system using mass-market mobile devices such as smart phones and tablets. In summary, the main objectives of the proposed project are: • The development of underwater AR-applications for entertainment and game based learning. • The development of an integrated underwater AR system based on mass-market mobile devices. • The development of realistic 3D contents dedicated for interactive underwater applications.
URL	http://www.fit.fraunhofer.de/en/fb/cscw/projects/areef.html http://www.areef.info
Information retrieved on	23rd January 2014

Name of the Project	SCIENCE CENTER TO GO A MIXED REALITY LEARNING ENVIRONMENT OF MINIATURE EXHIBITS (SCeTGo)
Category	Education
Description of the Project	The SCeTGo project aims at the integration of the AR technology in science teaching both in formal & informal educational environments in order to facilitate lifelong learning by offering to

	learners the opportunity to gain exposure to everyday science in a way that is appropriate to their individual level of understanding.
URL	http://www.fit.fraunhofer.de/content/dam/fit/de/documents/Science%20Center%20To%20Go%20HSCI2010.pdf

http://www.sctg.eu/about.asp |
| Information retrieved on | 23rd January 2014 |

Name of the Project	INTEGRATED PROJECT ON INTERACTION AND PRESENCE IN URBAN ENVIRONMENTS (IPCity)
Category	Navigation, Maintenance, etc. (multiple purposes)
Description of the Project	IPCity makes extensive use of Mixed Reality, a technology that adds (computergenerated) virtual elements – textual information, images, video, sound – to what a user sees or hears in her (physical) real environment. In final event (March 2010) Mixed Reality prototypes they use devices like mobile phones or PDAs, lightweight head-mounted displays or large-scale projections. To make Mixed Reality a truly satisfying experience, the IPCity project works on integrating advanced devices, extending IT frameworks for interactive prototyping of novel environments and for creating and managing virtual content.

Another major task for IPCity is to capture the position and direction of a user or a device in real time and with high precision – an absolute prerequisite for producing enjoyable advanced Mixed Reality experiences. |
| URL | http://www.fit.fraunhofer.de/content/dam/fit/de/documents/Final-Prototype-of-TimeWarp-application.pdf

http://www.fit.fraunhofer.de/en/fb/cscw/projects/ipcity.html

http://ipcity.fit.fraunhofer.de/?page_id=272 |
| Information retrieved on | 24th January 2014 |

Name of the Project	MOBILE AUGMENTED REALITY SYSTEMS (MARS)
Category	Navigation, Education along with learning

| Description of the Project | Birth of MARS project dates back to 1996. The aim of the project is to explore the synergy of two promising fields of user interface research: Augmented reality (AR), in which 3D displays are used to overlay a synthesized world on top of the real world and mobile computing. The increasingly small and inexpensive computing devices, which can be linked by wireless networks, allow the users to use computing facilities while they are roaming the real world.

In exploring user interfaces, systems software, and application scenarios for MARS, main focus of the project is on the following lines of research:

- Identifying generic tasks a mobile user would want to carry out using a context-aware computing system

- Defining a comprehensive set of reusable user interface components for mobile augmented reality applications.

- Making combined use of different display technologies, ranging from head-worn, to hand-held, to palm-top to best support mobile users. |
|---|---|
| URL | http://monet.cs.columbia.edu/projects/mars/mars.html

http://www.sciencedirect.com/science/article/pii/S009784939900103X |
| Information retrieved on | 24th January 2014 |

Name of the Project	CONNECT
Category	Education along with entertainment
Description of the Project	The European CONNECT project develops and uses leading edge information and communication technology to create an advanced learning environment, the virtual science thematic park. It connects 'informal' learning strategies and formal curricular activities in science education, involving students and teachers in extended episodes of playful learning.
URL	http://www.fit.fraunhofer.de/en/fb/cscw/projects/connect.html

http://www.ea.gr/ep/connect/main.asp?Cat_ID=426 |
| Information retrieved on | 24th January 2014 |

Name of the Project	VIRTUOSO
Category	Education with Entertainment
Description of the Project	Virtuoso is a collaborative educational game designed to show-case the possibilities of the hand-held Augmented Reality (AR) framework. The game's objective is to sort a collection of art-works according to their date of creation along a timeline drawn on a wall-mounted billboard. Every mark (fiducial) on the time-line carries one of the artworks, which are only visible through the player's AR PDA. The player can pick up any artwork with his PDA, by clicking on the artwork on the display and drop it on a free position by clicking on the free fiducial on the display. Since all positions are initially occupied, the game requires two or more players to cooperate in swapping or rearranging the se-quence of artworks.
URL	http://studierstube.icg.tugraz.at/handheld_ar/virtuoso.php
Information retrieved on	28th January 2014

Most of the above mentioned projects are or at least initially were focused on PDA's or even more powerful devices which can establish WLAN connections. This network connection is faster than a general GSM or UMTS connection and in most of the cases free or at least cheaper in cost.

In 2006 a doctoral thesis was conducted in University of Strasbourg having title "Study and imple-mentation of a protocol for mobile augmented reality system [Etude de protocole et réalisation d'un système de réalité augmentée mobile]" (Doctoral thesis by Feisst 2006) [25] . The aim of this project was to add support for 3D Virtual Reality data on mobile devices and additionally to pre-sent this information in monoscopic as well as in stereoscopic view. Due to device and telecom-munication network limitations such as- less processing power, limited bandwidth, bearer tech-nology (GSM, GPRS, EDGE, UMTS etc), transmission methodology (Circuit switch or packet switch) etc the project had to depend upon an intermediate step named VRML proxy server. Moreover the project is not cross platform. It is dedicated to Java enabled Symbian mobile devices and it was tested with Sony Ericsson k700i and Nokia 6680 models.

Today due to technological advancement in regard to software, hardware and network, plenty of 3D projects are available to serve different purposes. The latest and most amazing project in this

area is smart glasses. According to Mike Elgan smart glasses will be available in the market for public in 2014 [26]

A short list of wearable computing smart glasses currently in development is provided below in table 2.1

	Manufacturer/Model	Price	Articles with source
	EmoPulse nanoGlass-4	$25	EmoPulse nanoGlass-4 On Indiegogo http://www.smartglassesnews.org/emopulse-nanoglass-on-indiegogo/
	GlassUp AR Smart Glasses	$349	GlassUp AR Eyeglasses For Android And iOS http://www.smartglassesnews.org/glassup-ar-eyeglasses-for-android-and-ios/
	Google Glass	$1,500	Google Glass: Borgifying™ The Masses http://www.smartglassesnews.org/review-google-glass/
	ION Smart Glasses	$99	ION Glasses. First optical and sun smartglasses that interact with your devices http://www.indiegogo.com/projects/ion-glasses-first-optical-and-sun-smartglasses-that-interact-with-your-devices
	Meta AR Smart Glasses	$650	Meta Augmented Reality Smart Glasses http://www.smartglassesnews.org/meta-augmented-reality-smart-glasses/
	Pivothead Smart Glasses	$629	Pivothead Wearable Imaging Introduces SMART Eyewear at CES 2014 http://pivothead.com/news-archive/article/-pivothead-wearable-imaging-introduces-smart-eyewear-at-ces-2014

	Manufacturer/Model	Price	Articles with source
	Samsung Smart Glasses	unknown	Samsung Smart Glasses? Korean Patent Application Indicates Plans For Google Glass Competitor http://www.ibtimes.com/samsung-smart-glasses-korean-patent-application-indicates-plans-google-glass-competitor-1440172
	Vuzix M100	$500	Vuzix Smart Glasses M100 vs Google Glass Vuzix M100 Smart Glasses http://www.smartglassesnews.org/vuzix-m100-smart-glasses-vs-google-glass/

Table 2.1 Smart glases expected to be available in 2014

Nowadays it is common practice to have 3D view of any application such as Angry Bird go, Google Earth etc. Most of the time developers are focusing on operating system or platform while developing an application. Different tools are available in the market to develop so called cross platform application but they are not able to support any device with any operating system. A list is provided in following subchapter as example of tools to develop cross-platform application mainly for hand-held devices

2.3.1 Helping tool for developing cross-platform projects

There are plenty of tools available in the market as freeware, open source or commercial system. Below stated tools were chosen for their varied levels of complexity, price, features and documentation

I. SENCHA TOUCH 2

Knowledge required: HTML, CSS, JavaScript, general web development

Platform support: iOS, Android, Blackberry

Cost: Free under commercial and open source licenses (paid OEM license available)

Documentation: have examples, screencasts, and tutorials

URL: http://www.sencha.com/products/touch/

II. JQUERY MOBILE

Knowledge required: HTML, CSS, jQuery

Platform support: Most mobile browsers

Cost: Free (license: MIT and GPL)

Documentation: There are many demos on the site and books on jQuery Mobile

URL: http://jquerymobile.com/

III. TIGGZI

Knowledge required: HTML, CSS, JavaScript, XML

Platform support: Android, iOS, Mobile Web

Cost: $15-$50/month (there's a free plan); see Pricing page

Documentation: have lot of good tutorials, videos and other resources on their site

URL: http://appery.io/

IV. APPMAKR

Knowledge required: HTML and CSS would be helpful

Platform support: iOS, Android, Windows Phone

Cost: $79/month (free if you allow them to place ads in your mobile app)

Documentation: have good video tutorials including a walkthrough of the app store distribution process

URL: http://appmakr.com/

V. IBUILDAPP

Knowledge required: HTML and CSS would be helpful

Platform support: Options available for iOS and Android export

Cost: $9.99/month (free if you allow them to place ads in your mobile app)

Documentation: have good tutorials on how to use their service

URL: http://ibuildapp.com/

VI. WIDGETBOX

Knowledge required: HTML and CSS would be helpful

Platform support: Compatible with most modern web browsers

Cost: $25-100/month

Documentation: have support, FAQ, and a knowledgebase

URL: http://www.widgetbox.com/mobile/

VII. FONEFRAME

Knowledge required: HTML5, CSS3, JavaScript

Platform support: Compatible with most modern browsers and platforms

Cost: Free (license: Creative Commons Attribution 3.0 Unported License)

Documentation: The template has inline documentation

URL: http://www.qrdvark.com/foneFrame/

VIII.   <u>PHONEGAP</u>

Knowledge required: Web technologies such as HTML,CSS and JavaScript

Platform support: includes but not limited to iOS, Android, Blackberry, Windows, WebOS and Symbian

Cost: Free and open source

Documentation: have Blogs, Books, FAQ, docs and wiki

URL: http://phonegap.com/

IX.   <u>PHONEGAP BUILD</u>

Knowledge required: Web technologies such as HTML,CSS and JavaScript

Platform support: Currently supports iOS, Android™, Windows® Phone, Blackberry® 5/6/7 and webOS

Cost: Free and open source

Documentation: have Support, Blog, FAQ, and Docs

URL: https://build.phonegap.com/

X.   <u>APPCELERATOR</u>

Knowledge required: Web technologies such as HTML,CSS and JavaScript

Platform support: iOS, Android ,Windows and Blackberry OS

Cost: Free but proprietary software

Documentation: have resource center and training services

URL: http://software.intel.com/en-us/html5/tools

XI.   <u>THE INTEL® XDK</u>

Knowledge required: HTML5, CSS3 and JavaScript

Platform support: iOS, Android and Windows

Cost: Free

Documentation: They have Training, Blogs, and Documentation

URL: http://software.intel.com/en-us/html5/tools

Even though there are so many tools available to make the process of developing cross-platform application easier still there is no tool or project available, which literally supports application those are not dependent upon

- device in terms of type, capability and platform
- content format including 3D data and
- purpose in terms of business, private, education or a mixture of several purposes

2.4   Limitation of existing solutions

Subchapter 2.1 provides explanation of several projects where target was to establish an underlying media independent communication. Most of them were designed to serve a pre-decided commercial purpose for pre-decided types of devices. In adition, none of those projects except ICAT took independency of delivered data or content under consideration. As a major limitation of ICAT, lack of 3D content support has to be mentioned. Subchapter 2.3 provides explanation of several projects where 3D content were enabled either as virtual reality or as augmented reality. Most of those projects were designed for high-end devices but some of those considered mobile devices as well. There are supporting tools available as well to design theoretically a crossplatform application but practically all of those tools are dependent upon operating system and supported functionalities of the target device.

There is no project found that considered not only independency of the device but also independency of the content regardless of the purpose of the communication. Since support for as many different devices and data formats as possible is desired, committing to one of the above mentioned technique or direction is not an option. The approach of this thesis work is to take all the data gathered by the different communication systems, add some additional data, for example, certain capabilities of used device and generate information that supports the user far more than what was possible with previous systems. A platform for device and content independent communication and its personalization is thus provided which combines the advantages of all the approaches described above along with added features (see chapter 3).

Chapter 3   Proposed Solution

3.1   Purpose and Concept of the Thesis Along with a Proposed Solution

The goal of this research work is to find a way to solve the above stated problems by investigating system architectures to provide unconstrained, continuous and personalized access to the content and interactive applications everywhere and at anytime with different devices.

A solution has to be found to enable the automation of the content adaptation process. In order to realize such a system three major requirements have to be fulfilled:

- Identification of the connected device

- Generation, structuring and storage of generalized content

- Transformation process from general content to optimized and device dependent content

3.2   Originality of the Proposed Solution: an Architecture

As originality of this thesis can be listed as

- The proposed architecture aims to provide device and content format independent communication

- The proposed architecture does not focus any specific service or group. It could be used for academic purposes as well as business purposes, even for individual private use

- The proposed architecture aims to use tools those are either open source or at least free to use so that the prototype could be open for everybody

3.3   Scope of the Architecture

The scope of the thesis can be depicted as Figure 3.1. The creation of content and the processed content after being displayed at the end user device, how user is dealing with the content does not lie under the scope of this thesis. Besides the mobile devices lie under the lifetime of telecommunication system from 3rd Generation on, are considered under the scope of this thesis.

Even though the goal of this research work is to provide architecture of a system which is device and content independent, it should be kept in mind that there is no solution without limitation. To

have a clear understanding about the criteria of devices those will be supported and the reason behind this limitation, it is necessary to recall the life story of mobile telecommunication system.
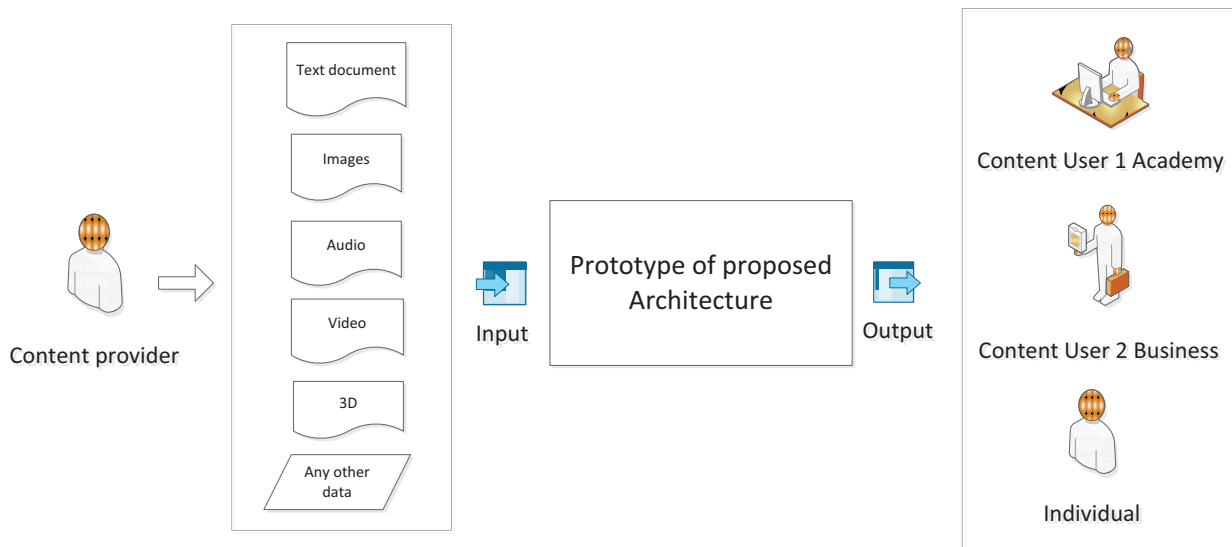
Figure 3.1 Pictorial representation of scope of the thesis

3.3.1 Life of telecommunication systems (0G to 4G)

The portrayal of a utopia of mobile phone in literature dates back to the year 1931. It is found in Erich Kästner's children's book Der 35. Mai oder Konrad reitet in die Südsee:

" A gentleman, who rode along the sidewalk in front of them, suddenly stepped off the conveyor belt, pulled a phone from his coat pocket, spoke a number into it and shouted: "Gertrude, listen, I'll be an hour late for lunch because I want to go to the laboratory. Goodbye, sweetheart!" Then he put his pocket phone away again, stepped back on the conveyor belt, started reading a book... "

—Erich Kästner

From the very beginning to till today life of mobile telecommunication system could be divided in eight different generations.

0G

0G (Zero Generation) is also known as Mobile Radio Telephone system. As this generation was invented prior to cellular system, it was mentioned as pre cellular system. In this system analog signals were used as carriers. Generally Mobile Radio Telephone system provides half duplex communication (voice signals can either be transmitted or received at a time). It consists of various technologies such as Advanced Mobile Telephone System (AMTS), Mobile Telephone Sys-

tem (MTS), and Mobile telephony system D (MTD), Offentlig Landmobile Telefoni or Public Land Mobile Telephony (OLT), Push to Talk (PTT) and Improved Mobile Telephone Service (IMTS). These mobile telephones were placed in vehicles (truck, cars etc). The mobile telephone instrument had two main parts those were transceiver (transmitter – receiver) and head containing display and dial keys. Transceiver was fixed in the vehicle trunk; head was fixed near the driver seat and both head and transceiver were connected to each other with wire. The device (telephone) would connect to local telephone network only if it is in the range of 20 Kms. Each city had a central antenna tower with 25 channels. It means that mobile transceiver should have a powerful transmitter with a transmitting range of 50-70 Kms. Only few people were able to use this device as only 25 channels were available. Roaming facility was not supported in this generation of analog cellular Mobile Radio telephone system. Mobile Radio telephone system was a commercial service under public switched telephone network with unique telephone numbers. Zero generation had seen different variants of two way radio telephones. Large number of limitations in this generation led to the advent of new generation.

0.5G

0.5G was the advance version of 0G (Zero Generation or Mobile Radio Telephone system). This 0.5G technology had introduced ARP (Autoradiopuhelin) as the first commercial public mobile phone network. This ARP network was launched in 1971 at Finland. ARP was operated on 8 Channels with a frequency of 150 MHz (147.9 – 154.875 MHz band) and its transmission power was in a range of 1 to 5 watts. ARP used half duplex system for transmission with manual switched system. This Network contains cells which are actually land area divided into small sectors and each of those sectors is known as cell. A cell is covered by a radio network with one transceiver. The size of the cell was 30 km. ARP did not support handover; as a result calls would get disconnected while moving from one cell to another. ARP provided 100% coverage which attracted many users towards it. It was successful and became very popular until the network became congested. The ARP mobile terminals were too large to be fixed in cars and were expensive too. These limitations led to the invention of Autotel. Autotel are also known as PALM (Public Automated Land Mobile). It is a radio telephone service which in terms of technology lies between MTS and IMTS. It used digital signals for messages like call step up, channel assignment, ringing, etc only voice channel was analog. This system used existent high-power VHF channels instead of cellular system. It was developed in Canada and Columbia.

The table 3.1 shows the most used technology of the above mentioned generations in different parts of the world [27] [28].

SYSTEM	USED IN
MTS	IN 1946 (MOTOROLA AND BELL SYSTEM) UNITED STATES
A-NETZ	IN 1952 WEST GERMANY
IMST	IN 1962 (BELL SYSTEM) UNITED STATES
ARP	IN 1971 FINLAND
B-NETZ	IN 1972 WEST GERMANY

Table 3.1 Use of mobile phone systems prior to 1G

1G

1G is also known as the First generation of wireless telecommunication technology, was introduced in 1980 but all developed systems were proprietary. The main difference between back then existing systems and 1G was invention of cellular technology and hence it is also known as First generation of analog cellular telephone. In 1G or First generation of wireless telecommunication technology the network contains many cells, so same frequency can be reused many times which results in great spectrum usage and thus increased the system capacity. Which means large number of users could be accommodated easily and simultaneously.

All of the standards in 1G used frequency modulation techniques for voice signals and all the handover decisions were taken at the Base Stations (BS). The spectrum within cell was divided into number of channels and every call is allotted a dedicated pair of channels. Data transmission between the wire part of connection and Packet Switched Telephone Network (PSTN) was done by using packet-switched network. With this development, it was possible to locate the mobile phone and handle incoming and outgoing calls as well as hand over calls between different cells [29]. The most used "First Generation" (1G) systems are shown in Table 3.2

System	Used in
Advanced Mobile Phone System (AMPS)	United States, Canada, Mexico, Australia, New Zealand, Taiwan, South Korea, Singapore, Hong Kong, Thailand, Brazil, Argentina, Bangladesh, China, Sri-lanka, Tajikistan, Vietnam,

	Pakistan, Brunei, Cambodia, Georgia, Nauru, etc
Total Access Communication System (TACS)	United Kingdom, Ireland, Spain, Italy, Austria, United Arab Emirates, Bahrain, Kuwait, Malta, Singapore
Nordic Mobile Telephone (NMT-450) System	Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Hungary, Poland, Russia, Spain, Sweden, Thailand, turkey and Ukraine.
Nordic Mobile Telephone (NMT-900) System	Cyprus, Denmark, Finland, France, Greenland, Netherlands, Norway, Switzerland and Thailand.
C-Net	Germany, Portugal and south Africa
Nippon Telephone and Telegraph (NTT)	Japan

Table 3.2 Use of mobile phone systems in 1G [30]

Even though use of cellular system in 1G resulted in great spectrum usage but it used analog transmission techniques which were basically transmitting voice signals only. Hence those devices are not supported by the proposed device and content format independent architecture.

2G

2G is also known as second generation wireless telephone technology. Three primary benefits of 2G networks over their predecessors were-

- phone conversations were digitally encrypted
- 2G systems were significantly more efficient on the spectrum allowing for far greater mobile phone penetration levels and
- 2G introduced data services for mobile, starting with SMS text messages

2G networks enabled the various mobile phone networks to provide the services such as text messages, picture messages and multimedia messages (MMS). All text messages sent over 2G are digitally encrypted, allowing for the transfer of data in such a way that only the intended receiver can receive and read it. After 2G was launched, the previous mobile telephone systems were retrospectively replaced.

In Europe, analogue cellular telephone systems were growing rapidly, particularly in Scandinavia and the United Kingdom, but also in France and Germany. Each country developed its own system, which was incompatible with the equipment and operation of any other system. This was an unsatisfactory situation because not only was the mobile equipment limited to operation within national boundaries which, in a unified Europe was becoming increasingly impractical, but also there was a very limited market for each type of equipment. Thus, economies of scale and the subsequent savings could not be realized in this situation.

Another driving influence to the development of the second generation systems (2G) was the development of new technologies in the semiconductor and microprocessor field. With these new technologies it was possible to realize digital systems. The USA enhanced the AMPS into D-AMPS. With the prospective establishment of a European Union, the European countries decided to develop a common mobile communication standard. In 1982, the Conference of European Posts and Telegraphs (CEPT) formed a study group called the "Groupe Special Mobile" (GSM) to study and develop a pan-european public land mobile system.

The proposed system had to meet certain criteria:

- Subjective good speech quality
- Low terminal and service costs
- Support for international roaming
- Ability to support hand-held terminals
- Support for range of new services and facilities
- Spectral efficiency
- ISDN compatibility

The developers of GSM chose a so far unproven digital system in contrast to the standard analogue cellular systems such as AMPS in the United States and TACS in the United Kingdom. They believed that advancements in compression algorithms and digital signal processors would allow the fulfillment of the original criteria and the continual improvement of the system in terms of quality and cost. With over 8000 pages of GSM recommendations, they tried to allow flexibility and competitive innovation among suppliers and also provide enough standardization to guarantee proper networking between the components of the system. This is realized by providing functional and interface descriptions for each of the functional entities defined in the system.

In 1989, the GSM responsibility was transferred to the European Telecommunication Standards

Institute (ETSI) and, in 1990, the first GSM specification was published. Commercial service was started in the middle of 1991, and by 1993 there were 36 GSM networks in 22 countries. Although standardized in Europe, GSM is not only a European standard. Over 200 GSM networks (including DCS1800 and PCS1900) are operational in 110 countries around the world. In the beginning of 1994, there were 1.3 million subscribers worldwide, which had grown to more than 55 million by October 1997. With North America making a delayed entry into the GSM field with a derivative of GSM called PCS1900, GSM systems exist on every continent, and the acronym GSM now stands for Global System for Mobile communications

2G technologies can be divided into Time Division Multiple Access (TDMA)-based and Code Division Multiple Access (CDMA)-based standards depending on the type of multiplexing used [29]. The main 2G technologies are shown in Table 3.3

System	Used in
GSM (TDMA based)	Europe and United states , later whole world
Interim Standard 95 (CDMA based)	North America
PDC (TDMA based)	Japan
iDen (TDMA based)	United states and Canada

Table 3.3 Use of mobile phone technologies in 2G [31]

2.5G

2.5G is also known as second and a half generation of mobile telecommunication technology. It is used to describe 2G-systems that have implemented a packet-switched domain in addition to the circuit-switched domain. In modern mobile communication systems, data services became more and more important. To meet these new requirements, compression/ de-compression algorithms were continuously improved. Since these improvements mainly consisted in new software algorithms or small hardware changes at the provider side, they were called the 2.5G.

From 2G to 3G is actually considered as evolution of GSM. It is because in the evolution towards the third generation systems, the GSM systems were enhanced. Those enhancements are:

- Intelligent network (IN) services
- Enhanced speech compression/decompression
- New transmission principles:

- High speed circuit switched data (HSCSD)

- General Packed Radio Service (GPRS)

- Enhanced Data rates for GSM Evolution (EDGE)

While the HSCSD improvement of the GSM system needs mainly a software update, the GPRS improvement needs an additional small hardware change on the operator side. Extending a GSM system to an EDGE system, new hardware is needed. Most European operators have upgraded to either GPRS or HSCSD or both of them in parallel. In the USA the system operators started to build up EDGE systems. Reason behind this development in the USA is that GSM systems are actually almost not in use in the United States, thus the frequency ranges are unused in contrast to frequency ranges for UMTS which are occupied by systems still in use. Thus an intermediate step towards 3G is the introduction of EDGE. In Europe, the system operators are planning EDGE systems only in rural areas where an UMTS system would be too uneconomic [32] [33].

In 2010, the GSM Association estimates that technologies defined in the GSM standard serve 80% of the global mobile market, encompassing more than 5 billion people across more than 212 countries and territories, making GSM the most ubiquitous of the many standards for cellular networks. Macau decided to fade out GSM network in July 2012 (only roaming service is kept), making it the first region to decommission GSM network [34].

SERVICES OF GSM EVOLUTION IN A NUTSHELL (2G & 2.5 G)

The GSM network supports speech, data (fax) and a Short Message Service (SMS), which allows the user to send ASCII characters to another mobile phone The network provides similar services as the ISDN network, which was a precondition for the network. The most important features of ISDN are:

- Handover, roaming (national and international)

- Identification (display coaling number, user profile, etc.)

- Knocking, halt the line, conferences

- Show the time/costs, data/time counter

- Data connections: GSM, GPRS/HSCSD, EDGE.

3G

3G is also known as third generation of wireless telecommunication technology. The necessity of this generation was driven by increased data rates and mobile access to

- Internet

- E-mail and

- Corporate networks

Birth of 3G could be depicted as figure 3.2. According to International Telecommunication Union (ITU) specification for IMT-2000, EDGE is recognized as UWC-136' and considered as first phase of 3G. It was initially developed for mobile network operators who failed to win spectrum for third generation networks, and is a cost-efficient way of migrating to full-blown 3G services. It gives incumbent GSM operators the opportunity to offer data services at speeds that are near to those available on UMTS networks. Hence in the scope of this thesis documentation EDGE is considered as 2.5 Generation technology [35].
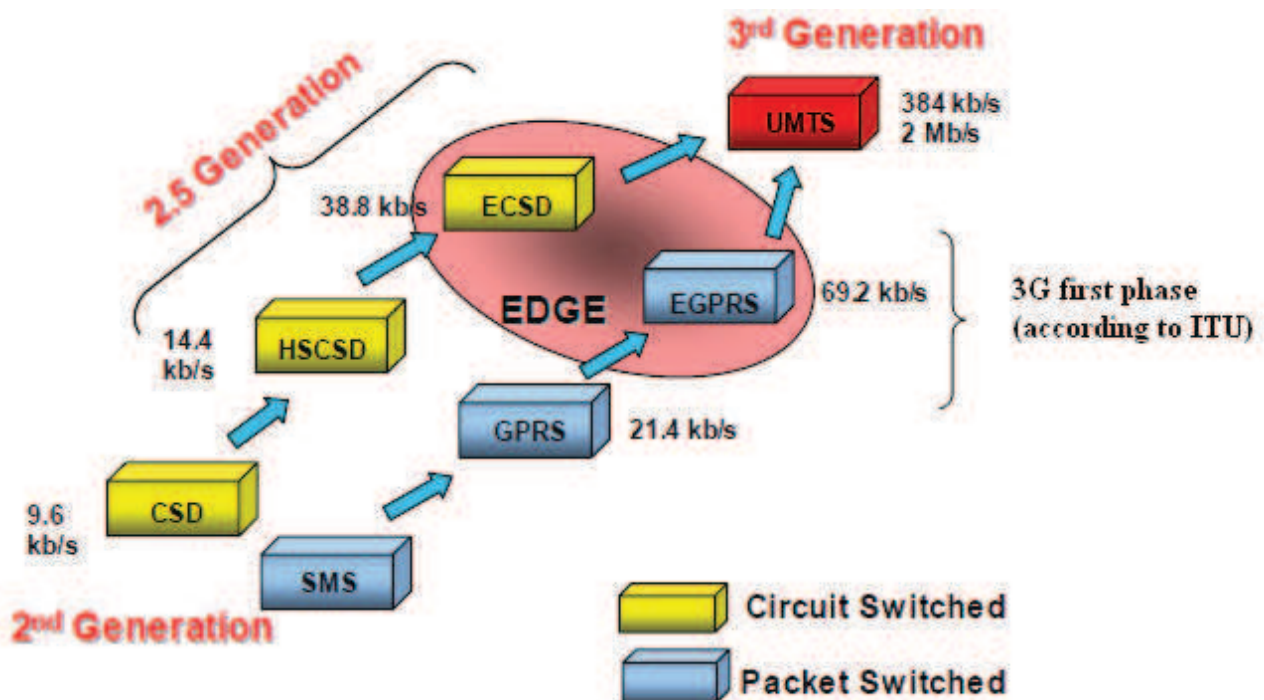
Figure 3.2 Evolution of GSM towards 3G (based on [25])

Benefits of 3G networks over their predecessors were-

- Multimedia support

- Quality of Service Implementation (QoS)

- Location based services

- Virtual home environment [36]

The bandwidth and location information available to 3G devices gives a rise to applications which were not previously available to mobile phone users. Some of the applications are:

- Mobile TV
- Video on demand
- Video Conferencing
- Telemedicine
- Location-based services
- Global Positioning System (GPS)

By using 3G, in Fixed Wireless Lans data can be transferred at the speed of 5-8 Mb per second, whereas for moving device the speed may decrease. A device, supporting 3G can work as computer, Television, Newspaper, ipod, Diary, credit card etc. 3G technologies made a mobile work just like a mini computer. The advantage of mobile over computer is – it is always switched on, nearby or at hand of the user and ready to be used anytime, anywhere.

The main 3G technologies are shown in Table 3.4 [37]

System	*Used in*
UMTS, W-CDMA	Europe, later whole world
CDMA 2000	North America
TD-SCDMA	China
DECT	Europe, US, Canada

Table 3.4 Use of mobile phone technologies in 3G

3.5G

3.5G is also known as enhanced 3G. It is basically an enhancement of UMTS in two steps to increase Downlink and Uplink speed of data.

High-Speed Downlink Packet Access (HSDPA) which is introduced in UMTS Release 5 of the 3GPP specifications; the UMTS standard was enhanced to be able to deliver much higher data rates per cell and per user than before. With data rates between 500 kbit/s and 3.6 Mbit/s per user, HSDPA allows network operators to compete head on with other high-speed Internet access technologies such as DSL.

Even with the introduction of HSDPA the uplink speed was still limited. As a solution to satisfy the increasing demand in the uplink direction, with UMTS Release 6 high-speed uplink packet access (HSUPA) was introduced. HSUPA increases uplink user data rates up to theoretical 5.8 Mbit/s. When taking realistic radio conditions into account, the number of simultaneous users, terminal capabilities, etc., user speeds 800 kbit/s and beyond can still be reached.

The advantages that 3.5G brought over 3G could be listed as [38]

- 3.5G is an improvement of 3G in terms of Speed and Quality of Transmission. It does not provide any other new feature to 3G.

- 3.5G has less latency compared to 3G.

- 3.5G is cheap because it only has to be upgraded from 3G without requiring any huge change in existing 3G infrastructures.

28 August 2009, 250 HSDPA networks have commercially launched mobile broadband services in 109 countries in all over the world [39]. As we know from the history, human need knows no bound. Some of the world population have already enjoyed the next generation of the mobile telecommunication technology after 3.5G.

4G

4G is also known as the fourth generation of wireless telecommunication technology. In March 2008, the International Telecommunications Union-Radio communications sector (ITU-R) specified a set of requirements for 4G standards, named the International Mobile Telecommunications Advanced (IMT-Advanced) specification, setting peak speed requirements for 4G service at 100 megabits per second (Mbit/s) for high mobility communication, such as from trains and cars and 1 gigabit per second (Gbit/s) for low mobility communication such as pedestrians and stationary users [40]. So An IMT-Advanced cellular system must fulfill the following requirements:

- be based on an all-IP packet switched network.

- have peak data rates of up to approximately 100 Mbit/s for high mobility such as mobile access and up to approximately 1 Gbit/s for low mobility such as nomadic/local wireless access.

- be able to dynamically share and use the network resources to support more simultaneous users per cell.

- using scalable channel bandwidths of 5–20 MHz, optionally up to 40 MHz. [32] [33]

- have peak link spectral efficiency of 15 bit/s/Hz in the downlink, and 6.75 bit/s/Hz in the uplink (meaning that 1 Gbit/s in the downlink should be possible over less than 67 MHz bandwidth).

- system spectral efficiency of up to 3 bit/s/Hz/cell in the downlink and 2.25 bit/s/Hz/cell for indoor usage. [32]

- smooth handovers across heterogeneous networks.

- the ability to offer high quality of service for next generation multimedia support.

Till end of 2012, release of 4G technologies did not fully comply with the IMT-Advanced requirements, but it was expected to be done within end of 2013. The main 4G technologies are shown in Table 3.5 [41]

System	Used in
LTE standardized by 3GPP	Europe, later whole world
WiMAX standardized by IEEE802.16	US, South Korea, Russia

Table 3.5 Use of mobile phone technologies in 4G

3.3.2   Supported generation by the proposed architecture

Among eight above mentioned generations five of them are currently in use. Table 3.6 shows a summary of the information about existing generations. Since the concept of this thesis work is mainly concentrating upon exchange of data with various formats, for business, academic or private purposes; speed of data transfer is one of the major parameter. Besides according to ITU (2011) 45 percent of the world´s population is now covered by a 3G mobile network. 3G network now has been launched in 159 countries. Some countries such as Sweden, Norway, Ukraine and United States are already moving to 4G [42]. Ericsson in June 2012 forecasts that, by 2017, 85 percent of the world's population will be covered by high-speed mobile networks. These networks will mostly be based on WCDMA/HSPA. LTE will be available at the high-end (led by deployments in the US, Korea, Japan and Western Europe) [43]. Almost 60 percent of operators have started to launch LTE-based services in 2013 (24.9 percent), according to a survey of mobile operators by Informa (May 2012) [44]. LTE, sometimes called 4G (forth-generation), networks allow increased download and upload speeds and faster response times, in particular for data-intensive services such as video. Strategy Analytics predicts (May 2012) LTE will rise sharply to one billion connections by 2017, when it will comprise 15 percent of all mobile connections [45].

Hence starting from 3G devices lie in the scope of this thesis work.

		Real World (avg)		Theoretical (max)		Availability
		Download	Upload	Download	Upload	
2.5G	GPRS	32-48Kbps	15Kbps	114Kbps	20Kbps	Today
2.75G	EDGE	175Kbps	30Kbps	384Kbps	60Kbps	Today
3G	UMTS	226Kbps	30Kbps	384Kbps	64Kbps	Today
	W-CDMA	800Kbps	60Kbps	2Mbps	153Kbps	Today
	EV-DO Rev. A	1Mbps	500Kbps	3.1Mbps	1.8Mbps	Today
	HSPA 3.6	650Kbps	260Kbps	3.6Mbps	348Kbps	Today
	HSPA 7.2	1.4Mbps	700Kbps	7.2Mbps	2Mbps	Today
Pre-4G	WiMAX	3-6Mbps	1Mbps	100Mbps+	56Mbps	Today
	LTE	5-12Mbps	2-5Mbps	100Mbps+	50Mbps	End 2010
	HSPA+	-	-	56Mbps	22Mbps	2011
	HSPA 14	2Mbps	700Kbps	14Mbps	5.7Mbps	Today*
4G	WiMAX 2 (802.16m)	-	-	100Mbps mobile / 1Gbps fixed	60Mbps	2012
	LTE Advanced	-	-	100Mbps mobile / 1Gbps fixed	-	2012+

Table 3.6 Use of mobile phone technologies in current time [41]

3.4 Presentation of the Proposed Solution: A New Architecture Named "Smoothie"

The word Smoothie is a noun, dedicated for a beverage. A smoothie is a blended emulsion of different products for example fruits; crushed ice, honey, syrup, yogurt, ice cream ingredients, soy milk, green tea, herbal supplements etc. and special cases can contain chocolate or peanut butter. So we can derive that a smoothie can contain any ingredient which is neither limited nor predictable, depends upon the recipe of an individual or a group such as board of director of a business institute who deals with food and beverages. The new architecture proposed as a solution of above mentioned problem has similar properties. It is an emulsion of different application and data formats. In terms of data formats anything could be used as an input to the proposed architecture which is neither limited nor predictable. As special case (one of the most important cases within the context of this thesis), any 3D data could also be provided as content into the architecture. Due to the similarity of properties in between the beverage smoothie and the proposed architecture, the architecture is named as "Smoothie".

Figure 3.3 shows the most simplified version of the system overview of Smoothie. First, it is detected whether the user is connecting to the system via mobile device or by desktop device by analyzing the HTTP-request header coming from the end client's device. Wireless Universal Re-

source File (WURFL) non commercial version is selected for the description of the features of mobile devices and browsers because it is an XML configuration file which contains information about capabilities and features of many mobile devices in the wireless world. Also, the repository of device in WURFL is updated every day by contributors in the world. So it is an up to date specification that brings reliability in device data manipulation. Proposed architecture works with a combination of WURFL and a local database saved in the server. Regarding content, providers do not have to care about the different mobile devices and to provide an optimized version of their file for every device. Content will be saved at first in its original format for safety reason then will be translated in a generalized format and stored in a XML database. In the prototype instead of XML database file system is used just for proof of concept. A decided generalized format is XML or XML extended format such as DAE. Based on the identified device capabilities generalized content format will be transformed into device dependent manner and displayed to the end client. So on the fly data conversion and serialization is happening in server side. For the prototype web browser is used along with the concept RWD (Responsive Web Design), as a container of the content to be presented at end client´s side. Any time it is possible to extend the serialization by implementing other available pipeline to process other output format such as PDF, WML etc. Web App is responsible for any kind of condition check and taking decision.
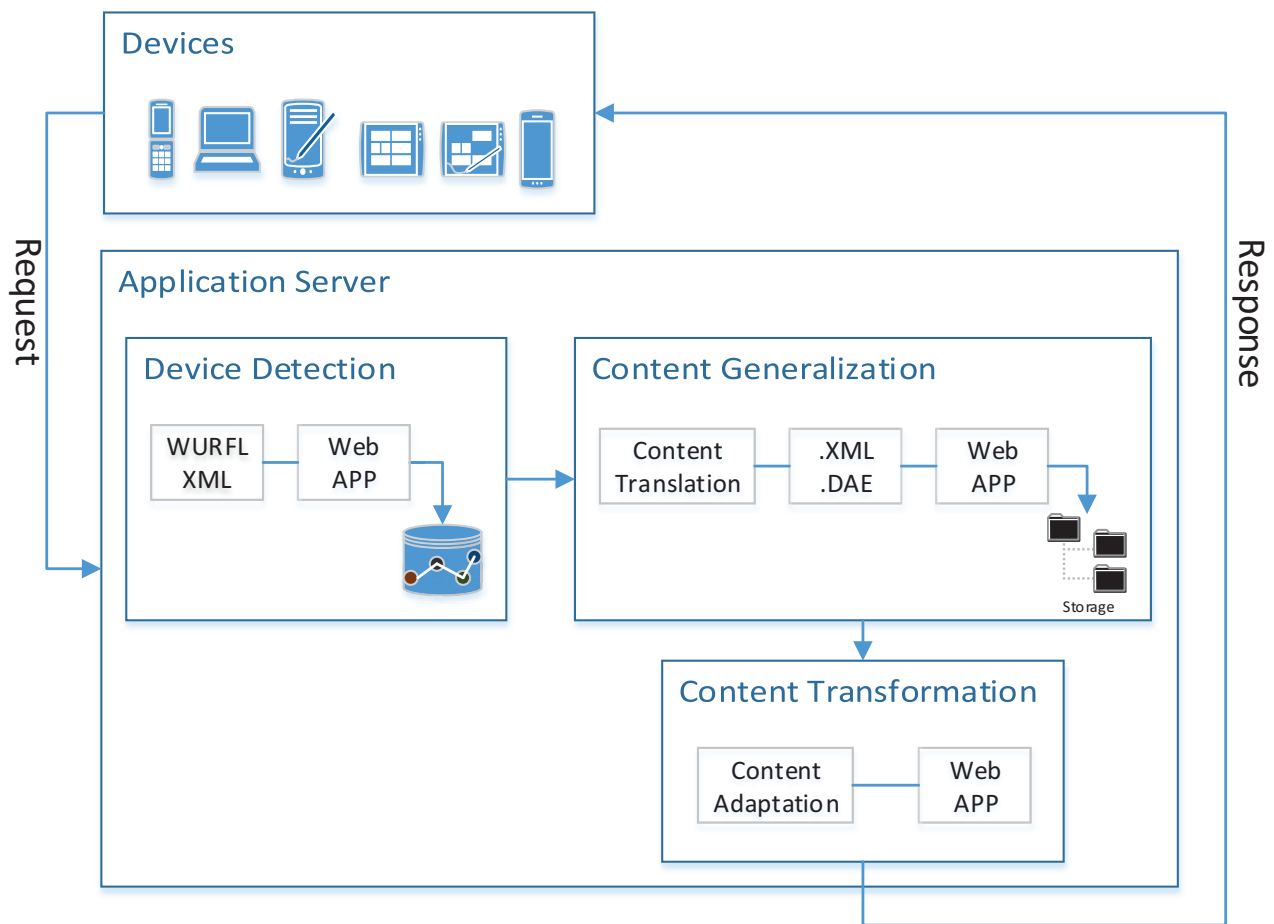
Figure 3.3 System overview of Smoothie

So, this thesis work is proposing an architecture named Smoothie where

- Browsers are chosen as a container of content at end user´s device

- the device independency is established along with content independency

- 3D and upcoming (XML compatible) data formats are included as content.

The goal of Smoothie is to establish a communication where

- Devices can access web content appropriate for their capabilities,

- Authors can create web content which is deliverable across different devices

- Content can be accessed from different kinds of device with different capabilities

Next chapter of this document will provide the theory and plan to achieve above mentioned goal (see chapter 4).

Chapter 4   Theoritical Concepts Behind Smoothie

4.1   Overview of major tasks of Smoothie

The very first responsibility of such a system is to identify the end user´s device along with device capabilities as soon as the user logs in to the system. Under the scope of this thesis it is considered as a basic right of the user to be able to see the desired data on any of their used devices available at hand. There might be a necessity of transforming the data based on the end user´s device capabilities. In this regard the responsibilities of the system or major tasks of the system could be divided into three phases consisting of seven actions

- Phase 1-Identification of the connected device along with recognition of device capabilities
- Phase 2-Collection of data/content then prepare and save generalized content in a predefined format
- Phase 3-Based on user request retrieve generalized format of content then transform and send the content in a destination device dependent manner

As it is depicted in figure 3.3, above mentioned three phases are divided in three different modules, i.e. each of them is distinct but interrelated unit from which Smoothie is built up and it´s complex activity can be analysed and enhanced. In future if any further improvement in the quality, value, or extent of existing code is necessary in any of those modules, it can be done without interrupting others. Besides, for software architectural pattern, Model View Controller (MVC) architecture is followed so that the user interface, storage and business logic are interrelated but independent. For example, any required change in user interface can be done without having any effect on storage or business logic and vice versa. The prototype of Smoothie is using Browser for data serialization at end user´s device and MVC together with JavaScript (JavaScriptMVC) extends libraries with MVC architecture. In addition JavaScriptMVC does not depend on server components i.e. it can be combined with any web service interface and server side language including Java. Hence JavaScriptMVC was selected as a design pattern for Smoothie. The combination of JavaScriptMVC and separation of modules based on major tasks of the system (three parts) make

the architecture (figure 3.3) future proof and most suitable to reach the goal of establishing a device and content independent communication.

The below stated picture (Figure 4.1) depicts an overview of the major tasks of Smoothie. The rest of the chapter will provide an elaborate discussion of above mentioned three phases as significant tasks of Smoothie.
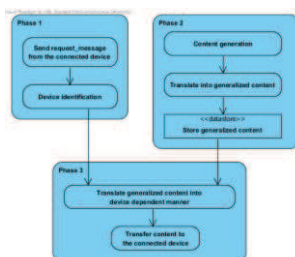
Figure 4.1 Overview of the major tasks of Smoothie

4.2   Identification of Device

The very first responsibility of the system i.e. Smoothie is to detect the connected device. Wireless Universal Resource File (WURFL) was selected for the description of the features of mobile devices and browsers, because the WURFL model is an XML configuration file that contains information about the capabilities and features of many mobile devices in the wireless world [46]. Also, the repository of devices listed in WURFL is updated every day by contributors in the world. Therefore, it is an up-to-date specification that brings reliability in device data manipulation. A closer look on WURFL and the reasons behind selecting it is explained below. Under the scope of this thesis, proposed system works with a combination of WURFL and a local database. Figure 4.3 shows a detailed proposal for identifying end user´s device.

Explanation of the involved steps in the process

Device recognition (high/low end devices) process is carried out at server end

- As soon as server receives request from the client device, analyses its header information and decides whether it is low end (mobile client/thin client) or high end (desktop /thick client)

  - Note: low end device means, a device which is unable to show the desired output with its capabilities, vice versa for high end device

- For high end devices, formation of content is not required because device is capable enough to render the content

- If a device is low end, then application fetches the capabilities and process the content accordingly so, for fetching capabilities WURFL API is used (in sub-section 4.2.1 it is described why as Device Description Repository WURFL is proposed to be used in this application)

- If it is a new device then capabilities information is fetched from WURFL.xml and stored in a Local database preferred Object Oriented Database (reason behind is explained in sub-section 4.2.2 of this chapter), because if in future again same user requests the application, then no need to fetch it again from the WURFL.xml application just collects it from Local database
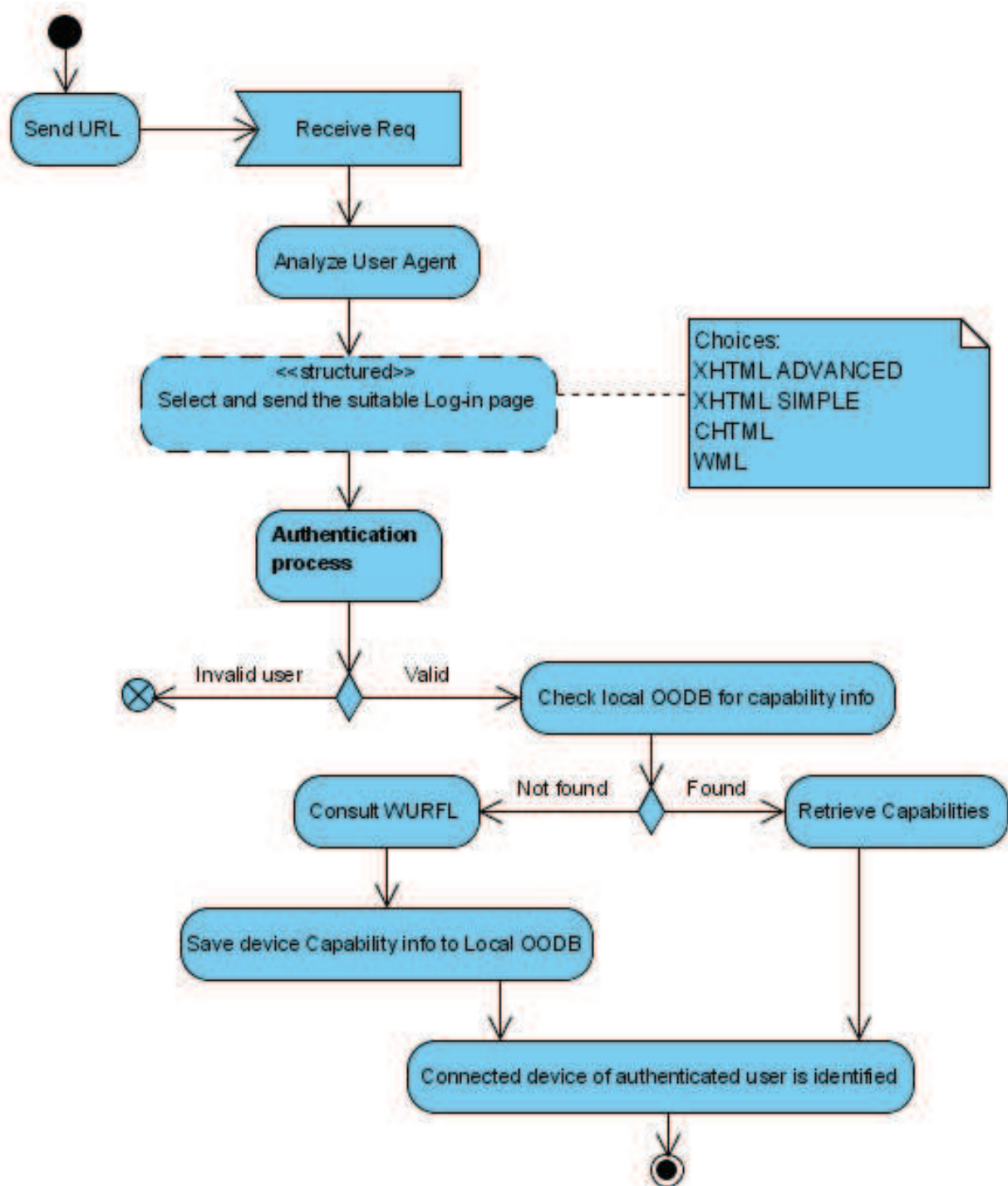
Figure 4.2 Process of device identification

4.2.1 *Comparison among different Device Description Repository (DDR)*

DDR provides device descriptions to enable device-aware applications. Device description provides both knowledge to be able to identify a device from its requests for content, and data about the device attributes so that content adaptation can be effectively performed [47].

There are many different DDR available in the market. The list below is chosen based on ease of use, reliability, potentiality and provided documentation for developers:

- WURFL

- Apache DeviceMap

- 51degrees.mobi

- DeviceRight

WURFL

The WURFL was an open source Device Description Repository (DDR) project run by ScientiaMobile that aims to be an independent central repository of mobile device information. It is used by Facebook and Google as well as many other organizations both large and small [48]. Valuable device information such as display size, version of the mobile browser being used, type of the device, which operating system is installed, whether it can support Flash, and much more can be found within the WURFL database.

In its simplest form, the WURFL is a repository of thousands of mobile devices and their respective capabilities. The DDR consists of an XML configuration file matched with a set of APIs that allows to retrieve a vast amount of information about a given mobile device with little effort. Table 4.1 shows some selected capabilities information of a desktop as sample.

WURFL values:
ContentType: text/html
Brand Name: chrome
Deck Size: 100000
Screen Height: 400
Screen Width: 400
Max Img Height: 600
Max Img Width: 600
Cookie Support: true
AJAX/JS Support: true
Device OS Version:
Device OS:
Is Tablet: false
PDF Support: true
Gif Animated: true
Resolution Height: 600
Resolution Width: 800
Device Id: google_chrome_2
Markup Support: XHTML_ADVANCED

Table 4.1 Sample of WURFL device cababilities information (desktop)

APACHE DEVICEMAP

Apache DeviceMap is an incubating project to create a data repository containing device information, images and other relevant information for all sorts of mobile devices, e.g. Smartphone

and Tablet. It is still under development phase, developers are experimenting with prototypes to gather device data.

```
DeviceMap values:
screen.width =1280
screen.height =1024
screen.availHeight =1024
screen.availWidth =1280
screen.colorDepth =16
screen.pixelDepth =16
navigator.vendor =GoogleInc.
navigator.platform =Win32
this.gives.NOVALUE =null
html5.audio.play =true
html5.video.play =true
dummy.video.foo =false
platform.name =Chrome
platform.version =25.0.1364.172
platform.product =null
platform.layout =WebKit
platform.os =WindowsXP
platform.description = Chrome 25.0.1364.172 on Windows XP
```

Table 4.2 Sample of DeviceMap device cababilities information (desktop)

51DEGREES.MOBI

51Degrees.mobi Device Data – enables a web site to identify a user's web browser and physical device type. For example, web developers can:

- Redirect a customer to a mobile website or improve the customer experience by utilizing other properties, such as screen size and input method;

- Validate a user's device type and software version before offering downloadable content or attempting to interact;

- Tailor the experience by using other functionality on the device, such as the type of keyboard or camera;

Key features of this DDR includes but not limited to – high performance with their APIs (.NET, C, Java and PHP), XML format provides easy access to data, the setup and deployment is very easy, and it is secure, reliable, always available usage installed in user´s environment, unlike cloud services.

DetectRight

DetectRight is a programming library that allows a program to access a database of tens of thousands of mobile devices and browsers (desktop and mobile).

DetectRight's core purpose is taking a Hash map of HTTP Headers (or just a user agent, TAC or UAProfile URL) and using them to create a customised capability profile of the device or browser. Advanced use cases allow maintaining device catalogue lists and keeping them synchronized with DetectRight.

	WURFL	Apache Device Map	DetectRight	51degrees.mobi
Advantages:	Plenty of capability information; Easy to configure; Reliable; Clean API A non commercial version is still available	Open source; Less waiting time	User agent, and UAProfile -URL analysis; Uses all HTTP headers	Easy to use administration tool; Detects over 11000 mobile devices
Disadvantages	Commercial; Waiting time	Still under developing phase	Commercial	Commercial
URL	http://www.ibm.com/developerworks/library/mo-wurfl-1/	http://incubator.apache.org/devicemap/	http://www.detectright.com/	http://51degrees.mobi/Products/DeviceDetection.aspx

Table 4.3 Comparison of four different DDR

Reliability, availability of noncommercial version and ease of use made WURFL suitable for the system. Hence WURFL is selected to be used as DDR for Smoothie. For the justification of the decision made, a closer look on WURFL is provided in following part of this subchapter.

A closer look on WURFL

Smoothie will be using WURFL XML file to collect saved information of a mobile device in it and WURFL Java API for collection process. Nokia 3300 is a randomly selected device used for experimental validation of WURFL.

View of WURFL XML

The wurfl.xml file consists of a single <devices> tag, which wraps a large number of <device> tags. Each individual <device> tag represents a wireless device, <device> tag has three attributes -- id, user_agent, and fall_back -- and a number of <group> tags. The id attribute simply defines an identifier (or ID) for the device. The user_agent attribute wraps a user-agent string that will be included in the HTTP request that a device sends to a Web application (See table 4.4). For a view of a request coming from a low-end device such as Nokia 3300 see table 4.5.

```
<devices>
<device id="generic" user_agent="" fall_back="root">
<group id="markup">
<capability name="preferred_markup" value="wml_1_1"/>
<capability name="wml_1_1" value="true"/>
</group>
<group id="wml_ui">
<!—Device capabilities -->
</group>
<group id="xhtml_ui">
<!—Device capabilities-->
</group>
<group id="chtml_ui">
<!—Device capabilities-->
</group>
<!—Other groups of capabilities-->
</device>
<device id="nokia_generic" user_agent="Nokia" fall_back="generic">
<!—Groups of device capabilities-->
</device>
<device id="nokia_3300_ver1_sub425" user_agent="Nokia3300/1.0 (4.25) Profile/MIDP-1.0
Configuration/CLDC-1.0" fall_back="nokia_3300_ver1"/>
<!—Groups of device capabilities-->
</device>
<!—Other devices-->
</devices>
```

Table 4.4 Device information tags in wurfl.xml

An HTTP request coming from a Nokia 3300 device

```
GET /MultiServeRestaurantMenu/multiServeMenuCategories.jsp HTTP/1.0

User-Agent: Nokia3300/1.0 (4.25) Profile/MIDP-1.0 Configuration/CLDC-1.0

Host: localhost:8080

Accept: text/vnd.wap.wml, image/vnd.wap.wbmp,

application/vnd.wap.wmlc, application/vnd.wap.wmlscriptc
```

Table 4.5 Sample of request sent by a user device

Nokia3300/1.0 (4.25) Profile/MIDP-1.0 Configuration/CLDC-1.0 string is included in the User-Agent parameter of the HTTP request. The same string forms the value of the user_agent attribute of the second <device> tag (see table 4.4). Immediately after receiving the request device identification process will start.

Device Identification steps

WURFL API identifies which wireless device is sending an HTTP request by following these steps:

- It looks at the User-Agent parameter

- It finds a <device> tag with a matching user_agent attribute value in the wurfl.xml file

- WURFL's accompanying Java API provides this function of processing HTTP requests and consulting the wurfl.xml file to learn device capabilities

- WURFL has devised a mechanism to define generic devices and then use generic device descriptions to specify the capabilities of real devices. This mechanism works using the fall_back attribute

The first device tag in Table 4.4 (whose id attribute value is generic) represents the most generic device. The fall_back attribute value of the second <device> tag in Table 4.4 is generic. This means that the second <device> tag uses (or specializes) device descriptions of the generic device.

Also notice that the id attribute value of the second <device> tag in Table 5 is nokia_generic. This means that the second <device> tag serves as the most generic form of all Nokia devices. Actual Nokia devices use device descriptions of the generic Nokia device. When a device specifies a generic device in the wurfl.xml file, it only needs to describe its capabilities that are either missing in the generic device or are different from the capabilities defined in the generic form. Therefore, you need to follow a "fall-back path" within the wurfl.xml file to learn the device capabilities. In order to follow the fall-back path, you can read the fall_back attribute value and then find the more generic device description by finding a <device> tag with a matching ID.

Use of WURFL Java API

Table 4.6 below shows the parsed output for the device Nokia 3300. Source code to detect the device through WURFL Java API is provided in Appendix B

```
Total Matches for "nokia3300": 7
Id: nokia_3300_ver1
UserAgent: Nokia3300
FallBack: nokia_generic_series40
Actual Device Root: true
  [product_info]
  model_name: 3300
  uaprof: http://nds1.nds.nokia.com/uaprof/N3300r200.xml
  [xhtml_ui]
  xhtml_format_as_css_property: true
  xhtml_supports_css_cell_table_coloring: true
  xhtmlmp_preferred_mime_type: application/xhtml+xml
  xhtml_supports_table_for_layout: true
  xhtml_readable_background_color1: #99CCFF
  xhtml_make_phone_call_string: wtai://wp/mc;
  xhtml_table_support: true
  [markup]
```

```
preferred_markup: html_wi_oma_xhtmlmp_1_0
```
Table 4.6 Parsed output of the device Nokia 3300

After collecting device information, next step is to save it in a local storage. Each time when same device sends request to the server, retrieving device capabilities from the WURFL.xml is resource intensive and time consuming. So it is advantageous to store these capabilities information in a local database such as Object Oriented Database (reason behind choosing OODB is explained in sub-section 4.2.2 Analysis of OODB), which has less foot print and faster data processing mechanism.

To show how exactly the data is being stored in OODB, see Appendix C

4.2.2   Analysis of Object Oriented Database (OODB)

Object oriented database management system in which information is stored in the form of Objects, using object oriented programming languages. Bertino, Martino (1991) says Object-oriented database technology combines the expressive power of programming languages with effective support for persistent data typical of database management systems [49]. In relational database model, developer has to follow a technique to convert data between incompatible systems Objects to specific relational table using ORM tools such as hibernate etc. Object oriented database paradigm can reduce the effort of the programmer by persisting object itself, without any mappings. Moreover Object Oriented database can handle simultaneously varieties of data types. Use of OODB improves productivity and specially provides ease of data access [50].

Since the proposed architecture under the scope of this thesis is implemented by using Object Oriented Programming, to avoid object-relational impedance mismatch, faster access in the database, very less foot-print compare to relational database management systems, and the efficiency of dealing with massive amounts of data about one item such as a end user device - Object oriented database was chosen to save device information collected from WURFL database.

To be specific, in this architecture Object oriented database is used to persist capabilities information of the remote device in server end, here device information is gathered and processed within fraction of second (Sample data is provided in Appendix B section iii), in order to display the content according to device capabilities. The fastest and simplest way of achieving this task is by using OODB. Below a list is provided to show a comparison among some of the available OODB in the market followed by their features:

Features of Object Oriented Database model defined by Hurson, Pakzad & Cheng (1993) [51]

- An object is a package of information (data) and descriptions of its manipulation (procedures).

- A class is a description of one or more similar objects.

- A message is specification for an object's manipulations.

- A method is a procedure-like entity that describes a sequence of actions to be performed upon receiving a message.

- An instance is an object described by a particular class.

- The basic object modeling primitives are object, literal, unique identifier.

- The object state is defined by the values carried by its attributes or relationships.

- The object behavior is defined by the method.

- The objects and literals are categorized by their type.

4.2.2.1 Available ope source OODB

According to Wikipedia, the free encyclopedia there are 19 notable Object Oriented Databases available in the market as listed in the link http://en.wikipedia.org/wiki/Comparison_of_object_database_management_systems, retrieved on 11.02.2014. Below a list of open source or under certain rules free to use object oriented database is provided along with brief introduction of the database and its supporting features. The goal of this list is to find out the most suitable database for Smoothie.

1) **ConceptBase** is a deductive and object-oriented database management system developed at University of Aachen and Tilburg University. It is mainly used for conceptual modeling and metamodeling in the domain of software engineering and related domains [52]. (URL : http://conceptbase.sourceforge.net/ visited on 12th February 2014)

Features of ConceptBase:

- Object-oriented concepts such as classes and inheritance

- Deductive rules evaluated by a Datalog engine

- Active rules conforming to the event condition action (ECA) paradigm

- Recursive function definitions

- Meta-modeling with arbitrarily many abstraction levels (metaclasses, meta metaclasses)

- Has CBQL (is a query language) support based on Datalog

2) **Twig** is an object persistence interface built on Google App Engine's low-level datastore API. (URL: https://code.google.com/p/twig-persist/ visited on 12th February 2014).

Twig data store interface supports:

- Parallel Asynchronous Commands
- Plain old Java object data models with no datastore dependencies
- Merged OR queries on multiple properties

3) **Db4o** (database for objects) is an embeddable open source object database for Java and .NET developers. It is developed, commercially licensed and supported by Versant. Db4o can run on any operating system that supports Java or .NET. DB4o is offered under the GNU General Public License (GPL), the DB4o Opensource Compatibility License (dOCL), and a commercial license for use in proprietary software. (URL: http://www.db4o.com/ visited on 12th February 2014)

Features of Db4o:

- Single command object storing: DB4o contains a function to store any object with a single command: "objectContainer.store(new SomeClass());" To handle special cases is also simple, through writing custom type handlers.
- Embeddable: designed to be embedded in clients or other software components completely invisible to the end user
- Client-server mode: Db4o uses TCP/IP for client-server communication and allows configuring port number. Communication is implemented through messaging.
- Dynamic schema evolution: Db4o supports automatic object schema evolution for the basic class model changes such as field name deletion/addition etc.
- Replication: It is a special feature of DB4o which makes it a better choice for the desired system
  - Replication in DB4o uses hibernate functionality, that synchronizes DB4o database with hibernate enhanced RDBMS (such as SQL) database

Advantage of this replication is, if DB4o is crashed or if the DB4o generated database file is lost, our data is safe inside the relational database.

4) **ObjectDatabase++ (ODBPP)** is an embeddable object-oriented database designed for server applications that require minimal external maintenance. (URL: http://www.ekkysoftware.com/ODBPP/ visited on 12th February 2014)

Features of ODBPP

- Full multi-process and multi-thread transaction control
- Auto real-time database recovery

- Hierarchical object data design

- Native code and script access

- Static hash index on object IDs

- Numerous supported index methods including full-text

- Drawback:

    - there is no support for java

5) **ODABA** is a terminology-oriented database management system, which is a conceptual extension of an object-oriented database system, and implements concepts defined in a terminology model. (URL: http://www.odaba.com/content/tools/odaba/ visited on 12th February 2014)

Features of ODABA:

- Database access is supported via an application program interface for C++ or .NET programming languages and via the ODABA Script Interface

- Beside standard models (object model, functional model and dynamic model), ODABA supports a documentation model and an administration model. In order to be terminology model compliant, several conceptual extensions are supported as set relations, multilingual attributes, weak-typed collections or hierarchical enumerations (classifications).

- ODABA supports semi-automatic conversion from terminology models to object models and schema conversion from object model to relational models (MS SQL Server, MySQL, Oracle) which allows storing or mirroring ODABA data in relational databases or in XML files.

6) **Perst** is an open source, dual license, object-oriented embedded database management system (ODBMS). Both the Java programming language, and the C# programming language versions are compact and Perst has been implemented on smart phones running the Android and Windows Phone (WP7) operating systems (URL: http://www.mcobject.com/perst)

Features of Perst:

- Compact: Perst's core consists of only five thousand lines of code. The small footprint imposes minimal demands on system resources.

- Garbage collection

- Schema evolution

- Multi-threaded

- Support for Language Integrated Query (LINQ)

- XML import/export utilities

- Full text search

- Transparent persistence: Perst is distinguished by its ease in working with Java and C# objects, and suitability for aspect-oriented programming with tools such as AspectJ and JAssist. The result is greater efficiency in coding.

- Replication: It is a special feature of Perst which makes it a choice for the desired system. It has master/slave replication support, with the option to run read-only queries on slave nodes which is used mainly for two purposes:

    - Fault tolerance

    - Load balancing

Decision regarding Suitable Object Oriented Database for the Application

Db40 is chosen as preferred database for this application because of its support on replication and Object Manager Enterprise (OME). OME is a special feature in DB4o, which is a plugin for eclipse that makes a visual presentation of Objects in the database. The reasons behind this preference are

- For experimental validation of Smoothie it is planned to be integrated with a Flashcard based learning system named Language Learning Game (LLG) [53]. It is designed and developed in Offenburg University of Applied Sciences, Germany in 2010 as a Master thesis project. The goal of this game is to help all foreign student of the institute in acquisition of a foreign language such as German. Here it is neither predictable that, what kind of devices will be used by the student for accessing the learning content; nor predictable that, what kind of data formats the content provider will be uploading as a learning material. The existing database used for LLG is MYSQL. So the ease of the replicating process and communication between DB4o and MYSQL made DB4o a better candidate for Smoothie.

- Perst supports replication as well but it is designed for master-slave replication support. In LLG everything is controlled by only one server that is the master node, which means it supports only one master node. There is no existence of slave node in the design of the game. That is why Perst´s master-slave replication support is not suitable for this application.

After identification and saving identified device capabilities in case of first time use of the device, next step to be explained is how to prepare generalized content. Device identification and preparing generalized content are completely two independent processes to be done separately.

4.3   Preparing Generalized Content including 3D Data

This section will provide a discussion and proposal of the process for collection of data/content then prepare and save generalized form of that content as generalized content. As depicted in figure 3.3 Smoothie is receiving unpredictable format of data as document from different domain and it has to send those documents as renponse in device dependent format in different domains which are not predictable either. To meet this design goal Smoothie has to translate received data in a generalized form as an intermediate step. Such a generalized form of data has to be

- Domain-independent

- Self describing

- Machine and human readable

- Can be used for any kind of data received by Smoothie as input

- Possible to transform i.e. many views of the one data

- Java compatible

- Supported by web-standard

- Open and extensible

- Complete integration of all traditional databases and data formats and above all

- Widely adopted by the computer industry as a common document exchange format

As generalized form of content, Extensible Markup Language (XML) and for 3D content, an xml extended format namely Digital Asset Exchange (DAE) have been chosen. The driving forces behind the decision are given below respectively.

4.3.1   Why XML

There are three data exchange formats found those can fit to the above mentioned requirements namely `YAML Ain't Markup Language` (YAML) [54], `JavaScript Object Notation` (JSON) [55], and `Extensible Markup Language` (XML) [56]. The table 4.7 below shows comparison among them to find the most suitable one to be used as the format of generalized content by Smoothie.

Feature	JSON	YAML	XML
Availability of Schema	Partial	Partial	XML Schema
Availability of Specification	RFC7159	RFC2822	W3C recommendations

Flexiblity	No	No	Yes
Web standard	Yes	No	Yes
Transformations	No	No	Yes
Lightweight	Yes	Yes	No
Human readable	Yes	Yes	Yes
Compatibility with Java	Yes	Yes	Yes

Table 4.7 Comparison among data exchange formats

Nomenclature

- Schema - Whether the language definition is available in a computer interpretable form.

- Specification – Explicit set of requirements to be satisfied

- Flexiblility - Whether the language enables extension of the semantic expression capabilities without modifying the schema.

- Web or ISO standard - Organization that endorsed the language as a standard.

- Transformations - Whether the language includes a translation to other standards.

- Lightweight - Whether a lightweight version is available, in addition to a full version.

- Human readable - Whether expressions in the language are readable by humans without training.

- Compatibility - Which other tools are possible or required when using the language.

According to last specification of YAML Version 1.2, 3rd edition, JSON is a subset of YAML [57]. Moreover JSON is accepted by web standard whereas YAML is not (see table 4.7). Hence a deeper discussion on comparison between JSON and XML is provided below in table 4.8 [58].

Feature	JSON	XML
Simplicity	Simpler	Simple
Extensibility	No	Yes
Interoperability	Yes	Yes
An exchange format to enable users to move their data between similar applications	Yes	Yes
Provides a structure to data so that it is richer in information	Yes	Yes

Availability of reusable software to programmers to handle, so re-invention of code is not necessary	JSON, being a simpler notation, needs much less specialized software. In the languages JavaScript and Python, the JSON notation is built into the programming language; no additional software is needed. In other languages, only a small amount of JSON-specific code is necessary. For example, a package of three simple classes that makes JSON available to Java. The package is available for free from JSON.org.	Yes
Separates the presentation of data from the structure of that data.	No. JSON structures are based on arrays and records. That is what data is made of.	Yes. XML requires translating the structure of the data into a document structure. Those structures are based on elements (which can be nested), attributes (which cannot), raw content text, entities, DTDs, and other meta structures.
A common exchange format	JSON is a better data exchange format	XML is a better document exchange format
Many views of the one data	JSON does not provide any display capabilities because it is not a document markup language.	Yes
Self-Describing Data	Yes	Yes
Complete integration of all traditional databases and data formats	JSON does not have a <[CDATA[]]> feature, so it is not well suited to act as a carrier of sounds or images or other large binary payloads. JSON is optimized for data. Besides, delivering executable programs in a data-interchange system could introduce dangerous	XML documents can contain any imaginable data type - from classical data like text and numbers, or multimedia objects such as sounds, to active formats like Java applets or ActiveX components.

		security problems.
Internationalization/Encoding	Use Unicode	Use Unicode
Open and extensible	Possible through migration from XML to JSON	XML has open structure that allows adding other state-of-the-art elements when needed. That means it is always possible to adapt the system to embrace industry-specific vocabulary.
Object-oriented	JSON is data-oriented. Can be mapped more easily to object-oriented systems	XML is document-oriented. Easy to map to object oriented systems
Being widely adopted by the computer industry	JSON is just beginning to become known.	Yes
Transmission times and resource utilizations	Faster and uses fewer resources than XML [59].	XML is slower and uses more resource compare to JSON in client/server program operational environment.

Table 4.8 comparison between JSON and XML

From the above stated table it can not be derived that JSON has advantages over XML or vice versa. Which one is more advantageous to use depends upon the design goal of the system.

According to the design, Smoothie has to deal with document as input and output in and out of the architecture (see figure 3.1). XML is a better document exchange format compare to JSON; it separates the presentation of data from the structure of that data so it facilitates many views of the one data; it is open and extensible; widely adopted by the computer industry and it has easy integration possibly to all traditional databases and data formats. Hence use of XML would make Smoothie future proof. For justification of the study a closer look on XML is provided below.

The initial design goals for XML were:

- XML shall be straightforwardly usable over the Internet.

- XML shall support a wide variety of applications.

- XML shall be compatible with SGML.

- It shall be easy to write programs which process XML documents.

- The number of optional features in XML is to be kept to the absolute minimum, ideally zero.

- XML documents should be human-legible and reasonably clear.

- The XML design should be prepared quickly.

- The design of XML shall be formal and concise.

- XML documents shall be easy to create.

- Terseness in XML markup is of minimal importance.

W3C provided XML specification, together with associated standards (Unicode [Unicode] and ISO/IEC 10646 [ISO/IEC 10646] for characters, Internet BCP 47 [IETF BCP 47] and the Language Subtag Registry [IANA-LANGCODES] for language identification tags), provides all the information necessary to understand XML Version 1.0 and construct computer programs to process it [60]. Besides

- Several research results have shown that XML is adequate and a powerful technology to store content in a presentation independent manner

- It helps the author to generate learning material for different devices in an efficient and structured way

- The material can be used in different formats such as XHTML, PDF, etc. as well as with different technologies such as browser, applet, MIDlet, Ajax, etc.

- XML is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, Now XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

Moreover, according to Jon Bosak (1997) the applications that will drive the acceptance of XML can be divided into four broad categories [61]:

- Applications that require the Web client to mediate between two or more heterogeneous databases.

- Applications that attempt to distribute a significant proportion of the processing load from the Web server to the Web client.

- Applications that require the Web client to present different views of the same data to different users.

- Applications in which intelligent Web agents attempt to tailor information discovery to the needs of individual users.

Since the goal of the thesis is to support variety of devices along with variety of data formats as document having different kinds of database as storage, XML is the most suitable format to meet the project requirements. Why .dae was selected as a generalized content format for 3D computer graphics is explained in sub-section 4.3.2 with an analysis of 3D graphic formats.

4.3.2 Why DAE for 3D data

As it is explained in chapter 2 sub-section 2.3 Research on 3D Enabled Projects, there are several sectors where 3D data format is used for most useful applications. Those sectors include not only academy but also industry and business to meet group as well as individual purposes. There are plenty of 3D data formats available out there in the market. To create and render those 3D data formats there are plenty of application available as well. Below a list of 3D modeler/renderer application is provided based on the features of the raw software itself without additional plug-in (the research was conducted in February 2014).

Application	Supported Platforms	Main usage	License
3ds Max [62]	Microsoft windows	Modeling, Animation for video games, Lighting, Rendering	Proprietary (A 30 day trial & Free student software license for 3 years)
3d View [63]	Microsoft windows Mac OS X, Chrome OS	Visualization, Sharing, Quality control	Proprietary (a student trial version for free)
AC3D [64]	Microsoft windows Mac OS X, Linux	Modeling	Proprietary
AOI [65]	Mac OS X, Windows, Linux, Unix/Other	Modeling, Animation, Lighting, Rendering	**GNU GENERAL PUBLIC LICENSE** [66]
Blender [67]	AmigaOS 4, MorphOS, FreeBSD, Linux, Mac OSX & Microsoft Windows	Modeling, Animation, Lighting, Rendering, Video Game Creation, Visual 3D effects, Sculpting, Basic Post-Production Video Editing	**GNU GENERAL PUBLIC LICENSE** [66]
Bryce [68]	Windows NT 4, 2000, XP, Vista & 7; Mac OS X Tiger, Leopard & Snow Leopard	Modeling, Animation, Landscape Fractal Geometry	Proprietary
Carrara [69]	Windows XP, Vista & 7; Mac OS X Tiger, Leopard & Snow Leopard	Modeling, Animation	Proprietary

Cheetah 3D [70]	Mac OS X	Modeling, Animation	Proprietary
Cinema 4D [71]	Apple Mac OS X 10.7.5~(and higher), Windows 8.1 64-bit Windows 8 64-bit Windows 7 64-bit Window 2012 Server 64-bit	Modeling, Animation, Lighting, Visual 3D effects, Rendering, Simulation	Proprietary (Free student software license for 18 months)
CityEngine [72]	Mac OS X, Microsoft Windows & Linux	Procedural modeling of 3D cities	Proprietary (a 30day free trial is available)
Clara.io [73]	Browser base	Modeling, Animation, Rendering	Freemium
Cobalt [74]	Microsoft Windows, Mac OSX	Modeling, Animation, Computer Aided Design	Proprietary
Electric Image Animation System [75]	Microsoft Windows, Mac OSX	Lighting, Animation, Rendering, Film, Television, Visual 3D effects	Proprietary
from-Z/from-Z Renderzone Plus [76]	Microsoft Windows, Mac OSX	Lighting, Animation, Rendering, Modeling, Fabrication, Set design	Proprietary
Hexagon [77]	Windows 2000 & XP; Mac OS X v10.2Tiger, Leopard, Snow Leopard & Panther	Subdivision Modeling, UV Mapping	Proprietary
Houdini/Houdini FX [78]	Mac OS X, Microsoft Windows & Linux	Animation, Lighting, Modeling, Visual 3D effects	Proprietary
LightWave 3D [79]	Mac OS X, Microsoft Windows	Animation, Lighting, Modeling, Rendering, Film and Television Previews, Videogame Asset Creation	Proprietary (30 day trial version)
MakeHuman [80]	Mac OS X, Microsoft Windows & Linux	Modeling of human figures	**AFFERO GENERAL PUBLIC LICENSE** [81]
Massive [82]	Microsoft Windows	Crowd related visual effects &	Proprietary

	& Linux	Autonomous character animation	
Maya [83]	Microsoft Windows 7 (SP1), Windows 8 and 8.1 Professional operating system Apple Mac OS X 10.8.5 and 10.9.x operating system Red Hat Enterprise Linux 6.2 WS operating system Fedora 14 Linux operating system CentOS 6.2 Linux operating system	Modeling, Animation(Video), Lighting, Rendering, Visual 3D effects	Proprietary(A 30 day trial & Free student software license for 3 years)
Metasequoia [84]	Microsoft Windows	Modeling	Shareware [85]
MicroStation [86]	Microsoft Windows	2D & 3D drafting and design, Modeling	Proprietary (30 day trial version)
Modo [87]	Microsoft Windows, Linux & Mac OS X	Modeling, Animation, Rendering	Proprietary (15 day trial)
POV-Ray [88]	Microsoft Windows, Linux & Mac OS X	Lighting, Visual 3D effects	**AFFERO GENERAL PUBLIC LICENSE** [89]
PTC Creo [90]	Microsoft Windows & Unix	Modeling	Proprietary (30 day trial)
Remo 3D [91]	Microsoft Windows & Linux	Creating 3D models for real-time visualization	Proprietary (A time limited trial)
Rhinoceros 3D [92]	Microsoft Windows	Modeling, Computer Aided Design	Proprietary (A 25 times save trial)
Sculptris [93]	Microsoft Windows, Mac OS X	Sculpting, Texturing	**FREE FOR ANY KIND OF USE**
Shade 3D [94]	Microsoft Windows, Mac OS X	Sculpting, Texturing	Proprietary (A time limited trial)

Silo [95]	Microsoft Windows, Mac OS X	Modeling	Proprietary (A 30 day trial)
SketchUp/SketchUp Pro [96]	Microsoft Windows, Mac OS X	Computer Aided Design	Proprietary (A freeware version named SketchUp Make)
Softimage [97]	Microsoft Windows, Linux	Modeling, Animation, Video-game creation, Lighting, Rendering, Visual 3D effects	Proprietary (A freeware 3 year student software)
Solid Edge [98]	Microsoft Windows	Computer Aided Design	Proprietary (A 45 day trial)
SolidThinking [99]	Microsoft Windows & Mac OS X	Computer Aided Design	Proprietary (A time limited trial)
SolidWorks [100]	Microsoft Windows	Computer Aided Design	Proprietary (A time limited trial)
SpaceClaim [101]	Microsoft Windows	Computer Aided Design	Proprietary (A time limited trial)
Swift 3D [102]	Microsoft Windows & Mac OS X	Modeling, Animation	Proprietary (A time limited trial)
E-on Vue [103]	Microsoft Windows & Mac OS X	Modeling, Animation	Proprietary (A 30 day trial)
Wings3D [104]	Microsoft Windows Mac OS X & Linux	Modeling, Sculpting, UV mapping	**BSD LICENSE** [105]
ZBrush [106]	Microsoft Windows & Mac OS X	Modeling, Texturing, Lighting, Rendering	Proprietary

Table 4.9 List of 3D Applications available and active in February 2014

From the above stated summary of 3D applications, it can be derived that there are only six applications available for free. And three more applications have free student license for a certain period of time. To meet the design goal of Smoothie, the next questions to be answered are – among those nine applications which application is

- able to support maximum variety of operating systems as supported platforms
- covering versatile area as main usage
- able to import maximum variety of other 3D data formats and

- free and easy to use for Smoothie according to the policy of License

In addition which 3D data format is the most common exchangeable export format of those nine applications? (Refer to table 4.19 to find the answers of the above stated questions. Answers are derived from table 4.9 to table 4.18)

1) **Wings 3D-** cross-platform, open source, loads and saves models in its own format (.wings). It supports several standard 3D formats as import and export, see table below.

Imports	Exports
- Nendo (.ndo) 3D Studio (.3ds) - Adobe Illustrator (.ai) - Lightwave/Modo (.lwo/.lxo) - Wavefront (.obj) - Adobe PostScript (.ps/.eps) - Stereolithography (.stl) .svg Path	- Nendo (.ndo) 3D Studio (.3ds) - BZFlag (.bzw) - Cartoon Edges (.eps) **Collada (.dae)** - Kerkythea (.xml) - Lightwave or Modo (.lwo/.lxo) **Wavefront (.obj)** - POV-Ray (.pov) - Renderware (.rwx) - Stereolithography (.stl) - VRML 2.0 (.wrl) - DirectX (.x)

Table 4.10 Export and import formats of Wings 3D

2) **Sculptris** has compatibility with Microsoft windows and Mac OS X, freeware, loads and saves models in its own format (.sc1). It supports some standard 3D formats as import and export, see table below.

Imports	Exports
- Mesh files - Wavefront (.obj) - ZBrush (GoZ)	**Wavefront (.obj)** - ZBrush (GoZ)

Table 4.11 Export and import formats of Sculptris

3) **POV-Ray** has support for Microsoft Windows, Linux & Mac OS X. It is open-source, loads and saves models in its own format (.pov). See table below to find other data formats it supports as import and export

Imports	Exports
• .inc • .mac • .mcr • .txt • .ini	• .inc • .mac • .mcr • .txt • .ini

Table 4.12 Export and import formats of POV-Ray

None of those formats are considered as either standard 3D data format or digital asset exchange format.

4) **MakeHuman** is compatible with Microsoft Windows, Linux & Mac OS X, open-source, loads and saves models in its own format (.mhm). It supports some standard 3D formats as import and export, see table below.

Imports	Exports
• Not Available	• .mhx • .fbx • **.dae** • **.obj** • .stl • Ogre3D

Table 4.13 Export and import formats of Make-Human

5) **Blender** 2.67 support is currently available for Microsoft windows, Mac OS X and GNU/Linux. It is open source, loads and saves models in its own format (.blend) but exports and imports many other 3D file formats. See table below to have a list of possible export and import formats of Blender

Imports	Exports
• Scalable Vector Graphics (.svg)	• **Collada (.dae)**

• GIMP Image to Scene (.xcf, .xjt)	• Save As Game Engine Runtime
• Images as Planes	• Acclaim Motion Capture (.amc)
• Collada (.dae)	• Motion Capture (.bvh)
• Acclaim Skeleton File (.asf)	• Cameras & Markers (.py)
• Acclaim Motion Capture (.amc)	• Nuke (.chan)
• Motion Capture (.bvh)	• Adobe After Effects (.jsx)
• Graphics Lab Motion Capture (c3d)	• Autocad (.dxf)
• Nuke (.chan)	• Pointcache (.pc2)
• HiRISE DTM from PDS IMG (.img)	• Skeleton Mesh (.psk)
• Autocad (.dxf)	• Skeleton Animation data (.psa)
• LightWave Object (.lwo)	• Protein Data Bank (.pdb)
• MakeHuman (.mhx)	• Stanford (.ply)
• Skeleton Mesh (.psk)	• Raw Faces (.raw)
• Skeleton Animation data (.psa)	• Stl (.stl)
• Protein Data Bank (.pdb)	• 3D Studio (.3ds)
• Stanford (.ply)	• Autodesk FBX (.fbx)
• Raw Faces (.raw)	• Quake MAP (.map)
• Stl (.stl)	• MilkShape 3D (.ms3d)
• 3D Studio (.3ds)	• **Wavefront (.obj)**
• MilkShape 3D (.ms3d)	• VRML2 (.wrl)
• Wavefront (.obj)	• DirectX (.x)
• X3D Extensible 3D (.x3d/.wrl)	• X3D Extensible 3D (.x3d)
• Lightwave Point Cache (.mdd)	• Lightwave Point Cache (.mdd)

Table 4.14 Export and import formats of Blender

6) **AOI** cross-platform, open source, loads and saves models in its own format (.aoi). It supports very few standard 3D formats as import and export, see table below.

Imports	Exports
• Wavefront (.obj)	• **Wavefront (.obj)**
	• VRML
	• Povray V3.5 (.pov)

Table 4.15 Export and import formats of AOI

7) **3ds Max** 2014 has support for Microsoft windows,loads and saves models in int own format (.3ds). See table below to get a list of other supported data formats by 3ds Max as import and export

Imports	Exports
• Autodesk (.FBX)	• Autodesk (.FBX)
• 3D Studio Mesh (.3DS, .PRJ)	• 3D Studio Mesh (.3DS)
• Adobe Illustrator (.AI)	• Adobe Illustrator (.AI)
• Autodesk Packet Files (.APF)	• ASCII Scene Export (.ASE)
• ProEASM(.ASM)	• **Autodesk Collada (.DAE)**
• Catia V5 (.CATPART, .CGR, .CATPRODUCT)	• Publish to DWF (DWF)
• Autodesk Collada (.DAE)	• AutoCAD (.DWG, .DXF)
• LandXML/DEM/DDF(.DEM, .XML,.DDF)	• Flight Studio OpenFlight (.FLT)
• AutoCAD Drawing (.DWG, .DXF)	• Motion Analysis HTR Files (.HTR)
• Legacy AutoCAD (.DWG)	• IGES (.IGS)
• Flight Studio OpenFlight (.FLT)	• **gw: .Obj-Exporter (.OBJ)**
• Motion Analysis HTR Files (.HTR)	• PhysX and APEX (.PXPROJ)
• IGES (.IGE, .IGES, IGS)	• ACIS SAT (.SAT)
• Autodesk Inventor (.IPT, .IAM)	• StereoLitho (.STL)
• JT (.JT)	• Shockwave 3D Scene Export (.W3D)
• Catia V4 (.MODEL, .DLV4, .DLV3, .DLV, .EXP, .SESSION, .MDL)	• Autodesk Alias (.WIRE)
• Obj-Importer (.OBJ)	• VRML97 (.WRL)
• ProE(.PRT)	
• UG-NX(.PRT)	
• ACIS SAT (.SAT)	
• 3D Studio Shape (.SHP)	
• Google SketchUp (.SKP)	
• SolidWorks (.SLDPRT, .SLDASM)	

StereoLitho(.STL)	
STEP (.STP, .STEP)	
Motion Analysis TRC Files (.TRC)	
Autodesk Alias (.WIRE)	
VRML (.WRL, .WRZ)	
VIZ Material XML Import (.XML)	

<div align="center">Table 4.16 Export and import formats of 3ds Max</div>

8) **Maya** 2014 has support for Microsoft windows, Linux and Mac OS X. It loads and saves models in its own format (.ma). See table below to get a list of other supported data formats by Maya as import and export

Imports	Exports
• Maya Scenes	• Maya ASCII
• Maya ASCII	• Maya Binary
• Maya Binary	• mel
• mel	• move
• obj	• editMA
• audio	• editMB
• move	• Autodesk Packet File
• EPS	• SPF_DCE
• Adobe Illustrator	• STL_DCE
• image	• UG_DCE
• fluidCache	• IGES_DCE
• editMA	• DWG_DCE
• editMB	• DXG_DCE
• Autodesk Packet File	• ZPR_DCE
• JT_DC	• lv2
• PTC_DC	• GE2
• SPF_DC	• **OBJexport**
• STEP_DC	• FBXexport
• STL_DC	• mentalRay

• SW_DC	• vrml2
• UG_DC	• rtg
• IV_DC	• OpenFlight
• CSB_DC	• animExport
• IGES_DC	• atomExport
• DWG_DC	• bullet
• DXF_DC	• **OpenCOLLADA exporter**
• CATIAV4_DC	
• CATIAV5_DC	
• IPT_DC	
• Alembic	
• studioImport	
• FBX	
• DAE_FBX	
• OpenFlight	
• animImport	
• atomImport	
• OpenCOLLADA importer (free plugin)	

Table 4.17 Export and import formats of Maya

9) **Cinema 4D** R15.057 has support for Microsoft windows and Mac OS X. It loads and saves models in its own format (.c4d). See table below to get a list of other supported data formats by Cinema 4D as import and export

Imports	Exports
• .3ds	• 3ds
• .abc	• .abc
• .bvh	• .xml (Allplan)
• .c4d	• .bullet
• .dae (collada 1.4 &1.5)	• .xml (Cinema 4D)
• .dem	• **.dae (collada 1.4 &1.5)**
• .dwg	• .x

• .dxf	• .dxf	
• .fbx	• .fbx	
• .lws	• .ai	
• .ai	• .rib	
• .igs	• .stl	
• .stl	• **obj**	
• .wrl	• .wrl	
• .obj		

Table 4.18 Export and import formats of Cinema 4D

Feature (Total_number)	Wings 3D	Sculptris	POV-Ray	MakeHuman	Blender	AOI	3ds Max	Maya	Cinema 4D
Supported Platforms (at least)	3	2	3	3	4	3	1	3	2
Main usages	3	2	2	1	8	4	4	5	6
Import formats	8	3	5	0	24	1	29	37	15
Opensource	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
Exports .obj	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Exports .dae	Yes	No	No	Yes	Yes	No	Yes	Yes	Yes

Table 4.19 Summary of features of selected 3D applications

From the above stated features of nine, either open-source or freeware 3D applications, which are the most possible cases to be used by the individuals or by the companies to create their content; it is seen that

- most supported data format is .obj and second most supported format is .dae

- Blender supports maximum number of heterogeneous platforms, covers maximum number of main usages including modelling, and most importantly has source code open.

- Maya imports maximum number of heterogeneous import formats but closed source.

- Wings 3D suppots equal number of platforms as Maya but has less capability in terms of import formats and main usages compare to maya. Having open source code is an advantage of Wings 3D.

For a comparison among Blender, Maya and Wings 3D in terms of suitability with the design goal of Smoothie refer to table 5.3 to 5.10. A further investigation on .obj and .dae is provided below to find out the generalized content format for 3D data.

A closer look on OBJ

It is a geometry definition file format for 3D models, initially developed by Wavefront Technologies [107]. The OBJ file format is a simple data-format that represents 3D geometry with the position of each vertex, the UV position of each texture coordinate vertex, vertex normals, and the faces that make each polygon defined as a list of vertices, and texture vertices. While loading first vertices, (depicted in the picture below figure 4.3 with red dots) then faces defining which vertices to connect (connecting vertices, depicted in the figure 4.4 with blue and light blue line) has to be loaded. A valid vertex index starts from 1 and matches the corresponding vertex elements of a previously defined vertex list. Each face can contain three or more vertices.
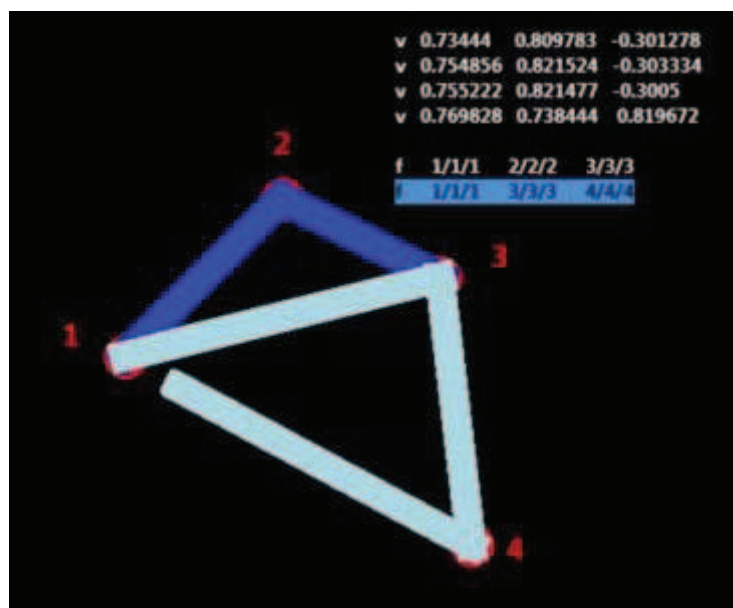
Figure 4.3 Vertices and faces of OBJ

After loading all the vertices and faces a model will look like a wire connected object depicted below in figure 4.4
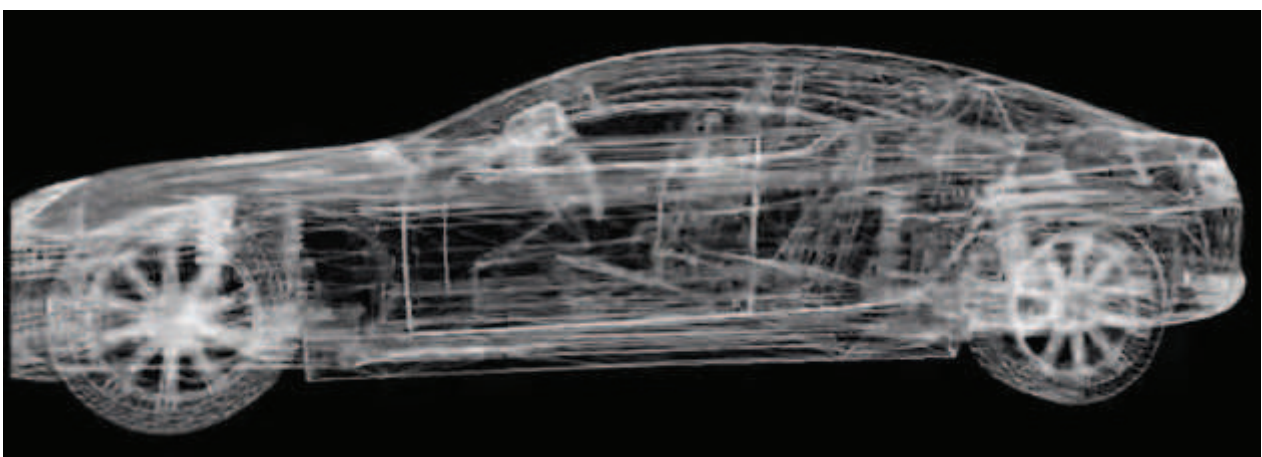
Figure 4.4 3D model of a car in OBJ with vertices and faces

To reflect the light on the model Vertex Normal has to be added and then the above stated model will look like figure 4.5 below.
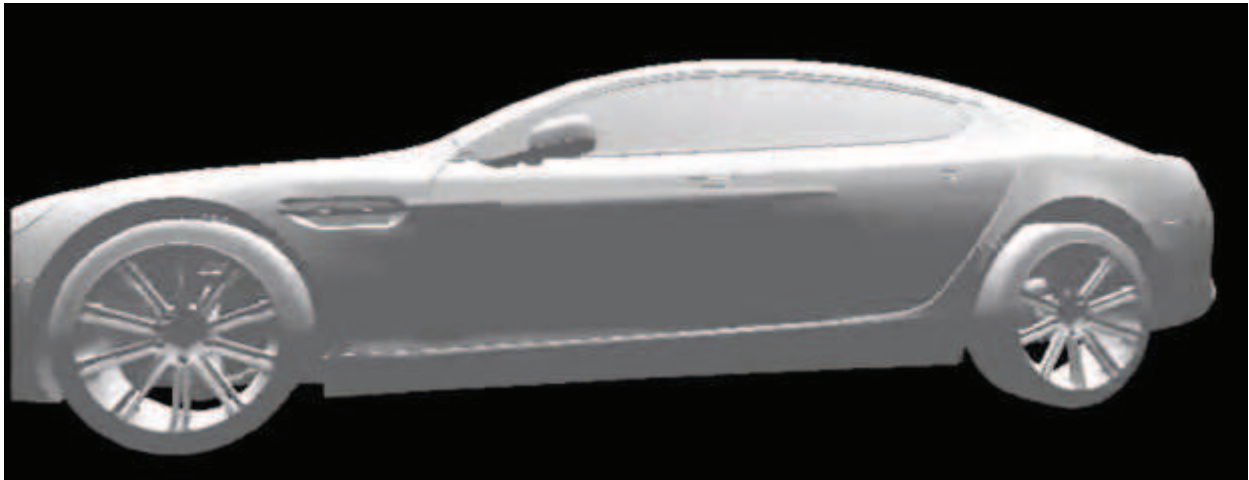
Figure 4.5 3D model of a car in OBJ with vertices, faces and vertex normal

Texture has to be loaded to make the model look more realistic and it is called Vertex Texture Co-ordinates. It could also be done while defining faces. After adding texture the model depicted in figure 4.5 will look like figure 4.6 below. OBJ coordinates have no units [108], but OBJ files can contain scale information in a human readable comment line.

Figure 4.6 3D model of a car with vertices, faces, vertex normal and vertex texture coordinates

A single texture can make some part of the model look real but it is not equally applicable every-where. To apply more color and texture in a model, OBj referfs to one or more external Materials file having extension .mtl. If right material is connected with right texture, the above stated model in figure 4.6 will look like figure 4.7 below.
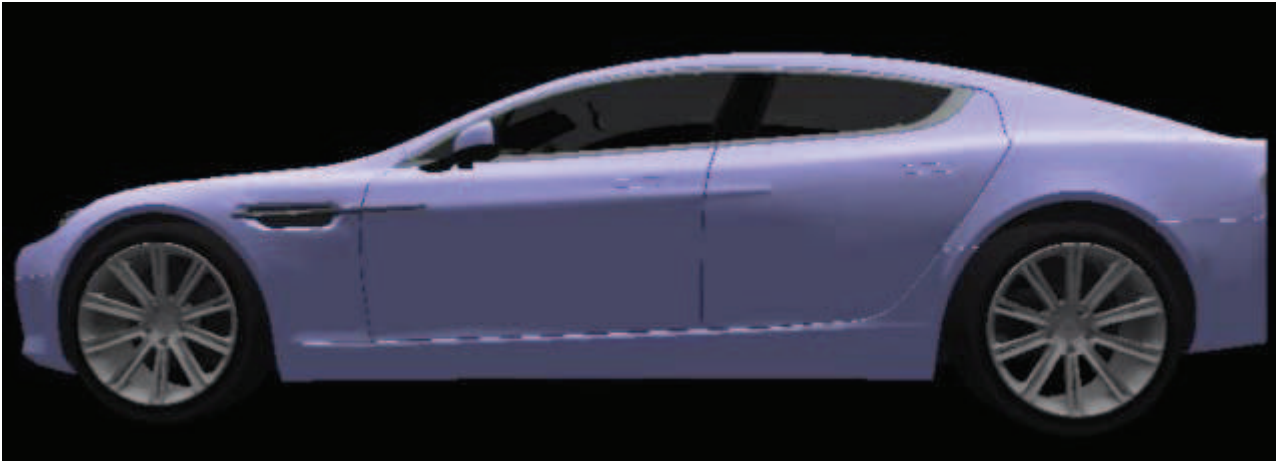
Figure 4.7 Model with vertices, faces, vertex normal, vertex texture coordinates and material file

OBJ is open to use, text based but not XML compatible. For exchanging basic model it is adopted by many 3D graphics applications. In this regard it could be called as universally accepted format. OBJ files support higher-order surfaces using several different kinds of interpolation, such as Taylor and B-splines, but support for those features in third party file readers makes OBJ being far from universal [109]. Obj files also do not support mesh hierarchies or any kind of animation or deformation, such as vertex skinning or mesh morphing. It is not entirely human readable. While importing and exporting .mtl has to accompany .obj, which is a problem for some 3D Digital Content Creation application such as Blender. Above all the design goal of OBJ was not for data exchange. Hence OBJ is not satisfying the appropriate conditions to be the format of generalized content for 3D data in Smoothie. Refer to Appendix G for an example of a OBJ file with MTL.

A closer look on DAE

As derived from table 4.19, the second most supported data format is COLLADA having file extension .dae. It is an interchange file format for interactive 3D applications, managed by the nonprofit technology consortium named the Khronos Group and has been adopted by ISO as a publicly available specification through ISO/PAS 17506 [110]. It is XML extended format which means COLLADA documents that describe digital assets are XML files. Since COLLADA was originally intended as an intermediate format for transporting data from one digital content creation (DCC) tool to another application, the most amount of available 3D graphics format could be converted to this format [111]. It has all the properties of OBJ depicted in figures 4.3 to 4.7 namely vertex attributes, materials and Textures and it is not a must to have external file mapping for multiple texture. Refer to Appendix H for an example of a DAE file having same model described in Appendix G (OBJ file with MTL)

According to COLLADA specification, the goal of COLLADA is to

- liberate digital assets from proprietary binary formats into a well-specified, XML-based, royalty free, open-standard format

- provide a standard common language format so that COLLADA assets can be used directly in existing content tool-chains, and to facilitate this integration

- be adopted by as many digital-content users as possible

- provide an easy integration mechanism that enables all the data to be available through COLLADA

- be a basis for common data exchange among 3D applications

- be a catalyst for digital-asset schema design among developers and DCC, hardware, and middleware vendors [112].

Moreover, there are several libraries available to read and write COLLADA files based on necessity through programmatic control, such as COLLADA DOM, FColloda, OpenCOLLADA, pycollada, GLGE, Three.js etc. An abridge comparison between OBJ and DAE is provided in a tabular presentation below (table 4.20).

Feature	OBJ	DAE
XML compatible	No	Yes
Needs external material file for multiple texture	Yes	No
Widely accepted for basic model exchange	Yes	Yes
Extensible	No	Yes
Open	Yes	Yes
Complexity level **A model created in sketchup Pro**	Simple OBJ exported file size 10 MB; loaded by Blender in 3 seconds	Complex DAE exported file size 38.6 MB; crashed Blender, was unable to load. Work in progress to enable rapid delivery and loading of 3D content [113]
Availability of specification	Yes	Yes
Native support for animation	No	Yes

Status	Wavefront does not exist anymore [107]	COLLADA Working Group is still active [114]

Table 4.20 Comparison between DAE and OBJ

Hence COLLADA having file extension .dae was chosen as the format of generalized content for 3D data.

After having the end user´s device identified or recognized next step is to support the user by providing requested content in such a manner that the used device is able to display the content. Exactly at this point transformation of content based on device capability comes under consideration.

4.4   Transformation of Content Including 3D Data

Content adaptation is used as a synonym of transformation of content under the scope of this thesis. In order to optimize the content presentation on different devices, the generalized content has to be adapted or translated in to a device dependent manner. To prove the concept of this thesis, Browser is used as a container for the content to be displayed at the end user´s device. Either web browser or mobile browser is responsible to be the container based on the type of the client device.

Initially according to W3C recommendation there were three defined ways for content adaptation [115].

- Server side
- Proxy side
- Client side

For the first decade of the mobile web there was a clear distinction between the mobile web and the desktop web and only one technique available to make content work well across multiple devices: server-side adaptation. This means logic in the server side that would detect the device, make changes to the delivered content to ensure that it worked well on the client side. In fact, server-side adaptation was usually necessary to make the content work at all; failing to do so would render the content unviable on many devices. However, nowadays the capabilities gap between mobile and desktop browsers has been filled by an array of smart phones of all shapes and sizes and tablets. Even the humble feature phone is inheriting rich browser features as WebKit based browsers, start to show up in lower-end phones. This has led to two outcomes:

- There is no longer a clear distinction between mobile and desktop devices

- Since many devices have rich, capable browsers with JavaScript support, there are new techniques available for adapting content for those new devices

The following table lists the main techniques in use today.

Name	Responsive design [116]	Mobile-first responsive design	Progressive Enhancement [117]	Server-side adaptation	Hybrid
Principle	Serve same HTML to every device - flexible grid - media queries - flexible images	Similar to responsive design but defaulting to serve mobile-friendly version	Serve base HTML to device, use JavaScript to build up functionality as far as possible	All logic on server, client receives only what is required.	Use device detection to serve initial device-sensitive HTML payload, use client-side progressive enhancement to build up
Content adaptation	No	No	Yes	Yes	Yes
Resolution independence	Yes	Yes	Yes	Yes	Yes
Context inspection	No	No	Yes	No	Yes

Advantages	It makes sites less dependent upon browser resolution & it doesn´t require device database to be maintained.	Removes image inefficiency issue.	Possibility of full adaptation. Due to high reliability used by most top internet brands such as Facebook, Google etc.	Full adaptation possible. Fine-grained controls	Full adaptation possible, but can also utilise client-side knowledge (e.g. config settings, real-time data) to add to experience
Disadvantages	Inefficient image handling.	Achieves only resolution independence.	There might be limitation since same base is used to build mobile and desktop functionality	Device detection library required (which is usually commercial)	Device detection library required. Complex. With reservations. All clients get initial page quickly but subsequent enhancement may impose delays
Summary	Achieves resolution independence but not content adaptation	Only really useful variant of responsive design for building a mobile website	Effectively moves adaptation logic to client. Often used in conjunction with responsive design	Standard technique for 10+ years Full content adaptation possible but with cost of having to use server-side library	It is very complex to build but the best solution. Used by browser.nokia.com

Table 4.21 The main techniques in use for content adaptation

It is intended to use server side, client side, proxy side or Hybrid approach based on necessity. For example for a a virtual reality scene, it is advantageous to use proxy based approach [25]. When a mobile device requests a specific 3D virtual reality file, the device has to inform the proxy about the device capabilities, such as processing power, screen resolution, supported sound formats, and so on. On the proxy side, the specific file is requested from a target server. By parsing this file, the proxy creates an object oriented representation of that scene. Additional resources (e.g. links

to other files) can be detected and preloaded. The proxy will remove unneeded and unsupported content according to the mobile device specification. If the identified device has lower processing power, then from a 3D virtual reality scene additional content such as fog, texture, and light source could be excluded. If the mobile device does not have the required capabilities to provide this information it does not make sense to transmit the data over the wireless network. This information can be removed at the proxy without any loss at the client side. The 3D virtual reality scene is transformed into the correct modeling language while mentioned methods are taken into account to reduce the content. Now, if the client device has the capability to translate the content from generalized format to it´s own compatible format in an efficient way, it is a better and safer choice regarding loss of information to use client side adaptation. If the device is partially capable of transforming the content then it makes more sense to use Hybrid approach. Hence it is decided to use any one or combination of client side, proxy side, server side and Hybrid approach based on device capability, connectivity and requested content.

Chapter 4 provided the theoretical overview of the architecture named Smoothie along with detailed explanation of its three steps. Chapter 5 will explain the practical implementation of the architecture.

Chapter 5   Experimental Validation

Experimental validation of Smoothie is devided into two steps namely implementation and evaluation. The lower part of this chapter will be evaluating Smoothie by integrating it with a collaborative language learning game (LLG) followed by the explanation of the implementation of three dominant part of smoothie exactly in same order as justified in chapter 4.

5.1   Implementation

Figure 5.1 below shows an overview of the implementation where model view controller architecture was used. The delegate layer is responsible to accept request from user devices and support them by providing response. From the request being processed to till the response being served is a complete task. This task can be divided in three major phases namely

1) Identification of device alias phase 1

2) Preparing generalized content alias phase 2

3) Transformation of content alias phase 3

For the explanation of proposed architecture named Smoothie´s implementation process same order will be followed as above provided list.

The system overview of Smoothie depicted in figure 3.3 is technically implemented as figure 5.1. Delegate layer is responsible to receive the request from heterogeneous client devices residing in View and pass it to the Application server. Delegate layer is also responsible to deliver the processed response back to the end client device. Web Components residing in Application server is holding all business logics therefore is responsible to analyse a request and then make all necessary decision. For example where to store or from where to retrieve the data; which API to call to process a certain request; Each time after receiving message from Delegate Layer, Web Components appoint a defined API to deal with a defined request, then wait for the job to be done. The appointed API replies and gives back the control to Web Components immediately after finishing the job. From left to right of figure 5.1, only the View is visible at client side; Delegate Layer, Application Server along with Storage are residing in Server side. The later part of this chapter explains

in detail the correlation between above mentioned three phases and the System implementation overview (figure 5.1).
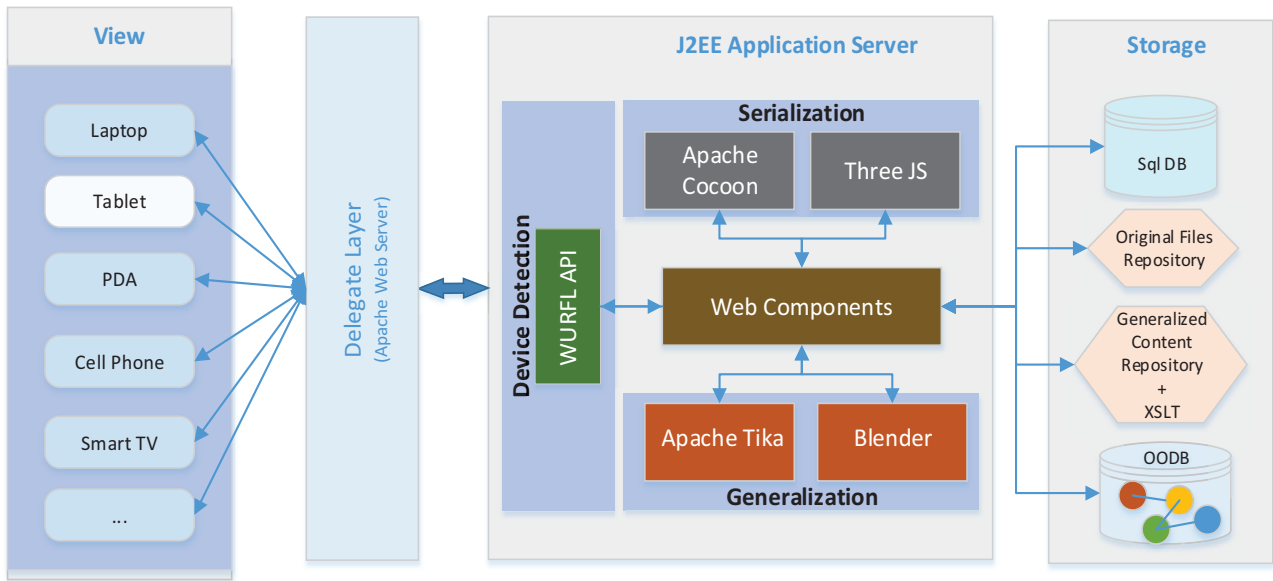
Figure 5.1 Simplified version of System implementation overview of Smoothie

5.1.1 *Evaluation and selection of open source software packages*

Below a short description of the software packages used as tools along with main purpose being served in the implementation of Smoothie is provided followed by table 5.1 introducing those tools along with their version number.

Name	Version	Used in		
		Phase 1	Phase 2	Phase 3
Apache Cocoon	2.2.0			▓
Apache Tika	1.4		▓	
Apache Tomcat	7.0	▓	▓	▓
Blender	2.67		▓	
DB4o	8.0.249.16098	▓	▓	
Eclipse	Kepler Service release 1	▓	▓	▓
Hibernate	4.2.8	▓	▓	▓
Java	1.7.0.21	▓	▓	▓
Microsoft Visio	2013	▓	▓	▓

MySql	5.6	For pictorial representation		
ThreeJS	R 66			
Visual Paradigm	3.1	For pictorial representation		
WURFL	1.5.0.1			

Table 5.1 List of tools used along with their version

1) Apache Cocoon

It is open source software, based on the Apache License 2.0. A well known web application framework built around the concept of pipeline, separation of concerns and component-based web development. The framework focuses on XML and XSLT publishing and is built using the Java programming language. The flexibility afforded by relying heavily on XML, allows rapid content publishing in a variety of formats including HTML, PDF andWML.

Main component used: Generator, transformers and serializers as pipeline, Sitemap as core of cocoon.

URL: https://cocoon.apache.org

2) Apache Tika

Tika is an open source toolkit, strives to offer the necessary functionality required for dealing with the heterogeneity of modern information content by detecting and extracting metadata and structured text content from various documents using existing parser libraries. Design goal of Tika was-Unified parsing interface, low memory footprint, fast processing, flexible metadata, easy parser integration, MIME database, provide flexible MIME detection and provide language detection [118].

Main component used: a mechanism to bridge the content type diversity that exists and to deal with file types in a uniform way.

URL : https://tika.apache.org

3) Apache Tomcat

it is an open source software implementation of the Java Servlet and JavaServer pages specifications, developed in an open and participatory environment and released under the Apache License version 2. Apache Tomcat provides a pure java HTTP web server environment for java code to run

in. It includes tools for configuration and management, but can also be configured by editing XML-configuration files.

Main component used: It is holding the whole application inside, as web server.

URL : http://tomcat.apache.org

4) Blender

For detailed investigation Autodesk Maya, Wings 3D and Blender was selected as modeler based on their licece, provided documentation and ease of use. Some relevant test results are provided below in tables from 5.2 to 5.10. From the result of table 5.7 it can be derived that displaying of a 3D data vary based on not only renderer but also modeller.

As it is found out from the test results

- Wings 3D has support for very few 3D data formats as import (See table 4.7). It is still under early stage of development.

- Maya follows the specification provided by collada while preparing generalized content for 3D data (DAE). It is confirmed by coherency test (validated against Expression Schema 1.5.0). Two collada files were used to make this test named dominos.dae and astroBoy_walk.dae. Those were created by following the schema 1.4.1 by using Maya as a modeler. Then those files were exported as .dae from Blender and Maya 2014. The coherency test results of those exported files are shown below in table 5.2. Due to closed source code of Maya, it was not suitable for Smoothie.

- It is also visible from the same table (table 5.2) that Blender exported DAE does not pass coherency test. Blender does not remove or change anything but it adds extra information inside the .dae file compare to declared specification. That is why failed status of coherency test does not have an impact on visibility of the same data.

3D extenstion	.dae prepared by	Choherency test	API	Displayed
domminos.dae (807 KB)	Maya	Passed	Three.Js	Yes
			Glge	Yes (better shading)
	Blender	Failed (Passed after modification)	Three.Js	Yes (Small)
			Glge	Yes (Small) (better shading)
Expression Schema 1.4.1	Exported from Maya 8.0 (Relased 2006)	Passed	Three.Js	Yes
			Glge	Yes (Better illumination)

astroBoy_walk.dae (765 KB)	Maya	Failed	Three.Js	No
			Glge	Yes
	Blender	Failed	Three.Js	No
			Glge	Yes (without color and animation)
Expression Schema 1.4.1	Exported from Maya 2008	Passed	Three.Js	No
			Glge	Yes(without color)

Table 5.2 Coherency test result of Maya and Blender

Since Blender is a free and open source 3D animation suite under GNU GPL licence. It supports the entirety of the 3D pipeline—modeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing. Alongside the modeling features it also has an integrated game engine. Blender is cross-platform and runs equally well on Linux, Windows and Macintosh computers. Its interface uses OpenGL to provide a consistent experience. Hence Blender is decided as the most suitable tool for Smoothie.

Main component used: Import and export functions of python based Blender

URL : http://www.blender.org

5) DB4o

It is an embeddable open source object database, for Java and .NET developers to store and retrieve any application object with only one line of code, provides respective APIs under multiple license. The DB4o engine is free when licensed under the General Public License (GPL) or distributed with non-GPL open source projects under the DB4o Opensource Compatibility License (dOCL). The DB4o Community offers free developer resources incl. forums, documentation, and code samples. A commercial license is required if it is embedded in a non-GPL, commercial product or if commercial support is desired. One of its main goals is to provide an easy and native interface to persisence for object oriented programming languages. Development with DB4o database does not require a separate data model creation, the application's class model defines the structure of the data in DB4o database. DB4o attempts to avoid the object/relational impedance mismatch by eliminating the relational layer from a software project.

Main component used : ObjectManager Enterprise (OME), a database visualizer and IDE plugin for Eclipse to store collected end user´s device capabilities.

URL : http://www.db4o.com

6) Eclipse

It is an integrated development environment (IDE). It contains a base-workspace and an extensible plug-in system for customizing the environment, written in Java, cross-platform and distributed under Eclipse Public License.

Main component used : Eclipse Java EE IDE, a tool for creating Java EE and Web applications.

URL : http://www.eclipse.org

7) Hibernate

Hibernate is an object-relational mapping library for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. It is free software, distributed under the GNU Lesser General Public License. Features of hibernate are object-relational mapping, Java Persistence API (JPA) provider, idiomatic persistence, high performance in terms of developer productivity and runtime performance, scalability, reliability and extensibility.

Main component used : Hibernate ORM for the reliable communication between Java and MySql databse .

URL : http://hibernate.org

8) Java

It is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), conveys code that runs on one platform does not need to be recompiled to run on another one. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine(JVM) regardless of computer architecture.

Main component used: JDK 7, Servlet API 2.4 as a part of J2EE

URL : http://www.oracle.com/technetwork/java

9) Microsoft Visio

it is a commercial software for Creating professional diagrams to simplify complex information with updated shapes, collaboration tools and data-linked diagrams.

Main component used: Flowchart from different categories

URL : http://office.microsoft.com/en-001/visio

10) MySql

It is an open-source relational database management system (RDBMS), distributed under GPL 2 or Commercial license. MySql is a popular choice of database for use in web applications such as Wikipedia [119], Google (though not for searches) [120], Facebook [121], Twitter [122], Flickr [123] and YouTube [124].

Main Component used: MySQl Enterprise Edition to save metadata of the content

URL: http://www.mysql.com/

11) ThreeJS

Based on suitability with WebGL and OpenGL ES, provided documentation, policy of lisence and ease of use three different tools were selected as renderer namely C3DL [125], GLGE [126] and ThreeJS [127]. Several tests were also conducted to find the most suitable one in terms of visibility of 3D data. Refer to the tables from 5.3 to 5.10 for sample of test results.

Nomenclature

- 3D extension – provides the file name of the 3D model with it´s extension.
- .dae prepared by – shows which application was used to export the model as .dae.
- Coherency test - validation against Expression Schema 1.5.0 provided by COLLADA
- API – which API was used to render prepared .dae
- Displayed – If the API was able to render the .dae file on the screen

3D exten-sion	.dae prepared by	Coherency test	API	Displayed
Collada .dae	Wings 3D Unable to import .dae	_	C3dl	-
			GLGE	-
			Three.js	-
	Maya	passed	C3dl	Yes (no color)

			GLGE	Yes (Grey head)
			Three.js	Yes (grey head)
	Blender	Failed	C3dl	No
			GLGE	Yes (with coler & texture but very small)
			Three.js	Yes (grey head)
Expression Schema-1.4	Exported from- Maya 8.5	Passed	C3dl	Yes (no color)
			GLGE	Yes (Grey head)
			Three.js	Yes (grey head)

Table 5.3 Result set-1 to select suitable Modeller and Renderer for 3D data

3D extension	.dae prepared by	Coherency test	API	Displayed
knot.3ds	Wings 3D	Failed (See log file)	C3dl	Yes(no color)
			GLGE	No
			Three.js	Yes (White knot)
	Maya	Unable to import .3ds	C3dl	
			GLGE	
			Three.js	
	Blender	Failed	C3dl	Yes (very small pic)
			GLGE	Yes (White very small knot)
			Three.js	Yes (very small pic)

Table 5.4 Result set-2 to select suitable Modeller and Renderer for 3D data

3D extension	.dae prepared by	Coherency test	API	Displayed
Cow.obj	Wings 3D	Failed	C3dl	Yes(no color)
			GLGE	No
			Three.js	Yes (no color)
	Maya	Passed	C3dl	Yes (no color)
			GLGE	Yes (with coler & texture)
			Three.js	Yes (with coler & texture)
	Blender	Failed	C3dl	Yes (very small pic)

			GLGE	Yes (with color & texture)
			Three.js	Yes (with color & texture)

Table 5.5 Result set-3 to select suitable Modeller and Renderer for 3D data

3D extension	.dae pre-pared by	Coherency test	API	Displayed
Big_dodge.ply	Wings 3D Not support-ed		C3dl	
			GLGE	
			Three.js	
	Maya Not support-ed		C3dl	
			GLGE	
			Three.js	
	Blender	Failed	C3dl	Yes (very small pic)
			GLGE	Yes (with texture)
			Three.js	Yes (with texture)

Table 5.6 Result set-4 to select suitable Modeller and Renderer for 3D data

3D extension	.dae prepared by	Coherency test	API	Displayed
alien.x3d	Wings 3D Not support-ed		C3dl	
			GLGE	
			Three.js	
	Maya Not support-ed		C3dl	
			GLGE	
			Three.js	
	Blender	Failed (see log file)	C3dl	Yes (very big pic)
			GLGE	Yes (very big pic, needed scaling)
			Three.js	Yes (very big pic, needed scaling)

Table 5.7 Result set-5 to select suitable Modeller and Renderer for 3D data

3D extension	.dae prepared by	Coherency test	API	Displayed
P1_wedge.fbx (too big for all API (9.1 MB))	Wings 3D Not supported		C3dl	
			GLGE	
			Three.js	
	Maya	Passed	C3dl	Yes (too big grayscale pic)
			GLGE	Yes (too big pic with texture and changed color)
			Three.js	Yes (with texture and major changed color)
	Blender Not supported (imported p1_wedge.dae by Maya)	Failed	C3dl	Yes (too small no color)
			GLGE	Yes (too small major color change)
			Three.js	Yes (too small greyscale)

Table 5.8 Result set-6 to select suitable Modeller and Renderer for 3D data

3D extension	.dae prepared by	Coherency test	API	Displayed
Porsche .stl 7.4 kb	Wings 3D	Failed	C3dl	Yes (no color)
			GLGE	No
			Three.js	Yes (no color)
	Maya No plugin for .stl Used exported .dae from blender	Passed	C3dl	Yes (too big, no color)
			GLGE	Yes (too big, no color)
			Three.js	Yes (too big, no color)
	Blender	Failed	C3dl	Yes (no color)
			GLGE	Yes (very big pic, had to scale down)
			Three.js	Yes (very big)

Table 5.9 Result set-7 to select suitable Modeller and Renderer for 3D data

3D extension	.dae prepared by	Coherency test	API	Displayed
Microscope.wrl 122 KB	Wings 3D Not supported		C3dl	
			GLGE	
			Three.js	
	Maya Not supported Used .dae export-ed from blender	Passed	C3dl	Nothing is visible.
			GLGE	Nothing is visible
			Three.js	Nothing is visible
	Blender	Failed	C3dl	Nothing is visible
			GLGE	Nothing is visible
			Three.js	Nothing is visible

Table 5.10 Result set-8 to select suitable Modeller and Renderer for 3D data

From table 5.9 Blender exported .dae file was imported in Maya and then was exported as .dae again. Maya exported .dae passed coherency test where as Blender exported .dae failed. Hence it is proved that Blender does not remove or modify any information while preparing generalized content format (.dae) for 3D data, it only adds some more tags inside the xml file. Those added tags are removed by Maya while preparing generalized content for 3D that is why exactly the same file passed coherency test. Table 5.7 proves the same by showing exactly opposite result of table 5.9 regarding coherency tests.

To achieve the result set shown above in tabular form (table 5.3 to 5.10), test was conducted in April 2013 with 57 test cases. A tabular presentation of the derived summary from those tables is provided below (table 5.11).

Feature	C3DL	GLGE	ThreeJS
Visibility	92.98 % of 57 test cases were visible	91.23 % of 57 test cases were visible	92.98 % of 57 test cases were visible
Texture & color	0 % of 57 test cases were visible	Better than ThreeJS See figure 5.2	Worse than GLGE See figure 5.3

Last release (until April 2013)	C3DL 2,2 June 2011	GLGE v0.9 February 2012	Three.js R 58
Status	Dead [125]	Dead [128]	Active [129] Last stable release July 2014 Three.js R 68

Table 5.11 Comparison among C3DL, GLGE and ThreeJS

Hence ThreeJS is proven to be a better choice as renderer. Moreover, with the later version of threeJS, problem in terms of texture and color was solved.
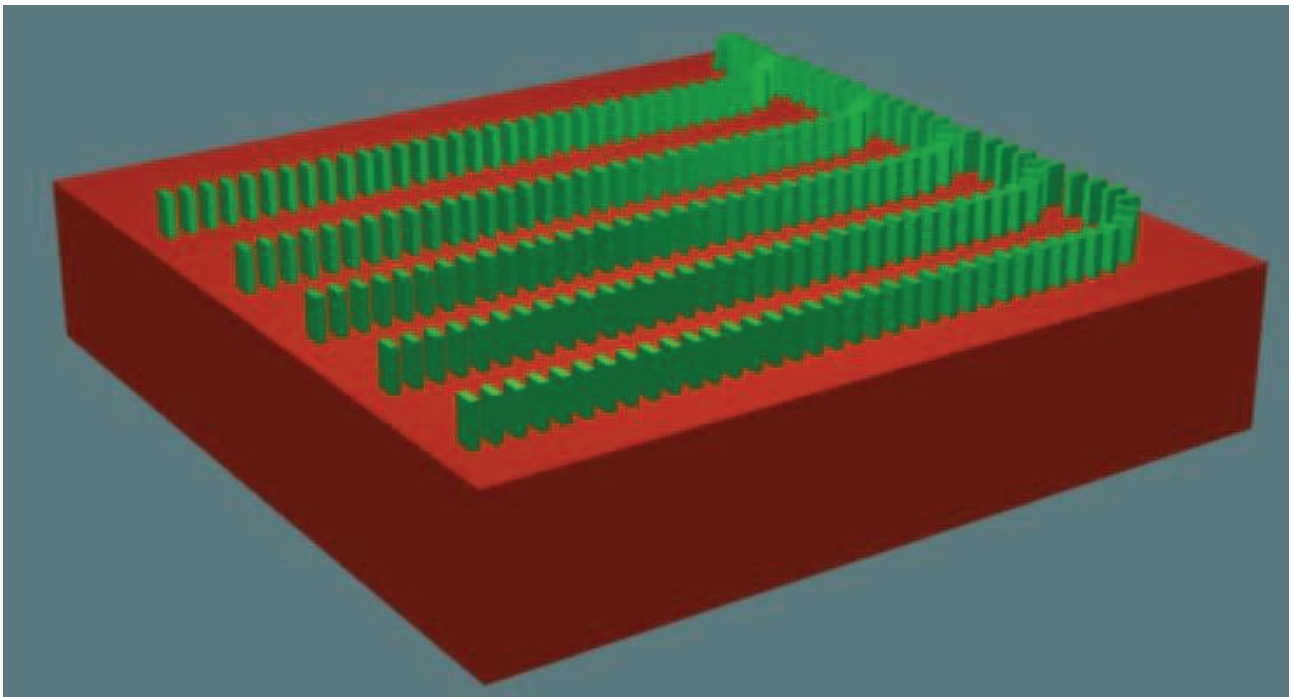
Figure 5.2 DAE file rendered by GLGE v 0.9
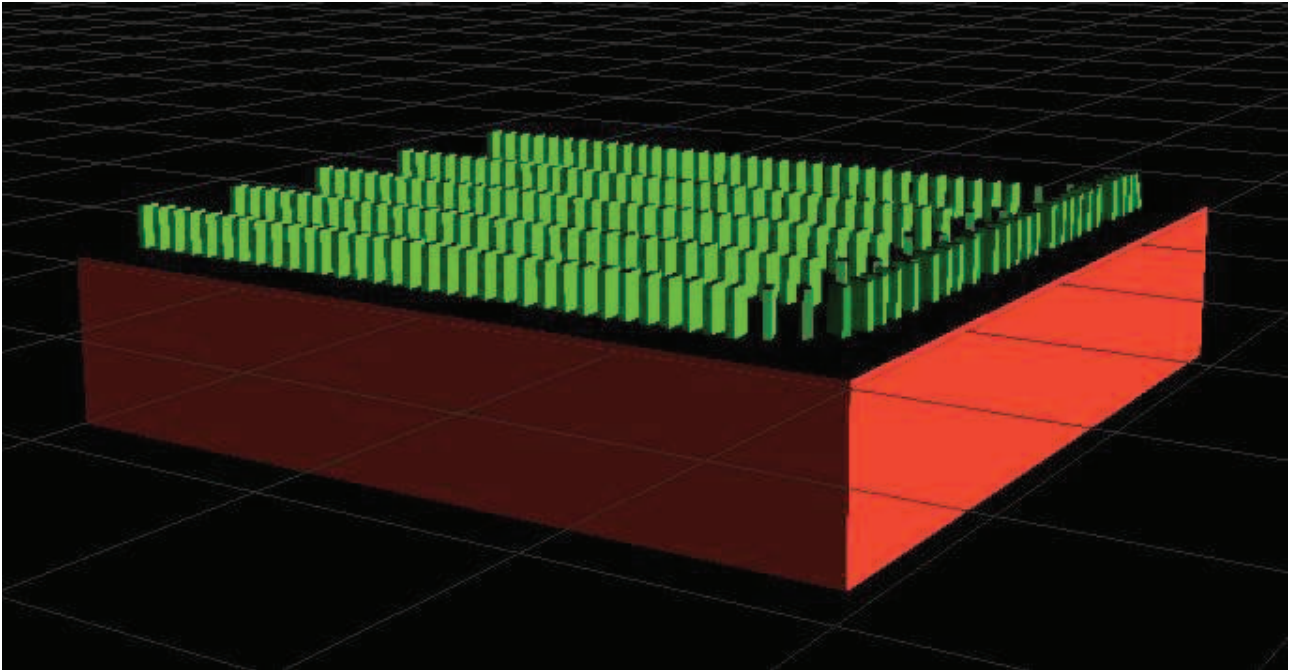
Figure 5.3 DAE file rendered by ThreeJS R58

Main component used: display generalized 3D computer graphics on a Webgl supported Web browser of end client´s device

URL : https://github.com/mrdoob/three.js

12) Visual Paradigm

It is a commercial software that facilitates building software and systems that excel in user experience by supporting effective use case identification, requirements gathering, flow of events, wire framing, requirement specification generation, etc.

Main component used: UML diagrams

URL: http://www.visual-paradigm.com

13) WURFL

Wireless **U**niversal **R**esource **Fi**Le, is a Device Description Repository (DDR), in other words a software component that maps HTTP Request headers to the profile of the HTTP client such as Desktop, Mobile Device, Tablet, etc. that issued the request. WURFL is a set of proprietary application programming interfaces (APIs) and an XML configuration file which contains information about device capabilities and features for a variety of mobile devices [130]. Until version 2.2, WURFL was released under an open source license [131]. Prior to version 2.2, device information was contributed by developers around the world and the WURFL was updated frequently, reflecting new wireless devices coming on the market. In June 2011, the founder of the WURFL project, Luca

Passani, and Steve Kamerman, the author of Tera-WURFL, a popular PHP WURFL API, formed ScientiaMobile, Inc to provide commercial mobile device detection support and services using WURFL [132]. As of August 30, 2011, the ScientiaMobile WURFL APIs are licensed under a dual-license model, using the AGPL license for non-commercial use and a proprietary commercial license.

Main component used: WURFL Java API with WURFL XML configuration file

URL: http://wurfl.sourceforge.net/apis.php

So far all the tools and technologies being used to implement the proposed architecture are introduced above. The rest of this chapter will show by using those tools and technologies how the implementation phase works.

5.1.2   *Implementation of device identification (Phase 1)*

The very first step of the proposed architecture is to identify the end user´s device along with it´s capabilities. Simultaneously with device detection it is important to make sure that the issued request is coming from a valid user. Implementation of device identification is divided into two steps

- User authentication
- Device identification

5.1.2.1   User Authentication

Figure 5.2 below depicts the user authentication process. Right now there are two different roles of user implemented in the system namely

1) Author, who is providing content and
2) Student, or content user

While authentication process takes place it is also decided what kind of user role this user has and based on the role different User Interface (UI) and access rights are provided. For example if a user has a role of a student or content user he does not have the right to change uploaded learning content by the author or content provider.

The action named "check authenticity of the user" in figure 5.4 may include different actions or a set of actions depending upon the policy of the institute using the proposed architecture. As an example figure 5.5 below depicts how the log-in process is personalized for Offenburg University of Applied Sciences Germany to provide access right for Language Learning Game (see sub-chapter 5.2 for detail information of Language Learning Game). Offenburg University of Applied Sciences

Germany is using Lightweight Directory Access Protocol (LDAP) server to save all information about each user such as user name, e-mail address, department, birth date, role in the institute, dependency, specific rights in other used system etc. Hence the authenticity process has to communicate with LDAP server for access right, role and above all validity of the user.
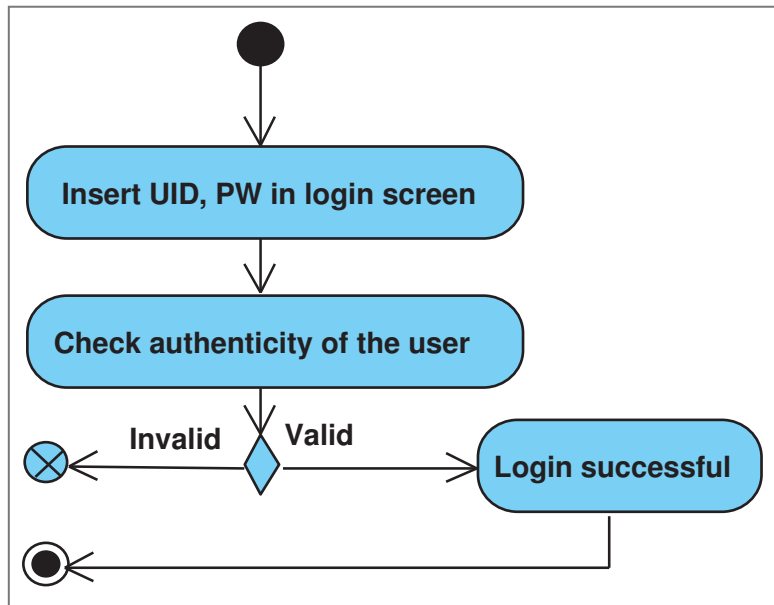
Figure 5.4 User Log-in process shown in an activity diagram
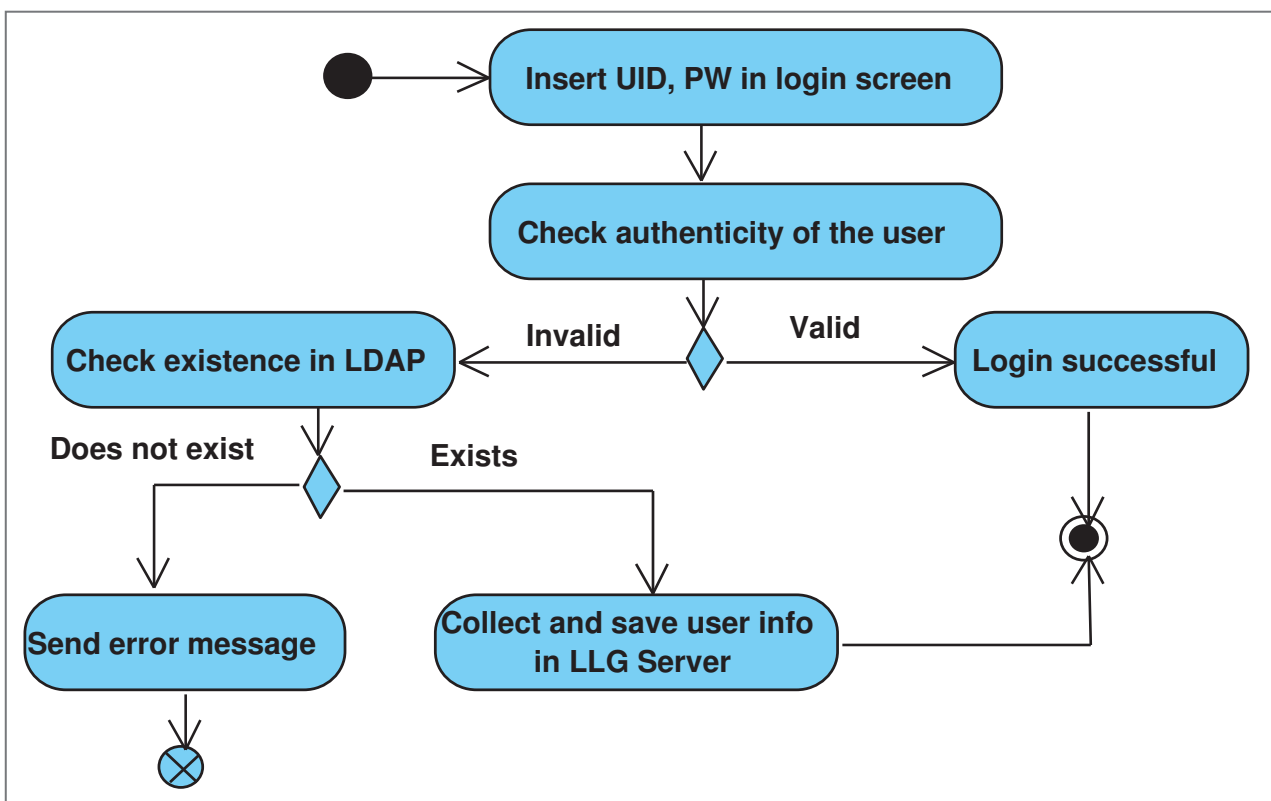
Figure 5.5 Simplified version of user Log-in process of LLG

5.1.2.2 Device identication

Figure 5.4 below depicts the process of device identification. As soon as the user requests for the log-in page which is an Http Request, Delegate layer (see figure 5.1) accepts the request and involve Web Components. Web Components appoint HttpServlet to communicate WURFL database (if necessary) to collect device capability information. Then collected capability information is being saved in DB4o database as object.

The three most important classes and interfaces to perform the job of collecting device capabilities, storing and retrieving the capability information as objects in object oriented database are called, DeviceRequestProcessor{}, WurflWebConstants{}, Db4oServletContextListener{}, RetrieveCapabilities{} and StoreCapabilities{} . To have a closer look at source codes, please refer to Appendix D.
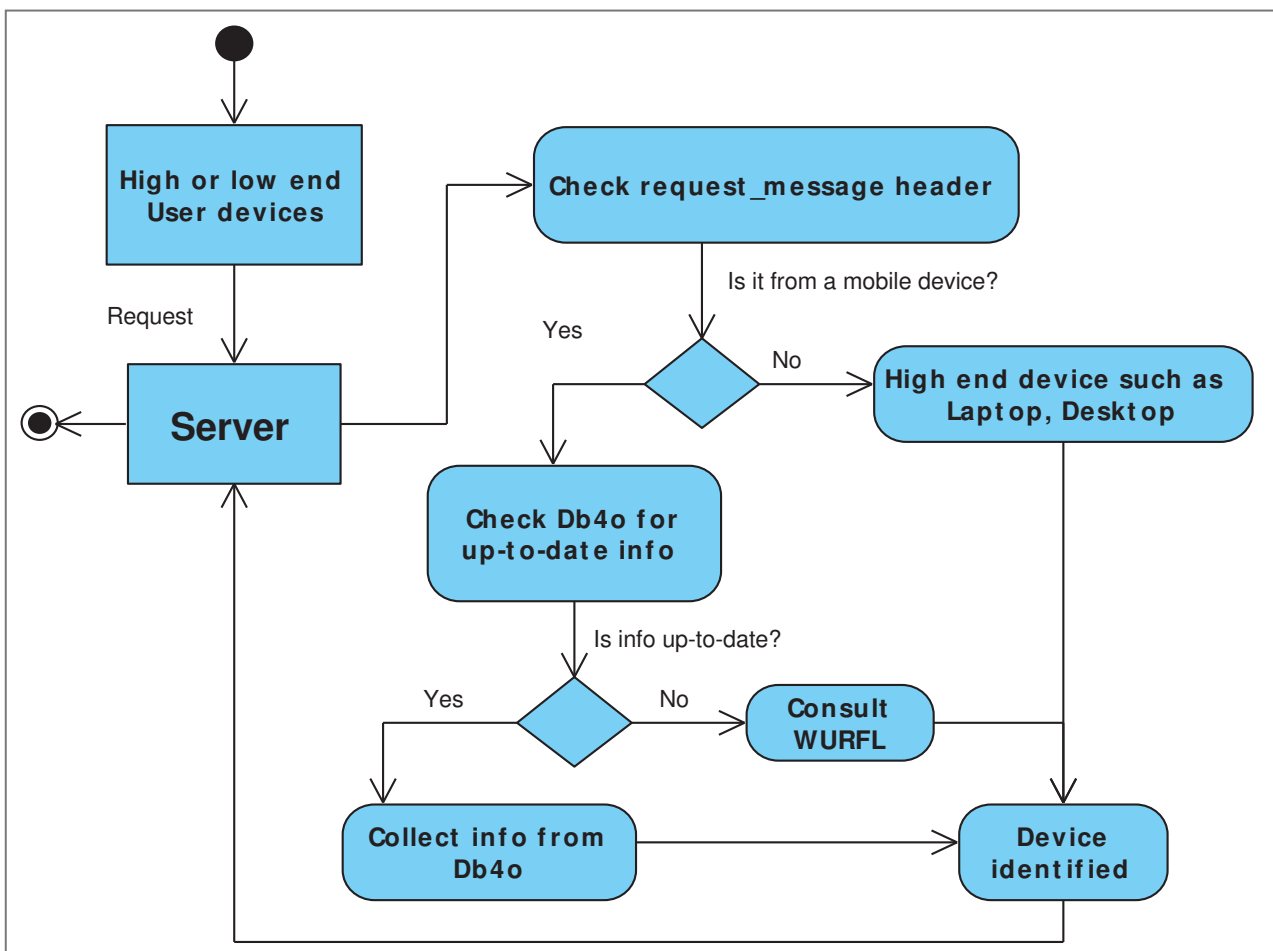
Figure 5.6 Device identification process shown in a flowchart

Device identification and user authentication takes place simultaneously and according to code implementation they are extensible but not separable. In the above considered example when the

user requests for the log-in page of Language Learning Game from the browser of the device by using pre defined Uniform Resource Locator (URL), immediately the execution of Device Identification process starts to take place; then the device dependent view of log-in page is sent. After submission of user id and password User Authentication process will take place. For data security, monitoring, to maintain institute defined policy, to identify role and corresponding rights over the data and the system, User Authentication is an obligatory part of Smoothie. Based on the needed level of assurance and accountability for different scenario, User Authentication process can be enhanced without having any effect on Device identification process. The complete process can be depicted as figure 5.7.
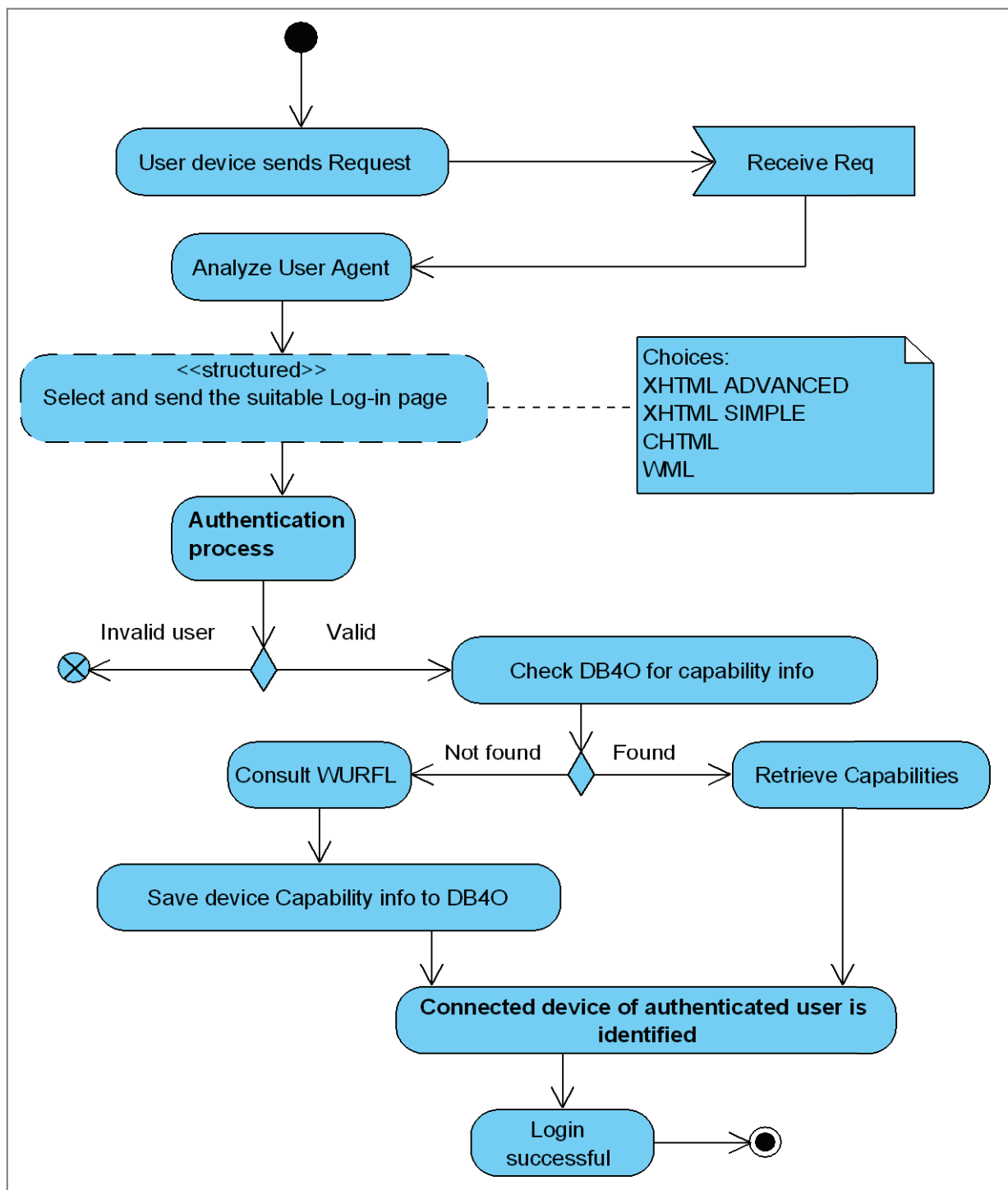
Figure 5.7 Device identification along with user authentication shown in an activity diagram

5.1.3 *Implementation of preparing generalized content (Phase 2)*

If the user has a role of content provider or author, after a successful login user will be provided a UI with the possibility of uploading content. As soon as the author uploads content, independent of its format or purpose, will be saved in the file system named Original Files Repository. The reason behind saving all content in its original form is safety, if anything goes wrong for example data gets corrupted or lost during any phase of the transformation process, the system can anytime start it over from the very beginning on with the originally uploaded data. As a next step Web Components appoints either Appache Tika or Blender to transform content in generalized format based on the MIME (Multipurpose Internet Mail Extensions) Type of the uploaded content. As output, XML and DAE are provided by Tika and Blender respectively. Immediately after receiving output from Tika, the core component of Apache Cocoon named Sitemap has to be updated for serialization later on. Then generalized format of content is saved in Generalized Content Repository and the metadata of the content is saved in SqlDB. The most important classes to carry out these responsibilities are ManageFiles{}, ContentGeneralization{}, Content{}, ContentOperation{} and CocoonSiteMapService{}. To have a closer look at source codes, please refer to Appendix E. The process as a whole could be depicted in a flowchart (figure 5.8)
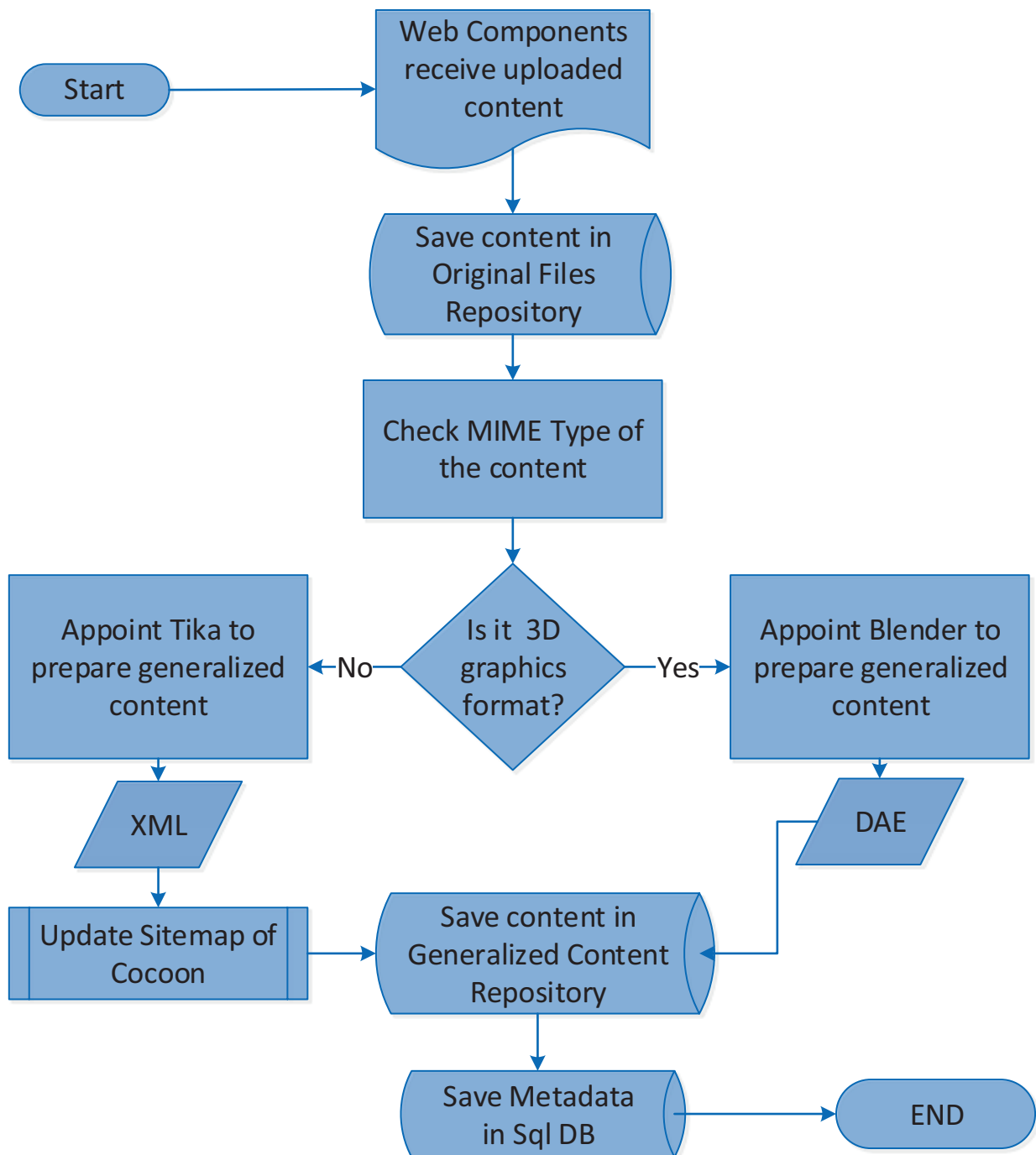
Figure 5.8 Implementation of Phase 2, shown in a flowchart

5.1.4 *Implementation of transformation of content (Phase 3)*

This step will take place if a user is requesting for an uploaded content. As soon as the request is received by the system, end user´s device identification and authentication processes will take place. If the mentioned processes are successful, metadata of the requested content will be retrieved from the database. Based on the ID which is a part of the link of the metadata different framework or library will be called for data transformation and visualization. For example for any kind of generalized document file apache cocoon will be appointed for serialization, for 3D generalized data ThreeJS will be appointed for rendering, any other data formats will be retrieved from the repository and at the end the content will be visualized on the browser of the end user´s device. A pictorial representation of the whole process is presented below in a flowchart (figure 5.9)

Since the capabilities gap between mobile and desktop browsers is becoming blur and blur and has been filled by an array of smart phones of all shapes and sizes and tablets. Even the humble feature phone is inheriting rich browser features as WebKit based browsers, start to show up in lower-end phones. Moreover in terms of processing power as well, today a smart phone is no where inferior compare to a desktop. This has led to two decisions:

- There is no longer a clear distinction between mobile and desktop devices
- Since many devices have rich, capable browsers with good JavaScript support, there is no obvious reason to use proxy support. It is also not necessary to have a complete dependency on server.

Prototype of Smoothie implements server side, and client side content adaptation techniques; and easily extensible to use Hybrid content adaptation technique based on the capabilities of the end device along with it´s connectivity and the type and volume of the requested data. For example,

- a document file type transformation is taking place in server side,
- a 3D data adaptation is taking place in client side and
- for a file having bigger volume such as video file, Hybrid technique of content adaptation could be used.

The most important class involved in this process is Content{}, ContentOperation{}, and PopulateTable{}. To have a closer look at source codes, please refer to Appendix F.
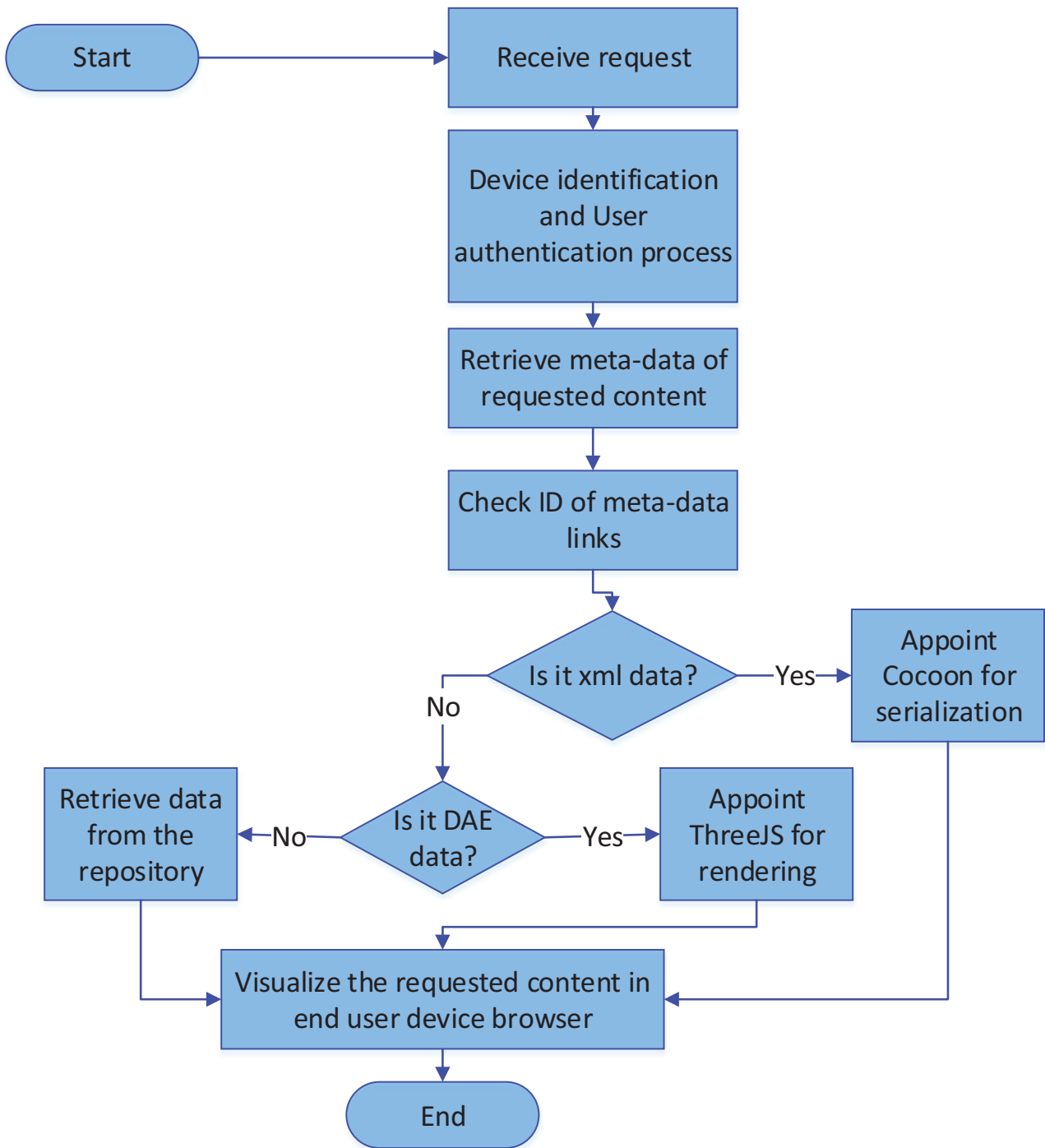
Figure 5.9 Implementation of phase 3 shown in a flowchart

5.2   Evaluation

As an example of Smoothie´s academic use it is integreted with Language Learning Game. As a method of evaluation of Smoothie, LLG was carefully chosen because of below stated reasons

- As a learning content language teachers may use any kind of data including 3D
- All the students having different kinds of devices which is not possible to predict

5.2.1   *Integration to Language Learning Game (LLG)*

Language Learning Game is an application developed in Offenburg University of applied sciences Germany as a master thesis by Razia Sultana under supervision of Professor Dr. Andreas Christ and Dr. Markus Feisst in 2010. Enhancement has been made to make it cooperate with Smoothie [133]. Figures 5.10 and 5.11 below show a simplified version of this integration i.e. the game process along with device detection (Phase 1), content generalization (Phase 2) and content transformation (Phase 3), in flowchart.

The focus of the game is on adult learner, to support them by providing a tool which is anytime and anywhere accessible, for foreign language acquisition. LLG is offered by Language department of Offenburg University of Applied Sciences Germany for international master program students.

5.2.1.1   Pedagogical concept

To learn foreign language interactivity with others in a way that facilitates practice of the desired language is very important because learners acquire fluency by using whatever skills they have. They should be presented with activity-based approaches that engage them in interactive experiences and support the development of intrapersonal and interpersonal skills. As the learners develop their ability to understand, appreciate, and relate positively to others using the target language, they learn to demonstrate constructive attitudes and values through participation in challenging real-life situations. So there learning should not be hindered by limiting them with time, place, device or format of learning content.

5.2.1.2   Brief description of the game

The LLG allows language learners to practice the target language using a mobile device and requires a supervisor who has very good knowledge of the foreign language to perform an overall check at the very end. New language learning needs a lot of interaction with other people. LLG can meet this entire requirement by implementing the proposed system on a mobile phone without asking for a physical presence on a certain place and without keeping the participant waiting for

an action actively. The instructor or supervisor will create the game and upload instructions as learning content having any data format; decide about the total number of group members in each group, and the game end condition. Participants will write sentences and upload supporting content and the aim is to build a short story in the desired language. When a group member writes and sends a sentence the other members of the group will have a chance to dispute by proposing another version of the sentence or they can simply agree with it. They are only allowed by the rule of the game, to change spelling or grammatical mistakes in their proposed version. Afterwards all group members will receive the proposed sentence and possible corrected versions. At this stage, every group member has to vote for the version the individual prefers. The sentence with the most votes gets selected. In case of equal votes the first submitted one wins. Each group member has to create one sentence per round. Typically the game is finished after 15-20 rounds. The final short story is sent to all group members as well as to a supervisor who will perform an overall correction. After that, all the participants will receive the supervisor's corrections of all the mistakes that occurred during the game. This overall correction is necessary, because there might be a situation where the majority of group members agree with an incorrect sentence. At the end, the corrections made by the supervisor will help them to learn and improve their level in desired language. [134]

5.2.1.3   Advantege of integrating Smoothie with LLG

The implementation of Smoothie made LLG a real device independent application and added freedom as a very necessary feature. It has provided a freedom to the content provider to be able to use any kind of data format without having any thought in mind regarding the end client device. At the same time it has also given a freedom to the end client to use any kind of device whenever and wherever they want based on the availability and their personal preference.
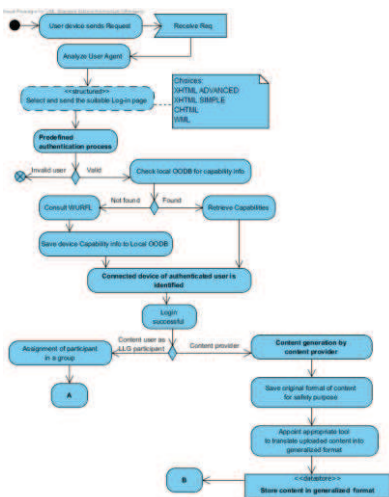
Figure 5.10 : Implementation of Phase 1 & 2 along with predefined user authentication process
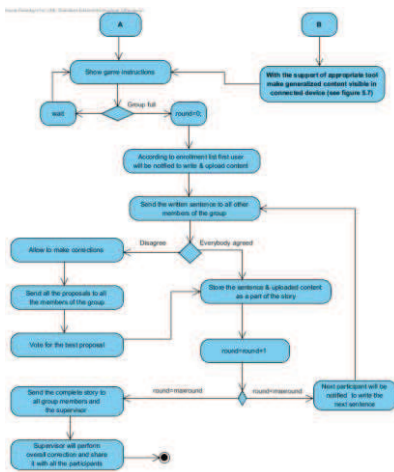
Figure 5.11 Implementation of Game logic with Phase-3 shown in a flowchart

5.2.2 *Test methods and results*

Smoothie was tested with several data types/ Internet media types having registered in Apache.org mime type list [135]. All those are registered as described in RFC 4288. This is the document that defines procedures for the specification and registration of media types for use in MIME and other Internet protocols. The registry could be found in http://www.iana.org/assignments/media-types/. The initial check was being done by using below stated mime types in Windows 8.1 desktop, and Android 4.4 tablet having Mozilla Firefox and Google Chrome installed on both the devices.

3D MIME Types

- application/x-blender
- model/x3d+xml
- application/octet-stream
- application/x-3ds
- application/blender
- application/sla
- application/vnd.sketchup.skp
- model/u3d
- model/vnd.collada+xml

Document MIME types

- application/pdf
- application/vnd.ms-excel
- application/vnd.ms-powerpoint
- application/vnd.openxmlformats-officedocument.wordprocessingml.document
- application/vnd.oasis.opendocument.text

Table 5.2 below shows performance of Smoothie in terms of visibility of different types of data in heterogeneous devices having different capabilities regarding hardware and software.

Content 1: Subject-Physics, Title-Light

File Name & Size	Mime Type	Generalized File Size	Operating System	Device Name	Browser	Data Visibility
IntrusionDetectionTechniques.pdf 503 KB	application/pdf	78 KB	Windows 8.1	Desktop (Dell Optiplex 755)	Google Chrome 35.0	Yes
					IE 11	Yes

ClassDiagram.png 174 KB	image/png	174 KB	Windows 8.1	Desktop (Dell Optiplex 755)	Google Chrome 35.0	Yes
					IE 11	Yes
Lantern1.obj 1598 KB	application/ x-tgif	5492 KB	Windows 8.1	Desktop (Dell Optiplex 755)	Google Chrome 35.0	No
					IE 11	Yes
IntrusionDetectionTechniques.pdf 503 KB	application-tion/pdf	78 KB	Windows 8	Laptop (Dell Latitude E6330)	Google Chrome 35.0	Yes
					IE 10	Yes
ClassDiagram.png 174 KB	image/png	174 KB	Windows 8	Laptop (Dell Latitude E6330)	Google Chrome 35.0	Yes
					IE 10	Yes
Lantern1.obj 1598 KB	application/ x-tgif	5492 KB	Windows 8	Laptop (Dell Latitude E6330)	Google Chrome 35.0	Yes
					IE 10	No
IntrusionDetectionTechniques.pdf 503 KB	application-tion/pdf	78 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 35.0 / Mozilla Firefox 30.0	Yes
ClassDiagram.png 174 KB	image/png	174 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 35.0 / Mozilla Firefox 30.0	Yes

Lantern1.obj 1598 KB	application/ x-tgif	5492 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 35.0	No
					Mozilla Firefox 30.0	Yes
IntrusionDetectionTechniques.pdf 503 KB	application/pdf	78 KB	Android 4.4	Tablet (Nexus 7)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	Yes
ClassDiagram.png 174 KB	image/png	174 KB	Android 4.4	Tablet (Nexus 7)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	Yes
Lantern1.obj 1598 KB	application/ x-tgif	5492 KB	Android 4.4	Tablet (Nexus 7)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	Yes
IntrusionDetectionTechniques.pdf 503 KB	application/pdf		iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	Yes
					Google Chrome 33.0	Yes
ClassDiagram.png 174 KB	image/png		iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	Yes
					Google Chrome 33.0	Yes

File Name & Size	Mime Type	Generalized File Size	Operating System	Device Name	Browser	Data Visibility
Lantern1.obj 1598 KB	application/ x-tgif		iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	No
					Google Chrome 33.0	No

Table 5.12 Sample of test results set 1

Content 2: Subject-IT, Title-Software

File Name & Size	Mime Type	Generalized File Size	Operating System	Device Name	Browser	Data Visibility
3DContent Adapta-tion.doc 750 KB	application/ msword	7 KB	Windows 8.1	Desktop (Dell Optiplex 755)	Google Chrome 35.0	Yes (im-age was not visi-ble)
					IE 11	
TestAudio1. wma 58 KB	audio/x-ms-wma	58 KB	Windows 8.1	Desktop (Dell Optiplex 755)	Google Chrome 35.0	Yes (im-age was not visi-ble)es
					IE 11	
R8.3ds 643 KB	image/x-3ds	3371 KB	Windows 8.1	Desktop (Dell Optiplex 755)	Google Chrome 35.0	No
					IE 11	Yes
3DContent Adapta-tion.doc 750 KB	application/ msword	7 KB	Windows 8	Laptop (Dell Latitude E6330)	Mozilla Firefox 30.0	Yes (im-age was not visi-ble)
TestAudio1. wma 58 KB	audio/x-ms-wma	58 KB	Windows 8	Laptop (Dell Latitude E6330)	Mozilla Firefox 30.0	Yes
R8.3ds 643 KB	image/x-3ds	3371 KB	Windows 8	Laptop (Dell Latitude E6330)	Mozilla Firefox 30.0	Yes

					Google Chrome 35.0	
3DContent Adapta-tion.doc 750 KB	application/ msword	7 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 34.0	Yes
					Mozilla Firefox 30.0	
TestAudio1. wma 58 KB	audio/x-ms-wma	58 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 34.0	Yes
					Mozilla Firefox 30.0	
R8.3ds 643 KB	image/x-3ds	3371 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 34.0	No
					Mozilla Firefox 30.0	Yes
3DContent Adapta-tion.doc 750 KB	application/ msword	7 KB	Android 4.4	Tablet (Nexus 7)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	Yes
TestAudio1. wma 58 KB	audio/x-ms-wma	58 KB	Android 4.4	Tablet (Nexus 7)	Google Chrome 35.0	No
					Mozilla Firefox 30.0	No
R8.3ds	image/x-3ds	3371 KB	Android 4.4	Tablet	Google Chrome	Yes

643 KB				(Nexus 7)	35.0	
					Mozilla Firefox 30.0	Yes
3DContent Adapta-tion.doc 750 KB	application/ msword	7 KB	iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	Yes
					Google Chrome 33.0	Yes
TestAudio1. wma 58 KB	audio/x-ms-wma	58 KB	iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	No
					Google Chrome 33.0	No
R8.3ds 643 KB	image/x-3ds	3371 KB	iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	No
					Google Chrome 33.0	No

Table 5.13 Sample of test results set 2

Content 3: Subject-Electrical, Title-Current

File Name & Size	Mime Type	Generalized File Size	Operating System	Device Name	Browser	Data Visibility
TestVideo1. mp4 8170 KB	video/mp4	8170 KB	Windows 8	Laptop (Dell Latitude E6330)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	
Gold-berg.stl 1706 KB	application/v nd.ms-pki.stl	2611 KB	Windows 8	Laptop (Dell Latitude E6330)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	

AllenDutto n.x3d 297 KB	model/vnd.c ollada+xml	728 KB	Windows 8	Laptop (Dell Latitude E6330)	Mozilla Firefox 30.0	Yes
					Google Chrome 35.0	
TestVideo1. mp4 8170 KB	video/mp4	8170 KB	Windows 8.1	Desktop (Dell Optiplex 755)	IE 11	Yes
					Google Chrome 35.0	
Gold- berg.stl 1706 KB	application/v nd.ms-pki.stl	2611 KB	Windows 8.1	Desktop (Dell Optiplex 755)	IE 11	Yes
					Google Chrome 35.0	
AllenDutto n.x3d 297 KB	model/vnd.c ollada+xml	728 KB	Windows 8.1	Desktop (Dell Optiplex 755)	IE 11	Yes
					Google Chrome 35.0	No
TestVideo1. mp4 8170 KB	video/mp4	8170 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 34.0	No
					Mozilla Firefox 30.0	Yes
Gold- berg.stl 1706 KB	application/v nd.ms-pki.stl	2611 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 34.0	Yes
					Mozilla Firefox 30.0	Yes
AllenDutto n.x3d 297 KB	model/vnd.c ollada+xml	728 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 34.0	Yes
					Mozilla	Yes

					Firefox 30.0	
AllenDutton.x3d 297 KB	model/vnd.collada+xml	728 KB	Ubuntu 12.04	Laptop (Dell Latitude E6330)	Mozilla Firefox 23.0	Yes
					Google Chrome 35.0	Yes
TestVideo1.mp4 8170 KB	video/mp4	8170 KB	Android 4.4	Tablet (Nexus 7)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	Yes
Gold-berg.stl 1706 KB	application/vnd.ms-pki.stl	2611 KB	Android 4.4	Tablet (Nexus 7)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	Yes
AllenDutton.x3d 297 KB	model/vnd.collada+xml	728 KB	Android 4.4	Tablet (Nexus 7)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	Yes
TestVideo1.mp4 8170 KB	video/mp4	8170 KB	iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	Yes
					Google Chrome 33.0	Yes
Gold-berg.stl 1706 KB	application/vnd.ms-pki.stl	2611 KB	iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	No
					Google Chrome 33.0	No

File Name & Size	Mime Type	Generalized File Size	Operating System	Device Name	Browser	Data Visibility
AllenDutton.x3d 297 KB	model/vnd.collada+xml	728 KB	iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	No
					Google Chrome 33.0	No

Table 5.14 Sample of test results set 3

Content 4: Subject-Electrical, Title-Car

File Name & Size	Mime Type	Generalized File Size	Operating System	Device Name	Browser	Data Visibility
Big_dodge.ply 535 KB	application/octet-stream	1239 KB	Windows 8	Laptop (Dell Latitude E6330)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	
Big_dodge.ply 535 KB	application/octet-stream	1239 KB	Windows 8.1	Desktop (Dell Optiplex 755)	Google Chrome 35.0	No
					IE 11	Yes
Big_dodge.ply 535 KB	application/octet-stream	1239 KB	LinuxMint 16	Desktop (HP DC5750)	Google Chrome 34.0	No
					Mozilla Firefox 30.0	Yes
Big_dodge.ply 535 KB	application/octet-stream	1239 KB	Android 4.4	Tablet (Nexus 7)	Google Chrome 35.0	Yes
					Mozilla Firefox 30.0	Yes
Big_dodge.ply 535 KB	application/octet-stream	1239 KB	iOS7.1	Smart phone (iPhone 5S)	Safari 6.0	No
					Google Chrome	No

					33.0	

Table 5.15 Sample of results set 4

5.2.2.1 Analysis of test results

From above stated tables showing results set 1 to results set 4 including the other entire test, it can be summerized that test has been conducted by using

- 4 different operating systems namely Windows 8 & 8.1, Android 4.1 & 4.4, iOS7.1 and LinuxMint 16 & Ubuntu 12.04

- 4 different browsers namely Mozilla Firefox 27 & 30; Google Chrome 33 &35; Safari 6.0; and Internet Explorer (IE) 10 & 11

- 4 different types of devices namely desktop (Linux), laptop (windows), tablet (android 4.4), and mobile phone (Android 4.1, iOS7.1)

- 3D data formats namely 3ds, obj, dae,stl,x3d,ply, lwo, ma, blend, fbx, wrl

- document and image formats namely doc, pdf, txt, jpeg, docx, odt, rtf

- audio and video data formats namely wma, mp4

Among above stated test parameters most of them were successful (in terms of data visibility) except iOS7.1, IE 10 and Safari in Windows machine, for 3D data due to lack of default WebGL support. Work around is possible to refine the functionality of Smoothie. For example use of Fallback in Smoothie. It facilitates an image-based fallback displaying a 360° view of the model where WebGL is not available. According to the idea provided by Sketchfab, a set of automatically generated screenshots of a 3D model can be taken, and put them together to create the equivalent of a turntable. It is generated as soon as the content author uploads the model. The fallback uses the set rotation axis of the model, so it might not always provide the ideal 360° view, depending on the initial view setup for the model. [136].

Even though according to the scope of the thesis (figure 3.1) and the design of the architecture (figure 3.3 and 5.1), it is able to deal with any kind of data; it can not be guaranteed that the end user´s device is able to display the received data. Below table 5.16 which, is derived from results set 1 to reselts set 4 (table 5.12 to 5.15) shows some example of irregular behavior of devices in terms of data visibility or bring the data into play. This irregular behavior was mainly observed in traditional high-end devices for 3D files and video file in a desktop having Linux as operating system.

Operating System	Device Name	File Name	Browser	Data Visibility
Windows 8.1	Desktop (Dell Optiplex 755)	Lantern1.obj.dae	Google Chrome 35.0	No
			IE 11	Yes
Windows 8.1	Desktop (Dell Optiplex 755)	AllenDutton.x3d.dae	Google Chrome 35.0	No
			IE 11	Yes
Windows 8.1	Desktop (Dell Optiplex 755)	R8.3ds.dae	Google Chrome 35.0	No
			IE 11	Yes
Windows 8.1	Desktop (Dell Optiplex 755)	Big_dodge.ply.dae	Google Chrome 35.0	No
			IE 11	Yes
Windows 8	Laptop (Dell Latitude E6330)	Lantern1.obj.dae	Google Chrome 35.0	Yes
			IE 10	No
LinuxMint 16	Desktop (HP DC5750)	Big_dodge.ply.dae	Google Chrome 34.0	No
			Mozilla Firefox 30.0	Yes
LinuxMint 16	Desktop (HP DC5750)	R8.3ds.dae	Google Chrome 34.0	No
			Mozilla Firefox 30.0	Yes
LinuxMint 16	Desktop (HP DC5750)	TestVideo1.mp4	Google Chrome 34.0	No
			Mozilla Firefox 30.0	Yes

Table 5.16 a derivation of irregular performance from results set 1- 4

Considering the first row from table 5.16 : according to the device capabilities of Dell Optiplex 755 having Windows 8.1 as operating system, it has the ability to display 3D data. The reason behind, exactly on the same device Lantern1.obj.dae was not displayed by Google Chrome 35.0 but was displayed by IE 11 is not found. It can not be a problem regarding Browser or it´s version because Google Chrome 35.0 was able to display Lantern1.obj.dae in Dell Latitude E6330.

Considering the last row from table 5.16 : It is also not explainable why in HP DC5750 a video file named TestVideo1.mp4 was being played by Mozilla Firefox 30.0 but was not possible to play by using Google Chrome 34.0.

Besides, from entire test results most of the negative results in terms of data visibility or bring the received data into play, were produced by Google Chrome. As "known issues" published by google it is found that When Chrome runs as a Windows 8 app, not all plug-ins are supported. Only plug-ins using Pepper API such as Flash, Native Client, and Chrome PDF Viewer will work. Relaunching Chrome in Windows 8 mode might be a solution depending upon below stated issues [137]-

Windows issues

- Outdated software causes Chrome to crash, freeze, or display blank pages
- Input Method Editor (IME) functionality does not work in Windows 8 mode
- Windows 8 mode doesn't work without hardware acceleration.
- Windows 8 mode doesn't work on hi DPI devices.

It can not be taken for granted that the very same device which was not able to display a certain kind of data once will not be able to display the data ever or a device which was able to display a certain kind of data once will always be displayed later. The visibility of data can be hindered by operating system, installed browser and it´s version, settings of the browser in use, volume of data and above all lack of WebGL support.

Chapter 5 explained the implementation of the theoretical plan provided in chapter 4 along with experimental validation of Smoothie by integrating it with a collaborative language learning game named LLG. It also mentioned some relevant test results. The next chapter will provide a general conclusion with summary of acheived result and open issues.

Chapter 6   Conclusion

Nowadays enormous content is being created everyday and there are plenty of communicative devices available out there. As a part of the world population it is desired by everybody, independent of place, age, sex and social status, to have access to information. This innocent desire is being resisted by creating intentional barrier. The proposed architecture Smoothie is the first step towards the betterment of the world by opening a door to be able to communicate with everybody and anybody so that different schemes such as Education For All (EFA), Anytime Anywhere Communication, etc., can literally come true. Smoothie is an integrable architecture, could be integrated to any other application

- where the application has to support varieties of devices along with varieties of data formats and those are not predictable

- independent of purpose, such as business, education or individual interest.

It is equally applicable for private use, academic purposes as well as business purposes to meet requirement either for a group or for individual interest. The design goals of Smoothie were to establish a communication where

- Devices can access web content appropriate for their capabilities,

- Authors can create web content which is deliverable across different devices

- Content can be accessed from different kinds of device with different capabilities

- used tools for implementation are either open source or at least free to use so that the prototype could be open for everybody

- enhancement of the system to cope with future will not require huge effort or resource

To meet the design goal the complete process of Smoothie was divided into three major phases namely

- Identification of device

- Preparing generalized content and

- Transformation of content

Identification of device is the very first responsibility of the system. In this phase end user device is being identified along with recognition of it´s capabilities with the help of WURFL. For later use,

safety reason and to make the process faster collected device capabilities from WURFL are being saved in an Object Oriented Database. Among available opensource object oriented databases DB4o was chosen for it´s replication through use of hibernate functionality, that synchronizes DB4o database with hibernate enhanced RDBMS (such as SQL) database. Another important feature of this phase is user authentication. The parameters of user authentication are dependent upon the application with which Smoothie will be integrated. Through user authentication, Smoothie is able to define the role and corresponding user rights over the data and the system of a valid user. User authentication and device identification are two different sub modules, correlated, not separable but easily extensible without requiring any change on the other sub module.

Preparing generalized content phase is independent from identification of device and Transformation of content phases. As soon as the content author uploads content this phase will statrt to be executed. Content author alias content provider may upload any kind of content/data in Smoothie independent of its format. Immediately after content upload Smoothie will save the contents in it´s original format in a file system for safety reason. As next step, based on the MIME Type of the uploaded content Smoothie will appoint either Appache Tika or Blender to prepare generalized content. Pre defined format of generalized content is XML for document type data and DAE for 3D data. Apache Tika and Blender are providing XML and DAE respectively as output. Generalized content is being saved in Generalized Content Repository and metadata of content is being saven in SqlDB, prepared to be transformed in device dependent manner based on necessity.

The third and last phase of Smoothie is transformation of content and it is dependent up on identification of device and preparing generalized content phases, even though it is a stand alone module according to the design. Immediately after receiving a request the device has to be identified along with it´s capabilities then the requested content has to be retrieved from generalized content repository based on the metadata of the content saved in SqlDB. Afterwards, based on several parameters such as volume of requested data, connectivity, device capability etc. transformation of content phase will take place to provide the reply of a corresponding request in a device dependent manner. Depending upon the ID which is a part of the link of the metadata different framework or library will be called for data transformation and visualization. For example in prototype of Smoothie apache cocoon will be appointed for serialization of XML data; ThreeJS will be appointed for rendering of DAE data; etc. Prototype of Smoothie is using Server side and client side transformation of content but it is easily extensible to use Hybride technique as well.

Above mentioned three phases of Smoothie are divided in three different modules, i.e. each of them is distinct but interrelated unit from which Smoothie is built up and it´s complex activity can be analysed and enhanced. In future if any further improvement in the quality, value, or extent of existing code is necessary in any of those modules, it can be done without interrupting others. A pictorial representation of the divition and relation of mudules is shown below
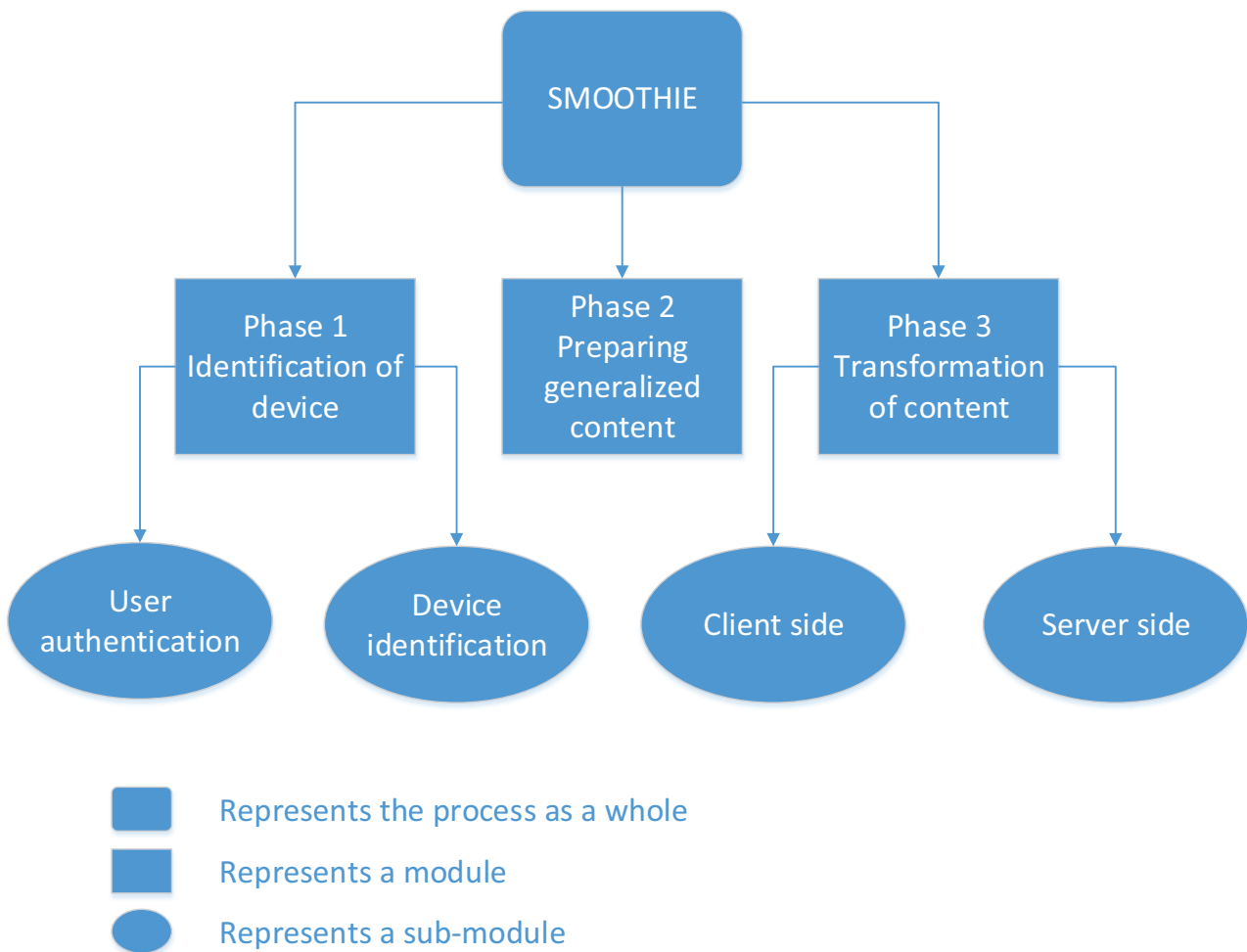
Figure 6.1 Pictorial representation of different modules of Smoothie

6.1   Achieved results & difference between theory and practice

As we know "there is no solution without limitation" Smoothie is also not out of the scope of this wise saying. Theoretically Smoothie is able to deal with any data independent of format and purpose. Practically, Import and Export functions of renowned open source software namely Blender is used to prepare generalized 3D data format. So, the proposed architecture Smoothie is dependent upon the given functionalities of Blender. Now if a 3D data has such a format that Blender is not able to import it, Smoothie won´t be able to deal with the data. Besides, each device behaves different based on operating system, installed application such as different browser or different version of installed applications. As we can see from the test results (sub section 5.2.3) it can not

be guaranteed that the very same device which was able to display a certain kind of data once will always be able to show the data or vice versa. How to extend given Import functionalities of Blender so that any 3D data format along with supporting files could be imported and how to overcome the dependency on installed application on a device are still open questions.

Moreover, even though the proposed architecture is allowing a device independent communication, the mobile devices lie under the lifetime of telecommunication system from 3rd Generation on, are considered under the scope of this thesis due to necessary speed of data transfer capability.

6.2   Future development

Smoothie was not designed to show what is technologically possible but what is technologically useful. It also does not want to describe what is coming next but what should come next. This thesis provides a prototype to prove the concept but not a product. So naturally there are open issues and scope for further research and development. Some of those are listed below but not limited to those

- Test could be done by using other available browsers, operating systems, different data formats and different types of devices as many as possible.

- How to make loss less compression of generalized format of data, when it is necessary to be send through internet

- How to make the efficient use of the server capabilities so that the system is less dependent upon the capabilities of end client´s device. At this point facilities of cloud services could be introduced and implemented along with Smoothie.

- Import and Export functions of renowned open source software namely Blender is used to prepare generalized 3D data format. So, the proposed architecture is dependent upon the given functionalities of Blender. How to extend given functionalities of Blender so that any 3D data formats along with supporting files could be imported.

- Smoothie is concentrating now on displaying different formats of content but not on presentation of content with right style. It is dependent upon provided facilities of Apache Cocoon for serialization.

- Smoothie is dependent upon provided facilities of Apache Tika for document type content generalization.

Further research and developing time is necessary to overcome or improve the level of dependencies.

References

[1]   M. Framingham, "Tablet Shipments Forecast to Top Total PC Shipments in the Fourth Quarter of 2013 and Annually by 2015, According to IDC," IDC press release, 11 September 2013. [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS24314413. [Accessed 12 June 2014].

[2]   Visual Networking Index (VNI), "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018," CISCO, 2013. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html. [Accessed 20 June 2014].

[3]   IDC Press Release, "Worldwide Mobile Phone Market Forecast to Grow 7.3% in 2013 Driven by 1 Billion Smartphone Shipments, According to IDC," IDC Analyze the Future, Online, 2013.

[4]   D. Evans, "The Internet of Things, How the Next Evolution of the Internet Is Changing Everything," CISCO IBSG, April 2011. [Online]. Available: http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf. [Accessed 2 November 2012].

[5]   Mobi Thinking, "Global mobile statistics 2014 Part A: Mobile subscribers; handset market share; mobile operators," mobiThinking, May 2014. [Online]. Available: http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a. [Accessed 20 June 2014].

[6]   p. Ned Freed, s. Mark Baker and s. Bjoern Hoehrmann, "Media Types," IANA, 13 June 2014. [Online]. Available: http://www.iana.org/assignments/media-types/media-types.xhtml. [Accessed 20 June 2014].

[7]   "The Source for File Extensions Information," File-Extensions.org, 5 February 2013. [Online]. Available: http://www.file-extensions.org/. [Accessed 12 June 2014].

[8]   K. Pulli, A. Baksheev, K. Kornyakov and V. Eruhimov, "Real-Time Computer Vision with OpenCV," *Communications of the ACM,* vol. 55, no. 6, pp. 61-69, 2012.

[9]   Wikipedia, "Responsive web design," wikipedia.org, 6 August Last modified 2014. [Online]. Available: http://en.wikipedia.org/wiki/Responsive_Web_Design. [Accessed 6 August 2014].

[10]   G. Appenzeller, M. Baker, K. Lai, P. Maniatis, M. Roussopoulos, E. Swierk and X. Zhao, "The Mobile People Architecture," *Mobile Computing and Communications Review,* vol. 3, no. 3, pp. 36-42, 1999.

[11]   H. Wang, B. Raman, R. Biswas, C. Chuah, R. Gummadi, B. Hohlt, X. Hong, E. Kiciman, Z. Mao, J. Shih, L. Subramanian, B. Zaoh, A. Joseph and R. Katz, "ICEBERG: An Internet-core Network Architecture for Integrated Communications," *IEEE Personal Communications: Special Issue on IP-based Mobile Telecommunications,* vol. 7, no. 4, pp. 10-19, 2000.

[12]   B. Thai, R. Wan, A. Seneviratne and T. Rakotoarivelo, "Integrated Personal Mobility Architecture: A Complete Personal Mobility Solution," *Special Issue of MONET Journal on Personal Environment Mobility in MultiProvider and Multi-Segment Networks,* vol. 8, pp. 27-36, 2003.

[13]   M. Youssef, A. Agrawala and U. Shankar, "WLAN location determination via clustering and probability distributions," in *IEEE International Conference on Pervasive Computing and Communications*, Dallas, 2003.

[14]   R. Zimmermann, "Lokalisierung Mobiler Geraete," Seminar Mobile Computing – SS 2001 Gruppe für verteilte Systeme ,

Departement Informatik, ETH Zürich, 2001. [Online]. Available: http://www.vs.inf.ethz.ch/edu/SS2001/MC/beitraege/07-location-rep.pdf. [Accessed 8 anuary 2011].

[15] Q. Wang and M. A. Abu-Rgheff, "INTEGRATED MOBILE IF' AND SIP APPROACH FOR ADVANCED LOCATION MANAGEMENT," in *4th International Conference on 3G Mobile Communication Technologies*, London, 2003.

[16] F. J. Gonzalez-Castano and J. Garcia-Reinoso, "Bluetooth location networks," in *IEEE Global Telecommunications Conference*, Taipei, 2002.

[17] ZigBee Alliance, "ZigBee Document 053474r06," 14 December 2004. [Online]. Available: http://www.datasheetarchive.com/pdf/Datasheet-035/DSA0012926.html. [Accessed 21st June 2014].

[18] ISO/IEC 18092:2013, "nformation technology -- Telecommunications and information exchange between systems -- Near Field Communication -- Interface and Protocol (NFCIP-1)," 2004. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=38578. [Accessed 20 June 2014].

[19] A. Christ and M. Feisst, "SW-Architecture for Device Independent Mobile Learning," in *Architectures for Distributed and Complex M-Learning Systems Applying Intelligent Technologies*, Hershey, Information Science Reference (an imprint of IGI Global), 2010, pp. 72-93.

[20] T. Glover and J. Davies, "Integrating device independence and user profiles on the Web.," *BT Technology Journal,* vol. 23, no. 3, pp. 239-248, 2005.

[21] R. A. Earnshaw, M. A. Gigante and H. Jones, Virtual Reality Systems, London: Academic Press Inc., 1993.

[22] I. Sutherland, "A head-mounted three dimensional display," in *Fall Joint Computer Conference*, San Francisco, 1968.

[23] S. Feiner, B. Macintyre and D. Seligmann, "Knowledge-based augmented reality," *Communications of the ACM,* vol. 36, no. 7, pp. 53-62, 1993.

[24] S. Mann, "Wearable Computing:A First Step Toward Personal Imaging," *Computer,* vol. NA, no. NA, pp. 25-32, 1997.

[25] M. Feisst, "Etude de protocole et réalisation d'un système de réalité augmentée mobile," Serveur des thèses de l'Université de Strasbourg, Strasbourg, 2006.

[26] M. Elgan, "Computerworld, Why 2014 is the 'year of smart glasses'," IDG, 23 December 2013. [Online]. Available: http://news.idg.no/cw/art.cfm?id=4004D866-0906-6ABB-6844C1D913815616. [Accessed 22 January 2014].

[27] Clear Your Doubts, "What is Mobile Radio Telephone System or 0G (Zero Generation) ?," clearyourdoubts.com, 2010. [Online]. Available: http://www.cleardoubts.com/technology/what-is-mobile-radio-telephone-system-or-0g-and-what-is-0-5g/. [Accessed 3 August 2013].

[28] Wikipedia, "Mobile radio telephone," Wikipedia.org, 26 April last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/Mobile_radio_telephone. [Accessed 3 August 2013].

[29] K. Siegmund, M. K. Weber and M. W. Oliphant, An Introduction to GSM, Tucson: Artech House Publishers, 1995.

[30] Clear your Doubts, "Technologies used in 1G or First generation of Wireless Telecommunication Technology," clearyourdoubts.com, 2010. [Online]. Available: http://www.cleardoubts.com/index.php?s=1G. [Accessed 3 September 2013].

[31] Wikipedia, "2G," Wikipedia.org, 12 June last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/2G. [Accessed 1 October 2013].

[32] A. Huber and J. F. Huber, UMTS and mobile computing, London: Artech House, 2002.

[33] H. Holma and A. Toskala, WCDMA for UMTS: Radio Access for Third Generation Mobile Communications, New York: John Wiley & Sons Inc., 2001.

[34] GSM Association, "Gsm world statistics," gsmworld.com, 2010. [Online]. Available: gsmworld.com. [Accessed 8 June 2010].

[35] GSM Case Study, "3G Mobile Licensing Policy: From GSM TO IMT-2000- A COMPARATIVE ANALYSIS," 2000. [Online]. Available: www.itu.int/osg/spu/ni/3G/casestudies/GSM-FINAL.doc. [Accessed 1 October 2013].

[36] ITU, "ITU defines the future of mobile communications, ITU Radiocommunication Assembly approves new developments for its 3G standards," ITU, 23 October last update 2007. [Online]. Available: http://www.itu.int/newsroom/press_releases/2007/30.html. [Accessed 2 October 2010].

[37] J. Korhonen, Introduction to 3G Mobile Communications, Boston: Artech House, 2003.

[38] M. Sauter, Communication Systems for the Mobile Information Society, Hoboken: John Wiley & Sons, 2006.

[39] Wikipedia, "High-Speed Downlink Packet Access," wikipedia.org, 04 June last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/High-Speed_Downlink_Packet_Access. [Accessed 2 October 2013].

[40] ITU Radiocommunication Bureau, Circular Letter , "IEEE WirelessMAN 802.16," 7 March 2008. [Online]. Available: http://wirelessman.org/liaison/docs/L80216-08_008.pdf. [Accessed 4 October 2013].

[41] J. Vilches, "Everything You Need To Know About 4G Wireless Technology," TechSpot Inc., 29 April 2010. [Online]. Available: http://www.techspot.com/guides/272-everything-about-4g/. [Accessed 4 October 2013].

[42] ITU Telecom World, ICT Data and Statistics Division, "One third of the world's population is online,45% of Internet users below the age of 25," 2011. [Online]. Available: http://www.itu.int/ITU-D/ict/facts/2011/material/ICTFactsFigures2011.pdf. [Accessed 4 June 2014].

[43] Ericsson, "Ericsson Mobility Report," Online, 2012.

[44] P. Lambert, "Press release: 70% of operators believe now is the right time to launch LTE (4G) services," Informa telecoms and media, Online, 2012.

[45] Strategy Analytics, "One Billion LTE Connections by 2017, Strong Smartphone Growth to Propel 4G Market Forward," Strategy Analytics, Boston, 2012.

[46] Scientia Mobile Commercial Support & Licensing, "Welcome to WURFL," Scientia Mobile Inc., Last update 2014. [Online]. Available: http://wurfl.sourceforge.net/. [Accessed 24 January 2012].

[47] W3C, "Device Description Landscape," W3C, 31 October 2007. [Online]. Available: http://www.w3.org/TR/dd-landscape/. [Accessed 11 March 2013].

[48] C. Russel, "Device-aware mobile sites using PHP, JavaScript, and WURFL, Part 1: Get started using the WURFL PHP API Taking the guesswork out of mobile device capabilities," IBM, 7 February 2012. [Online]. Available: http://www.ibm.com/developerworks/library/mo-wurfl-1/mo-wurfl-1-pdf.pdf. [Accessed 18 March 2013].

[49] E. Bertino and L. Martino, "Object-oriented database management systems: concepts and issues," *Computer,* vol. 24, no. 4, pp. 33-47, April 1991.

[50] P. Suri and M. Sharma, "A Comparative study between the performance of relational & object oriented database in data warehousing," *International Journal of Database Management Systems (IJDMS),* vol. 3, no. 2, pp. 116-127, May 2011.

[51] A. R. Hurson, S. H. Pakzad and J. Cheng, "Object-Oriented Database Management Systems: Evolution and Performance Issues," *Computer,* vol. 26, no. 2, pp. 48-58, February 1993.

[52] M. Jarke, R. Gallersdörfer, M. A. Jeusfeld, M. Staudt and S. Eherer, "ConceptBase — A deductive object base for meta data management," *Journal of Intelligent Information Systems,* vol. 4, no. 2, pp. 167-192, 1 March 1995.

[53] R. Sultana, M. Feisst and A. Christ, "Collaborative Language Learning Game as a Device Independent Application," in *Towards Learning and INstruction in Web 3.0; Advances in Cognitive and Educational Psychology*, New York, Springer, 2012, pp. 73-87.

[54] Wikipedia, "YAML," 2 August last modified 2014. [Online]. Available: http://en.wikipedia.org/wiki/YAML. [Accessed 7 August 2014].

[55] wikipedia, "JSON," wikipedia.org, 6 August Last modifies 2014. [Online]. Available: http://en.wikipedia.org/wiki/JSON. [Accessed 7 August 2014].

[56] wikipedia, "XML," wikipedia.org, 31 July last modified 2014. [Online]. Available: http://en.wikipedia.org/wiki/XML. [Accessed 7 August 2014].

[57] O. Ben-Kiki, C. Evans et I. döt Net, «YAML Ain't Markup Language (YAML™) Version 1.2,» 1 October 2009. [En ligne]. Available: http://yaml.org/spec/1.2/spec.html. [Accès le 7 August 2014].

[58] D. Crockford, "JSON: The Fat-Free Alternative to XML," Jason.org, [Online]. Available: http://www.json.org/xml.html. [Accessed 7 August 2014].

[59] N. Nurseitov, M. Paulson, R. Reynolds and C. Izurieta, "Comparison of JSON and XML Data Interchange Formats: A Case Study," 2009. [Online]. Available: http://www.cs.montana.edu/izurieta/pubs/caine2009.pdf. [Accessed 7 August 2014].

[60] W3C, "Extensible Markup Language (XML) 1.0 (Fifth Edition)," 26 November 2008. [Online]. Available: http://www.w3.org/TR/xml/. [Accessed 15 February 2014].

[61] J. Bosak, "XML, Java, and the Future of the Web," 2 October 1997. [Online]. Available: http://www.xml.com/pub/a/w3j/s3.bosak.html. [Accessed 15 February 2014].

[62] Autodesk, "3D modeling, animation, and rendering software," Autodesk, last update 2014. [Online]. Available: http://www.autodesk.com/products/3ds-max/overview. [Accessed 7 February 2014].

[63] U.S. Satellite Laboratory Inc., "Project 3D VIEW (Virtual Interactive Environmental Worlds)," U.S. Satellite Laboratory Inc. Sponsored by NASA, last update 2014. [Online]. Available: http://www.3dview.org/. [Accessed 7 February 2014].

[64] inivis, "AC3D™," INIVIS, 20 March latest release 2014. [Online]. Available: http://www.inivis.com/. [Accessed 5 August 2014].

[65] P. Eastman, «Art of Illusion,» AOI, 23 September latest release 2013. [En ligne]. Available: http://www.artofillusion.org/. [Accès le 5 August 2014].

[66] Wikipedia, "GNU General Public License," Wikipedia.org, 8 July last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/GNU_General_Public_License. [Accessed 5 August 2014].

[67] Blender Foundation, "Blender," Blender Foundation, 26 June latest update 2014. [Online]. Available: http://www.blender.org/. [Accessed 5 August 2014].

[68] Daz Productions, Inc., "Bryce 7 The First Name in 3D Landscapes," Daz3D, 2011. [Online]. Available: http://www.daz3d.com/products/bryce/bryce-what-is-bryce. [Accessed 5 August 2014].

[69] Daz Productions, Inc., "Carrara 8 The Next Dimension in 3D Art," Daz3D, 2011. [Online]. Available: http://www.daz3d.com/products/carrara/carrara-what-is-carrara. [Accessed 5 August 2014].

[70] MW3D Solutions, «The amazingly powerful 3D modeling, rendering and animation software for Apple Mac OS X,» MW3D Solutions, 2001. [En ligne]. Available: http://www.cheetah3d.com/. [Accès le 5 August 2014].

[71]   Maxon Computer, "3D for the real World," Maxon Computer, 13 April last release 2014. [Online]. Available: http://www.maxon.net/products/new-in-cinema-4d-r16/overview.html. [Accessed 5 August 2014].

[72]   Esri, "Esri City Engine Transform 2D GIS Data into Smart 3D City Models," Esri, 13 April last release 2012. [Online]. Available: http://www.esri.com/software/cityengine. [Accessed 5 August 2014].

[73]   Exocortex Technologies, Inc, "Clara.io Model. Animate. Render. Online.," Exocortex Technologies, Inc, 9 June last release 2013. [Online]. Available: https://clara.io/. [Accessed 5 August 2014].

[74]   Ashlar-Vellum, "Cobalt™ 3D Modeling The ultimate 3D modeling product design and development tool," Ashlar-Vellum, 11 December last release 2009. [Online]. Available: http://www.ashlar.com/3d-modeling/3d-modeling-cobalt.html. [Accessed 5 August 2014].

[75]   EIAS3D, "eias3D EIAS 9 Overview," EIAS3D, June last release 2013. [Online]. Available: http://www.eias3d.com/. [Accessed 5 August 2014].

[76]   AutoDesSys, "form Z The Fastest, Easiest, Most Flexible 3d Modeling Software Available," AutoDesSys, August last release 2012. [Online]. Available: http://www.formz.com/. [Accessed 5 August 2014].

[77]   Daz3D, "Hexagon 2.5 Model Your Inspiration Everything you need to create complete, detailed, textured 3D models ready to be rendered," DAZ Productions, Inc, 20 May last release 2008. [Online]. Available: http://www.daz3d.com/products/hexagon/hexagon-what-is-hexagon. [Accessed 5 August 2014].

[78]   Side Effects Software Inc., "Houdini 3D Animation Tools," Side Effects Software Inc., 31 October last release 2013. [Online]. Available: http://www.sidefx.com/. [Accessed 5 August 2014].

[79]   NewTek LightWave Group, "Welcome to Lightwave," NewTek Inc., 1 November 2013. [Online]. Available: https://www.lightwave3d.com/. [Accessed 5 August 2014].

[80]   MakeHuman team, "Makehuman Open Source Tool For Making 3D Characters," MakeHuman team, 3 February 2014. [Online]. Available: http://www.makehuman.org/. [Accessed 5 August 2014].

[81]   Wikipedia, "Affero General Public License," wikipedia.org, 19 July last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/Affero_General_Public_License. [Accessed 5 August 2014].

[82]   Massive Software, "Massive Simulating Life," Massive Software, 31 July last release 2013. [Online]. Available: http://www.massivesoftware.com/. [Accessed 5 August 2014].

[83]   Autodesk Inc., "MAYA Comprehensive 3D animation software," Autodesk Inc., 25 September last release 2013. [Online]. Available: http://www.autodesk.com/products/maya/overview. [Accessed 5 August 2014].

[84]   tetraface Inc., "Metasequoia 4," tetraface Inc., 18 July last release 2014. [Online]. Available: http://metaseq.net/en/index.html. [Accessed 5 August 2014].

[85]   Wikipedia, «Shareware,» wikipedia.org, 6 July last update 2014. [En ligne]. Available: http://en.wikipedia.org/wiki/Shareware. [Accès le 6 August 2014].

[86]   Bentley Systems, Incorporated, "MicroStation V8i 3D CAD Design and Modeling Software for Architecture, Engineering, Construction and Operations," Bentley Systems, Incorporated, 6 November last release 2008. [Online]. Available: http://www.bentley.com/en-US/Products/MicroStation/. [Accessed 6 August 2014].

[87]   The Foundra, "MODO The next generation of 3D content creation," Luxologx LLC., 1 January last release 2014. [Online]. Available: http://www.thefoundry.co.uk/products/modo/. [Accessed 6 August 2014].

[88]   Wikipedia, "POV-Ray," Persistence of Vision Raytracer Pty. Ltd., 9 November latest release 2013. [Online]. Available:

http://en.wikipedia.org/wiki/Pov-ray. [Accessed 4 February 2014].

[89] Wikipedia, "Affero General Public License," Wikimedia Foundation, Inc., 19 July last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/Affero_General_Public_License. [Accessed 4 Frbruary 2014].

[90] PTC Inc., "PTC Creo is Award Winning 2D and 3D CAD Software," PTC Inc., 5 March last release 2012. [Online]. Available: http://www.ptc.com/product/creo. [Accessed 6 August 2014].

[91] Remograph AB, "Remo 3D," Remograph AB, 13 January latest release 2014. [Online]. Available: http://www.remograph.com/products.php#remo3d. [Accessed 6 August 2014].

[92] Robert McNeel & Associates, "Rhinoceros design, model, present, analyze, realize...," Robert McNeel & Associates, 17 June latest release 2014. [Online]. Available: http://www.rhino3d.com/. [Accessed 6 August 2014].

[93] Pixologic, Inc., "SCULPTRIS, Focus as an artist on pure creativity," Pixologic, Inc., 22 July Latest release 2011. [Online]. Available: http://pixologic.com/sculptris/. [Accessed 6 August 2014].

[94] Proactive International, LLC, "Shade 3D 14 Model | Render | Animate | Print to 3D: Creativity Unleashed," Proactive International, LLC, 6 March Latest release 2014. [Online]. Available: http://mirye.net/shade-3d. [Accessed 6 August 2014].

[95] Nevercenter Ltd. Co., "Silo 3D modeling zen.," Nevercenter Ltd. Co., latest release 2010. [Online]. Available: http://www.nevercenter.com/silo/. [Accessed 6 August 2014].

[96] Trimble Navigation Limited, "SketchUp The easiest way to draw in 3D," Trimble Navigation Limited, Latest release 2014. [Online]. Available: http://www.sketchup.com/products/sketchup-pro. [Accessed 6 August 2014].

[97] Autodesk Inc., "SOFTIMAGE," Autodesk Inc., 14 April Expected last release 2014. [Online]. Available: http://www.autodesk.com/products/softimage/overview. [Accessed 6 August 2013].

[98] Siemens Product Lifecycle Management Software Inc., "Solid Edge Design better," Siemens Product Lifecycle Management Software Inc., 26 June Latest release 3013. [Online]. Available: http://www.plm.automation.siemens.com/en_us/products/velocity/solidedge/. [Accessed 4 February 2014].

[99] solidThinking, Inc., "solidThinking," solidThinking, Inc., 18 June Latest release 2013. [Online]. Available: http://www.solidthinking.com/. [Accessed 3 February 2014].

[100] Dassault Systèmes SolidWorks Corp., "SOLIDWORKS Solutions Cover All Aspects of Your Development Process," Dassault Systèmes SolidWorks Corp., 26 November Latest release 2013. [Online]. Available: http://www.solidworks.com/. [Accessed 4 February 2014].

[101] SpaceClaim Corporation, "SpaceClaim solves geometry problems, fast.," SpaceClaim Corporation, 10 December Latest Release 2013. [Online]. Available: http://www.spaceclaim.com/. [Accessed 4 February 2014].

[102] Electric Rain, Inc., "Swift 3D Standalone 3D Modeling and Animation Software," Electric Rain, Inc., June Latest Release 2009. [Online]. Available: http://www.erain.com/products/swift3d/. [Accessed 4 February 2014].

[103] e-on software, inc, "Vue, Solutions for Your Digital Nature," e-on software, inc, Latest release 2014. [Online]. Available: http://www.e-onsoftware.com/products/vue/. [Accessed 6 August 2014].

[104] B. Gustavsson, D. Gudmundsson and a. others, "WINGS 3D," April Latest Release 2014. [Online]. Available: http://www.wings3d.com/. [Accessed 6 August 2014].

[105] Wikipedia, "BSD licenses," wikipedia.org, 23 July last modified 2014. [Online]. Available: http://en.wikipedia.org/wiki/BSD_License. [Accessed 6 August 2014].

[106] Pixologic, Inc. , "ZBRUSH AT A GLANCE," Pixologic, Inc. , June Latest Release 2013. [Online]. Available: http://pixologic.com/zbrush/. [Accessed 9 January 2014].

[107] Wikipedia, "Wavefront Technologies," Wikimedia Foundation, Inc., 14 January last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/Wavefront_Technologies. [Accessed 8 August 2014].

[108] Wikipedia, "Unit vector," Wikimedia Foundation, Inc, 6 July last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/Unit_normal. [Accessed 8 August 2014].

[109] Wikipedia, "Wavefront .obj file," Wikimedia Foundation, Inc., 14 June last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/Wavefront_.obj_file. [Accessed 8 August 2014].

[110] ISO, "ISO/PAS 17506:2012, Industrial automation systems and integration -- COLLADA digital asset schema specification for 3D visualization of industrial data," 13 July 2012. [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59902. [Accessed 27 February 2014].

[111] Dassault Systemes, "3DVIA Announcement: Now supporting COLLADA .dae Downloads," 3DVIA, 11 August 2009. [Online]. Available: http://www.3dvia.com/blog/3dvia-announcement-now-supporting-3d-dae-download/. [Accessed 8 August 2014].

[112] M. Barnes and E. L. Finch, "COLLADA – Digital Asset Schema Release 1.5.0 Specification," The Khononos Group Inc., Sony Computer Entertainment Inc., April 2008. [Online]. Available: https://www.khronos.org/files/collada_spec_1_5.pdf. [Accessed 6 March 2014].

[113] Khoronos Group, "Runtime asset format for WebGL, OpenGL ES, and OpenGL," Khoronos Group, Unknown. [Online]. Available: http://www.khronos.org/gltf. [Accessed 8 August 2014].

[114] Khronos Group, "COLLADA," Gold Standard Group, 14 July last update 2014. [Online]. Available: http://collada.org/. [Accessed 8 August 2014].

[115] W3C Recommendation, "Mobile Web Best Practices 1.0 Basic Guidelines," July 2008. [Online]. Available: http://www.w3.org/TR/mobile-bp/. [Accessed 1 March 2014].

[116] E. Marcotte, "A List Apart, Responsive Web Design," 25 May 2010. [Online]. Available: http://alistapart.com/article/responsive-web-design. [Accessed 1 March 2014].

[117] S. Champeon, "Progressive Enhancement and the Future of Web Design," 21 March 2003. [Online]. Available: http://www.hesketh.com/thought-leadership/our-publications/progressive-enhancement-and-future-web-design. [Accessed 1 March 2014].

[118] J. L. Zitting and C. A. Mattmann, Tika In Action, Online: Manning Publications, 2011.

[119] Wikimedia, "Wikimedia servers," WIkimedia.org, 25 June Last update 2014. [Online]. Available: http://meta.wikimedia.org/wiki/Wikimedia_servers. [Accessed 26 June 2014].

[120] TheOpenForce, "Google Runs MySQL," TheOpenForce.com, 13 December 2005. [Online]. Available: http://zurlocker.typepad.com/theopenforce/2005/12/googles_use_of_.html. [Accessed 3 August 2010].

[121] M. Callaghan, Director, *MySQL at Facebook.* [Film]. California: MySQL Conference & Expo, 2010.

[122] J. Cole, Director, *Big and Small Data at @ Twitter.* [Film]. California: MySQL Conference & Expo, 2011.

[123] MySQL, "MySQL Customer: Flickr," MySQL.com, [Online]. Available: http://www.mysql.com/customers/view/?id=720. [Accessed 10 September 2013].

[124] MySQL, "MySQL Customer: YouTube," MySQL.com, [Online]. Available: http://www.mysql.com/customers/view/?id=750.

[Accessed 17 September 2012].

[125] C3DL.org, "Canvas 3d JS Library," C3DL, 22 March 2011. [Online]. Available: http://www.c3dl.org/. [Accessed 10 February 2013].

[126] GLGE, "GLGE: WebGL for lazy," GLGE, 12 February latest update 2012. [Online]. Available: http://www.glge.org/. [Accessed 10 February 2013].

[127] J. Dirksen, Learning Three.js: The JavaScript 3D Library for WebGL, Birmingham: Packt Publishing Ltd, 2013.

[128] Google, "groups.google.com," google, 27 August 2013. [Online]. Available: https://groups.google.com/forum/#!topic/glge/e8t_6SSuk7Y. [Accessed 10 February 2014].

[129] Wikipedia, "Three.js," wikipedia.org, 12 June last update 2014. [Online]. Available: http://en.wikipedia.org/wiki/Threejs. [Accessed 26 June 2014].

[130] R. Wong, "In Mobile, Fragmentation is Forever. Deal With It.," TechCrunch, 4 March 2010. [Online]. Available: http://techcrunch.com/2010/03/04/mobile-fragmentation-forever/. [Accessed 26 June 2014].

[131] L. Passani, "New WURFL Snapshot available, BUT WITH IMPORTANT CHANGE," Yahoo Groups, 30 August 2011. [Online]. Available: https://groups.yahoo.com/neo/groups/wmlprogramming/conversations/topics/34311. [Accessed 26 June 2014].

[132] Scientiamobile Press Release, "ScientiaMobile Offers WURFL Image Tailor for Easy to Use Mobile Image Resizing," Scientiamobile.com, 6 June 2011. [Online]. Available: http://www.scientiamobile.com/page/category/press-releases. [Accessed 26 June 2014].

[133] R. Sultana, A. Christ and P. Meyrueis, "Architecture of a device independent collaborative language learning game," in *Handbook of Mobile Learning*, L. B. Zane and L. M. Mulienburg, Eds., New York, Abingdon, Routledge, 2013, pp. 545-560.

[134] R. Sultana, M. Feisst and A. Christ, "Collaborative Language Learning Game as a Device Independent Application," in *Towards Learning and Instruction in Web 3.0 Advances in Cognitive and Educational Psychology*, New York, London, Springer, 2012, pp. 73-88.

[135] Apache.org, "Mime Types," Apache.org, 12 March 2014. [Online]. Available: http://svn.apache.org/viewvc/httpd/httpd/branches/2.2.x/docs/conf/mime.types?view=annotate. [Accessed 20 June 2014].

[136] Sketchfab, "Sketchfab Blog," Sketchfab and users of Sketchfab, 11 August last update 2014. [Online]. Available: http://blog.sketchfab.com/post/47164072537/fallback-release. [Accessed 19 August 2014].

[137] Google, "Known issues," Google, last update 2014. [Online]. Available: https://support.google.com/chrome/known-issues/15036?hl=en. [Accessed 19 August 2014].

Appendices

A Some of the most used 3D computer graphics formats

I. List that allows to build models in real-time or non real-time 3D rendering

3DMF — QuickDraw 3D Metafile (.3dmf)

3DS, MAX — 3D Studio Max Model (.max, .3ds)

3DT — 3D Topicscape The database in which the meta-data of a 3D Topicscape is held. A 3D Topicscape is a form of 3D concept map (like a 3D mind-map) used to organize ideas, information and computer files.

ATY — 3D Topicscape file, produced when an association type is exported by 3D Topicscape. Used to permit round-trip (export Topicscape, change files and folders as desired, re-import them to 3D Topicscape).

AC — AC3D Model (.ac)

AN8 — Anim8or Model (.an8)

AOI — Art of Illusion Model (.aoi)

B3D — Blitz3D Model (.b3d)

BLEND — Blender (.blend)

C4D — Cinema 4D (.c4d)

Cal3D — Cal3D (.cal3d)

CAG — Linear Reference System.

CFL — Compressed File Library (.cfl)

COB — Caligari Object (.cob)

CTM — OpenCTM (.ctm)

DAE — COLLADA (.dae)

DTS — Torque Game Engine (.dts)

EGG — Panda3D Engine

FACT — Electric Image (.fac)

FBX — Autodesk FBX (.fbx)

FES — 3D Topicscape file, produced when a fileless occurrence in 3D Topicscape is exported to Windows. Used to permit round-trip (export Topicscape, change files and folders as desired, re-import them to 3D Topicscape).

G — BRL-CAD geometry (.g)

GLM — Ghoul Mesh (.glm)

LWO — Lightwave Object (.lwo)

LWS — Lightwave Scene (.lws)

LXO — Luxology Modo (software) file (.lxo)

MA — Autodesk Maya ASCII File (.ma)

MB — Autodesk Maya Binary File (.mb)

MD2 — Quake 2 model format (.md2)

MD3 — Quake 3 model format (.md3)

MDX — Blizzard Entertainment's own model format (.mdx)

MESH — New York University(.m)

MESH — Meshwork Model (.mesh)

MM3D — Misfit Model 3d (.mm3d)).

MMP — Mind Manager mind map file.

TPC — 3D Topicscape file, produced when an inter-Topicscape topic link file is exported to Windows. Used to permit round-trip (export Topicscape, change files and folders as desired, re-import them to 3D Topicscape).

NIF — Gamebryo NetImmerse File (.nif)

OBJ — OBJ (.obj)

OFF — OFF Object file format (.off)

PRC — Adobe PRC (embedded in PDF files)

POV — POV-Ray Document (.pov)

RWX — RenderWare Object (.rwx)

SIA — Nevercenter Silo Object (.sia)

SIB — Nevercenter Silo Object (.sib)

ANEE — 3D Flash Gallery Model (.u3d)

SKP — Google Sketchup file (.skp)

SLDASM — SolidWorks Assembly Document (.sldasm)

SLDPRT — SolidWorks Part Document (.sldprt)

SMD — Valve's format. (.smd)

U3D — Universal 3D file format (.u3d)

W3D—Westwood 3D object format

WINGS — Wings3D (.wings)

X — DirectX 3D Model (.x)

X3D — Extensible 3D (.x3d)

Z3D — Zmodeler (.z3d)

II. List that allows Computer aided design to assist engineers, architects and other design professionals in project design

3DMLW - 3D Markup Language for Web files

3DXML - Dassault Systemes graphic representation

ACP — VA Software VA - Virtual Architecture CAD file

AR — Ashlar-Vellum Argon - 3D Modeling

ART — ArtCAM model

ASC — BRL-CAD Geometry File (old ascii format)

ASM — Solidedge Assembly, Pro/ENGINEER Assembly

BIM — Data Design System DDS-CAD

CCC — CopyCAD Curves

CCM — CopyCAD Model

CCS — CopyCAD Session

CAD — CadStd

CATDrawing — CATIA V5 Drawing document

CATPart — CATIA V5 Part document

CATProduct — CATIA V5 Assembly document

CATProcess — CATIA V5 Manufacturing document

CGR — CATIA V5 graphic representation file

CO — Ashlar-Vellum Cobalt - parametric drafting and 3D modeling

DRW — Caddie Early version of Caddie drawing - Prior to Caddie changing to DWG

DWG — AutoCAD and Open Design Alliance applications

DFT — Solidedge Draft

DGN — MicroStation design file

DGK — Delcam Geometry

DMT — Delcam Machining Triangles

DOTXSI— An ascii file format of Softimage corporation for storing scene data

DXF — ASCII Drawing Interchange file format - AutoCAD

DWB — VariCAD drawing file

DWF — AutoDesk's Web Design Format; AutoCAD & Revit can publish to this format; similar in concept to PDF files;

AutoDesk Design Review is the reader

EMB— Wilcom - Wilcom ES Designer Embroidery CAD file

ESW — Agtek format

EXCELLON— Excellon file

FM — FeatureCAM Part File

FMZ — FormZ Project file

G — BRL-CAD Geometry File (.g)

GBR— Gerber file

GRB — T-FLEX CAD File

GTC — GRAITEC Advance file format

IAM — Autodesk Inventor Assembly file

ICD — IronCAD 2D CAD file

IDW — Autodesk Inventor Drawing file

IFC — buildingSMART for sharing AEC and FM data

IGES— Initial Graphics Exchange Specification

ISFF— Intergraph's Intergraph Standard File Formats

IPN — Autodesk Inventor Presentation file

IPT — Autodesk Inventor Part file model - CATIA V4 part document

PAR — Solidedge Part

PRT — NX (recently known as Unigraphics), Pro/ENGINEER Part, CADKEY Part

PLN — ArchiCad project

PLY—Polygon file format, principally used by the 3D scanner

PSM — Solidedge Sheet

PSMODEL — PowerSHAPE Model

PWI — PowerINSPECT File

PYT — Pythagoras File

SKP— SketchUp Model

RLF — ArtCAM Relief

RVT — AutoDesk Revit project files

RFA — AutoDesk Revit family files

SLDASM — SolidWorks Assembly drawing

SLDDRW — SolidWorks drawing

SLDPRT — SolidWorks part model

STEP — Standard for the Exchange of Product model data

STL— Stereo Lithographic data format, used by various CAD systems and stereo lithographic printing machines.

TCT —TurboCAD drawing template

TCW — TurboCAD for Windows 2D and 3D drawing

VC6 — Ashlar-Vellum Graphite - 2D and 3D drafting

VLM — Ashlar-Vellum Vellum, Vellum 2D, Vellum Draft, Vellum 3D, DrawingBoard

VS — Ashlar-Vellum Vellum Solids

WRL— Similar to STL, but includes color. Used by various CAD systems and 3D printing rapid prototyping machines. Also used for VRML models on the web.

XE —Xenon - for Associative 3D Modeling used by Ashlar-Vellum

Sources

[http://en.wikipedia.org/wiki/Alphabetical_list_of_file_formats_(S%E2%80%93Z)]

[http://artist-3d.com/free_3d_models/graphics-file-formats.php]

B   Most important part of source code to detect the device through WURFL Java API

I.   Redirecting between different markup languages for the very first view of User Interface

```java
/** Servlet problems handles initial request from the client and decides the markup language to use */
public class RequestHandler extends HttpServlet implements WurflWebConstants{

        /** Serial */
        private static final long serialVersionUID = 10L;

        private static final String XHTML_ADV = "xhtmladv.jsp";
        private static final String XHTML_SIMPLE = "xhtmlmp.jsp";
        private static final String CHTML = "chtml.jsp";
        private static final String WML = "wml.jsp";

        /** Logger */
        private final Logger log = LoggerFactory.getLogger(getClass());

        /**
         * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
         * methods.
         *
         * @param request
         *         servlet request
         * @param response
         *         servlet response
         */
        protected void processRequest(HttpServletRequest request,
                          HttpServletResponse response) throws ServletException, IOException {

                WURFLHolder holder = (WURFLHolder) getServletContext()
                                .getAttribute(WURFL_HOLDER_KEY);

                WURFLManager mgr=holder.getWURFLManager();
                Device device = mgr.getDeviceForRequest(request);

                log.debug("Device: " + device.getId());
                log.debug("Capability: " + device.getCapability("preferred_markup"));

                MarkUp markUp = device.getMarkUp();
                System.out.println("Mark up is"+markUp);
                String jspView = null;

                if (MarkUp.XHTML_ADVANCED.equals(markUp)) {
                        jspView = XHTML_ADV;
                } else if (MarkUp.XHTML_SIMPLE.equals(markUp)) {
                        jspView = XHTML_SIMPLE;
                } else if (MarkUp.CHTML.equals(markUp)) {
                        jspView = CHTML;
                } else if (MarkUp.WML.equals(markUp)) {
                        jspView = WML;
                }

                log.info("View: " + jspView);

                // MIME type is decided by JSP. Only in case of XHTML we will need to
                // multi-serve
                // i.e text/html vs application/xml+xhtml vs
                // application/vnd.wap.xml+xhtml
                if (markUp == MarkUp.XHTML_ADVANCED || markUp == MarkUp.XHTML_SIMPLE) {
                        String contentType = "text/html";
                        try {
                                contentType = device
```

```java
                                .getCapability("xhtmlmp_preferred_mime_type");
                } catch (CapabilityNotDefinedException e) {
                        throw new RuntimeException(
                                "Somethingh is seriously wrong with your WURFL:"
                                        + e.getLocalizedMessage(), e);
                }
                request.setAttribute("contentType", contentType);
                log.debug("ContentType: " + contentType);
        }
        request.setAttribute("markUp", markUp);
        request.setAttribute("device", device);

        request.getRequestDispatcher("WEB-INF/jsp/" + jspView).forward(request,
                        response);
}
protected void doGet(HttpServletRequest request,
                HttpServletResponse response) throws ServletException, IOException {
        processRequest(request, response);
}
protected void doPost(HttpServletRequest request,
                HttpServletResponse response) throws ServletException, IOException {
        processRequest(request, response);
}}
```

```java
/** Program collects device capabilities information from the Wurfl.xml file  */
import java.util.Hashtable;
import java.util.Vector;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.Attributes;
import org.xml.sax.helpers.DefaultHandler;

public class WurflParser extends DefaultHandler
{
    public  Vector<Device> deviceList;
    private Device currentDevice;
    private Group  currentGroup;
    private String keyword;

    public static void main(String[] args)
    {
/** nolian95 is the input to the Wurfl Parser */

        WurflParser wp = new WurflParser("nokian95");

        System.out.println("Total Matches for \"" + wp.keyword + "\": " + wp.deviceList.size() + "\n");
        for(int i = 0; i < wp.deviceList.size(); i++)
            System.out.println(wp.deviceList.elementAt(i));
    }
/** this program uses SAX parser for parsing xml data, because wurfl.xml is an xml file we are using java xml parser methods to
fetch the device details */

    public WurflParser(String keyword)
    {
        this.keyword = keyword.toLowerCase().trim();
        this.deviceList = new Vector<Device>();

        try
        {
            SAXParserFactory spf = SAXParserFactory.newInstance();
            SAXParser sp = spf.newSAXParser();
            sp.parse("res/wurfl.xml", this);
        }
        catch (Exception e)
        {
            System.out.println("Exception while parsing the file. " + e);
        }
    }

    public void startElement(String uri, String localName, String qName,
        Attributes attributes)
    {
      if (qName.equalsIgnoreCase("device") && attributes.getLength() > 0)
      {
        if (!isValidDevice(attributes.getValue("user_agent")))
          return;
        this.currentDevice = new Device();
        this.currentDevice.id = attributes.getValue("id");
        this.currentDevice.fallBack = attributes.getValue("fall_back");
        this.currentDevice.actualDeviceRoot = Boolean.parseBoolean(attributes.getValue("actual_device_root"));
        this.currentDevice.userAgent = attributes.getValue("user_agent");
      }
      else if (qName.equalsIgnoreCase("group") && (attributes.getLength() > 0)
          && (this.currentDevice != null))
      {
```

```java
            this.currentGroup = new Group();
            this.currentGroup.id = attributes.getValue("id");
        }
        else if (qName.equalsIgnoreCase("capability") && (attributes.getLength() > 0)
            && (this.currentGroup != null))
        {
            this.currentGroup.addAttribute(attributes.getValue("name"),
                attributes.getValue("value"));
        }
    }
    public void endElement(String uri, String localName, String qName)
    {
        if (qName.equalsIgnoreCase("device") && (this.currentDevice != null))
        {
            this.deviceList.add(this.currentDevice);
            this.currentDevice = null;
        }
        else if (qName.equalsIgnoreCase("group") && (this.currentGroup != null))
        {
            this.currentDevice.addGroup(this.currentGroup);
            this.currentGroup = null;
        }
    }
    private boolean isValidDevice(String userAgent)
    {
        boolean status = (userAgent != null) && (userAgent.toLowerCase().indexOf(this.keyword) > -1);
        if (status)
            System.out.println("Comparing [" + status + "] " + userAgent);
        return (status);
    }
    public static class Device
    {
        public String id;
        public String userAgent;
        public String fallBack;
        public boolean actualDeviceRoot;
        public Vector<Group> group;

        public void addGroup(Group group)
        {
            this.group = (this.group == null)
                ? new Vector<Group>()
                : this.group;
            this.group.add(group);
        }

        public String toString()
        {
            StringBuilder stringBuilder = new StringBuilder();

            stringBuilder.append("Id: " + this.id + "\n");
            stringBuilder.append("UserAgent: " + this.userAgent + "\n");
            stringBuilder.append("FallBack: " + this.fallBack + "\n");
            stringBuilder.append("Actual Device Root: " + this.actualDeviceRoot + "\n");

            if (this.group != null)
            {
                for(int i = 0; i < this.group.size(); i++)
                    stringBuilder.append(this.group.elementAt(i));
            }
            return (stringBuilder.toString());
        }
    }
}
/** Wurfl.xml is well organised xml, through this class it is organizing the capabilities information as in the xml file */
```

```java
public static class Group
{
    public String id;
    public Hashtable<String, String> attributes = null;

    public void addAttribute(String key, String value)
    {
        this.attributes = (this.attributes == null)
            ? new Hashtable<String, String>()
            : this.attributes;
        this.attributes.put(key, value);
    }
    public String toString()
    {
        StringBuilder stringBuilder = new StringBuilder();

        if (this.attributes != null)
        {
            stringBuilder.append("  [" + this.id + "] \n");
            for (String key : this.attributes.keySet())
            {
                stringBuilder.append("  " + key + ": "
                        + this.attributes.get(key) + "\n");
            }
        }
        stringBuilder.append("\n");
        return (stringBuilder.toString());
    }
}
}
```

III.   Example of Collected device capabalities of a desktop and a tablet from WURFL.XML by using WURFL Java API

Capabilities information retrieved from a desktop	Capabilities retrieved from a Nexus 7 Tablet
callericon: false	callericon: false
amr: false	amr: false
mms_wml: false	mms_wml: false
Screen Height: 400	Screen Height: 60
Screen Width: 400	Screen Width: 40
voicexml: false	voicexml: false
sckl_vcard: null	sckl_vcard: null
html_wi_imode_compact_generic: false	html_wi_imode_compact_generic: false
xhtml_document_title_support: true	xhtml_document_title_support: true
nokia_series: 0	nokia_series: 0
access_key_support: false	access_key_support: false
xhtml_support_wml_namespace: null	xhtml_support_wml_namespace: null
PDF Support: true	PDF Support: true
connectionless_service_indication: false	connectionless_service_indication: false
oma_v_1_0_separate_deliveryString: null	oma_v_1_0_separate_deliveryString: null
xhtml_support_wml2_namespace: false	xhtml_support_wml2_namespace: false
: false	: false
xhtml_select_as_dropdown: true	xhtml_select_as_dropdown: false
xhtml_select_as_dropdown: true	xhtml_select_as_dropdown: false
times_square_mode_support: false	times_square_mode_support: false
ajax_xhr_type: standard	ajax_xhr_type: standard
ringtone_midi_monophonic: false	ringtone_midi_monophonic: false
mms_vcard: false	mms_vcard: false
pdf_support: true	pdf_support: true
devId: google_chrome_2	devId: generic_android
picture_jpg: false	picture_jpg: false
accept_third_party_cookie: true	accept_third_party_cookie: true
cookieSupport: true	cookieSupport: true
picturemessage: false	picturemessage: false
mobile_browser_version:	mobile_browser_version:
ems_odi: false	ems_odi: false
chtml_can_display_images_and_text_on_same_line: false	chtml_can_display_images_and_text_on_same_line: false
midi_monophonic: false	midi_monophonic: false
transparent_png_alpha: false	transparent_png_alpha: false
picture_inline_size_limit: 0	picture_inline_size_limit: 0
max_deck_size: 100000	max_deck_size: 65535
built_in_recorder: false	built_in_recorder: false
wizards_recommended: false	wizards_recommended: false
operatorlogo: false	operatorlogo: false
largeoperatorlogo: false	largeoperatorlogo: false
transcoder_ua_header: user-agent	transcoder_ua_header: user-agent
wml__String: null	wml__String: null
xhtml_format_as_attribute: false	xhtml_format_as_attribute: false
nokia_edition: 0	nokia_edition: 0
picture_preferred_height: 0	picture_preferred_height: 0
proportional_font: false	proportional_font: false
midi_polyphonic: false	midi_polyphonic: false
html_wi_imode_html_4: false	html_wi_imode_html_4: false
handheldfriendly: false	handheldfriendly: false
qcelp: false	qcelp: false
receiver: false	receiver: false
html_wi_oma_xhtmlmp: null	html_wi_oma_xhtmlmp: null
connectionoriented_unconfirmed_service_indication: false	connectionoriented_unconfirmed_service_indication: false
physical_screen_width: 400	physical_screen_width: 40
emptyok: false	emptyok: false
mms_rmf: false	mms_rmf: false

C   Source code to use Object Oriented Database

    I.   Configuring OODB

```java
/** This servlet configures the Object oriented database by creating ServletContext object */

import javax.servlet.ServletContext;
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;

import com.db4o.Db4o;
import com.db4o.ObjectServer;

public class Db4oServletContextListener implements ServletContextListener {
        public static final String KEY_DB4O_FILE_NAME = "db4oFileName";

        public static final String KEY_DB4O_SERVER = "db4oServer";

        private ObjectServer server = null;
/** database.db4o file is created in WEB-INF/db folder in the project and context listener places in context of the servlet, so that
every other servlet can access the information */
        @Override
        public void contextInitialized(ServletContextEvent event) {
                close();
                ServletContext context = event.getServletContext();
                String filePath = context.getRealPath("WEB-INF/db/"+ context.getInitParameter(KEY_DB4O_FILE_NAME));
                server = Db4o.openServer(filePath, 0);
                context.setAttribute(KEY_DB4O_SERVER, server);
                context.log("db4o startup on " + filePath);
                }

        public void contextDestroyed(ServletContextEvent event) {
                ServletContext context = event.getServletContext();
                context.removeAttribute(KEY_DB4O_SERVER);
                close();
                context.log("db40 Shutdown");
        }

        private void close() {
                if (server != null) {
                        server.close();
                }
                server = null;

        } }
```

II.   Storing collected device capabilities in OODB

```java
public class StoreCapabilities extends HttpServlet {
        static final long serialVersionUID = 1L;

        /**
         * source to store the capabilities list in Object Oriented Database
         */
        protected void service(HttpServletRequest request,
                                HttpServletResponse response) throws ServletException, IOException {
                // TODO Auto-generated method stub

                PrintWriter out = response.getWriter();

                HttpSession session = request.getSession();

                // Registering Servlet Context
                ServletContext context = getServletContext();
                ObjectServer server = (ObjectServer) context.getAttribute("db4oServer");
                // Opening client
                ObjectContainer db = server.openClient();

                String callericon = (String) session.getAttribute("callericon");
.. /**
         * many capabilities more than 150 are listed in the original source
         */
CapabilitiesVO cap = new CapabilitiesVO();

                cap.setCallericon(callericon);
                cap.setAmr(amr);
                cap.setMms_wmlc(mms_wml);
…

      db.store(cap);
                db.close();
}


}
```

D Most important part of source code for identification of device (Phase 1)

I. Class name : DeviceRequestProcessor.java

```java
package com.hs.service.wurfl;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import net.sourceforge.wurfl.core.Device;
import net.sourceforge.wurfl.core.MarkUp;
import net.sourceforge.wurfl.core.WURFLHolder;
import net.sourceforge.wurfl.core.WURFLManager;
import net.sourceforge.wurfl.core.web.WurflWebConstants;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

/**
 *
 * Device Request Processor class will handles the users requests, when
 * index.jsp page is requested by the user, it redirects to
 * DeviceRequestProcessor class and navigates the users to their respective jsp files.
 *
 */
public class DeviceRequestProcessor extends HttpServlet implements
                    WurflWebConstants {

        /** Serial */
        private static final long serialVersionUID = 10L;

        private static final String XHTML_ADV = "xhtmladv.jsp";
        private static final String XHTML_SIMPLE = "xhtmlmp.jsp";
        private static final String CHTML = "chtml.jsp";
        private static final String WML = "wml.jsp";

        /** Logger */
        private final Logger log = LoggerFactory.getLogger(getClass());

        /**
         * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
         * methods.
         *
         * @param request
         *        servlet request
         * @param response
         *        servlet response
         */
        protected void processRequest(HttpServletRequest request,
                        HttpServletResponse response) throws ServletException, IOException {

                WURFLHolder holder = (WURFLHolder) getServletContext().getAttribute(
                                WURFL_HOLDER_KEY);

                WURFLManager mgr = holder.getWURFLManager();
                Device device = mgr.getDeviceForRequest(request);

                log.debug("Device: " + device.getId());
                log.debug("Capability: " + device.getCapability("preferred_markup"));

                MarkUp markUp = device.getMarkUp();
```

```java
                System.out.println("Mark up is" + markUp);
                String jspView = null;

                if (MarkUp.XHTML_ADVANCED.equals(markUp)) {
                        jspView = XHTML_ADV;
                } else if (MarkUp.XHTML_SIMPLE.equals(markUp)) {
                        jspView = XHTML_SIMPLE;
                } else if (MarkUp.CHTML.equals(markUp)) {
                        jspView = CHTML;
                } else if (MarkUp.WML.equals(markUp)) {
                        jspView = WML;
                }

                log.info("View: " + jspView);

                // MIME type is decided by JSP. Only in case of XHTML we will need to
                // multi-serve
                // i.e text/html vs application/xml+xhtml vs
                // application/vnd.wap.xml+xhtml
                if (markUp == MarkUp.XHTML_ADVANCED || markUp == MarkUp.XHTML_SIMPLE) {
                        String contentType = "text/html";
                        try {
                                contentType = device
                                                .getCapability("xhtmlmp_preferred_mime_type");
                        } catch (Exception e) {
                                throw new RuntimeException(
                                                "Something is seriously wrong with your WURFL:"
                                                                + e.getLocalizedMessage(), e);
                        }

                        request.setAttribute("contentType", contentType);
                        log.debug("ContentType: " + contentType);
                }

                request.setAttribute("markUp", markUp);
                request.setAttribute("device", device);

                request.getRequestDispatcher(jspView).forward(request,
                                response);
        }

        protected void doGet(HttpServletRequest request,
                        HttpServletResponse response) throws ServletException, IOException {
                processRequest(request, response);
        }

        protected void doPost(HttpServletRequest request,
                        HttpServletResponse response) throws ServletException, IOException {
                processRequest(request, response);
        }}
```

II.   Interface name : WurflWebConstants.java

```java
/**
 * Copyright (c) 2011 ScientiaMobile Inc.
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as
 * published by the Free Software Foundation, either version 3 of the
 * License, or (at your option) any later version.
 *
 * Refer to the COPYING file distributed with this package.
 */

package net.sourceforge.wurfl.core.web;

import net.sourceforge.wurfl.core.WURFLHolder;

/**
 * Default web constants.
 * @author Ermanno Franco
 */
public interface WurflWebConstants {


    /**
     * Servlet context parameter contains the name of WURFLHolder parameter
     */
    String WURFL_HOLDER_KEY_PARAM = "wurflHolderKey";

    /**
     * Default Servlet Context attribute key for WURFLHolder: net.sourceforge.wurfl.core.WURFLHolder
     */
    String WURFL_HOLDER_KEY = WURFLHolder.class.getName();

    /**
     * ServletContext wurfl main file parameter
     */
    String WURFL = "wurfl";


    /**
     * Default location for wurfl root file, it is /WEB-INF/wurfl.zip
     */
    String WURFL_DEFAULT_LOCATION = "/WEB-INF/wurfl.zip";


}
```

```java
package com.hs.service.db4o.servlets;

import javax.servlet.ServletContext;
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;

import com.db4o.Db4o;
import com.db4o.ObjectServer;

/**
 * Db4oServletContextListener
 */

public class Db4oServletContextListener implements ServletContextListener {

        public static final String KEY_DB4O_FILE_NAME = "db4oFileName";
        public static final String KEY_DB4O_SERVER = "db4oServer";
        private ObjectServer server = null;

/**
 * contextInitialized method is called automatically by the application        *server when the application is initialized. A
Db4o server is opened, and a   *reference to this server is stored as an attribute of the servlet context.
 */

        @Override
        public void contextInitialized(ServletContextEvent event) {
                close();
                ServletContext context = event.getServletContext();
                String          filePath          =          context.getRealPath("WEB-INF/db/"+          con-
text.getInitParameter(KEY_DB4O_FILE_NAME));
                server = Db4o.openServer(filePath, 0);
                context.setAttribute(KEY_DB4O_SERVER, server);
                context.log("db4o startup on " + filePath);
                }
/**
 * contextDestroyed method is called by the application
*server when the application is destroyed. A Db4o server is closed.
 */

        public void contextDestroyed(ServletContextEvent event) {
                ServletContext context = event.getServletContext();
                context.removeAttribute(KEY_DB4O_SERVER);
                close();
                context.log("db4o shutdown");
        }

        private void close() {
                if (server != null) {
                        server.close();
                }
                server = null;
        }

}
```

```java
package com.hs.service.db4o.servlets;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.db4o.ObjectContainer;
import com.db4o.ObjectServer;
import com.db4o.ObjectSet;
import com.hs.service.db4o.vo.*;

/**
 * Servlet implementation class RetrieveCapabilities
 */
public class RetrieveCapabilities extends HttpServlet {
        private static final long serialVersionUID = 1L;

        /**
         * @see HttpServlet#HttpServlet()
         */
        public RetrieveCapabilities() {
                super();
                // TODO Auto-generated constructor stub
        }

/**
 * RetrieveCapabilities servlet obtains Db4oServer from the Servlet context using the key "db4oServer", which gets
 the server reference and opens the server.
 */
        protected void service(HttpServletRequest request,
                               HttpServletResponse response) throws ServletException, IOException {
                int countObjects = 0;
                response.setContentType("text/html;charset=UTF-8");
                PrintWriter out = response.getWriter();
                ServletContext context = getServletContext();
                ObjectServer server = (ObjectServer) context.getAttribute("db4oServer");
                ObjectContainer db = server.openClient();
        /**
         * queryByExample() method returns the ObjectSet result
         */
                ObjectSet results = db.queryByExample(CapabilitiesVO.class);
                db.close();
                out.println("<html>");
                out.println("<head>");
                out.println("<title>Servlet RetrieveServlet</title>");
                out.println("</head>");
                out.println("<body>");
                out.println("<h1>Capabilities objects retrieved</h1>");
                while (results.hasNext()) {
                        CapabilitiesVO cap = (CapabilitiesVO) results.next();
                        out.println("<b>Object : </b>" + (countObjects++));
                        String callericon = cap.getCallericon();
                        String amr = cap.getAmr();
                        String mms_wml = cap.getMms_wmlc();
        ...
                        out.println("<p>callericon: " + callericon + "<br/>");
                        out.println("amr: " + amr + "</p>");
```

```java
                out.println("mms_wml: " + mms_wml + "</p>");
...
            }
            out.println("<a href=\"/ContentService/index.jsp\">Click here to go home</a>");
            out.println("</body>");
            out.println("</html>");
            out.close();
        }

}
```

V.   Servlet name : StoreCapabilities.java

```java
package com.hs.service.db4o.servlets;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.db4o.ObjectContainer;
import com.db4o.ObjectServer;
import com.db4o.ObjectSet;
import com.hs.service.db4o.vo.*;

/**
 * Servlet implementation class StoreCapabilities
 */

public class StoreCapabilities extends HttpServlet {
        static final long serialVersionUID = 1L;

        /**
         * @see HttpServlet#HttpServlet()
         */
        public StoreCapabilities() {
                super();
                // TODO Auto-generated constructor stub
        }

/**
* StoreCapabilities servlet obtains Db4oServer from the Servlet context     using the key "db4oServer", which gets
the server reference and opens the server.
*/
        protected void service(HttpServletRequest request,
                                HttpServletResponse response) throws ServletException, IOException {

                PrintWriter out = response.getWriter();
                HttpSession session = request.getSession();
                // Registering Servlet Context
                ServletContext context = getServletContext();
                ObjectServer server = (ObjectServer) context.getAttribute("db4oServer");
                ObjectContainer db = server.openClient();

                String callericon = (String) session.getAttribute("callericon");
                String amr = (String) session.getAttribute("amr");
                String mms_wml = (String) session.getAttribute("mms_wml");
        ...
                CapabilitiesVO cap = new CapabilitiesVO();
                cap.setCallericon(callericon);
                cap.setAmr(amr);
                cap.setMms_wmlc(mms_wml);
        ...
                db.store(cap);
                db.close();

        }
}
```

145

E   Most important part of source code for preparing generalized content (Phase 2)

I.   Class Name: ManageFiles.java

```java
package com.hs.service;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Iterator;

import com.hs.vo.ContentVO;
import com.hs.vo.Generalized3DVO;
import com.hs.vo.GeneralizedAttachmentFile;
import com.hs.vo.GeneralizedAttachmentLinks;
import com.hs.vo.GeneralizedAttachmentType;
import com.hs.vo.GeneralizedDocVO;
import com.hs.vo.OriginalAttachmentFile;
import com.hs.vo.OriginalAttachmentLinks;
import com.hs.vo.OriginalAttachmentType;
import com.hs.vo.OriginalContentVO;

/**
* ManageFiles class is responsible for managing attached files by author.
*/

public class ManageFiles {

        SimpleDateFormat ft = new SimpleDateFormat("yyyy.MM.dd 'at' HH:mm:ss");
        String date = ft.format(new Date());

        public ManageFiles() {

        }
/**
* ManageFiles constructor collects the MULTIPART_FORM_DATA and Original files saved locations, this information
is stored it in a ContentVO (Model) object inorder to persist it in Database.
*/

        public ManageFiles(ArrayList<String> contentTypes,
                        ArrayList<String> fileNames, ArrayList<String> contentLinks,
                        ArrayList<String> feildNames, ArrayList<String> feildValues) {

            ContentVO OC = new ContentVO();

            Iterator<String> itrFeilds = feildNames.iterator();
            Iterator<String> itrFileName = feildValues.iterator();


            while (itrFeilds.hasNext() && itrFileName.hasNext()) {
                    Object feildElement = itrFeilds.next();
                    Object StringElement = itrFileName.next();
                    if (feildElement.toString().equals("userName"))
                            OC.setUname(StringElement.toString());
                    else if (feildElement.equals("con_title"))
                            OC.setContentTitle(StringElement.toString());
                    else if (feildElement.equals("con_auth_name"))
                            OC.setAuthorName(StringElement.toString());
                    else if (feildElement.equals("textarea"))
                            OC.setContentInfo(StringElement.toString());
                    else if (feildElement.equals("subject"))
                            OC.setSubjectName(StringElement.toString());
                    System.out.println(feildElement + ", " + StringElement);
            }

            Iterator<String> fIterator = fileNames.iterator();
```

```java
        Iterator<String> cIterator = contentLinks.iterator();
        Iterator<String> conIterator = contentTypes.iterator();

        ArrayList set1 = new ArrayList();
        ArrayList set2 = new ArrayList();
        ArrayList set3 = new ArrayList();

        while (fIterator.hasNext()) {
                Object fElement = fIterator.next();
                set1.add(new OriginalAttachmentFile(fElement.toString()));
        }
        while (cIterator.hasNext()) {
                Object cElement = cIterator.next();
                set2.add(new OriginalAttachmentLinks(cElement.toString()));
        }
        while (conIterator.hasNext()) {
                Object contElement = conIterator.next();
                set3.add(new OriginalAttachmentType(contElement.toString()));
        }

        Generalized3DVO g3DVO = new Generalized3DVO();
        GeneralizedDocVO gDVO = new GeneralizedDocVO();
        OriginalContentVO oCVO = new OriginalContentVO();
        /*
         * ContentGeneralization class generalizes the attached files. *For example if attached content
type is 3D format, then it converts into *generalized xml format to ".dae". if attachment type is document format
*then generalized format be ".xhtml".
         */

        ContentGeneralization cg = new ContentGeneralization(contentTypes, fileNames, contentLinks);

        g3DVO = cg.fetchGeneralized3DContentDetails();
        gDVO = cg.fetchGeneralizedDocContentDetails();
        oCVO = cg.fetchOriginalContentDetails();

        ArrayList<String> generalized3DLinks = g3DVO.getGeneralized3DContentLinks();
        ArrayList<String> generalized3DTypes = g3DVO.getGeneralized3DContentTypes();
        ArrayList<String> generalized3DFNames = g3DVO.getGeneralized3DFileNames();

        ArrayList<String> generalizedDOCLinks = gDVO.getGeneralizedDocContentLinks();
        ArrayList<String> generalizedDOCTypes = gDVO.getGeneralizedDocContentTypes();
        ArrayList<String> generalizedDOCFNames = gDVO.getGeneralizedDocFileNames();

        ArrayList<String> originalContentTypes = oCVO.getOriginalContentTypes();
        ArrayList<String> originalContentLinks = oCVO.getOriginalContentLinks();
        ArrayList<String> originalContentNames = oCVO.getOriginalContentNames();

        generalized3DFNames.addAll(generalizedDOCFNames);
        generalized3DLinks.addAll(generalizedDOCLinks);
        generalized3DTypes.addAll(generalizedDOCTypes);

        originalContentLinks.addAll(generalized3DLinks);
        originalContentNames.addAll(generalized3DFNames);
        originalContentTypes.addAll(generalized3DTypes);

        Iterator<String> g3DLIterator = originalContentLinks.iterator();
        Iterator<String> g3DCIterator = originalContentTypes.iterator();
        Iterator<String> g3dConIterator = originalContentNames.iterator();

        ArrayList set4 = new ArrayList();
        ArrayList set5 = new ArrayList();
        ArrayList set6 = new ArrayList();

        while (g3DLIterator.hasNext()) {
                Object lElement = g3DLIterator.next();
```

```java
                set4.add(new GeneralizedAttachmentLinks(lElement.toString()));
        }
        while (g3DCIterator.hasNext()) {
                Object tElement = g3DCIterator.next();
                set5.add(new GeneralizedAttachmentType(tElement.toString()));
        }
        while (g3dConIterator.hasNext()) {
                Object contElement = g3dConIterator.next();
                set6.add(new GeneralizedAttachmentFile(contElement.toString()));
        }

        /* Content class is responsible for persisting the Original and Generalized meta information in a
database. */

        Content content = new Content();
        Integer conID1 = content.addContentInfo(OC.getUname(), OC.getAuthorName(),
                        OC.getContentTitle(), date, OC.getContentInfo(),
                        OC.getSubjectName(), set1, set2, set3, set4, set5, set6);
        /* CocoonSiteMapService class is responsible for updating generalized document information
(e.g. xml location, serialization type) in a SiteMap */
        CocoonSiteMapService cSMS = new CocoonSiteMapService();
        System.out.println("Sitemap Call status: "+cSMS.getSiteMapUpdate());

    }

}
```

II.   Class Name: ContentGeneralization.java

```java
package com.hs.service;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Iterator;

import com.hs.vo.Generalized3DVO;
import com.hs.vo.GeneralizedDocVO;
import com.hs.vo.OriginalContentVO;


/* ContentGeneralization class is responsible for generalizing 3D and document type data into XML or XML-extended
format. For this process we used Blender and Apache Tika API's. Blender is responsible for converting 3D formatted
data, whereas Apache Tika is responsible for converting document type data */

public class ContentGeneralization {


        ArrayList<String> generalized3DFileNames = new ArrayList<>();
        ArrayList<String> generalized3DContentLinks = new ArrayList<>();
        ArrayList<String> generalized3DContentTypes = new ArrayList<>();


        ArrayList<String> generalizedDocFileNames = new ArrayList<>();
        ArrayList<String> generalizedDocContentLinks = new ArrayList<>();
        ArrayList<String> generalizedDocContentTypes = new ArrayList<>();

        ArrayList<String> originalContentTypes = new ArrayList<>();
        ArrayList<String> originalContentLinks = new ArrayList<>();
        ArrayList<String> originalContentNames = new ArrayList<>();

        public ContentGeneralization() {

        }

        public ContentGeneralization(ArrayList<String> contentTypes,
                        ArrayList<String> fileNames, ArrayList<String> contentLinks){

                /* Generalizing content records */

                Iterator<String> fNames = fileNames.iterator();
                Iterator<String> cLinks = contentLinks.iterator();
                Iterator<String> cTypes = contentTypes.iterator();


                ArrayList<String> blendValues = new ArrayList<>();
                ArrayList<String> tikaValues = new ArrayList<>();

                while (fNames.hasNext() || cLinks.hasNext() || cTypes.hasNext()) {
                        Object fileNameElement = fNames.next();
                        Object linkElement = cLinks.next();
                        Object conTypeElement = cTypes.next();

                        if (conTypeElement.equals("application/x-blender")||
                                        conTypeElement.equals("application/vnd.ms-pki.stl")
                                        || conTypeElement.equals("model/x3d+xml")
                                        || conTypeElement.equals("application/octet-stream")
                                        || conTypeElement.equals("application/x-3ds")
                                        || conTypeElement.equals("application/blender")
                                        || conTypeElement.equals("application/sla")
                                        || conTypeElement.equals("application/vnd.sketchup.skp")
                                        || conTypeElement.equals("model/u3d")
```

```java
                                    || conTypeElement.equals("model/vnd.collada+xml")){
                                System.out.println("method call");
                                this.blendConvert(linkElement.toString(), fileNameElement.toString());
                            }
                        else if (conTypeElement.equals("application/pdf")
                                        || conTypeElement.equals("application/vnd.ms-excel")
                                        || conTypeElement.equals("application/vnd.ms-powerpoint")
                                        || conTypeElement
                                            .equals("application/vnd.openxmlformats-
officedocument.wordprocessingml.document")
                                        || conTypeElement

        .equals("application/vnd.oasis.opendocument.text")){
                                this.tikaDocumentConvert(linkElement.toString(),              fileNameEle-
ment.toString());
                }else{
                                String genLocation = "ConentService/OriginalFiles/"+fileNameElement.toString();
                        originalContentLinks.add(genLocation);
                        originalContentNames.add(fileNameElement.toString());
                        originalContentTypes.add(conTypeElement.toString());
                        }
                        }

        }

        public void blendConvert(String fileLoc, String fName) {
                /*
                 *blendConvert() method converts some of the 3D models into generalized ".DAE format"
                 */
                String aLocation = fileLoc;
                String       bLocation     =     "E:/tomcat/webapps/ThreeJSWeb/webFiles/models/collada/"     +
fName+".dae";

                String genLocation = "ThreeJSWeb/webFiles/webgl_loader_collada.jsp?model=" + fName+".dae";
                ArrayList<String> returnValues = new ArrayList<>();

                System.out.println("blend Location \t" + bLocation);
                String st = "--background --python E:/eclipse/convert_blend_to_obj.py -- "
                                + aLocation + " " + bLocation;
                String[] cmd = { "cmd.exe", "/c", "blender " + st };
                System.out.println("Second Command" + cmd[2]);

                generalized3DContentLinks.add(genLocation);
                generalized3DFileNames.add(fName+".dae");
                generalized3DContentTypes.add("model/vnd.collada+xml");

                Process p;
                try {
                        p = Runtime.getRuntime().exec(cmd);
                        BufferedReader input = new BufferedReader(new InputStreamReader(
                                        p.getInputStream()));
                        String temp = "Sucess";
                        while ((temp = input.readLine()) != null)
                                System.out.println(temp);
                        input.close();
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }

        public void tikaDocumentConvert(String fileLoc, String fileName) {
                /*
                 *tikaDocunmentConvert() method converts some of the document formatted files into general-
ized .xml or .xhtml format"
                 */
```

```java
                    String aLocation = fileLoc;
                    String  bLocation  =  "E:/tomcat/webapps/CocoonAPI/ContentManagement/uploads/"  +  file-
Name+".xml";

                    String genLocation = "CocoonAPI/ContentManagement/"+ fileName+".xml";
                    ArrayList<String> returnValues = new ArrayList<>();

                    System.out.println("tika Location \t" + bLocation);
                    String st = " java -jar E:/downloads/tika/tika-app-1.4.jar -x "
                                    + aLocation + " > " + bLocation;
                    String[] cmd = { "cmd.exe", "/c " + st };
                    System.out.println(cmd[1]);

                    generalizedDocContentLinks.add(genLocation);
                    generalizedDocFileNames.add(fileName + ".xml");
                    generalizedDocContentTypes.add("application/xhtml+xml");

                    Process p;
                    try {
                            p = Runtime.getRuntime().exec(cmd);
                            BufferedReader input = new BufferedReader(new InputStreamReader(
                                            p.getInputStream()));
                            String temp = "Sucess";
                            while ((temp = input.readLine()) != null)
                                    System.out.println(temp);
                            input.close();
                    } catch (IOException e) {
                            e.printStackTrace();
                    }
            }

    /*    fetchGeneralizedDocContentDetails(),fetchOriginalContentDetails(),and                        fetchGeneral-
ized3DContentDetails() methods are responsible for returning Generalized and non-converted (i.e. Original files)
information to the ManageFiles Class.
        */

        public GeneralizedDocVO fetchGeneralizedDocContentDetails(){

                GeneralizedDocVO gDVO = new GeneralizedDocVO();
                gDVO.setGeneralizedDocContentLinks(generalizedDocContentLinks);
                gDVO.setGeneralizedDocContentTypes(generalizedDocContentTypes);
                gDVO.setGeneralizedDocFileNames(generalizedDocFileNames);

                return gDVO;
        }

        public OriginalContentVO fetchOriginalContentDetails() {
                OriginalContentVO oCVO = new OriginalContentVO();
                oCVO.setOriginalContentLinks(originalContentLinks);
                oCVO.setOriginalContentNames(originalContentNames);
                oCVO.setOriginalContentTypes(originalContentTypes);

                return oCVO;
        }
        public Generalized3DVO fetchGeneralized3DContentDetails(){

                Generalized3DVO g3DVO = new Generalized3DVO();
                g3DVO.setGeneralized3DContentLinks(generalized3DContentLinks);
                g3DVO.setGeneralized3DContentTypes(generalized3DContentTypes);
                g3DVO.setGeneralized3DFileNames(generalized3DFileNames);

                return g3DVO;
        }

}
```

```java
package com.hs.service;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import org.hibernate.HibernateException;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.Transaction;

import com.hs.util.HibernateUtil;
import com.hs.vo.ContentVO;

    /*      Content Class implements the ContentOperations interface.
         */


public class Content implements ContentOperations{

                protected int contentName;

/*        deleteContent() method delete the content information from the mysql database
          */

                @Override
                public int deleteContent(String contentTitle) {
                        // TODO Auto-generated method stub
                        Session hsession = HibernateUtil.getSessionFactory().openSession();
                        org.hibernate.Transaction tx = hsession.beginTransaction();
                        Query query = hsession
                                        .createQuery("delete com.hs.vo.ContentVO where content_title = '"
                                                        + contentTitle + "' ");
                        int result = query.executeUpdate();
                        tx.commit();
                        hsession.close();
                        System.out.println("Query Fired" + contentTitle);
                        return result;
                }

/*        addContentInfo() persist the form information and attachments information to the mysql database.
          */

                @Override
                public int addContentInfo(String uname, String authorName,
                                String contentTitle, String submittedDate, String contentInfo,
                                String subjectName, ArrayList<String> set1, ArrayList<String> set2,
                                ArrayList<String> set3, ArrayList<String> set4, ArrayList<String> set5,
                                ArrayList<String> set6) {
                        Session session = HibernateUtil.getSessionFactory().openSession();
                        Transaction tx = null;
                        Integer contentID = null;
                        try {
                                tx = session.beginTransaction();
                                ContentVO content = new ContentVO(uname,
                                                authorName, contentTitle, submittedDate, contentInfo,
                                                subjectName);
                                content.setFileName(set1);
                                content.setContentLink(set2);
                                content.setContentType(set3);
                                content.setGenFileName(set6);
                                content.setGenContentLink(set4);
```

```java
                                content.setGenContentType(set5);
                                contentID = (Integer) session.save(content);
                                tx.commit();
                        } catch (HibernateException e) {
                                if (tx != null)
                                        tx.rollback();
                                e.printStackTrace();
                        } finally {
                                session.close();
                        }
                        return contentID;
                }

/*      updateContentInfo() method updates the information added by the content author.
        */

                @Override
                public void updateContentInfo() {

                }

/*      listContentInfo() method returns the ArrayList of type ContentVO to the PopulateTable Class.
        */

                @Override
                public ArrayList<ContentVO> listContentInfo() {
                        ArrayList<ContentVO> contentList = new ArrayList<ContentVO>();

                        Session session = HibernateUtil.getSessionFactory().openSession();
                        Transaction tx = null;
                        tx = session.beginTransaction();
                        List contentInfo = session.createQuery("FROM ContentVO").list();
                        System.out.println(contentInfo.size());

                        for (Iterator iterator1 = contentInfo.iterator(); iterator1.hasNext();) {
                                ContentVO OC = (ContentVO) iterator1.next();
                                ContentVO cVO = new ContentVO();

                                cVO.setContentId(OC.getContentId());
                                cVO.setAuthorName(OC.getAuthorName());
                                cVO.setContentInfo(OC.getContentInfo());
                                cVO.setContentTitle(OC.getContentTitle());
                                cVO.setSubjectName(OC.getSubjectName());
                                cVO.setSubmittedDate(OC.getSubmittedDate());
                                cVO.setGenContentLink(OC.getGenContentLink());

                                contentList.add(cVO);
                        }

                        return contentList;
                }

}
```

IV.   Interface Name: ContentOperations.java

```java
package com.hs.service;

import java.util.ArrayList;
import com.hs.vo.ContentVO;

/*        ContentOperations performs operations on content
By
1) adding, 2) deleting 3) updating and 4) displaying and many more if needed are added here.
          */

public interface ContentOperations {


        int deleteContent(String contentTitle);

    int addContentInfo(String uname, String authorName,
                        String contentTitle, String submittedDate, String contentInfo,
                        String subjectName, ArrayList<String> set1, ArrayList<String> set2,
                        ArrayList<String> set3, ArrayList<String> set4, ArrayList<String> set5,
                        ArrayList<String> set6);



    void updateContentInfo();


        ArrayList<ContentVO> listContentInfo();



}
```

V.   Class Name: CocoonSiteMapService.java

```java
package com.hs.service;

import java.util.Iterator;
import java.util.List;

import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.Transaction;
import com.hs.util.HibernateUtil;
import com.hs.util.SiteMapUpdate;
import com.hs.vo.ContentVO;
import com.hs.vo.GeneralizedAttachmentFile;
import com.hs.vo.GeneralizedAttachmentLinks;
import com.hs.vo.GeneralizedAttachmentType;


/*      CocoonSiteMapService class fetches the last updated content information from the content_information
table for adding the last generalized document information to the SiteMap.xamp in the Cocoon API.
        */
public class CocoonSiteMapService {
        public boolean getSiteMapUpdate(){
        SiteMapUpdate su = new SiteMapUpdate();
        Session session = HibernateUtil.getSessionFactory().openSession();
        Transaction tx = null;
        try {
                tx = session.beginTransaction();
                List contentInfo = session
                                .createQuery(
                "FROM ContentVO as OC WHERE contentId in (select MAX(OC.contentId) from OC)")
                                .list();
                for (Iterator iterator1 = contentInfo.iterator(); iterator1
                                .hasNext();) {
                        ContentVO OC = (ContentVO) iterator1.next();
                        List conLinks = OC.getGenContentLink();
                        List conTypes = OC.getGenContentType();
                        List coninfo = OC.getGenFileName();
                        for (Iterator iterator2 = conLinks.iterator(), iterator3 = conTypes
                                        .iterator(), iterator4 = coninfo.iterator(); iterator2
                                        .hasNext()
                                        || iterator3.hasNext()
                                        || iterator4.hasNext();) {
                                GeneralizedAttachmentLinks conLink = (GeneralizedAttachmentLinks) itera-
tor2
                                                .next();
                                GeneralizedAttachmentType conType = (GeneralizedAttachmentType) itera-
tor3
                                                .next();
                                GeneralizedAttachmentFile conFile = (GeneralizedAttachmentFile) iterator4
                                                .next();
                                if (conType.getContentType()
                                                .equals("application/xhtml+xml")) {
                                        su.createNewItem(conFile.getFileName());
                                }
                        }
                }
                tx.commit();
        } catch (HibernateException e) {
                if (tx != null)
                        tx.rollback();
                e.printStackTrace();
        } finally {
                session.close();
        }
        return true;
        }}
```

F   Most important part of source code for transformation of content (Phase 3)

I.   Class Name: PopulateTable.java

```java
package com.hs.service;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.google.gson.Gson;
import com.google.gson.JsonArray;
import com.google.gson.JsonElement;
import com.google.gson.reflect.TypeToken;
import com.hs.vo.ContentVO;

/*       PopulateTable class gets the list of type ContentVO object from the Content class by using method listCon-
tentInfo(), this list object then added to the Gson object to convert it into a JSON format. JSON formated object is
returned to the JSP page.
         */

@WebServlet("/PopulateTable")
public class PopulateTable extends HttpServlet {
         private static final long serialVersionUID = 1L;

         public PopulateTable() {

         }

         protected void doGet(HttpServletRequest request,
                             HttpServletResponse response) throws ServletException, IOException {
                 request.setAttribute("org.apache.catalina.ASYNC_SUPPORTED", true);

                 final PrintWriter writer = response.getWriter();
                 Content content = new Content();
                 List contentL = content.listContentInfo();
                 Gson gson = new Gson();
                 JsonElement element = gson.toJsonTree(contentL,
                                 new TypeToken<List<ContentVO>>() {
                                 }.getType());
                 JsonArray jsonArray = element.getAsJsonArray();
                 response.setContentType("application/json");
                 response.getWriter().print(jsonArray);
         }

         protected void doPost(HttpServletRequest request,
                             HttpServletResponse response) throws ServletException, IOException {
         }
}
```

G   Example of a OBJ file containing a 3D model of a ball followed by material file used DCC- 3ds Max

OBJ FILE NAMED b_ball.obj having file size 64 KB, compressed 13 KB
```
=========================================================
# 3ds Max Wavefront OBJ Exporter v0.97b - (c)2007 guruware
# File Created: 13.06.2014 13:24:00
# File size 64 KB
# Compressed file size 13 KB

mtllib b_ball.mtl

#
# object Sphere001
#
v  1.1819 25.1630 -4.9346 v  1.1819 24.8972 -7.6336 v  0.6554 24.8972 -7.5817
v  0.1491 24.8972 -7.4281 v  -0.3175 24.8972 -7.1787 v  -0.7265 24.8972 -6.8431
v  -1.0622 24.8972 -6.4341 v  -1.3116 24.8972 -5.9675 v  -1.4652 24.8972 -5.4611
v  -1.5171 24.8972 -4.9346 v  -1.4652 24.8972 -4.4081 v  -1.3116 24.8972 -3.9017
v  -1.0622 24.8972 -3.4351 v  -0.7265 24.8972 -3.0261 v  -0.3175 24.8972 -2.6905
v  0.1491 24.8972 -2.4411 ...................... # 482 vertices

vn 0.0000 1.0000 0.0000 vn -0.0000 0.9796 -0.2010 vn -0.0392 0.9796 -0.1971
vn -0.0769 0.9796 -0.1857 vn -0.1117 0.9796 -0.1671 vn -0.1421 0.9796 -0.1421
vn -0.1671 0.9796 -0.1117 vn -0.1857 0.9796 -0.0769 vn -0.1971 0.9796 -0.0392
vn -0.2010 0.9796 0.0000 vn -0.1971 0.9796 0.0392 vn -0.1857 0.9796 0.0769
vn -0.1671 0.9796 0.1117 vn -0.1421 0.9796 0.1421 vn -0.1117 0.9796 0.1671
vn -0.0769 0.9796 0.1857 ......................... # 482 vertex normals

vt 0.0000 1.0000 0.0000 vt 0.0000 0.9375 0.0000 vt 0.0313 0.9375 0.0000
vt 0.0313 1.0000 0.0000 vt 0.0625 0.9375 0.0000 vt 0.0625 1.0000 0.0000
vt 0.0938 0.9375 0.0000 vt 0.0938 1.0000 0.0000 vt 0.1250 0.9375 0.0000
vt 0.1250 1.0000 0.0000 vt 0.1563 0.9375 0.0000 vt 0.1563 1.0000 0.0000
vt 0.1875 0.9375 0.0000 ...................... # 559 texture coords

g Sphere001
usemtl wire_228184153
s 1
f 1/1/1 2/2/2 3/3/3 f 1/4/1 3/3/3 4/5/4 f 1/6/1 4/5/4 5/7/5 f 1/8/1 5/7/5 6/9/6
f 1/10/1 6/9/6 7/11/7 f 1/12/1 7/11/7 8/13/8 f 1/14/1 8/13/8 9/15/9
f 1/16/1 9/15/9 10/17/10 f 1/18/1 10/17/10 11/19/11 f 1/20/1 11/19/11 12/21/12
f 1/22/1 12/21/12 13/23/13 f 1/24/1 13/23/13 14/25/14 f 1/26/1 14/25/14 15/27/15
f 1/28/1 15/27/15 16/29/16 f 1/30/1 16/29/16 17/31/17 f 1/32/1 17/31/17 18/33/18
............................. # 448 polygons - 64 triangles


MATERIAL FILE NAMED b_ball.mtl having file size 1KB
===================================================
# 3ds Max Wavefront OBJ Exporter v0.97b - (c)2007 guruware
# File Created: 13.06.2014 13:24:00
# File size 1 KB (original and compressed)

newmtl wire_228184153
        Ns 32
        d 1
        Tr 0
        Tf 1 1 1
        illum 2
        Ka 0.8941 0.7216 0.6000
        Kd 0.8941 0.7216 0.6000
        Ks 0.3500 0.3500 0.3500
```

H   Example of a COLLADA file containing a 3D model of a ball in XML editor view used DCC- 3ds Max

DAE FILE NAMED b_ball.dae having file size 159 KB, compressed 28 KB
==

```xml
<?xml version="1.0" encoding="UTF-8"?>
- <COLLADA version="1.4.1" xmlns="http://www.collada.org/2005/11/COLLADASchema">
  - <asset>
    - <contributor>
        <author/>
        <authoring_tool>FBX COLLADA exporter</authoring_tool>
        <comments/>
      </contributor>
      <created>2014-06-13T11:23:12Z</created>
      <keywords/>
      <modified>2014-06-13T11:23:12Z</modified>
      <revision/>
      <subject/>
      <title/>
      <unit name="centimeter" meter="0.001000"/>
      <up_axis>Y_UP</up_axis>
    </asset>
  - <library_geometries>
      + <geometry name="Sphere001Mesh" id="Sphere001-lib">
    </library_geometries>
  - <library_animations>
    - <animation name="Sphere001" id="Sphere001-anim">
      - <animation>
        - <source id="Sphere001-Matrix-animation-input">
            <float_array id="Sphere001-Matrix-animation-input-array" count="101">
              0.000000 0.033333 0.066667 0.100000 0.133333 0.166667 0.200000
              0.233333 0.266667 0.300000 0.333333 0.366667 0.400000 0.433333
              0.466667 0.500000 0.533333 0.566667 0.600000 0.633333 0.666667
              0.700000 0.733333 0.766667 0.800000 0.833333 0.866667 0.900000
              0.933333 0.966667 1.000000 1.033333 1.066667 1.100000 1.133333
              1.166667 1.200000 1.233333 1.266667 1.300000 1.333333 1.366667
              1.400000 1.433333 1.466667 1.500000 1.533333 1.566667 1.600000
              1.633333 1.666667 1.700000 1.733333 1.766667 1.800000 1.833333
              1.866667 1.900000 1.933333 1.966667 2.000000 2.033333 2.066667
              2.100000 2.133333 2.166667 2.200000 2.233333 2.266667 2.300000
              2.333333 2.366667 2.400000 2.433333 2.466667 2.500000 2.533333
              2.566667 2.600000 2.633333 2.666667 2.700000 2.733333 2.766667
              2.800000 2.833333 2.866667 2.900000 2.933333 2.966667 3.000000
              3.033333 3.066667 3.100000 3.133333 3.166667 3.200000 3.233333
              3.266667 3.300000 3.333333</float_array>
          - <technique_common>
            - <accessor count="101" source="#Sphere001-Matrix-animation-input-
              array">
                <param name="TIME" type="float"/>
              </accessor>
            </technique_common>
          </source>

- <source id="Sphere001-Matrix-animation-output-transform">
    <float_array id="Sphere001-Matrix-animation-output-transform-array"
      count="1616"> 1.000000 0.000000 0.000000 1.181931 0.000000
      0.000000 1.000000 11.328453 0.000000 -1.000000 0.000000 -4.934601
      0.000000 0.000000 0.000000 1.000000 1.000000 0.000000 0.000000
      1.181931 0.000000 0.000000 1.000000 11.504532 0.000000 -1.000000
      0.000000 -4.934601 0.000000 0.000000 0.000000 1.000000 1.000000
      0.000000 0.000000 1.181931 0.000000 0.000000 1.000000 12.013421
```
.... # 154 lines of float_array values in between

158

```xml
                0.000000 0.000000 1.181931 0.000000 0.000000 1.000000 10.574814
                0.000000 -1.000000 0.000000 -4.934601 0.000000 0.000000 0.000000
                1.000000</float_array>
              - <technique_common>
                - <accessor count="101" stride="16" source="#Sphere001-Matrix-animation-
                  output-transform-array">
                    <param type="float4x4"/>
                  </accessor>
                </technique_common>
            </source>
          - <source id="Sphere001-Interpolations">
              <Name_array id="Sphere001-Interpolations-array" count="101"> LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR LINEAR
                LINEAR</Name_array>
              - <technique_common>
                - <accessor count="101" source="#Sphere001-Interpolations-array">
                    <param type="name"/>
                  </accessor>
                </technique_common>
            </source>
          - <sampler id="Sphere001-Matrix-animation-transform">
              <input source="#Sphere001-Matrix-animation-input" semantic="INPUT"/>
              <input source="#Sphere001-Matrix-animation-output-transform"
                semantic="OUTPUT"/>
              <input source="#Sphere001-Interpolations" semantic="INTERPOLATION"/>
            </sampler>
            <channel source="#Sphere001-Matrix-animation-transform"
              target="Sphere001/matrix"/>
        </animation>
      </animation>
  </library_animations>

- <library_visual_scenes>
  - <visual_scene name="b_ball" id="b_ball">
    - <node name="Sphere001" id="Sphere001" sid="Sphere001">
        <matrix sid="matrix">1.000000 0.000000 0.000000 1.181931 0.000000
          0.000000 1.000000 11.328453 0.000000 -1.000000 0.000000 -4.934601
          0.000000 0.000000 0.000000 1.000000</matrix>
        <instance_geometry url="#Sphere001-lib"/>
      - <extra>
        - <technique profile="FCOLLADA">
            <visibility>1.000000</visibility>
          </technique>
        </extra>
      </node>
    - <extra>
      - <technique profile="MAX3D">
          <frame_rate>30.000000</frame_rate>
        </technique>
      - <technique profile="FCOLLADA">
          <start_time>0.000000</start_time>
          <end_time>3.333333</end_time>
        </technique>
      </extra>
    </visual_scene>
  </library_visual_scenes>
- <scene>
    <instance_visual_scene url="#b_ball"/>
  </scene>
</COLLADA>
```

List of Reviewed Publication (2010-2014)

Book Chapter in edited volume

1. Sultana, R.; Feisst, M.; Christ, A.: (2012) Collaborative Language Learning Game As A Device Independent Application. In Isaías, P.; Ifenthaler, D.; Kinshuk; Sampson, D. G.; Spector, J. M. (Ed.): Towards Learning and Instruction in Web 3.0. Advances in Cognitive and Educational Psychology. New York, USA : Springer, ISBN: 978-1-4614-1538-1

2. Christ, A., Meyrueis, P., & Sultana, R. (2013). Architecture of a device independent collaborative language learning game. In Z.L. Berge & L.Y. Muilenburg (Ed.), Handbook of Mobile Learning. New York, USA & Oxfordshire, UK: Routledge, ISBN: 978-0-415-50369-3

Paper in international journal

3. Marepalli, S. M., Sultana, R., & Christ, A. (2013). Research on System Architecture to Provide Maximum Security, End User Device Independency and User Centric Control over Content in Cloud. International Journal of E-Entrepreneurship and Innovation (IJEEI), 4(3), 38-52. doi:10.4018/ijeei.2013070103

4. Sultana, R., Christ, A., Meyrueis, P., System Architecture for Device and Content Independent Applications Including 3D Imaging as Content. International of 3-D Information Modeling (IJ3DIM), IGI Global (Under review)

5. Sultana, R., Christ, A., Meyrueis, P., System Architecture for Device and Content Independent Applications Including 3D Imaging as Content: Implemented in a Collaborative Mobile Learning Game. International Journal of Mobile and Blended Learning (IJMBL), IGI Global (Under review 2nd phase)

Reviewed Paper & Poster in international conference proceeding

1. Christ, A; Feißt, M.; Sultana, R.: Collaborative Language Learning as A Device Independent Application. CELDA (Cognition and Exploratory Learning in Digital Age) 2010 Timisoara, Romania (15. – 17. October 2010)

2. Sultana, R.; Christ, A.: An Architecture for Device Independent Mobile Learning Content Presentation. mLearn 2011, Beijing, China (18.-21. Okt. 2011)

3. Sultana, R.; Christ, A.: Research on System Architecture for Device Independent Applications for 3D Imaging and Virtual Reality.mLearn 2011, Beijing, China (18.-21. Okt. 2011)

4. Sultana R, Christ A, Meyrueis P ; Architecture of Device Independent Communication Applied in Language Learning Game. IDAACS-SWS'2012 ; 20-21 Sept 2012, Offenburg. Germany

5. Razia Sultana; Andreas Christ; Markus Feisst; Dan Curticapean, Learning in the cloud: A new challenge for a global teaching system in optics and photonics. ETOP 2013 University of Porto, Portugal (July 23-26)

6. Razia Sultana; Andreas Christ;Patrick Meyrueis; Diversity of devices along with diversity of data formats as a new challenge in global teaching and learning system. ETOP 2013 University of Porto, Portugal (July 23-26)

7. Razia Sultana; Andreas Christ;Patrick Meyrueis; System Architecture for Device and Content Independent Communication Including 3D Imaging. mLearn 2013 Doha, Qatar (Oct 22-24)

8. Sultana, R.; Christ, A.; Meyrueis, P.: System Architecture of Device and Content Independent Application including 3D Imaging and Virtual Reality Implemented in Collaborative Language Learning Game. Second World Congress on Information and Communication Technologies (WICT 2012), Trivandrum, India (30.-31. Oct, 1.-2. Nov. 2012)

9. Marepalli, S. M., Sultana, R., Christ, A., The Challenges of Collaborative Learning in Cloud and a Three Layered Architecture towards the Solution. The Third International Conference on Digital Information and Communication Technology and its Applications (DICTAP 2013). Ostrava, Czech Republic (8-10 July.2013) Available in:

HTTP://SDIWC.NET/DIGITAL-LIBRARY/THE-CHALLENGES-OF-COLLABORATIVE-LEARNING-IN-CLOUD-AND-A-THREE-LAYERED-ARCHITECTURE-TOWARDS-THE-SOLUTION.HTML

10. SULTANA, R., CHRIST, A., MEYRUEIS, P., SOFTWARE ARCHITECTURE AS A FREEDOM FOR 3D CONTENT PROVIDERS AND USERS ALONG WITH INDEPENDENCY ON PURPOSES AND USED DEVICES, SPIE PHOTONICS EUROPE 2014, BRUSSELS, BELGIUM (14-17 APRIL 2014)

11. MAREPALLI, S. M., SULTANA, R., & CHRIST, A. (2014). INTRODUCTION TO MAR PRINCIPLE: A LOG-BASED APPROACH TOWARDS ENHANCED SECURITY IN CLOUD. KONFERENZ SOFTWARE-TECHNOLOGIEN UND -PROZESSE (STEP 2014), FURTWANGEN, GERMANY, (20 MAY 2014)