A Novel Approach for Mobile Native App Development Using Ontological Design

Sasmita Pani\textsuperscript{1*} and Jibitesh Mishra\textsuperscript{2}

\textsuperscript{1}Department of Information Technology, College of Engineering and Technology, Ghatkia, Bhubaneswar, India
\textsuperscript{2}Department of Computer Science and Application, College of Engineering and Technology, Ghatkia, Bhubaneswar, India

\texttt{1}susmitapani@gmail.com 
\texttt{2}jmishra@cet.edu.in

Abstract

Mobile devices are different from desktop based systems in terms of particular execution environment, limited resources, small size and high mobility requirement. Various agile based methodologies are developed for mobile applications such as Mobile-D, Scrum, and MASAM etc. These methodologies are used in building mobile apps. Some of web engineering models or methodologies such as WebE, OOHDM and WebML are also used for developing web apps. But these methodologies are used only for designing web apps but not for mobile apps. These web and mobile apps development models don’t provide structured and semantic representation of data. They don’t provide knowledge of a set of concepts which are relationally arranged with each other. Due to this representation of a particular data is not structured and hence the efficiency of its retrieval is ineffective. These mobile app development methodologies retrieve information from semantic web resources. But they are not concerned with semantic design of data for a mobile native app. Using ontology, data and their properties are arranged and organized in terms of classes, subclasses and object properties. For this propose, Web ontology language (OWL) is used for modelling the classes, their subclasses and object properties. OWL DL provides various class axioms and property axioms for modelling data in a structured and semantic manner. The aim of this paper is to develop a model based approach that takes three main components in 3D approach such as context, generic ontology and developments phases for designing mobile application in a semantic way. Depending upon the semantic design, the mobile app will be developed so the user can get meaningful data. This paper also validates the entire 3D model using hypothesis testing against mobile-D and shows its acceptance against mobile-D native app development model.

Keywords: Context, generic ontology, conceptual design, architecture design navigation design

1. Introduction

Mobile devices are the ubiquitous systems embedded with various sensors and powerful processors which can provide information about any domain like agriculture, health care system and learning system. There are various platforms of mobile phone, ios phone, android phone which are open for third party services. Mobile applications are more effective and adaptive due to the providing of required information to users without taking so much attention of users. In a generic domain users need information about
different domain specific entities and this information can be given to users through any web services. The agile software methodology is used to develop any mobile apps but this approach is not taken consideration about semantic modelling of data under the generic mobile domain. Generally the agile process models are using HTML, HTML5, CSS, XML and JavaScript languages for designing the mobile apps. Here we have proposed a 3D approach which takes into account context, generic ontology and development phases as key elements for building generic mobile native apps. We have specified and identified the requirements for a generic mobile app and established conceptual relationships among these concepts using various axioms for obtaining semantic mobile apps. We have also identified the context elements and specified the involvement of contexts in design phases for developing generic mobile native apps. Here this proposed model can be worked in generic domain or any domain such as agriculture systems, health care systems and learning etc. and mobile native app will be developed accordingly.

The rest of the section is as follows. Section 2 discusses related works. Section 3 gives research methodology for developing the 3D based model. Section 4 discusses proposed 3D model and section 5 provides the validation of the proposed approach. Section 6 provides discussion. Section 7 discusses conclusion and future work respectively. Section 8 provides author contribution and section 9 elaborates case study.

2. Related Works

In this section we briefly explore the different mobile app development methodologies and limitations of these models in contrast to semantic mobile native app development. We then discuss various web app development methodologies.

2.1. Mobile App Development Methodologies

The end user demands the mobile apps should work satisfactorily although it works in a heterogeneous and resource limited environment. Compared to desktop apps, the mobile apps possess various constraints like small screen size, limited power supply and bandwidth variability etc. Having these constraints, the users demand that it should provide high availability of data in a short response time. Finally the rapid evolutions of mobile apps challenge various premises upon which agile software development methodologies have been developed. Agile Software Engineering Framework for mobile web application works great when the solution is targeting to more than one platform. The mobile web application works through any web standard like HTML5 and CSS3, XML and WML. There are various agile approaches, [1] in the software development of mobile apps such as Mobile-D, MASAM, Hybrid Methodology, Scrum, and Scrum Lean Six Sigma.

Mobile-D [2], is an agile software methodology is based on so extreme programming, crystal methodologies and rational unified process. The mobile-D is the combination of five phases such as explore, initialize, productionize, and stabilize and system text & fix. In the explore phase, the development team must produce a plan and identify the project characteristics. This phase has again sub divided into three stages as stakeholder establishment, scope definition and project establishment. It includes the tasks as customer establishment, initial project planning, requirements collection, and process establishment. In the Initialize phase, the development team and all active stakeholders have to understand the product in development and define the key resources necessary for production activities. The key resources are physical, technological and communication resources for the mobile app software production. The productionize phase includes implementation of software product development. This phase includes various activities such as development process prioritizing and analyzing requirements, writing code and
using test driven development with continuous integration. The last two phases of the methodology comprises the finalization and testing of the software product.

MASAM [3], is Mobile Application Software Agile Methodology which provides the process for developing the mobile applications easily using an agile approach. It is based on extreme programming, agile unified process, RUP and the software and systems process engineering meta-model. This MASAM development methodology provides a mobile application development cycle which is structured into four phases as preparation of the software product, embodiment, development and commercialization. The preparation phase defines a summary and a pre study about the product, and assigns roles and responsibilities among the developers. It also includes pre planning and project setup for the development of the product. The embodiment phase focuses on understanding user’s needs and defining the architecture of the software product. In this phase, the implementation of the software product is carried out through test driven development, pair programming, refactoring with continuous integration with a close relationship with iterative testing activities. Finally the commercialization phase includes methods to concentrate on product releases and selling tasks.

Scrum [4], is an iterative and incremental agile based framework commonly used in mobile app development. The Scrum uses flexible strategy where customer has more transparency on the current progress of the project and customer can change the project execution plan in the middle of the project according to the business values which is not possible with other traditional/sequential approach. It uses iterations of fixed duration which duration is one month or less than one month, called sprints. The product owner is responsible for managing the product backlog which has list of user requirements and is maintained for the app development. It is dynamic and constantly changes to identify what mobile app needs to be appropriate, competitive and useful. In the product backlog, high priority items get immediate development activities over lower priority items. development team is responsible for various activities such as analysis, design, implementation, test, and documentation of the final product in a sprint backlog. Sprint backlog is the list of selected items from the product backlog for the sprint. Sprint backlog makes visible all of the work that development team identifies as necessary to meet the sprint goal. After this, the scrum team decides how much work they will execute in the next sprint. By the end of the sprint, a functional product is delivered and the pending features of the new product will be forwarded to the development team in the next iteration.

SLeSS [5], is an agile approach that integrates Scrum and Lean Six Sigma that focuses on project management and process improvement for the development of embedded software for mobile phones. The use of SLeSS assists in the easy adaptation to requirement changes in the later stages of the project and with less overall impact than the traditional approach which helps in meeting deadlines, reduces overtime hours, and delivers more rapidly versions and shortening the development cycle. The execution of SLeSS assumes an incremental approach by first implementing the Scrum alone and once the Scrum is well defined in the organization, Lean Six Sigma should be implemented as a quality framework for the development of mobile apps. SLeSS picks up from the Scrum methodology through pursuing a combination of the effort and consistent deliveries of the Scrum sprints with the continuous process analysis and improvement model represented by the Six Sigma methodology.

The approach uses two types of product backlogs such as customization product backlog for customizing development projects and LSS (Lean Six Sigma) Product Backlog for process improvements for the development of embedded software for mobile device. The six sigma methodologies employs phases such as define, measure, analyze, improve and control for improving the mobile app. The sprint backlog is used not only to establish the objectives of the next iteration, but it is also carefully examined for statistic based process improvement purposes. This agile incremental approach enables the
achievement of performance and quality targets of real software development project increases productivity, improves process quality, helps in cost reduction, progressively improves the development process, management process and the outcome of the projects with fewer defects and failures for the development of the mobile app.

2.2. Web App Development Methodologies

Web engineering is the application of systematic, disciplined and quantifiable approaches to development, operation, and maintenance of web-based applications. It is both a proactive approach and a growing collection of theoretical and empirical research in web application development. It includes various design methodologies or design approaches for designing web apps such as UWE, WebE, OOHDM, Web ML, OOWS and WSDM.

UWE [6], stands for UML based web engineering, is composed of different models at different phases such as requirements engineering, analysis, design, and implementation of the development process which are used to represent different views of the web application that includes content view, navigation view and presentation view. UWE includes UML diagrams to represent structural aspect of different views and behavioural aspect of web apps. Requirement modelling is done at two steps. At first a rough description of the functionalities is produced by UML use case diagram. use cases are used to model typical user behaviour when interacting with a web application. Process use cases are used to describe business tasks that end users will perform with the web apps. Personalized use cases are used to provide customization services to user with in web apps. In a second step, a more detailed description of the functionalities is developed by UML activity diagrams that give the responsibilities and actions of the stakeholders. The detailed specification of requirements depends on the project risks and the complexity of the web application to be built. Analysis models provide a basis for designing the content in web apps. This content model provides a visual specification of the domain specific information for the web system which comprises the content of the web apps.

The navigation model comprises the navigation space model and the navigation structure model. The first specifies which objects can be visited by navigation through the web application and the second specifies how these objects are reached. The navigational space model specifies two modelling elements such as navigation class and navigation association. These elements are termed as page and link in web terminology. The navigation structure model describes how the navigation is supported by access elements such as indexes, guided tours, queries and menus. The presentation model, is based on the navigation model provides an abstract view of the user interface of a web app. The presentation model only describes the basic structure of the user interface by including UI elements such as text, images, anchors, forms. These UI elements are used to present the navigation nodes. The presentation model also specifies how each node or page is presented to a user and how the user can interact with them. The presentation model is independent of the actual techniques used to implement the web site and allows the stakeholders to evaluate the appropriateness of the presentation before implementation.

OOWS is an object oriented web solution method [7], which provides a strategy for going from the problem space, is presented as conceptual models to the solution space that is the final software product in an automatic way. The OOWS web app development methodology is composed of two main phases such as conceptual modeling and solution development. The conceptual modelling phase again decomposed into three sub steps such as functional requirements elicitation, classic conceptual modelling, navigation and presentation modelling. In the functional requirements elicitation phase, conceptual schema is built by applying use cases and scenarios of university department by taking it as case study. In the classic conceptual modelling sub phase, structural model, functional and dynamic models are used to capture the system structure and behaviour. The
navigational modeling is done in two phases which includes user identification and categorization step and navigational diagram specification step.

This user identification and categorization step provides a user diagram which specifies the system kind of users and the interaction among users. The diagram also specifies the accessibility of each kind of user to the system information and functionality. The navigational diagram specification phase provides a navigation model to capture the navigational requirements of web apps by defining a navigational view or navigational map for each kind of relevant users of the system. The presentation modelling provides various presentation patterns such as information paging pattern, circularity pattern and layout patterns which are used to specify navigation features and capture the essential requirements for the web app development. The specified information will be used by the model compiler to generate the web pages interface. The solution development is the second main step in the OOWS approach which is determined by the target platform and is chosen a specific architectural style. This phase uses a three tiered architecture style such as presentation tier, application tier and persistent tier which are providing clear structure web applications, making ease in their adaptability and extensibility and improving scalability. These architecture styles are applied for obtaining the final system.

WSDM [8], stands for web site design method, is an audience driven methodology which does the requirement analysis from audience point of view unlike the other web apps development methodologies like OOHDM, WebML, OOWS, UWE and WebE. The WSDM is a user centric web app development approach and it integrates with V-model for testing of each phase to enhance the maintainability and effectiveness of the web site. WSDM phases include mission statement phase, audience modelling phase, conceptual design phase, implementation design phase and implantation phase. The mission statement phase defines the subject, purpose and specifies the intended users and declares them as target audience. It ensures that the designer should clearly establish the borders of the design. The audience modeling phase provides users a general indication of the audiences involved in the web app. The users identified in the mission statement are taken as a starting point and classified into different audience classes based on their information and functional requirements.

The conceptual design phase describes the structure of the web app and models how the members from different audience classes will be able to navigate through the web apps. The navigational design built a navigational model that consists of nodes and information where this information are logically related. The implementation design phase consists of many sub phases. The site structure design sub phase decides the nodes in the navigation model and makes them groups to form pages. The presentation design sub phase provides aesthetic design and layout of web apps. In the style and template design sub phase, the designer specifies templates that will be used for different types of pages. In the page design sub phase, the designer describes where the information on a page should be positioned and how it should be laid out. It is also decided how and where the links from the navigational design should be presented. In the implementation phase, an implementation environment is chosen and then the result of the implementation design is converted to the chosen environment. At last, WSDM integrates with V-model for testing of various phases. WSDM applies various testing methods such as unit testing, integration testing, system testing and acceptance testing in its each phase starting from requirement analysis and ends in implementation.

3. Research Approach

In order to explore issues around generic mobile native app specification processes and what characteristics are typically included in these specifications, we have established four research questions (RQ1 to RQ4). These are as follows.
RQ1
How the requirement gathering is done in building mobile native generic apps under mobile domain?
This question RQ1 is established to identify and understand the requirements under different contexts in mobile domain which are included in the requirement specifications for building mobile native generic apps.

RQ2
How the different attributes or elements under mobile native app development are identified in order to give effective requirement analysis for building mobile native generic apps?
This question RQ2 is established to identify the different attributes or elements under mobile domain with a particular focus required attributes based on contexts for building mobile native generic apps.

RQ3
How the design models are defined and modeled in generic mobile native apps considered in existing process models under web app domain for effective design?
In order to defining the design models and modelling them in developing mobile native generic apps, RQ3 was established to examine the current process trends into designing web applications, with a particular focus developing the design phases in generic mobile native apps through mapping the current web app design models and its attributes into those of mobile native generic apps.

RQ4
To what extent the design model for semantic modelling generic mobile native apps considered in existing process models under mobile domain?
For specifying the design elements and modelling them in semantic way, RQ4 was established to identify the current design process used in developing mobile native applications with a particular focus towards providing a semantic approach for design and development of mobile native generic apps.

We undertook an extensive set of questionnaires and send these questionnaires to different mobile app software developers of different software industries. The questions were placed in Google forms and sent to different mobile app developers of different software industries in India having international offices as well. After getting their responses, we are able to find out different aspect or contexts in terms of requirements under mobile native app domain. The overall goal was to understand the best requirements for the conceptual design and semantic design of generic mobile native apps.

All these sending questionnaires and response processes were conducted over a period of 6 weeks, primarily during October/November 2015.

4. Proposed 3D Approach for Mobile Native Apps

There exists various design methodologies for web engineering models as we have discussed in Section 2. Also there exist various agile software methodologies to develop any mobile native apps. But these agility based software methodologies are not using contexts in mobile domain which is bottleneck of this framework. Also the web engineering process models are not providing semantic design of data in web applications which is the primary need of the design model.
Figure 1. A 3D Approach for Designing Mobile Native Apps

Here we are using a 3D approach for building the design model in a structured and semantic manner by taking three dimensions as contexts, Generic ontology and development stages of mobile native app. We are identifying contexts under mobile domain through research approach and modelled them in ontology through different development phases for a mobile native app.

4.1. Contexts

We have studied various contexts in ubiquitous systems and identified four contexts in mobile devices as determined from empirical analysis given in [Ph.D thesis]. These four contexts includes device context, user context, mobility context and social context in mobile domain. We have taken numerous activities [60], of various components in mobile native apps and organized them into a set of questions through Google forms and send to various mobile app developers of different software organizations for finding out the attributes of mobile contexts.

We have done a mapping of some activities into contexts under mobile domain and identifying them such as device context, user context, mobility context and social context which are required for design and development of mobile native apps [Ph.D thesis].

4.2. Generic Ontology

Ontology represents description of concepts and relationship that exist for an object or a group of objects. The term ontology is borrowed from philosophy [9], where an “Ontology is a systematic account of Existence”. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called
the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary [10], with which a knowledge-based program represents knowledge. Thus, we can describe the ontology of a program by defining a set of representational terms and this ontology can be termed as domain ontology. In such domain ontology, definitions associate the names of entities in the universe of discourse for e.g., classes, relations and functions etc. with human-readable text describing what the names mean, and formal axioms that provide constraints in the interpretation and well-formed use of these terms.

Here the generic ontology can be used in any domain such as agriculture domain, learning domain and healthcare domain etc. For creating agriculture ontology, the concepts such as crops, farmers, season, soil etc. associated in this domain are represented as objects and classes. Relationships can be established among these objects for representing the formal vocabulary in agriculture ontology. Formal axioms such as class axioms and property axioms can be established among the classes, and relationships to obtain constraints in ontology.

4.3. Development Phases for Mobile Native Apps

Here the approach is to build a process model for design of a generic mobile app so that the native user can get the required information in a systematic and semantic way. This model or the layered approach consists of four layers or sub models which are as follows.

- Requirement Analysis Model
- Design Model
- Testing
- Implementation

All these phases are discussed below and the requirements model specifies requirements based on which mobile native apps can be designed in a structured and semantic manner so that user can get data in a meaningful manner.

4.3.1. Requirement Analysis Model: The proposed 3D design approach or model provides the framework for constructing the semantic design and based on which the development of the mobile native apps can be built for any domain of the user application.

The device context consists of hardware specifications, technical specifications and functional specifications. The hardware characteristics or specifications [11], of a mobile device that includes screen size, screen resolution, overall physical dimensions, weight etc. The technical characteristic or specification of a mobile device includes processor speed, sensors and storage capabilities. The motion, posture and placement of a mobile device can be extracted from various sensors [12], which include 3D accelerometer, digital compass and skin resistance sensors.

The user context consists of user’s role and task, its profile etc. The user context also includes user’s preferences and its usefulness for accessing a generic mobile app. User and it’s role holds information about user as well as it’s activity [13], related to any generic mobile native application. The activity and task of a user is dependent on the type of mobile native apps. The user’s preferences determine the preferences or likeliness of the user towards a generic native mobile app. The user’s preferences include effectiveness, efficiency, satisfaction and memorability [14]. The user’s usefulness is determined frequency of usage of generic mobile native apps by users. There are certain concepts by which usefulness of a user determined such as perceived ease of use, perceived usefulness, intention to use and trust [15] and [16].

The social media tools, [17] and [18] enable the user to efficiently use them which are linked in a mobile native apps and get the required information from the mobile native
apps. Wikis are social context parameters which allow the users to contribute and edit the information available publicly to the community of users searching for their relevant information. Wikis provide information about specific topic which the users easily searches and views them. Mobility enables the flexibility and portability of a wireless mobile device in moving from one pale to another and continuing to access the data connection network facility throughout its location inside the network zone through using some wireless mechanisms such as WLAN, browser, GPS and Bluetooth, [19] which are shown in Figure 3. Wireless LAN uses the IEEE 802.11 standards of wireless technology popularly called Wi-Fi. The WLAN uses various versions of IEEE 802.11 standards for wireless bearer services with frequency of 5GHz ISM band and data rate of 54 Mbps speed. It is also used for power control, authentication, handoff, internetworking, quality of service and prioritization.

Mobile browsers are optimized for the effective display of user friendly screen interface compatible to portable mobile devices. A mobile browser is designed such that it can be accommodated in low memory capacity and low bandwidth wireless handheld devices. The mobile browser is connected to the cellular through WLAN connection or other types of mobile networks for accessing the web using the http and the TCP/IP network protocols. The mobile browser uses different layers of wireless application protocols for providing the various facilities of the internet browsing in mobile handheld devices.

4.3.2. Design Model for Mobile Native Apps: The design model for mobile native apps consist of various design phases such as:

4.3.2.1. Content Design: It focuses on organization of the contents in the screen of mobile native apps over a mobile device. The contents are constructed and designed in views or pages of mobile native apps irrespective of platform. The content in mobile native generic apps is organized according to the user context or user aspect irrespective of domain. Based on user’s role, profile and task, the content is developed in mobile native apps. The content design also concerns about user’s percieveness and usefulness in mobile native apps. Hence the user context will be placed in content design.

4.3.2.2. Architecture Design: The architecture design concerns about the hardware elements in mobile native generic apps. The architecture design includes mobility mechanisms, sensors, input and output mechanisms for building mobile native generic apps. The mobility mechanisms include mobility elements like GPRS, EDGE, Bluetooth and GPS etc. The mobile device elements include various sensors, keyboard and memory. Hence the device context and mobility context will be put under architecture design in mobile native generic apps.

4.3.2.3. Conceptual Design: The conceptual design is built with ontologies in the mobile native apps domain to describe the concepts and relationships that determine the semantics of the concepts in that domain. The contexts such as device, user, mobility and social under mobile domain are organized meaningfully through building mobile domain ontology and hence these are put under conceptual design. These contexts and their attributes are built as classes, subclasses and structured in a meaningful manner through specifying various axioms. Web ontology language or OWL is the modern tool available to create and represent the ontology having classes and relationships between the classes, clearly describes its meaning so that an application can be developed taking into consideration the four contexts namely the device context, mobility context, user context and social context.

The conceptual design or model will work as a hidden layer and based on it UI design will be done for building and developing mobile native apps. OWL DL stands for web
ontology language description logic which is a sublanguage of OWL and provides logics for formal description of concepts and roles. Here concepts in ontology describe a set of individuals and role defines the relationship/property holds among them. Semantically these logics are found in predicate logics and have efficient decidability to build knowledge base information system or ontology.

In the mobile domain ontology these classes are represented as classes and their attributes are represented as subclasses. The classes are interrelated with each other through various relationships identified as object properties and is defined through property axioms which are shown in sections below.

4.3.2.3.1. Building Class Axioms: The class mobility context contains the subclasses such as WLAN, GPRS, EDGE and GPS as shown below. To design them semantically and structurally, it is necessary that the instance of one class cannot be the same instance of another class. It can be achieved through taking class axiom “disjoint” among the classes as shown below in the following OWL structure.

```xml
<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#Bluetooth"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#Browser"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#EDGE"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#GPRS"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#GPS"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#WLAN"/>
  </owl:members>
</rdf:Description>
```

4.3.2.3.2. Building Property Axioms: The object property between the class device context and mobility context is denoted as “isPervadedBy” where the domain and range of this object property are identified as device context and mobility context. The domain, range and the property axioms are shown in the following OWL structure.

```xml
<owl:ObjectProperty
  rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#isPervadedBy"
  <rdf:type rdf:resource="&owl;AsymmetricProperty"/>
  <rdf:type rdf:resource="&owl;IrreflexiveProperty"/>
</owl:ObjectProperty>
```

4.3.2.4. User Interface Design: The user interface design takes place from user context point of view. The interface in mobile native generic apps is designed and developed based on user’s role, its task, profile, perceiviosity and usefulness towards using the mobile native generic apps. It concerns user context when designing interface in mobile native apps and hence it is put under user interface design. The user interface design takes into account various elements for designing mobile native apps specific to different platforms. The user interface design concerns about the design elements such as different layouts, widgets and containers for design and development of mobile native apps.
4.3.2.5. Navigation Design: The navigation design focuses on user’s perceived usefulness in native apps over a mobile device so that user can move or navigate among the screens efficiently. It is necessary to build the navigation design from the user’s perceived so that user can do movements within the native app over mobile device. The user can do different interactions or navigations such as across, into, back out from the different pieces of content in the generic mobile native app. Hence the user context will be put under navigation design.

4.3.2.6. Presentation Design: The presentation design also focuses on user’s perceived usefulness in native apps over a mobile device so that user can interact with the native app components. Hence user context will put under presentation design in mobile native generic apps for providing effective and efficient view or page on the mobile devices. The presentation design focuses on presenting a view or page in mobile native apps over a mobile device. It includes the elements used for presentation design are fade-in, fade-out, zoom-in and zoom-out.

4.3.2.7. Personification Design: The personification design enables customization of mobile native app user by allowing them to connect with social media tools. Hence if a mobile native app user wants to share and find its information, it can do this by linking with social media tools. Hence the social context will be put under personification design.

4.3.3. Testing:

4.3.3.1. Testing Conceptual Design: The ontology which is created by the protégé OWL DL creator, user tool called reasoner. The reasoner is used as an additional plug-in with protégé OWL DL checks and validates for the overall consistency of the created mobile domain ontology by parsing through each class and class axioms along with property characteristics and restrictions in the class hierarchy which is present in the mobile domain ontology. The ontology is said to be consistent and meaningful if it is successfully classified and validated by the reasoner. The reasoner can be of several versions which are compatible in protégé 5.0 beta such as pellet, hermi and shet++ etc.

4.3.3.2. Testing User Interface Design: The user interface is built and developed through different layouts, views, widgets and wrapped through various containers. These design elements are developed through java programming languages in android studio framework. Further, these design elements are implemented or tested in emulator for validation purpose through which user interface is developed. The emulators are different with respect to different platforms.

4.3.3.3. Testing Navigation Design: There exist different types of navigation such as top down navigation, front navigation, back navigation etc. and navigation in any direction if possible in mobile native apps. The navigation mechanism is specific to various platforms. In android platform, all the navigation mechanisms are handled by navigation controller. To develop the navigation mechanism, a navigation drawer is defined in the java activity file. The function of navigation drawer is to provide list of items present in the native mobile apps so that native mobile app user can select the desired option or item. Further these navigation design elements are tested or implemented in emulator so that navigation design can be developed. The emulators are different with respect to different platforms.

4.3.3.4. Testing Presentation Design: For presenting a view in or a number of views in the generic native mobile apps, various transition animation operations are used. The transition animation includes zoom in, zoom out, fade in and fade out. Further these
presentation design elements are implemented in emulators of specific platform for validation purpose. The duration of the transition animation is defined by the mobile native app developers based on the client requirements.

4.3.3.5. Testing Personification Design: For achieving customization and personification in generic native mobile apps, the mobile native app user should be connected with other users through the social media tools so that mobile native app user can share and find its desired information about any domain. All the design elements which are used in personification design are tested in emulators of specific platform to develop the personification design.

4.3.4. Implementation: In this work, we have developed an approach or design model which represents the important components in form of an ontology developed in protégé OWL DL language. The different relationships between the mobile domain contexts are modelled through OWL DL for organizing the data and attributes in a semantic way to develop a mobile native application.

The user context and the device context have the relationship “device usability” as the common parameter among the device context and user context. The user context and the social context have the relationship “social interaction” as the common parameter between the user context and social context classes. The device context and the mobility context have the relationship “pervasiveness” as the common parameter between these contexts and mobility context and the user context have the relationship “mobility interaction” as the common parameter between these contexts. These common parameters between the contexts govern, the development of a generic mobile native app in the semantic web environment, is shown in Figure 2.

Figure 2. Semantic Relationships among the Different Contexts under Mobile Domain

The relationships between the contexts is implemented in the OWL DL language which is used to create ontology and the axioms determined such as the relationship restrictions, the object property consistency with domain and range, and the class hierarchy consistency for a semantic mobile native application. The entire requirements for designing a mobile native app are shown in the VOWL in Figure 3.
Figure 3. Requirements for Designing a Mobile Native App in VOWL

The entire model or the layered approach will be used as base for development of mobile native apps. Here in this approach, we have specified the requirements as needed for mobile native apps and accordingly we have designed it semantically and building structured relationships among the various concepts for mobile native apps as shown in 4.3.2. This semantic design act as hidden layer and foundation for UI design in mobile native apps. If further, the client specifies different requirements, then the design process will be done accordingly as sown in Section 4 for mobile native apps.

5. Validation of the Proposed Model

We have formed questionnaires based on process areas with their specific goals and interviewed to six experts of different mobile app developers of different software industries in India. We have also taken the existing native mobile app development framework mobile-D as benchmark level and performed hypothesis testing through getting responses from the mobile native app developers. The Table 1. shows the different process areas with its specific goals in the 3D approach.
Table 1. Process Areas and Specific Goals in Proposed Final Design Model for Mobile Native Apps

<table>
<thead>
<tr>
<th>Process Areas</th>
<th>Specific Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Requirement analysis</td>
<td>N1 Establishment of process, project and customer</td>
</tr>
<tr>
<td></td>
<td>N2 Analyzing the requirement contextually with respect to device, mobility, user and social media tools</td>
</tr>
<tr>
<td></td>
<td>N3 Identification of the physical, technological and communication resources for the product development</td>
</tr>
<tr>
<td>2. Design model</td>
<td>N1 Design of the content in mobile native apps</td>
</tr>
<tr>
<td></td>
<td>N2 Providing architecture design</td>
</tr>
<tr>
<td></td>
<td>N3 Designing the context attributes conceptually</td>
</tr>
<tr>
<td></td>
<td>N4 Developing the user interfaces</td>
</tr>
<tr>
<td></td>
<td>N5 Providing navigation design</td>
</tr>
<tr>
<td></td>
<td>N6 Developing various transition animations for presenting a view</td>
</tr>
<tr>
<td></td>
<td>N7 Developing personification design</td>
</tr>
<tr>
<td>3. Testing</td>
<td>N1 Testing of content design phase</td>
</tr>
<tr>
<td></td>
<td>N2 Testing of architecture design</td>
</tr>
<tr>
<td></td>
<td>N3 Testing of conceptual or semantic design</td>
</tr>
<tr>
<td></td>
<td>N4 Testing of user interface</td>
</tr>
<tr>
<td></td>
<td>N5 Testing of navigation design</td>
</tr>
<tr>
<td></td>
<td>N6 Testing of presentation design</td>
</tr>
<tr>
<td></td>
<td>N7 Testing of personification design</td>
</tr>
<tr>
<td>4. Implementation</td>
<td>N1 Working of the final native app by configuring it into the system</td>
</tr>
<tr>
<td></td>
<td>N2 Release of the final native app</td>
</tr>
</tbody>
</table>

5.1. Statistical t-test

A t-test can be defined as a statistical hypothesis test in which the test static follows a student’s t distribution when the null hypothesis is supported. In this, a comparison is made between the means of two groups of observations. The observations must be made randomly assigned to each of the two. Here we have taken Mobile –D framework as null hypothesis which is used as benchmark level in this statistical analysis.

H₀: Null Hypothesis – Rd Process area is not a significant process in final model

H₁: Alternative Hypothesis – Rd Process area is a significant process in final model

The T-value of all the process areas are calculated and the values of each process areas are greater than 0.05 (> 0.05), hence accept H₁ with its all process areas. The t-values are shown in Table 2 below.

\[
T \text{ test formula} = \frac{\bar{X}_1 - \bar{X}_2}{S_1^2/n_1 + S_2^2/n_2} \tag{1}
\]

\[
\bar{X}_1 = \frac{\sum X_i}{n} \tag{2}
\]
\[ S_i = \frac{\sum (X_i - \bar{X})^2}{n-1} \]  

(3)

Similarly \( X_2 \) and \( S_2 \) can be calculated and the t value can be found as taking the formula given in (1). Here Table 2 and 3 shows the expert’s response belonging to different mobile app developers in India against the process areas. The tables for the other process areas are shown in annexure.

### Table 2. Requirement Analysis of the Proposed 3D Approach

<table>
<thead>
<tr>
<th>Process areas</th>
<th>Expert 1</th>
<th>Expert 2</th>
<th>Expert 3</th>
<th>Expert 4</th>
<th>Expert 5</th>
<th>Expert 6</th>
<th>Relevance score</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1 Establishment of process, project and customer</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>6</td>
</tr>
<tr>
<td>N2 Analyzing the requirement contextually with respect to device, mobility ,user and social media tools</td>
<td>Irrelevant (0)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Irrelevant (0)</td>
<td>4</td>
</tr>
<tr>
<td>N3 Identification of the physical, technological and communication resources for the product development</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>6</td>
</tr>
</tbody>
</table>

### Table 3. Requirement Analysis of the Mobile-D

<table>
<thead>
<tr>
<th>Process areas</th>
<th>Expert 1</th>
<th>Expert 2</th>
<th>Expert 3</th>
<th>Expert 4</th>
<th>Expert 5</th>
<th>Expert 6</th>
<th>Relevance score</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1 Establishment of process, project and customer</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>Relevant (1)</td>
<td>6</td>
</tr>
<tr>
<td>N2 Analyzing</td>
<td>Irrelevant</td>
<td>Relevant</td>
<td>Relevant</td>
<td>Irrelevant</td>
<td>Irrelevant</td>
<td>Irrelevant</td>
<td>2</td>
</tr>
</tbody>
</table>
the requirement contextually with respect to device, mobility, user and social media tools

<table>
<thead>
<tr>
<th>N3 Identification of the physical, technological and communication resources for the product development</th>
<th>Relevant (1)</th>
<th>Relevant (1)</th>
<th>Relevant (1)</th>
<th>Relevant (1)</th>
<th>Relevant (1)</th>
<th>Relevant (1)</th>
</tr>
</thead>
</table>
| From (2), we have calculated the mean of process area requirement analysis which is $\bar{X}_1$ in our proposed 3D approach and the value is 5.333. Similarly we have calculated the mean of process area requirement analysis which is $\bar{X}_2$ in Mobile-D and the value is 4.666. After that from (3), we have found out the standard deviations of the proposed 3D approach and mobile-D which are $S_1$ and $S_2$ respectively. The values of $S_1$, $S_2$ are 7.54 and 6.6. From (1), we have found out the value of $t$ which is equal to 0.11(0.11>0.05) is greater than 0.05. This states to accept the process area requirement analysis of our proposed 3D approach. The T-test value of each process areas are shown in Table 4.

Table 4. T – test Value of Different Process Areas in 3D Approach

<table>
<thead>
<tr>
<th>Sl no</th>
<th>Process areas</th>
<th>T-test value</th>
<th>Accepted/Rejected in final 3D model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Requirement analysis</td>
<td>0.11&gt;0.05</td>
<td>Accepted</td>
</tr>
<tr>
<td>2</td>
<td>Design model</td>
<td>0.14&gt;0.05</td>
<td>Accepted</td>
</tr>
<tr>
<td>3</td>
<td>Testing</td>
<td>0.20&gt;0.05</td>
<td>Accepted</td>
</tr>
<tr>
<td>4</td>
<td>Implementation</td>
<td>0.11&gt;0.05</td>
<td>Accepted</td>
</tr>
</tbody>
</table>

Here the T-test value of the process areas in 3D approach is greater than 0.05. Hence the null hypothesis Mobile-D is not a significant process and the proposed 3D approach or alternative hypothesis is a significant process for mobile native app development.

6. Discussion

By far the most significant observation from responses of questionnaires sending to the mobile app software developers of different software industries is that many aspects or contexts that would conventionally be regarded as "requirements" for developing mobile native apps in a structured and semantic manner. This is achieved by designing or modelling the four aspects or contexts in terms of requirements and their attributes using class axioms and property axioms through web ontology language (OWL) to achieve semantics in data. This semantic design will work as base for developing user interface and building generic mobile native apps.

If further, the requirements have changed that is the client specifies different requirements , then the design process will be done accordingly as sown in section 4 for mobile native apps.
7. Conclusion and Future Work

Previously various agile methodologies were used to develop mobile apps which are discussed in related works, but these are not focused on the context of mobile domain. Normally these mobile apps retrieve information from semantic web resources. Here our attempt is to build the mobile native apps in semantic environment so that user can get meaningful information. Our intention is to understand the basic optimum contexts or aspects under mobile domain and develop the model for mobile native application in semantic environment. In this work we have proposed an design model which is composed of many layers or sub model. The requirement model specifies different requirements for building a generic native mobile apps. Further these requirements are assimilated into different contexts such as device context, user context, mobility context and social context.

The conceptual model establishes establishes the structural relationships or semantic relationship among the different contexts or aspects for designing a generic mobile native app. Here we have modelled the requirements for building mobile native apps in an generic domain in semantic environment and based on these requirements, generic mobile native app will be built and developed. We have also identified the user interface design elements navigation design elements and presentation design element and social apps. These design elements can be designed in any framework like ios, windows and android etc and here we have taken android framework for design of these elements.

We have also validate our proposed model through hypothesis testing using statistical t-test and obtain its acceptance is better than the other mobile native app developments. This model will solve various requirements of the users. If further the requirements of user changes, then changes will occur in all the design phases of mobile native apps and accordingly better design model will be built and developed.

8. Author Contribution

Here we have identified and suggested different phases or process areas for mobile native app development and obtains expert’s views in our 3D approach. We have also obtained experts views in different phases of Mobile-D framework and done hypothesis testing against the proposed 3D approach and Mobile-D. We have found out that our proposed 3D approach have a greater acceptance than the Mobile-D framework. Hence from the statistical t-test, we conclude that the proposed 3D approach with different phases will be used for mobile native app development.

9. Case Study

There exists various E-agriculture systems such as e-choupal, m-kirishi, e-sagu etc. but these e-agriculture systems don’t provide semantic design of information so that the users can’t get meaningful data. We have taken agriculture domain as case study for semantic design in mobile native apps. For that, we have identified the phases of agriculture system in India and elements or objects that are associated with those phases. The phases of agriculture system in India include crop identification, crop production, crop protection etc.

The elements associated in those phases include crop, soil, season, farmer, weather etc. We have built agriculture ontology through taking objects, classes and specified class axioms and property axioms which is shown in Figure 4. We have established an relationship between the class agriculture system and farming equipment “applies” and the property axioms for this object property is shown in OWL DL code:
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2015/2/untitled-ontology-98#applies">
  <rdf:type rdf:resource="&owl;AsymmetricProperty"/>
  <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
  <rdf:type rdf:resource="&owl;IrreflexiveProperty"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2015/2/untitled-ontology-98#AgricultureSystem"/>
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2015/2/untitled-ontology-98#FarmingEquipments"/>
  <owl:inverseOf rdf:resource="http://www.semanticweb.org/ontologies/2015/2/untitled-ontology-98#isAppliedBy"/>
</owl:ObjectProperty>

Figure 4. An Ontology Graph for Agriculture System in India

References

[19] Introduction to WiFi Networking: Pacific schools and solar project.

Authors

Jibitesh Mishra, has more than 20 years of teaching and research experience. Currently, he is Associate Professor in the Department of Computer Science and Application, College of Engineering and Technology, Bhubaneswar, a constituent college of Biju Patnaik University of Technology, Odisha. He has authored many books of repute. He has written many papers and organized conferences in the area of web engineering and application. His research interests are fractal graphics, pattern recognition, quality web engineering and cloud computing.

Sasmita Pani, is doing Ph.D under Prof. Jibitesh Mishra and her research area includes mobile native app development, web engineering and semantic web.