Business Domain Modelling Using an Integrated Framework

Mohammed Salahat, Steve Wade

Abstract**—**This paper presents an application of a "Systematic Soft Domain Driven Design Framework" as a soft systems approach to domain-driven design of information systems development. The framework combining techniques from Soft Systems Methodology (SSM), the Unified Modelling Language (UML), and an implementation pattern known as "Naked Objects". This framework have been used in action research projects that have involved the investigation and modelling of business processes using object-oriented domain models and the implementation of software systems based on those domain models. Within this framework, Soft Systems Methodology (SSM) is used as a guiding methodology to explore the problem situation and to develop the domain model using UML for the given business domain. The framework is proposed and evaluated in our previous works, and a real case study "Information Retrieval System for academic research" is used, in this paper, to show further practice and evaluation of the framework in different business domain. We argue that there are advantages from combining and using techniques from different methodologies in this way for business domain modelling. The framework is overviewed and justified as multimethodology using Mingers multimethodology ideas.

Keywords**—**SSM, UML, domain-driven design, soft domain-driven design, naked objects, soft language, information retrieval, multimethodology.

I. Introduction

THE business domain for any organization accommodates the organization business process that must be well defined and modelled for the implementation. Business domain comprises the business process can be defined as 'the transformation of something from one state to another state through partially coordinated agents, with the purpose of achieving certain goals that are derived from the responsibility of the process owner' [16].

To support the business domain, good information systems software used to support the organization work by handling the internal business process and control all aspects affecting the execution of the process. The business process must be supported with good business process modeling (domain modeling) and implementation techniques that can analyze, model, and implement the business process in a professional way to achieve the organizational goals [18].

The failure of software support systems has been well documented over the years, and many of these failures have been attributed to poor business process modelling [1]. One of the main reasons for information systems failure is a tendency to concentrate on the technical aspects of design rather than understanding the business needs [2]. There is a need for a systematic approach for capturing the information required by business processes [1]. This suggests a need to bridge the gap between business process modelling, information systems modelling, and implementation. The bridging framework may be will enhance the development of proper information systems and the IS development process. It is named SDDD (Soft Domain-Driven Design) and it aims to investigate, analyze and model a business domain so that we can implement it as a software support system [4], [5], [21]-[24]. SDDD is a multimethodology systemic framework consisting of four phases with guiding procedures to steer the developer between the various compromises that need to be made throughout the development process. The paper will be presented as follows:

Section 2 reviews related work. Section 3 briefed the research methodology used. Section 4 is introduced the framework as a multimethodology approach. Section 5 is a brief description of a practical case study "An information retrieval system for academic research" in which the method has been applied. Section 6 presents a reflection on the framework and the learning process of applying it. Section 7 is conclusion and future works.

II. Related Works

A. Domain Driven Modelling (DDM)

The business domain for any organization accommodates the organization business process that must be well defined and modelled for the implementation. Business domain comprises the business process that can be defined as 'the transformation of something from one state to another state through partially coordinated agents, with the purpose of achieving certain goals that are derived from the responsibility of the process owner' [16]. There are many definitions of "business process", and the most of these definitions are based on the idea of a business process as a deterministic system that receives inputs and transforms into outputs following a series of activities. For example [17] defines business processes as "''structured sets of activities designed to produce a specified output for a particular customer or market''. Business processes are similar in different business domains running the same industry of business. To support the business domain, good information systems software is used to support the organization work by handling the internal business process and controlling all aspects affecting the execution of the process. The business process must be supported with good business process modeling (domain modeling) and implementation techniques that can analyze, model and implement the business process in a professional way to achieve the organizational goals [18].

Mohammed Salahat is a Lecturer in Ajman University of Science and Technology in UAE and a part-time-PhD student in the School of Computing and Engineering, University of Huddersfield, UK (e-mail: m.salahat@ajman.ac.ae, u0423855@hud.ac.uk).

Steve Wade is a senior Lecturer in the Informatics department, School of Computing and Engineering, University of Huddersfield (e-mail: s.j.wade@hud.ac.uk)

B. Domain-Driven Design

Domain-Driven Design can be used to model the business process as a business domain model [6]. A Ubiquitous Language (UL) is generated first as a communication tool between different stakeholders and the domain model will be generated and implemented based on UL. UML diagrams are sufficient tools for requirement modelling to support business process modelling in an object-oriented domain model [19]. When it comes to implementing the system we have made use of the DDD implementation pattern (i.e. Naked Objects or True View) to reflect the system interface directly from the domain model. Naked Objects [20] and True View Domain Modeler are used for exploring Business Domains and creating rapid prototypes using Domain Driven Design. It helps you to work with your Domain Experts to understand business entities, relationships and the business' ubiquitous language and to write classes using. NET and the Naked Objects or True View framework.

C. Soft Domain Driven Design

Soft Domain Driven Design [21] is an approach that seeks to model the system processes as a domain model and develop a software support system based on it. In DDD Ubiquitous Language was used to create the domain model by the developers and domain experts [6] and to facilitate the communication between different stakeholders. UML, as a part of SDDD, defines a number of diagrams that can be used to model the business process [7] but lacks the ability to explore the soft issues related to the problematic situation which can be handled using Soft System Methodology. SSM is an established means of problem solving that focuses on the development of idealized models of relevant systems that can then be compared with real world counterparts [8]-[10]. SSM is used in SDDD to model the business domain using rich pictures, root definition, and conceptual model. In our previous work [21], we have adapted the idea of a Ubiquitous Language into a “Soft Language” which incorporate certain artifacts of a SSM analysis into the model. The first step of the SDDD approach is to develop a ‘Soft Language’ as result of the application of Soft System Methodology. This language is an a compliment of the Ubiquitous Language described in Domain-Driven Design [6] which consists of different concepts, diagrams, and documents to facilitate the communications between the developers and domain experts. Some researchers have explored the relationship between SSM and object oriented analysis and design techniques in general [11] but less has been written about the application of these techniques in the context of the UML. An object-oriented domain model can be extracted from this Soft Language through a transition process from SSM Conceptual Model to UML Use Cases. We argue here that SSM helps the developer to gain a deep understanding of different stakeholders’ perspectives which will need to be represented in the Soft Language. In this paper we argue that this transition supported the students understanding of modeling the business domain and implementing the software support system based on that.

As described in our previous work [21] SDDD framework guides the developer into creating a “Soft Language” which consists of the output of the SSM stage to deal with the soft aspects which are not handled explicitly by Domain Driven Design. The SSM Conceptual Primary task Model (CPTM) is used to map human activity to a UML use-case model using a new elaboration technique. Use-cases, as abstractions of business activities, are used to model the business process in a domain model using UML diagrams and based on the philosophy of DDD which employs the idea of “Knowledge Crunching” during the different stages. To the best of our knowledge, this combination has not been applied in an intervention before, and an evaluation in teaching context and the application in business projects will be a contribution to this domain of research and software development.

D. Other Related Works

Previous works consider the SSM conceptual model as a focal point for linking SSM and UML by mapping the activities of an SSM conceptual model into UML use-cases [12], [13]. Recent examples of this approach can be found in SWfM [7] and our previous works [4], [5], [21]. Other researchers have made use of various extensions to the UML. For example [3] employed a systemic framework combining SSM and UML extensions proposed by [14] to model the business process of a manufacturing factory. Their framework is based on Mingers Multimethodology ideas [15] but does not encompass the software implementation phase of development.

III. Research Methodology

This research, as part of on-going research work, aims to answer the following research question:

How to model and implement the Business Domain Processes into a Domain-Driven Design System?

Both authors are involved in teaching in their Universities. This encouraged us to use the approach of Action Research since we are actors and part of any system in the education environment and for us it is the domain to apply any research. By teaching systems modelling and design for many years by both authors, we found from teaching and the literature review that many software systems failed and the reason of failures because of the tendency to focus more on the technical aspects rather than the Business Domain Processes modeling [1]. The majority of software development methodologies initiated from the software engineering science and not given sufficient attention to the business processes modeling for any given business domain. Involving the business experts with the technical people to investigate and model any domain needs a methodology or framework that can be used by different stockholders and facilitate the communication between them. Among this Domain Driven Design is dominant but still the communication depends on the technical system concepts which may be a problem for the business expert to understand. Soft System Methodology is well-established and known as an approach to explore problematic situation. Based on that, this ongoing research suggested the combination between SSM, UML as a modeling language, and an implementation pattern satisfied the philosophy of Domain-Driven Design as a dominant approach among others. The new approach is proposed and published in our previous work [21] and further evaluation from the development perspective taken place in this paper. To answer the above research question and to apply the research as it's designed, the following methodology followed:

1. Review the current situation of business domain processes modelling through teaching and literature review. A comparison between different approaches done based on that.
2. Formulate and propose a multimethodology framework considering soft and hard business domain aspects.
3. Evaluating the framework through different practical case studies as an undergraduate and Msc projects (Peer-Tutoring System,, University Students Associations System, and Information Retrieval system).
4. Reflect on the implementation and record learning from the methodology application in order to guide further applications.
5. Reflection on the framework as an approach of business domain modelling and implementation.

IV. The SDDD Framework

The SDDD framework [21] is briefed here to relate it with the evaluation in order to facilitate the understanding process of the reader. SDDD was developed into an action research intervention based on research of multimethodology, which justifies combining methods for the same business intervention [15]. It is a multi-method framework which intended to guide the developer through an investigation of a problematic situation. The purpose here is to insure that a comprehensive understanding is achieved in order to facilitate the modelling and implementation of the domain-driven business processes as a software support system. The modelling will produce an object-oriented domain- driven model as the bases of developing the software support system. As mentioned in the previous work [21], the framework was been developed through a series of “action research” case studies. Accordingly our case studies have involved development projects within our own school. In this paper, a *Information Retrieval System for academic research researchers* investigated as an MSc Project to evaluate the SDDD applicability in modelling the business domain processes into object-oriented domain model. The researchers are part of the school and they are participating in the daily activities related to the case studies. They supervised the students and guided them to the final stage of the projects and teaching courses related to business domain modelling and implementation.

The SDDDF Framework (Fig. 1) is focused on modelling and implementation of the domain-driven business process as a software support system. SSM is used as a guiding and learning methodology with techniques including UML and implementation pattern (Naked Object or True View) embedded within it. The DDD philosophy is adapted to generate a “Soft Language” (SL) as a compliment of “Ubiquitous Language” (UL) and it used as an input to the next stages. The implementation pattern is used after the generation of the final refined change report which is an input to the implementation process.

Using [15] generic model which discussed in [21], the SDDD framework consists of four phases and each phase consists of a group of activities. The framework satisfies the generic process of conducting an action research in the business intervention. SDDD represented in Fig.1, Fig. 2 represents the conceptualization of the framework, and Fig. 3 represents the logical processes embedded in it. For more details about these phases refer to our previous works [21], [22].

V. The development of Information Retrieval System using SDDD Framework

Since we have been engaged in an information systems development project using SSM and UML techniques within an agile framework to make recommendations about the development of an intranet for the academic school in which we are employed. At the beginning of the project the department had an operational intranet but this was not widely used. An information system strategy was initiated to investigate ways in which the intranet could be developed to support the university mission and departmental goals. Initially we used use cases as the primary fact-gathering technique but certain limitations in this approach led us to a more thorough SSM-based analysis of the situation. We argue that the techniques of SSM can help the developer to identify a richer set of use cases than would otherwise be possible but developers with a full use case model still have many challenges ahead of them. We are interested in object oriented design and the view that all business behavior identified in the use case model should be encapsulated as methods on domain objects. Thus, a Student object should not just be a collection of data about the Student; it should encapsulate all the behaviors that we need to apply to a student. In Domain-Driven Design these are often referred to as 'behaviorally-rich' domain objects. A number of software systems required supporting the department and one of these is Information Retrieval System for academic research researchers. The information retrieval system we proposed to design is that which provides academics the necessary facilities to aid in the precise and relevant information retrieval (IR) from databases across the internet. IR can be undertaken by the academics by using several different navigation paths returning the same significant data retrieval. The information retrieval system aims to provide academic students and staff to submit search criteria utilizing various options including general search, advanced searching, and Tag based operations. Furthermore students and staff can add tags to documents which can be used as another means of document navigation as well as information retrieval. The effectiveness and efficiency of this system is entirely dependent on the financial constraints of the Universities subscriptions and the ability of the academics to provide effective tagging.

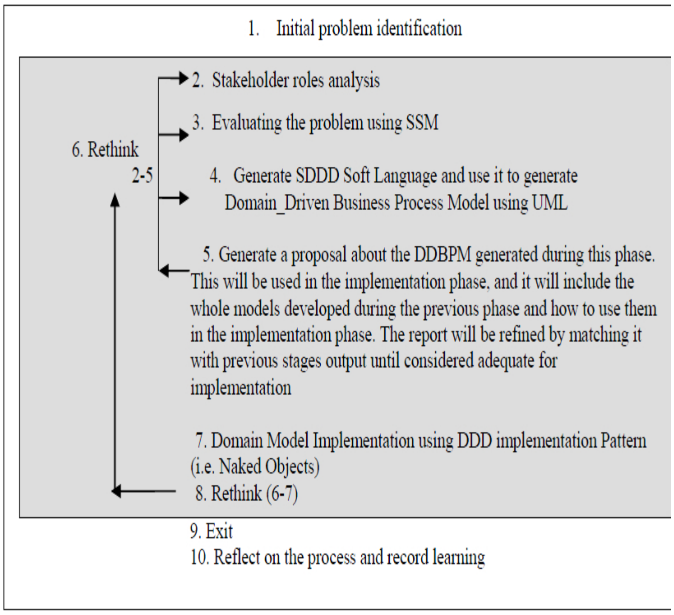
1. Initial problem identification
2. Stakeholder roles analysis
3. Evaluating the problem using SSM
6. Rethink
2-5
4. Generate SDDD Soft Language and use it to generate Domain_Driven Business Process Model using UML
5. Generate a proposal about the DDBPM generated during this phase. This will be used in the implementation phase, and it will include the whole models developed during the previous phase and how to use them in the implementation phase. The report will be refined by matching it with previous stages output until considered adequate for implementation
7. Domain Model Implementation using DDD implementation Pattern (i.e. Naked Objects)
8. Rethink (6-7)
9. Exit
10. Reflect on the process and record learning

Fig. 1 The SDDDF Model

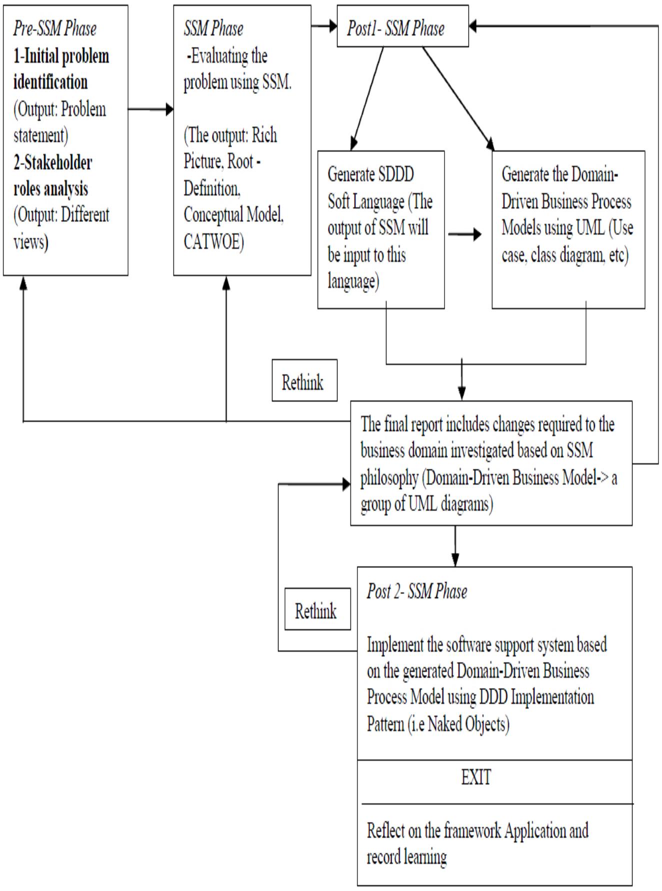
Pre-SSM Phase
1-Initial problem identification
(Output: Problem statement)
2-Stakeholder roles analysis
(Output: Different views)
SSM Phase
-Evaluating the problem using SSM.
(The output: Rich Picture, Root -Definition, Conceptual Model, CATWOE)
Post1- SSM Phase
Generate SDDD Soft Language (The output of SSM will be input to this language)
Generate the Domain-Driven Business Process Models using UML (Use case, class diagram, etc)
Rethink
The final report includes changes required to the business domain investigated based on SSM philosophy (Domain-Driven Business Model-> a group of UML diagrams)
Rethink
Post 2- SSM Phase
Implement the software support system based on the generated Domain-Driven Business Process Model using DDD Implementation Pattern (i.e Naked Objects)
EXIT
Reflect on the framework Application and record learning

Fig. 2 The conception of SSDDF

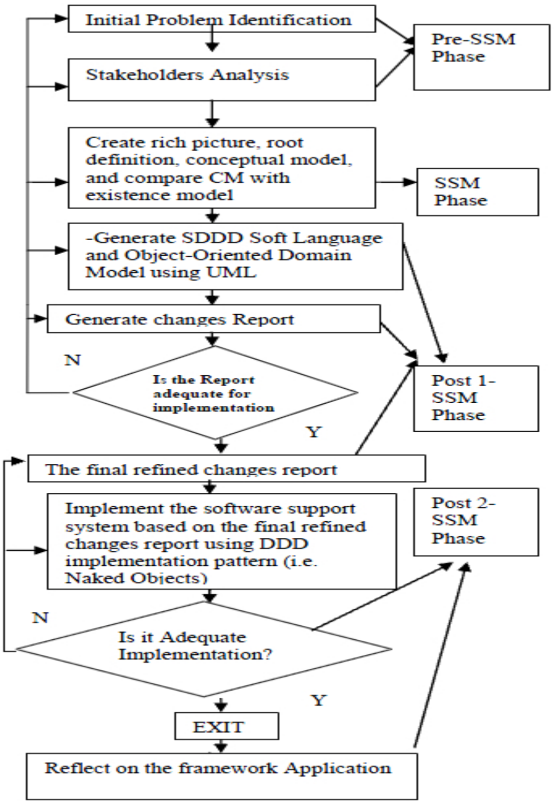
Initial Problem Identification
Stakeholders Analysis
Pre-SSM Phase
Create rich picture, root definition, conceptual model, and compare CM with existence model
SSM Phase
-Generate SDDD Soft Language and Object-Oriented Domain Model using UML
Generate changes Report
N
Is the Report adequate for implementation
Y
Post 1- SSM Phase
The final refined changes report
Implement the software support system based on the final refined changes report using DDD implementation pattern (i.e. Naked Objects)
Post 2- SSM Phase
N
Is it Adequate Implementation?
Y
EXIT
Reflect on the framework Application

Fig. 3 SSDDF logic Pre-SSM Phase

1. The Pproblem Identification:

It is mentioned in the previous work [21] that the Department of Informatics in the School of Computing and Engineering at the University of Huddersfield in UK needs to investigate and develop certain systems to support he staff and students In previous publications some of these applications investigated using SDDDF and this one will investigate the Information Retrievals System requested by the staff and students.. As an action researchers, we conducted the interviews in an informal way and done face to face so that the participants will be feeling comfortable during the interview as they can see who is interviewing them hence also giving their ideas and suggestions comfortably. With some ideas, participants were better explained by face to face interviews for example expressions which may not be fully explained in writing or over the phone. These actions were being noted throughout the interviews and appreciated unlike over the phone that these cannot be appreciated.

The interviews were targeted to collect the following information:

- What is the current system offering? (Lecturers & students).
- How far the text box can go in supporting them with their works in classroom hours? (Lecturers)
- What do they think of IRS? (both)
- How can IRS support them to do their assignments and projects? (students)
- Would they need more information to support on the work to increase their lecturing skills? (Lecturers).

2. Stakeholders Determination:

The stakeholders defined in this case are the people that will be using the system, and who will benefit from it. The stakeholders of the required IRS system were determined to be students, academic staff,, and department management. The stakeholders have different expectations of the system. The different stakeholders of this system expected that they can achieve the following from using it:

- Support them to search and retrieve latest information
- Minimize the time required for looking of extra information resources

B. SSM-Phase

1. Investigating the Problem Situation Using a Rich Picture

A rich picture is a drawing that graphically illustrates the issues expressed by people, processes involved in the transformation, people involved with or affected by the change (stakeholders), working climate, conflicts and structures within the change process [25].

In order to develop a rich picture of the situation under study, a number of information sources were used to capture views of Information Retrieval System from the perspective of the management the school of computing and engineering, lecturers, and students. Interviews with the school administration and groups of students were conducted to understand the problematic situation of getting the proper information required to deliver different courses and set out suggestions to solve the problems. The following Fig. (4) represent the Rich Picture of IRS.

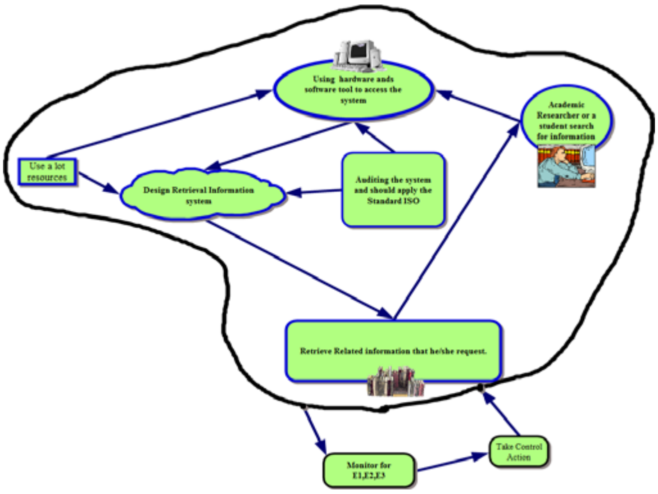
Using hardware ands software tool to access the system
Academic Researcher or a student search for information
Use a lot resources
Design Retrieval Information system
Auditing the system and should apply the Standard ISO
Retrieve Related information that he/she request.
Monitor for E1,E2,E3
Take Control Action

Fig. 4 IRS Rich Picture

2. Modelling the Relevant System Using SSM (Root Definition & Conceptual Model)

Modelling the system using Root Definition is described by [9] as a movement from the real world to systems thinking about the real world. [25] mentioned that during root definition stage, points of views from the different stakeholders are drawn out from the rich picture and have them in a structured development process. Root Definition (RD) of the IRS is as follows:

"*The information retrieval system aims to provide academic students and staff to submit search criteria utilizing various options including general search, advanced searching, and Tag based operations. Furthermore students and staff can add tags to documents which can be used as another means of document navigation as well as information retrieval. The effectiveness and efficiency of this system is entirely dependent on the financial constraints of the Universities subscriptions and the ability of the academics to provide effective tagging*".

Root definition has been used to extract the conceptual model which represents the different stakeholder views. So the conceptual model describes activities that might take place if the relevant root definition was to be an accurate representation of the work of a system. The following Conceptual Model of IRS (Fig. 5) represents the different stakeholders' views, the actions that must be taken based on their views, and also meeting their particular cultural, political and social requirements of the system. All of these issues are expressed in the rich picture and modelled using the conceptual model.

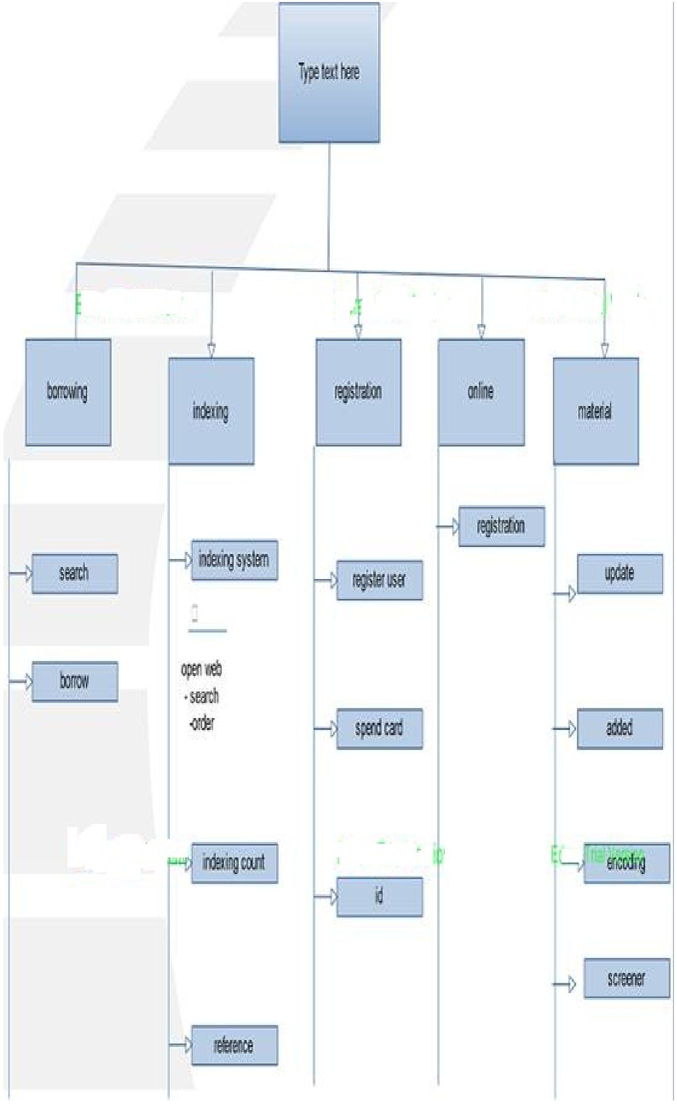

Fig. 5 The Conceptual Model of IRS

Rich picture, root definition, and conceptual models represents the output of the SSM phase, and it's called "Soft Language".

C. Post1-SSM Phase: Moving from Soft Language (SSM Phase) to Domain Model

Domain Model will be represented using UML. Domain modeling starts with the conversion from the Conceptual Model into Use Cases and Use Case modeling. The extracted

Use cases will be used to develop UML Sequence Diagram, Class diagram, and Activity Diagrams development. The next subsection will show the conversion from CM to Use Cases.

1. Moving from SSM Conceptual Model UML Use Cases

The conversion process is presented in Fig. 9. Any activity required software support will be selected as a use case. The stage of moving from an SSM conceptual model to a use cases l is not as straightforward as this high-level discussion would suggest. In thinking this through we have been pushed towards making a clear distinction between stakeholder goals, business activities and use cases. The following model (Fig. 6) shows the relationship between these key abstractions.

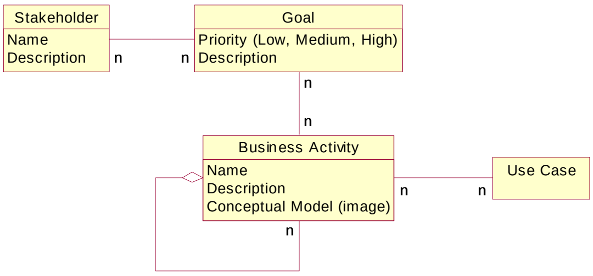

Fig. 6 Moving from an SSM to use case diagram

Using the above conversion algorithm, the conceptual models presented above converted into different use case. The following Use Case diagram (Fig. 7) presented as a result of the conversion process. This figure show the identified use cases and actors within the search function of the system. The 2 actors are the academics and the collection of databases. It can be clearly seen that the actor can do 2 types of search; query based and tag based. Query search can be further split up into its types of search, these are: general search and advanced search. The Boolean search is incorporated into the advanced search. The actor can type words into the tag search bar and these will be checked against the current "Doc Tag Index" to see if it matches. This is equivalent of having auto suggestive tags matching the partial string the actor is inputting. The actor can also click on existing tags in the cloud and these will automatically bring up the list of documents attached to that tag.

These use cases described using use case proforma which is a template to show the logic embedded in each use case. Fig. 8 show a sample of use case proforma to represent the general search process.

1. Generating the Aactivity Diagrams

Fig. 9 shows the activity diagram of the search functionality in the information retrieval system. The actors can search in various ways: general search, advanced search and tag based search. The general search is just a basic keyword search, the advanced search uses Boolean operation (AND mainly) in having more than one type of search criteria, i.e. searching for an Author AND Publication Year. Finally tag based searching firstly allows the actor to search through already processed tags attached to documents and they are used in auto completion when typing in the tag field.

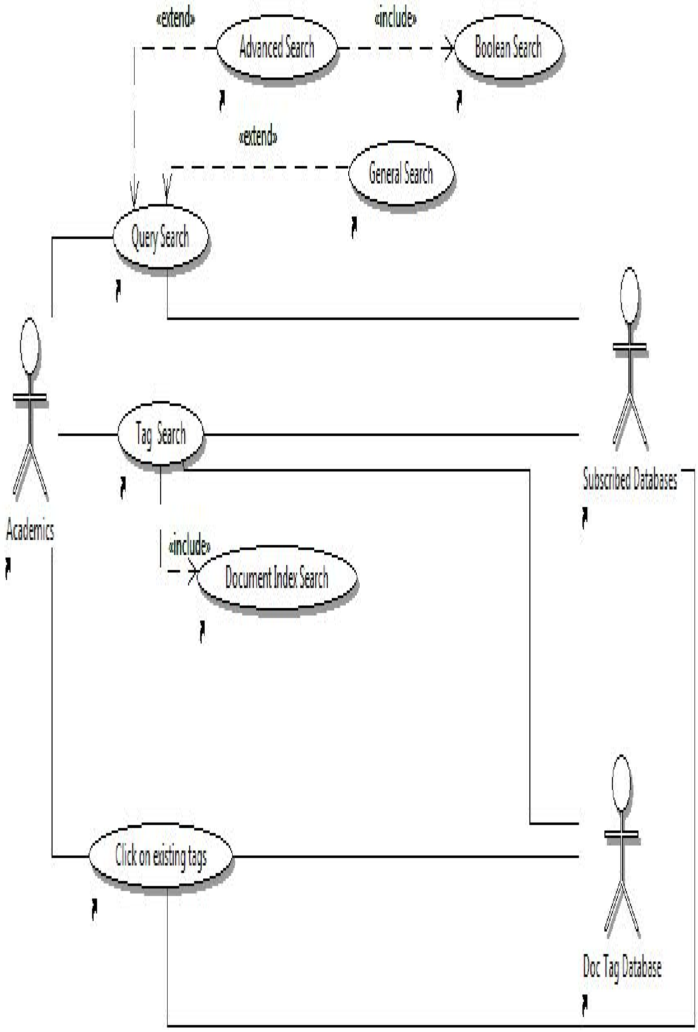

Fig. 7 The use case diagram of IRS

2. Generate Class Diagram

A class diagram is a representation of a basic structure of a system, it shows what classes will be present in the system, how the classes are going to link between themselves and how many links to one class there would be from another. It's a presentation of the system to more detail. Each use case presented using textual template, activity diagram, sequence diagram, and all use cases are combined in a use case diagram. The next step in the process is to take the business logic identified in the use cases and associate it with classes in a class diagram. We have followed the guideline that all important business logic must be implemented in classes in the domain model. Class diagram is the major part of the Domain Model that can be used to generate the programming code through the implementation pattern. Class diagram of RIS is presented in Fig. 10.

Use Case Number: 3	**Use Case Name: Tag based search**
Overview: This use case will be executed daily and concurrently by hundreds of actors a day. This use case allows the actors to perform searches by tags inputted. An auto suggestive tag function will match up the string the actor is inputting in real time.	
Actors: Academics.	
Frequency of execution: This use case will be run daily.	
Scalability This process will have several instances running at one time.	
Criticality: This use case is extremely critical again for this type of search. With this functionality down it may affect other critical systems for the tag cloud functionality.	
Primary Path: 1) Actor decides to do a tag search 2) Actor begins to enter tag into the text field. 3) Auto suggestive tagging searches tag index for matching tag. 4) Suggested tag matches actor input, 5) Actor clicks on tag, 6) Doc Tag Index is checked to gather all documents attached to the queried tag. 7) The list of documents are displayed to the actor and ready to view.	
Use Cases related to Primary Path: Document Index Search. This is because as the actor enters the tag, it automatically tries to match it up with an entry in the Doc Tag database.	
Alternatives: 4.1 – If no tags are matched then the auto suggestive tagger will simply display “No tags found” and allows the actor to perform another tag search or a different search type.	
Use Cases related to Alternatives: None	
Exceptions: None	
Use Cases related to Exceptions: None	

Fig. 8 Use case proforma of general search

3. Change Report Generation and Refinement

As shown in the framework (SDDD), there is a draw back to the previous stages to refine what’s done during Pre-SSM, SSM, and Post1-SSM. This refinement is essential to be sure that the exact changes required already modelled well as a domain model. As a guiding methodology, SSM focus on the generation of the required change report as a result to be recommended for the management actions [8]-[10].

So, before leaving this stage, Domain model supposed to be refined and ready for implementation.

A. Post2-SSM Phase: Software Implementation

SDDD framework considered the domain model as the base to extract the programming code using the implementation pattern. Naked objects [20] and True view are recommended as implementation patterns. For the purpose and size limit of this paper implementation not presented. The paper considered the modelling part.

VI. Reflections on the Framework

Our work in applying the framework to a series of real-world development projects has focused our attention on a number of issues that we had not considered at the outset. Some of these present difficulties for the further development of the framework which present opportunities for further research. Some of these will be briefly discussed in this section.

A. Role of Re-Use and Design Patterns in Domain Modelling

Our approach tries to preserve as much “soft” information as possible in the evolving domain models. Inevitably some of this information is lost as we move from approaches that try to model what “people” are doing (including activities that do not require software support) through to program code. At present our framework leads to development of a bespoke software system based on a rich object-oriented domain model. In practice many software developers make use of reusable software components or wish to design software with an eye to future reuse. There is clearly a tension between our emphasis on a bespoke solution and the software developers’ objective of developing generic, reusable software solutions.

B. Representation of Implicit Information in the Domain Model

The conceptual models in SSM do not have rigorous syntax. We have discovered that when developing the conceptual models people often include information in, for example, the sequence of activities or the knowledge required to carry out certain activities which is lost when we move into the use case and object models. We are attempting to develop clear guidelines for identifying this type of information and what should be done about it. One possibility is that we develop our own version of conceptual models that do include a more prescriptive notation.

C. Ambiguity in the Definition of “Business Process”

One of the issues that we have confronted is the lack of consensus about precisely what can be defined as a business process and what cannot. SSM has a number of techniques for capturing multiple stakeholder perspectives on what the key business processes are and how they should be monitored. We want to preserve these multiple perspectives for as long as possible into the development process. At present we take the Consensus Primary Task Model produced in SSM to be an objective description of what is required but we have found that it is often difficult to gain consensus in developing this model and then to preserve that consensus as we move on.

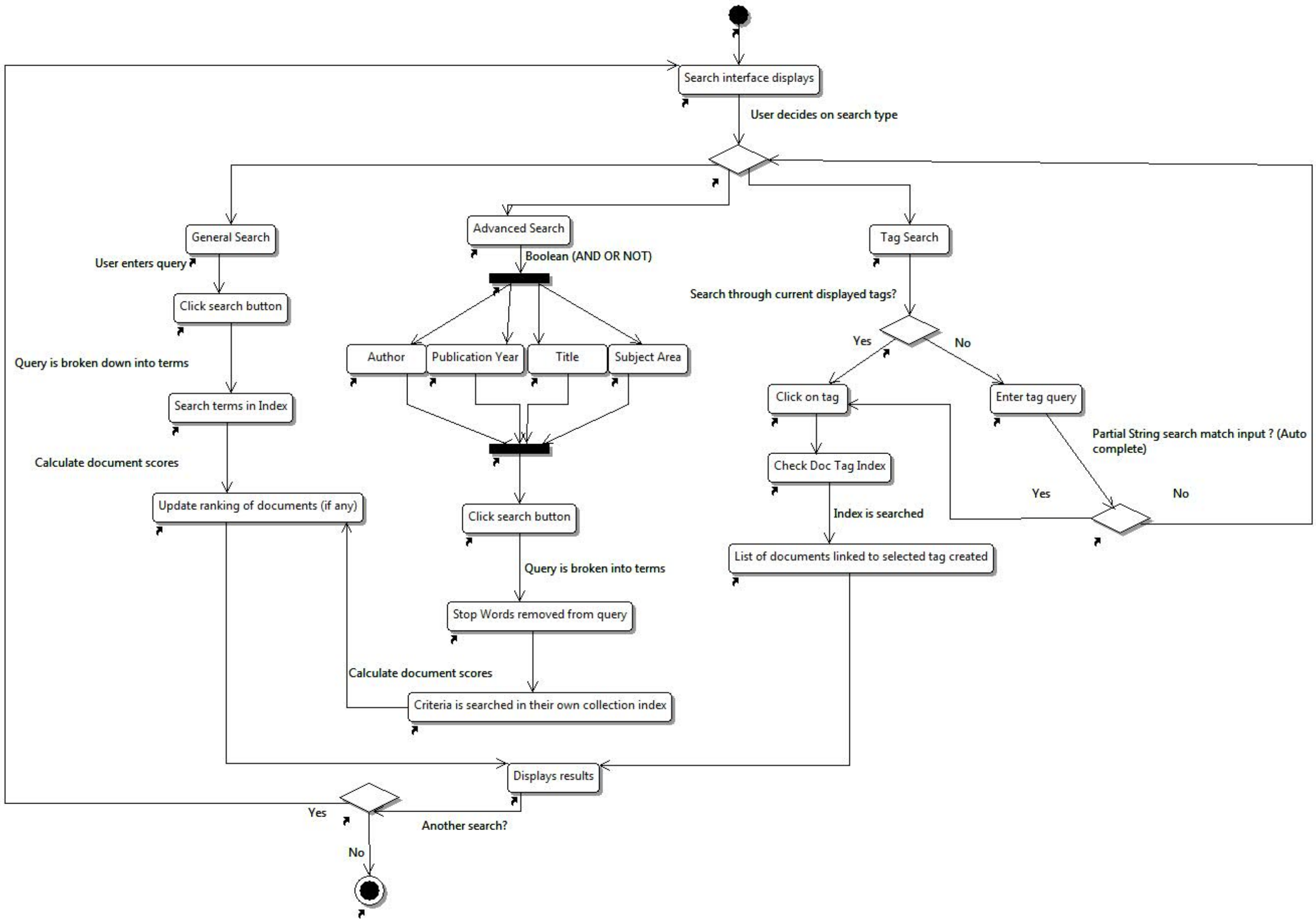
Search interface displays
User decides on search type
General Search
User enters query
Click search button
Query is broken down into terms
Search terms in Index
Calculate document scores
Update ranking of documents (if any)
Advanced Search
Boolean (AND OR NOT)
Author
Publication Year
Title
Subject Area
Click search button
Query is broken into terms
Stop Words removed from query
Calculate document scores
Criteria is searched in their own collection index
Tag Search
Search through current displayed tags?
Yes
No
Click on tag
Enter tag query
Check Doc Tag Index
Partial String search match input ? (Auto complete)
Yes
No
Index is searched
List of documents linked to selected tag created
Displays results
Yes
Another search?
No

Fig. 9 Activity Diagram

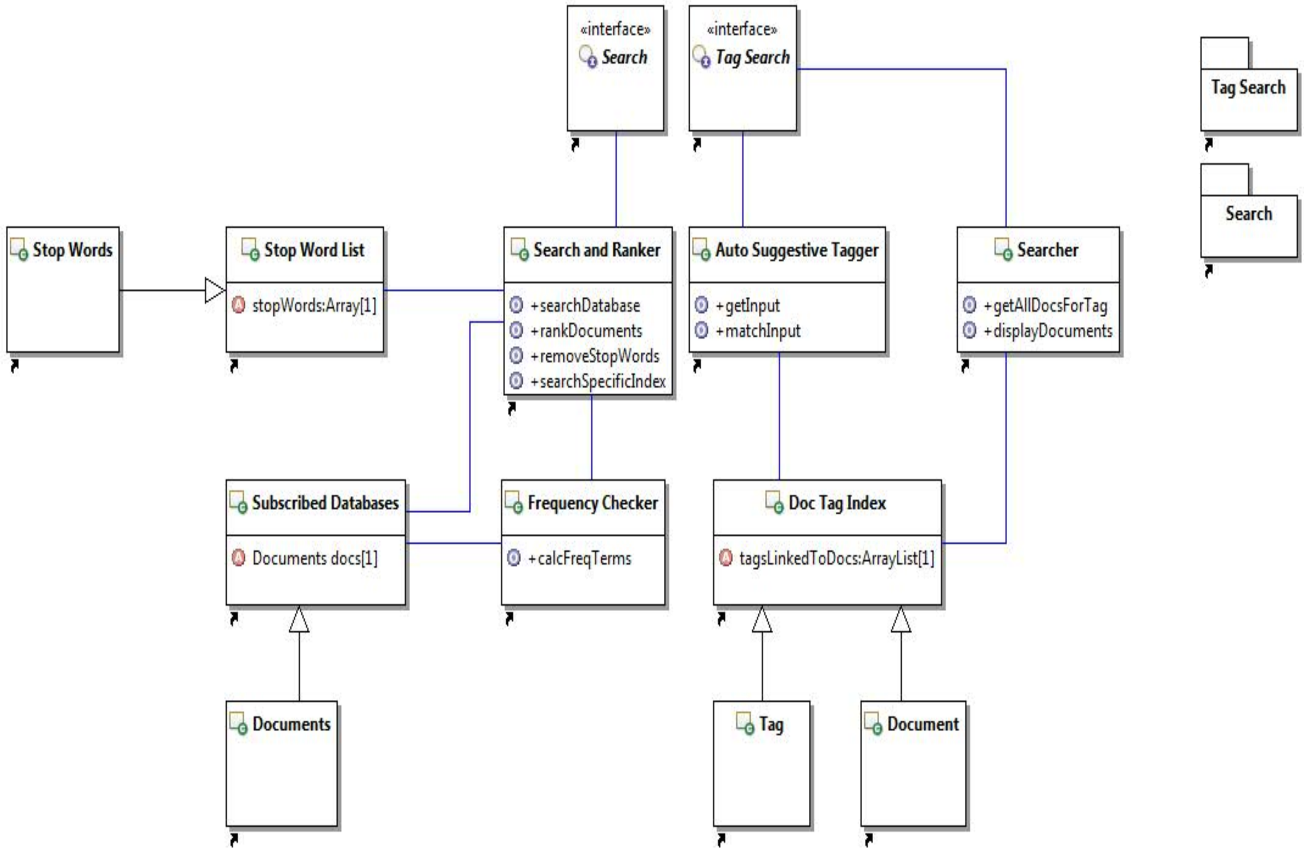

Fig. 10 The Class Diagram of RIS System

VII. Conclusion and Future Work

The work done in this paper reviewed and highlighted the need for a multimethodology framework that can handle both soft and hard issues of domain business process modelling and implementation as a software support system. The new proposed framework is developed based on the idea of Domain-Driven Design (DDD) and Soft Systems Methodology (SSM). We have added a “soft” perspective on DDD to form “Soft Domain-Driven Design”. The approach is described as a systemic framework for domain business process modelling and implementation. The framework is proposed and justified as a multimethodology framework, incorporating guiding steps through various key stages in the development process. The framework is being evaluated and further developed in an action research program. We briefly provided the example of a “Information Retrieval System” (RIS) case study to show how the proposed framework can be applied to a real problem situation. The evaluation already done before for other projects like “Peer Tutoring System”, “Combined Studies Program Development” (CSPD), and the “Placement Unit Management System” within our institutions. All evaluation results show the applicability of the framework as a domain modelling and implementation approach for ISD projects. Also pedagogical evaluations show the applicability of the framework as a teaching approach.

References

[1] Joseph Barjis, “The importance of business process modelling in software systems design”, Science of Computer Programming Journal, vol 71, pp 73–87, 2008.

[2] Alter, S., “The work system method: Connecting people, processes and IT for business results”, Work System Press, Larkspur, CA, 2007.

[3] Sewchurran, K. & Petkov D, “A systemic Framework for Business Process Modelling Combining Soft Systems Methodology and UML”, Information Resources Management Journal, 20, 3, IGI Publishing, PA,USA, P. 46-62., 2007.

[4] Salahat, M., Wade, S., Lu, J., A systemic Framework for Business Process Modelling and Implementation, In the proceeding of 5th International Conference on Innovations of Information Technology (Innovations’08), UAE University, Al Ain, UAE, in IEEE explore 978-1-4244-3397-1/08., 2008.

[5] Mohammed Salahat, Steve Wade. A Systems Thinking Approach to Domain-Driven Design. In the proceeding of UKAIS2009 conference, Oxford University, Oxford, UK, 2009.

[6] Eric Evan, Domain-Driven Design –Tackling Complexity in the Heart of Software, Addison Wesley, 2004.

[7] Al Humaidan, F., “Evaluation and Development Models for Business Processes”, PhD thesis, University of Newcastle, UK, 2006

[8] Checkland, P., and Poulter J., “Learning for Action. A short Definitive Account of Soft Systems Methodology and its use for Practitioners, Teachers and Students”, John Wiley and Sons Ltd, West Sussex, England, 2006.

[9] Checkland, P., “Systems Thinking, Systems Practice”, John Wiley and Sons Ltd, West Sussex, England, 1999.

[10] Checkland, P. and Holwell, S.E., “Information, Systems and Information Systems, Making sense of the field”, John Wiley and Sons Ltd, West Sussex, England, 1998.

[11] Bustard, D. W., Dobbin, T. J., and Carey, B. N., “Integrating Soft Systems and Object-Oriented Analysis”, IEEE International Conference on Requirements Engineering, Colorado Springs, Colorado, pp. 52-59, 1996.

[12] Wade, S. and Hopkins, J., “A Framework for Incorporating Systems Thinking into Object Oriented Design” Seventh CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD’02), Toronto, Canada, May, 27-28,2002.

[13] Al-Humaidan, F., & Rossiter, N.,” Business Process Modelling with OBPM combining soft and hard approaches”, in Proceeding of 1st Workshop on Computer Supported Activity Coordination (CSAC), 6th International Conference on Enterprise Information Systems, Porto,, pp 253-260, 13-14 April,, 2004.

[14] Erikksonn, H. E., & Penker, M., “UML business process modelling at work”, John Wiley and Sons, New York, 2000.

[15] John Mingers, “Combining IS Research Methods: Towards a Pluralist Methodology”, Information Systems Research, 12, 3, Institute for Operations Research and the Management Sciences (INFORMS), pp. 240-259, 2001.

[16] D. Platt, “Process Modelling and Process Support Environment to Design Management”, Department of Civil Engineering, Faculty of Engineering, University of Bristol, UK, 1994.

[17] Daveport, T. h. Process innovation: Reengineering work through information technology, Harvard Business School Press, Boston, Mass, 1993.

[18] Warboys, Brian, Kawalek, Peter, Robertson, Ian, and Greenwood, Mark, “Business Information Systems-A process approach”, McGraw-Hill, UK, 1999.

[19] Svatopluk Štolfa, Ivo Vondrák, “Mapping from Business Processes to Requirements Specification”, Retrieved on 7th Aug, 2008 from 85.255.195.219/conf/esm/esm2006/abstract.pdf

[20] Pawson R. & Mathews R., “Naked Objects”, John Wiley and Sons Ltd, West Sussex, England, 2002.

[21] Salahat, M., Wade, S., Ul-Haq, I.(2009) The Application of A systemic Soft Domain Driven Design Framework, WASET online Journal, Issue57,pp476-486.

[22] Salahat, M., Wade S. (2012) Pedagogical Evaluation of S systemic Soft Domain-Driven Design, In the proceeding of UKAIS2012 conference, Oxford University, Oxford, UK.

[23] Wade, S., Salahat, M., Wilson, D.(2012) A Scaffolded Approach to Teaching Information Systems Design. ITALICS Volume 11 Issue 1 June 2012 - ISSN: 1473-7507

[24] Salahat, M., Wade S. (2014) Teaching Information Systems Development Through an Integrated Framework, In the proceeding of UKAIS2014 conference, Oxford University, Oxford, UK.

[25] Williams B. (2005) *Soft Systems Methodology* [online] available at: <http://users.actrix.co.nz/bobwill/ssm.pdf> [accessed on 18th June 2010].