Domain Driven Design vs Soft Domain Driven Design Frameworks

Mohammed Salahat, Steve Wade

Abstract—This paper presents and compares the SSDDD "Systematic Soft Domain Driven Design Framework" to DDD "Domain Driven Design Framework" as a soft system approach of information systems development. The framework use SSM as a guiding methodology within which we have embedded a sequence of design tasks based on the UML leading to the implementation of a software system using the Naked Objects framework. This framework has 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, a comparison between SSDDD and DDD is presented in this paper, to show how SSDDD improved DDD as an approach to modelling and implementing business domain perspectives for Information Systems Development. The comparison process, the results, and the improvements are presented in the following sections of this paper.

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

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).

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 framework SDDD was presented in the previous works [21], [24] and in this paper a comparison between SDDD and DDD, as an "Information Systems Development Frameworks", will be presented.

The paper will be presented as follows: Section II reviews related work. Section III briefed the research methodology used. Section IV is introduced the framework as a multimethodology approach. Section V is a comparison between SSDDD and DDDD. Section VI 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].

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 DDD. They can help work with domain experts to understand business entities, relationships and the business' UL and to write classes using .NET and the Naked Objects or True View framework.

C. Soft Domain Driven Design

SDDD [21], [26] 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, UL 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 SSM. 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 [21], [26], the idea of a UL into a "Soft Language" has been adapted which incorporates 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 a compliment of the UL 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.

As described in [21], [26] 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 DDD. 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 [4], [5], [21], [26]. 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 it 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:

- What are the differences between SDDD and DDD as ISD frameworks?

The approach of Action Research is used since we can play the role of actors of any system in the education environment (the educational domain) to apply the research.

We found from teaching systems modelling and design and from the literature review that many software systems failed 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 without giving a sufficient attention to the business process modeling of the business domain. To investigate and model any business domain needs a methodology or framework that can be used by the business experts and the technical people (different stockholders) and facilitate the communication between them. Among this, DDD is dominant but the communication still 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 [21], [26] and evaluation from the development perspective

were taken place.

A comparison criterion, to compare between SDDD and DDDD, is identified and the comparison process and results are presented. The following briefed methodology was followed to apply the research as it is designed in order to answer the above research question.

1. Review the current situation of business domain processes modelling through teaching and literature review.
2. Formulate comparison criteria to compare between SDDD and DDD.
3. Use the SDDD evaluation results presented in the previous works [21], [22], [24], [26] to be used in the comparison process.
4. Show the achievement of SDDD to enhance DDD.
5. Reflect 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. 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 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; and Fig. 2 represents the conceptualization of the framework. For more details about these phases refer to [21], [22].
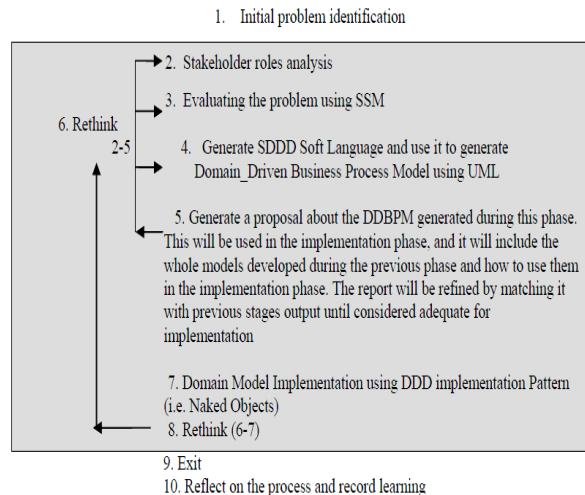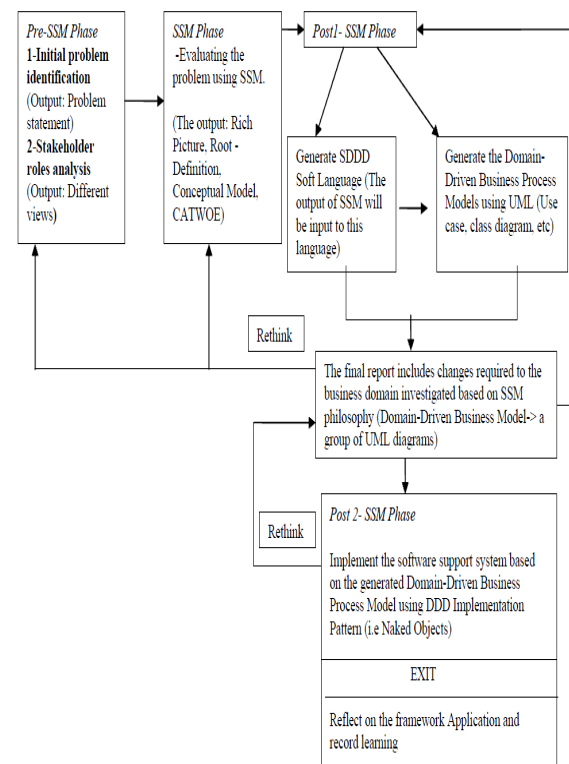

Fig. 1 The SDDDF Model

Fig. 2 The conception of SSDDF

V. THE COMPARISON BETWEEN SDDD AND DDD AS AN ISD FRAMEWORKS

In both frameworks, business domain perspectives are modelled and implemented into a software system to support different organizational functions. Business domains, and the software systems implementing them, consist of 'hard' and 'soft' perspectives. In order to make a comparison between DDD and SDDD, these perspectives have been formalized as described in Subection A. This formalization enables these perspectives to be used as the basis of the comparison, which

considers the frameworks as approaches for modelling and implementing the business process perspectives of any business domain. The comparison will be presented in Subsections *B, C, D* by identifying the business domain perspectives, and show how DDD and SSDDDF respectively handle each perspective through the modelling and implementation of a business domain.

A. Business Domain Perspectives

Different authors agree that the business process of any business domain comprises different perspectives [25], [18]. These perspectives are identified as functional, organizational, behavioral and informational perspectives. These perspectives have been adopted by other researchers and used to model and implement business processes of the business domain [7], [21]. This paper will briefly present these perspectives and introduce a new 'soft perspective', as suggested and used by [6] to model the business process as a workflow system. In this research, the business process has been modelled using SDDD as a 'business domain system' to be used for implementation. Then, the way in which these perspectives are handled by both DDD and SDDD will be presented in tabular form. The comparison will use these tabulations to reach a conclusion about the performance of DDD and SDDD as approaches to modelling and implementing the business process of the business domain. Table I represents business process perspectives 2-4, as presented by [25] and [18], then adds the soft perspective (no.1) proposed by [7] and [21], which includes SSM to model the soft perspective. In addition, the implementation perspective (no.6) is proposed by this research in order to include an implementation pattern. The soft and implementation perspectives included in this table are based on the notion of modelling and implementing the 'business process of the business domain' as 'a business domain system', as previously presented in [21].

B. Modelling and Implementing 'Business Domain' Perspectives Using DDD

Domain-Driven Design is a software development approach to the investigation, modelling and implementation of any investigated business domain. It consists of different layers, and aims to concentrate on the domain layer before starting the implementation. The different business process perspectives are presented in Table I, and DDD can handle these perspectives up to different levels. All the business perspectives, except implementation, belong to the domain layer. The other DDD layers (interface, application and infrastructure) belong to the implementation perspective. Thus, the domain layer contains the concepts of the business domain, business rules and use cases, the state and behavior of business entities and information about the business situation. The domain layer attempts to model the business domain into a 'domain model' that can be implemented through the implementation layer using any implementation pattern. Table II presents how DDD handles each of the business domain perspectives.

C. Modelling and Implementing 'Business Domain' Perspectives Using SDDD

SDDD is a proposed new framework designed to enhance the DDD approach by handling the soft issues of the business domain. The application of the framework, and how it handles the processes within the business domain perspectives, is presented in Table III. Based on this comparison of the two frameworks as a development approach, Subsection *D* will evaluate both approaches to determine whether the use of SDDD as an enhanced framework has achieved the intended improvement of DDD.

D. Comparing DDD with SDDD as an 'Information Systems Development' Approach

DDD and SDDD were compared on the basis of the modelling and implementation of 'business domain' perspectives. Table I presents a summary of these perspectives, Table II presents how DDD handles these perspectives, and Table III presents how they are handled by SDDD. The comparison between the two is presented in Table IV. The schema used to compare DDD and SDDD was developed based on the work of [7] and Likert scale values. The schema was defined as:

1- 4 points: if the framework handles all issues of the business domain perspective
2- 3 points: if the framework handles more than half of the issues of the business domain perspective
3- 2 points: if the framework handles at least half of the issues of the business domain perspective
4- 1 point: if the framework handles less than half of the issues of the business domain perspective
5- 0 points: if the framework does not handle any of the issues of the business domain perspective

First of all, neither approach can be considered as 100% perfect to do the job. Further improvements are still required and there is a need for future research to handle outstanding issues. The allocation of points is explained and justified below:

1- The soft perspective is handled completely through SSM techniques, which support the users' involvement in determining the problem and stakeholders' roles, and the investigation of the problem through the development of the rich picture, root definition, conceptual models and the CPTM. The use of feedback and acceptance of the models being developed is important before proceeding to UML modelling and DDD implementation patterns. Based on this, SDDD was given a score of 4. In contrast, DDD does not adopt SSM. Thus, while user involvement is still available, it cannot be guaranteed that users will be able to understand all the methods and techniques used to develop the domain model. It is estimated that users may be able to understand half of these but not all, so the score given here is 2.

2- The organizational perspective is handled by both DDD and SSDDD through UML modelling techniques. Since this perspective focuses on who will perform the business process activities and where (the organizational structure),

the use case diagram represents these activities and their actors. In addition, this perspective can be modelled using the class diagram by assigning tasks to users using the role concept. SDDD uses use case and class diagrams, while DDD uses class diagram only. Both approaches are therefore given 4 points because they model this perspective using UML tools.

3- The behavioral perspective is handled by SDDD through SSM and UML modelling techniques. Since this perspective deals with timing of the execution of business processes, the sequence diagram (timing) and activity diagram are used to model all activities depicted in the use case diagram. The SSM conceptual model deals with this perspective partially, but detailed modelling is done by UML (sequence and activity) diagrams. In contrast, DDD depends only on the class diagram, which can show the behavior of these activities but focuses more on data, such as entities, types of data, data structure, etc. For this reason, SSDDD is given 3 and DDD is given 2. This thesis believes that behavior cannot be standardized or fixed, as a variety of circumstances may occur which cause directions to be changed.

4- The informational perspective deals with the informational entities required (entities within the structure and their relationships), so the tabulation of activities presented in use case proformas and class diagram are used to model this perspective. Both DDD and SDDD use the UML class diagram to model this perspective. Based on this, 3 points are given for both approaches. As some information is still not recognized by either of the approaches, neither can be considered complete.

5- The functional perspective deals with business process activities and information flow, and these activities are depicted in SSM conceptual models and modelled using the UML activity diagram. The SDDD framework models this perspective using both SSM conceptual models and the UML activity diagram, but DDD depends on the class diagram, which partially or indirectly depicts these functions. Because of this, SDDD is given 4 points while DDD is given 3 points.

6- The implementation perspective deals with implementation of the domain model into a software support system using a DDD implementation pattern. SDDDD considers two DDD implementation patterns, Naked Objects and TrueView, while DDD leaves it open for users to select the implementation pattern from among a range of different available patterns. Based on this, both SDDD and DDD perform the implementation perspective and because of this, both are given 3 points. However, some of the students who developed projects during the evaluation period complained about SDDD restricting them to the use of these two implementation patterns; they said the choice of options should be kept open because it would take them more time to master new patterns.

Overall, SDDD earned 21 out of 24 points while DDD earned 16 out of 24 points. Therefore, the enhancement of DDD as a software development approach was achieved. The improvement percentage was calculated as follows:

The performance of SDDD was calculated as 21*100/24=87.5%, while that of DDD was calculated as 18*100/24=75%. Thus, the percentage of improvement to DDD by adopting the new SDDD framework as a software development approach is 87.5%-75% = 12.5%. There are various areas in which further improvement can be achieved, and these are presented in the following section in the form of recommendations and suggestions for future work.

VI. 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 DDD and 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. All evaluation results show the applicability of the framework as a domain modelling and implementation approach for ISD projects. In this paper, the developed framework SDDD is compared to DDD using a comparison criteria, and the comparison results are presented and it's shown that the SDDD improved the DDD framework as stated from the beginning in the research question. It's recommended that the framework SDDD will be more improved in the future by developing different pattern languages and try it in the business environment.

TABLE I
BUSINESS PROCESS PERSPECTIVES (PERSPECTIVES 2-4 BY [25] AND [18])

Perspective	Description
1-Soft	This perspective is added by this thesis to deal with soft aspects of the business process. For the SSDDD framework, this refers to the first two investigative phases: the pre-SSM phase to identify the problem and stakeholders' roles, and the SSM phase to evaluate the problem using SSM and produce 'soft language'. From this perspective, progression can be made to other perspectives through the transition process from SSM CPTM diagram to use cases. Different soft issues will be included, such as users' involvement in modelling and development of the system, different stakeholders' views, users' satisfaction, etc.
2-Organizational	This focuses on who will perform the business process activities and where (the organizational structure).
3-Behaviour	This perspective deals with the timing of the execution of business process activities (ordering), and how they can be executed.
4-Informational	Deals with the informational entities required (entities within the structure and their relationships).
5-Functional	Deals with business process activities and information flow.
6-Implementation	Deals with implementing the domain model into a software support system using a DDD implementation pattern.

TABLE II
HANDLING OF EACH PERSPECTIVE BY DDD

Perspective	How DDD handles each perspective
1-Soft	DDD partially handles soft issues through the usage of UL as a means of communication between team members to avoid misunderstanding and an inconsistent model. However, UL does not include any soft modelling tools to allow the users to participate in the development or to provide feedback and agreement about the system activities being modelled. It only facilitates communications between team members.
2-Organizational	This relates to DDD's domain layer modelling of the business processes. This perspective focuses on who will perform the business process activities and where (the organization's structure). The DDD approach achieves this by developing the domain model, which is represented by a class diagram. However, there is no indication in the class diagram of who will carry out the activities presented in the domain model.
3-Behaviour	This also relates to DDD's domain layer modelling of business processes, but this perspective deals with the timing of the execution of business processes. The DDD approach achieves this by developing the domain model, which is represented by the class diagram.
4-Informational	Again this relates to DDD's domain layer modelling of business processes, but this perspective deals with the informational entities required (entities within the structure and their relationships). The DDD approach achieves this by developing the domain model, which is represented by the class diagram.
5-Functional	This relates to DDD's domain layer modelling of business processes, and this perspective deals with business process activities and information flow. The DDD approach achieves this by developing the domain model, which is represented by the class diagram.
6-Implementation	This relates to the implementation layer and deals with implementation of the domain model into a software support system using a DDD implementation pattern. There are different DDD implementation patterns available, such as Ruby, Naked Objects, TrueView, JMatter, XT Framework, etc.

TABLE III
HANDLING OF EACH PERSPECTIVE BY SDDD

Perspective	How SSDDDF handles each perspective
1-Soft	Investigation starts with the pre-SSM phase to identify the problem and stakeholders' roles, followed by the SSM phase to evaluate the problem using SSM techniques and produce 'soft language'. These phases involve users by enabling them to express their views and participate in identifying the problem and the roles of stakeholders. They are then involved in analyzing the problem by constructing the rich picture and the root definition. Next, users are involved in the development of different conceptual models to represent different stakeholders' views (human activities), and in the construction of the consensus primary task model (CPTM) which includes all the activities agreed by different stakeholders. Users can recognize how the system is presented in the CPTM and compare it to what they have used in the real life system. If any amendments are required or they are not happy with this model, the team will modify it until the users are satisfied. This involvement will promote acceptance of the software system that will be developed based on the SSM modelling techniques, as it can be understood more easily by users than other, more technical methods. From this perspective, progression can be made to other perspectives through the transition process from SSM CPTM diagram to use cases. Different soft issues are handled, such as users' involvement in modelling and developing the system, determination of different stakeholders' views, users' satisfaction, etc.
2-Organizational	This perspective focuses on who will perform the business process activities and where (the structure), and the use case diagram represents these activities and their actors. In addition, this perspective can be modelled using the class diagram by assigning tasks to users using the role concept.
3-Behaviour	Since this perspective deals with the timing of the execution of business processes, the sequence diagram (timing) and activity diagram are used to model all activities depicted in the use case diagram. The SSM conceptual model deals with this perspective partially, but detailed modelling is done by UML (sequence and activity) diagrams.
4-Informational	This perspective deals with the informational entities required (entities within the structure and their relationships). The tabulation of activities, presented in use case proformas, and the class diagram are used to model this perspective.
5-Functional	Since this perspective deals with business process activities and information flow, these activities are depicted in SSM conceptual models and modelled using the UML activity diagram.
6-Implementation	This deals with implementation of the domain model into a software support system using a DDD implementation pattern. SSDDD recommends Naked Objects or TrueView as implementation patterns.

TABLE IV
COMPARISON BETWEEN DDD AND SDDD

Perspective	Business Domain Modelling and Implementation Approach/Framework	
	DDD	SSDDF
Soft	2	4
Organizational	3	4
Behaviour	2	3
Informational	3	3
Functional	3	4
Implementation	3	3
Total	16/24	21/24

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, World Academy of Science, Engineering and Technology 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] Curtis, B., et al. (1992). Process modelling. Communications of the ACM 35(9): 75-90.
[26] Salahat, M. and S. Wade (2015). Business Domain Modelling Using an Integrated Framework. World Academy of Science, Engineering and Technology organization conference, London,UK. International Science Index. Vol:9,No:6,2015.