Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 9 Issue 5, May - 2023

Architecture-Oriented Design Method For Smart Agriculture Innovative Service Systems

Shuh-Ping Sun
Department of Digital Media Design
I-Shou University
Kaohsiung, Taiwan

William S. Chao
SBC Architecture International
U.S.A

***Abstract*—This study adopts the structure-behavior coalescence (SBC) architecture as a design method for Smart Agriculture Innovative Service Systems. SBC architecture design method starts from the preparation phase and then goes through the planning, preliminary design, and detailed design phases of SBC architecture construction. SBC architecture design method uses Architecture Description Language (ADL) to formally design the essence of a Smart Agriculture Innovative Service System architecture and its details at the same time. In the planning phase, framework diagram (FD)-ADL is used. In the preliminary design phase, component channel diagram (CChD)-ADL is used. In the detailed design phase, interaction flow diagram (IFD)-ADL is used. SBC architecture design method helps integrate different stakeholders' works on the same track and unfold the backbone of Smart Agriculture Innovative Service System. Designer then can effectively design the structure, behavior, function, and data of Smart Agriculture Innovative Service System; resolve uncertainties and risks to improve the acceptance and effectiveness of Smart Agriculture Innovative Service System development.**

Keywords—Innovative Service System; Smart Agriculture; Architecture Description Language; Structure-Behavior Coalescence Architecture

I. INTRODUCTION

Due to the crisis of global warming, a smart agriculture service system is an important concern to decrease the farming carbon footprint [1]. In general, a Smart Agriculture Innovative Service System is exceptionally complex that it includes multiple views such as structure, behavior, function, and data views [2, 3]. The systems model designs the Smart Agriculture Innovative Service System multiple views possibly using two different methods. The first one is the non-architecture-oriented method and the second one is the architecture-oriented method [4][5]. Non-architecture-oriented systems model respectively picks a model for each view [13][14][15]. Architecture-oriented systems model, instead of picking many heterogeneous and unrelated models, will use only one single coalescence model [6, 7, 8, 9].

An architecture-oriented design method for Smart Agriculture Innovative Service System adopts the structure-behavior coalescence (SBC) architecture [7][8][9] as a systems model. With SBC architecture, we then can effectively design the structure, behavior, function, and data of Smart Agriculture Innovative Service System; resolve uncertainties and risks caused by those non-architecture-oriented design methods. Overall, SBC architecture design method helps integrate different stakeholders' works on the same track and unfold the backbone of Smart Agriculture Innovative Service System. The Smart Agriculture Innovative Service System design result of SBC architecture can be used as Smart Agriculture Innovative Service System design schemes to improve the acceptance and effectiveness of the development of Smart Agriculture Innovative Service System.

II. MATERIALS AND METHODS

A Smart Agriculture Innovative Service System consists of multiple views such as structure view, behavior view, function view, data view as shown in Figure 1. The systems model designs the Smart Agriculture Innovative Service System multiple views possibly using two different methods. The first one is the non-architecture-oriented method and the second one is the architecture-oriented method.

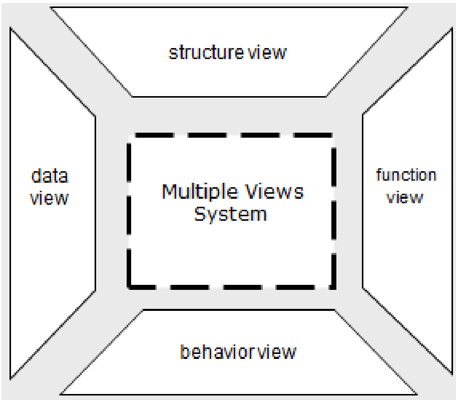

Figure 1 Multiple Views of a System

The non-architecture-oriented method respectively picks a model for each view as shown in Figure 2, the structure view has the structure model; the behavior view has the behavior model; the function view has the function model; the data view has the data model. These multiple models are heterogeneous and unrelated of each other, thus there is no way to put them into a conformity model [10][11].

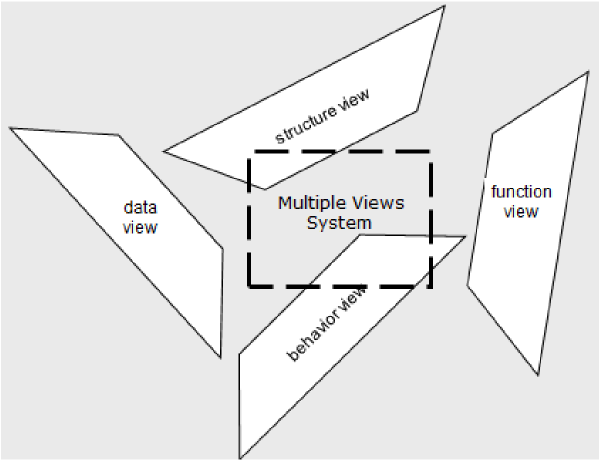

Figure 2 Non-Architecture-Oriented Method

The architecture-oriented method, instead of picking many heterogeneous and unrelated models, will use only one single coalescence model as shown in Figure 3. The structure, behavior, function, and data views are all integrated in this multiple view coalescence (MVC) systems model [4][5][6][7][8][9][12][15].

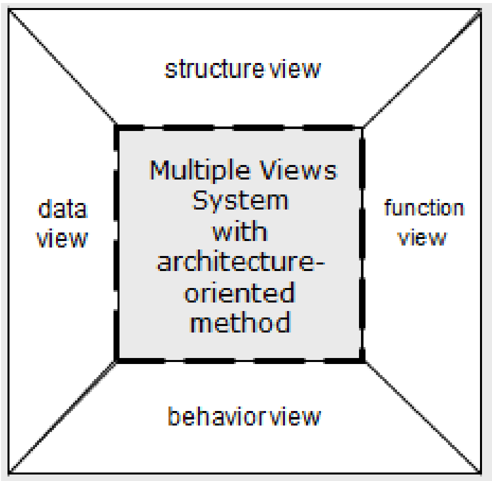

Figure 3 Architecture-Oriented Method

Figure 2 has many models. Figure 3 has only one model. Comparing Figure 2 with Figure 3, we unquestionably conclude that an integrated, holistic, united, coordinated, coherent, and coalescence model is more favorable than a collection of many heterogeneous and unrelated models.

Since structure and behavior views are the two most prominent ones among multiple views, integrating the structure and behavior views apparently is the best approach of integrating multiple views of a system. In other words, structure-behavior coalescence (SBC) facilitates [7, 8, 9] multiple view coalescence (MVC) as shown in Figure 4. Therefore, authors claim that SBC architecture is an architecture-oriented systems model.

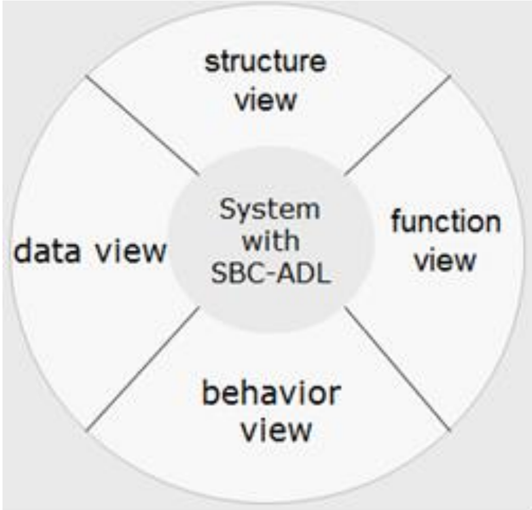

Figure 4 SBC facilitates MVC

III. RESULTS AND DISCUSSION

SBC architecture design method for Smart Agriculture Innovative Service System adopts the SBC architecture as a systems model. SBC architecture design method shall start from the preparation phase and then goes through the planning, preliminary design, and detailed design phases of SBC architecture construction. Each phase checks with the SBC architecture to make sure the constructed Smart Agriculture Innovative Service System is what the users want as shown in Figure 5.

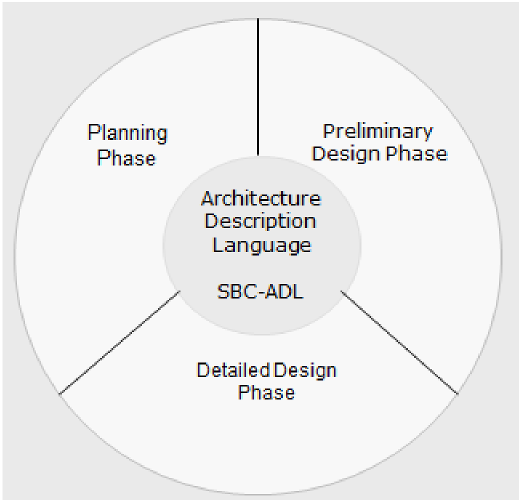

Figure 5 SBC architecture design method for Smart Agriculture Innovative Service System

SBC architecture design method uses Architecture Description Language (ADL) to formally design the essence of a Smart Agriculture Innovative Service System and its details at the same time. In the planning phase, framework diagram (FD)-ADL is used. In the preliminary design phase, component channel diagram (CChD)-ADL is used. In the detailed design phase, interaction flow diagram (IFD)-ADL will be used.

A. Planning Phase

The framework diagram (FD)-ADL designs the decomposition and composition of a Smart Agriculture Innovative Service System in a multi-layer manner. Only non-aggregated systems will appear in the FD-ADL. As an example, Figure 6 shows a FD-ADL of the Smart Agriculture Innovative Service System.

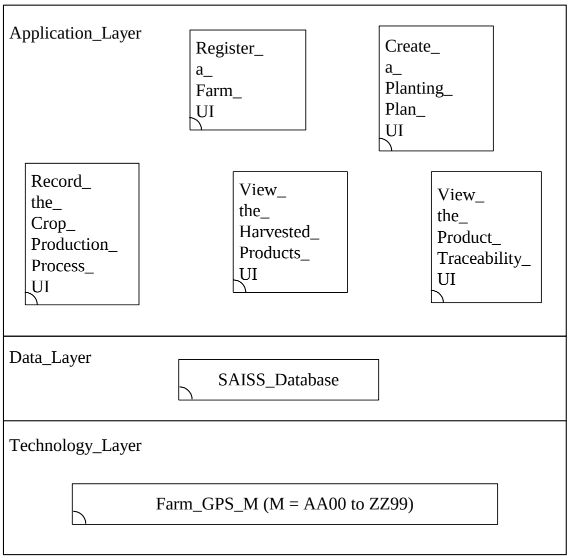

Figure 6 FD of the *Smart Agriculture Innovative Service System*

B. Preliminary Design Phase

For a Smart Agriculture Innovative Service System, using component channel diagram (CChD)-ADL to design all components' channels. Figure 7 shows a CChD-ADL of the Smart Agriculture Innovative Service System (SAISS). In the figure, component Register_Courier_Account_UI has one channel: Register_Courier_Account; component Place_an_Order_UI has three channels: Show_Restaurants_and_Meals_Call, Show_Restaurants_and_Meals_Return, Place_an_Order; component Accept_a_Delivery_Request_UI has three channels: Show_a_Delivery_Request_Call, Show_a_Delivery_Request_Return, Accept_a_Delivery_Request; component Pay_the_Order_UI has two channels: Pay_the_Order_Call, Pay_the_Order_Return; component Rate_the_Courier_UI has one channel: Rate_the_Courier; component SAISS_Database has six channels: SQL_Insert_Register_Courier_Account, SQL_Insert_Place_an_Order, SQL_Select_a_Delivery_Request, SQL_Insert_Accept_a_Delivery_Request, SQL_Insert_Pay_the_Order, and SQL_Insert_Rate_the_Courier; component Customer_GPS_P (P = AAA000 to ZZZ999) has one channel: Customer_GPS_Positioning.

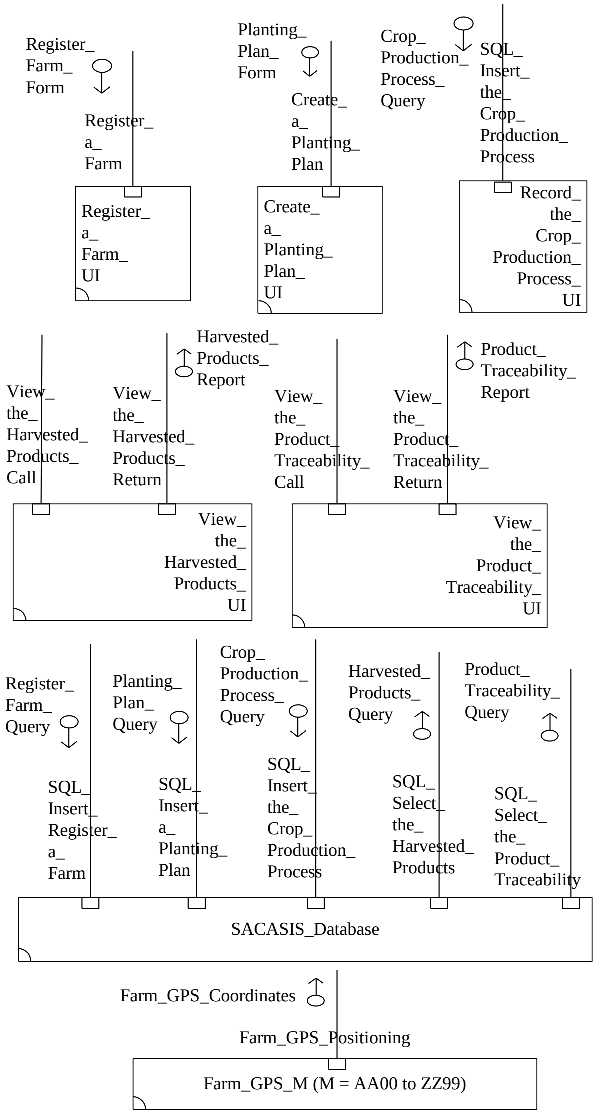

Figure 7 CChD of the *Smart Agriculture Innovative Service System*

C. Detailed Design Phase

In a Smart Agriculture Innovative Service System, if the components, and among them and the external environment's actors to interact, these interactions will lead to the systems behavior. That is, "interaction" plays an important factor in coalescing structures with behaviors for a Smart Agriculture Innovative Service System.

The overall behavior of a Smart Agriculture Innovative Service System consists of many individual behaviors. Each individual behavior represents an execution path. Author use interaction flow diagram (IFD)-ADL to design this individual behavior. The overall Smart Agriculture Innovative Service System's behavior includes five behaviors: Registering_a_Farm,Creating_a_Planting_Plan,Recording_the_Crop_Production_Process,Viewing_the_Harvested_Products,Viewing_the_Product_Traceability.

Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 9 Issue 5, May - 2023

Figure 8 shows an IFD-ADL of the Registering_a_Farm behavior. First, actor Farm_Owner interacts with the Register_a_Farm_UI component through the Register_a_Farm channel interaction, carrying the Register_Farm_Form input parameter. Next, component Register_a_Farm_UI interacts with the Farm_GPS_M (M = AA00 to ZZ99) component through the Farm_GPS_Positioning channel interaction, carrying the Farm_GPS_Coordinates output parameter. Finally, component Register_a_Farm_UI interacts with the SAISS_Database component through the SQL_Insert_Register_a_Farm channel interaction, carrying the Register_Farm_Query input parameter.Do not confuse “imply” and “infer.”

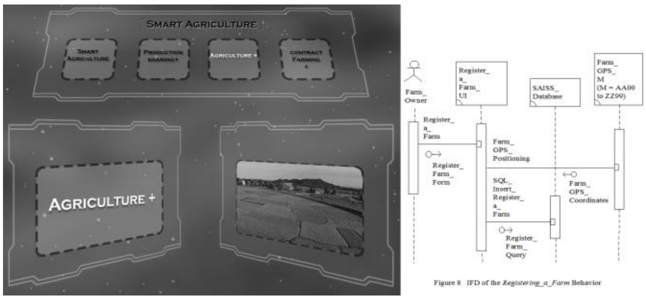

Fig.8 IFD of the Registering_a_Farm Behavior

Figure 9 shows an IFD-ADL of the Creating_a_Planting_Plan behavior. First, actor Farm_Owner interacts with the Create_a_Planting_Plan_UI component through the Create_a_Planting_Plan channel interaction, carrying the Planting_Plan_Form input parameter. Finally, component Create_a_Planting_Plan_UI interacts with the SAISS_Database component through the SQL_Insert_a_Planting_Plan channel interaction, carrying the Planting_Plan_Query input parameter.

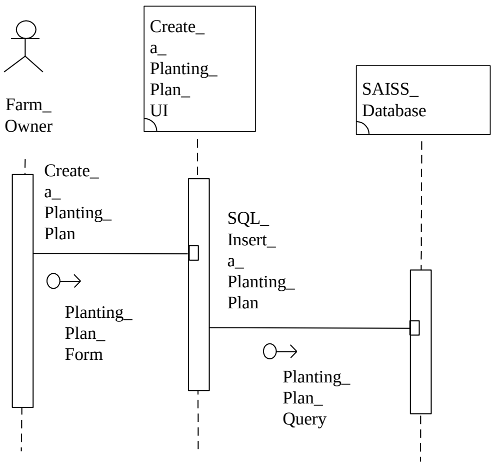

Figure 9 IFD of the *Creating_a_Planting_Plan* Behavior

Figure 10 shows an IFD-ADL of the Recording_the_Crop_Production_Process behavior. First, actor Farmer interacts with the Record_the_Crop_Production_Process_UI component through the Record_the_Crop_Production_Process channel interaction, carrying the Crop_Production_Process_Form input parameter. Finally, component Record_the_Crop_Production_Process_UI interacts with the SAISS_Database component through the SQL_Insert_the_Crop_Production_Process channel interaction, carrying the Crop_Production_Process_Query input parameter.

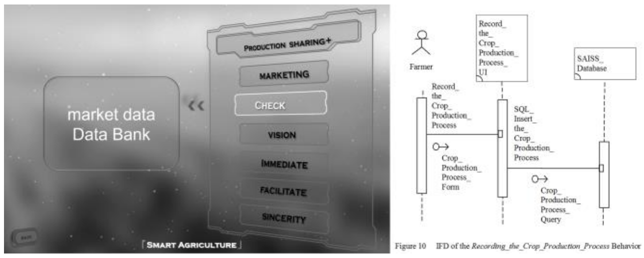

Fig.10 IFD of the Recording the Crop Production Process Behavior

Figure 11 shows an IFD-ADL of the Viewing_the_Harvested_Products behavior. First, actor Consumer interacts with the View_the_Harvested_Products_UI component through the View_the_Harvested_Products_Call channel interaction. Next, component View_the_Harvested_Products_UI interacts with the SAISS_Database component through the SQL_Select_the_Harvested_Products channel interaction, carrying the Harvested_Products_Query output parameter. Finally, actor Consumer interacts with the View_the_Harvested_Products_UI component through the View_the_Harvested_Products_Return channel interaction, carrying the Harvested_Products_Report output parameter.

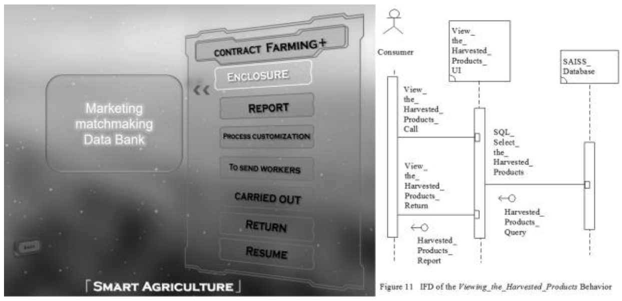

Fig.11 IFD of Viewing the Harvested Products Behavior

Figure 12 shows an IFD-ADL of the Viewing_the_Product_Traceability behavior. First, actor Consumer interacts with the View_the_Product_Traceability_UI component through the View_the_Product_Traceability_Call channel interaction. Next, component View_the_Product_Traceability_UI interacts with the SAISS_Database component through the SQL_Select_the_Product_Traceability channel interaction, carrying the Product_Traceability_Query output parameter. Finally, actor Consumer interacts with the View_the_Product_Traceability_UI component through the View_the_Product_Traceability_Return channel interaction, carrying the Product_Traceability_Report output parameter.

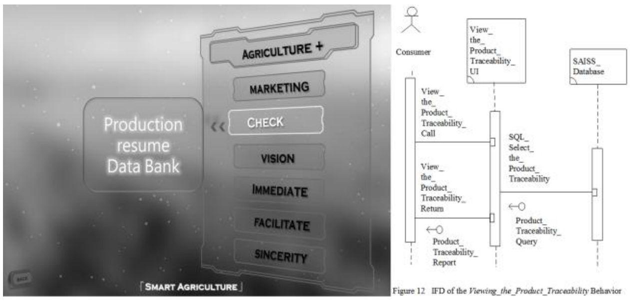

Figure 12 IFD of Viewing the Product Traceability Behavior

IV. CONCLUSION AND FUTURE WORKS

The Smart Agriculture Service Systems adopts the SBC architecture as a systems model to make sure the constructed Smart Agriculture Service Systems is what the users want. The Smart Agriculture Service Systems can optimize the crops farming footprint then decreasing greenhouse emission from farming footprint. The service system will match the buyer and farmer by contractual farming mechanism then farmer planted traceable agriculture products (TAP) deliver to the buyer directly. The Smart Agriculture Service platform model is shown in Fig 8-12 thus optimized the farming footprint. contractual farming and TAP are the efficiency way to decrease greenhouse emission from farming footprint [2, 3]. The efficiency of Smart Agriculture C Service Systems shall positive correlation to the reduction rate of farming carbon footprint.

A Smart Agriculture Service Systems includes multiple views such as structure, behavior, function, and data views. Non-architecture-oriented systems model respectively picks a model for each view. These multiple models are heterogeneous and unrelated of each other, thus there is no way to put them into a conformity model. Architecture-oriented systems model, instead of picking many heterogeneous and unrelated models, will use only one single coalescence model. The structure, behavior, function, and data views are all integrated in this multiple view coalescence (MVC) systems model.

SBC architecture design method for Smart Agriculture Service Systems adopts the SBC architecture as a systems model. SBC architecture design method uses Architecture Description Language (ADL) to formally design the essence of a Smart Agriculture Service Systems Architecture and its details at the same time. With Architecture Description Language (ADL), designer then can effectively design the structure, behavior, function, and data of Smart Agriculture Service Systems; resolve uncertainties and risks caused by those traditional non-architecture-oriented design methods. Overall, the Smart Agriculture Service Systems design schemes also can helps designers to do further adjustment and improve the effectiveness of Smart Agriculture Service Systems.

AUTHOR CONTRIBUTIONS

The structure and behavior views are the two most prominent ones among multiple views, integrating the structure and behavior views apparently is the best approach of integrating multiple views of a system. In other words, structure-behavior coalescence (SBC) facilitates multiple view coalescence (MVC). Therefore, the authors claim that SBC architecture is an architecture-oriented systems model.

REFERENCES

[1] Gilbert, Natasha, Summit urged to clean up farming, Nature News & Comment, nature.com, Nov.22, 2011.

[2] Thornton, P., Recalibrating Food Production in the Developing World: Global Warming Will Change More Than Just the Climate, CCAFS Policy Brief no. 6. (CGIAR Research Program on Climate Change, Agriculture and Food Security, 2012).

[3] Gilbert, Natasha, Climate-smart agriculture is needed, Nature News & Comment, nature.com, Mar.02, 2011.

[4] Qiu, Guo-Peng, Sun, Shuh-Ping, Wang, Fu-Tien, William S. Chao, "Architecture-Oriented Design Method for Smart Tourism City Internet of Things System", Advanced Information Sciences and Service Sciences (AISS), vol. 8, no. 4, 2016, pp. 8-199.

[5] Bass, L., Software Architecture in Practice, 2nd Edition, Addison-Wesley, 2003; pp. 164–186.

[6] Bernard, S., An Introduction To Enterprise Architecture, 2nd Edition, AuthorHouse, 2005; pp.66-86

[7] Chao, W. S. and Sun, S. P. and Harn, M. C., Systems Architecture: Hardware, Software, Enterprise, Knowledge, Thinking, I-Shou University System Architecture Research Center Publishing, 2013; pp.90-120

[8] Chao, W. S, Systems Modeling and Architecting: Structure-Behavior Coalescence for Systems Architecture, CreateSpace Independent Publishing Platform, 2014; pp.178-196

[9] Chao, W. S, System: Contemporary Concept, Definition, and Language, CreateSpace Independent Publishing Platform, 2016; pp.254-280.

[10] Clements, P., Documenting Software Architectures: Views and Beyond, 2nd Edition, Addison Wesley, 2013; pp.166-190.

[11] Dennis, A., Systems Analysis and Design, 4th Edition, Wiley, 2008; pp.266-286.

[12] Johnson, Pontus, "IT Management with Enterprise Architecture." KTH, Stockhol, 2014. pp.50-75

[13] Kendall, K. et al., Systems Analysis and Design, 8th Edition, Prentice Hall, 2010, pp.255-286.

[14] IACOB, Maria-Eugenia, From enterprise architecture to business models and back. Software & Systems Modeling, 2014, pp. 288-326.

[15] Maier, M. W., The Art of Systems Architecting, 3rd Edition, CRC Press, 2009, pp.155-195.G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)