A Business Architecture Capability Meta Model and Tool-set for Providing Function Point Estimation for Enterprise Architecture Management

Francois A. du Toit, Maureen Tanner

***Abstract*— Change impact analysis is error prone and time consuming when business architecture capabilities, enterprise service models, enterprise data models and UML design models are disconnected across multiple repositories. Software architecture rapid evolution is further complicating the management of this analysis especially since software sizing is not based on techniques like function point measurement. Whilst following the Design Science Approach, this study seeks to analyze, design and implement a software prototype which integrates business architecture capabilities and design models to facilitate change impact analysis. This paper specifically reports on the findings obtained from the first stage of this design science research cycles and proposes an approach to model Business Architecture Capabilities when design specifications and models are spread over more than one repository. It also presents the requirements for a prototype to implement the principles to model Business Architecture Capabilities of disconnected design models which are obtained and confirmed from literature and through observations and interviews of solution design specialists. Therefore this paper proposes by literature, interviews and observations: (1) The need to have a tool and meta-model to model from a Business Architecture Capabilities perspective when design specifications and models are spread over more than one repository, (2), a set of requirements outline this need (3), a high level design pattern for implementing a software toolset that integrates UML design models and enterprise architecture models using a unifying meta-model.**

***Index Terms*— Business Capability, Change Impact Analysis, Enterprise Architecture, Software Cost Estimation**

I. Introduction

Change impact analysis across multiple software design repositories are error prone and time consuming [10]. When Software Architecture Design models are spread over different repositories, they can easily become out of sync with each other. The design models end up disconnected with no traceability between them as different teams work on different artifacts in parallel. Traceability is a crucial element in the change impact analysis process [47]. While doing a change impact analysis, the software engineer is seeking to identify the relationship between all traceable elements [30]. For example, changes applied to one repository must maintain consistency and integrity towards other design models horizontally [18] or vertically [17] in the end to end solution of the software architecture.

The largest part of software total cost of ownership is concerned with the change and evolution of software [9]. There is thus a need from software clients to have more accurate software estimates from software providers when change impact analysis is conducted. When traceability and dependency information is not visible or captured, then the change impact analysis estimate is prone to error [10] and [78]. Change Impact analysis is very dependent on the accuracy of current software architecture documentation. As the software architecture changes and evolves, the changes in documentation must also be synchronized [10].

Change impact analysis and traceability are two aspects that go hand in hand with each other. To do proper change impact analysis, the software engineer has to trace the relationship the requirement has towards other requirements and then determine if there would be an impact [77]. Similarly, for the same requirement, the relationship towards software components has to be found and the impact of change determined. In addition, the relationship between the software components should also be identified [77]. Prior research has also shown the need to have traceability of the requirements towards software architecture design models [17], [18], [24], [30], [31] and [78]. Automated impact analysis of UML models were proposed by [17] and [31] to improve the traceability and dependency analysis when requirements enforce changes to software solution models. These proposed solution look at the perspective of a single repository and not of those when enterprise architecture models are spread across various repositories maintained by different stakeholders.

II. Research Objectives

Identifying and managing enterprise capabilities to align with business strategy are considered to be valuable means of supporting the coordination between business strategies and IT [61]. This is to facilitate how organizations can continuously derive and leverage value through IT. Enterprise architecture (EA) captures the essentials of the business, IT and its evolution [42]. The importance to have proper strategic information systems in place that support enterprise asset management were pointed out by [74]. One of the main criteria for a strategic information system to

Manuscript received December 9, 2014; revised January 9, 2015. Mr. F.A. du Toit is with the University of Cape Town, Cape Town, South Africa. He is a Masters student at the Department of Information Systems (e-mail: francoisadt@gmail.co.za). Dr. M. Tanner is with the University of Cape Town, Cape Town, South Africa. She is a lecturer at the Department of Information Systems (e-mail: m.tanner@uct.,ac.za; phone: +27 (021)-650-4860).

support IT Asset planning was to have the ability to represent information gathered via different planning models and software design models. Limited research was conducted to investigate change impact analysis across heterogeneous software architecture repositories where the software solution design models are spread across multi-disciplinary teams.

The purpose of the paper is firstly to propose an integrated model of the Enterprise Architecture (EA) as Business Capabilities [13]. This is accomplished by using the EA model constructs that represent IT systems and organize them in a perspective that shows how IT Assets enable business strategy and objectives. These IT assets and resources that execute these strategies are called “Business Capabilities”. When these “Business Capabilities” are described within the context of EA models, they are then called “Business Architecture Capabilities” [13] to describe an architecture building block which are assigned to a business capability concept. Secondly, to propose a high level integration and transformation components that will enable the development and implementation of the toolset.

Following [38] design science research cycle, the overall research process is iterative in nature. Based on the research objectives described in this section, a design science approach is proposed as the research methodology for this study as shown in Table I. The methodology concepts were drawn from design science research methodology approaches applied towards information systems research area proposed by [39], [56] and [60]. Phase 1 of the research takes the form of a single in-depth case study, during which the need for the tool is ascertained and confirmed and the tool requirements are analyzed (See phase 1 in Table I). During phase 1 a literature review, observations and interviews have been conducted, the result of which are presented in this paper. During phase 2, the tool will be designed and developed based on the requirements identified in phase 1. Thereafter, another case study will be conducted during phase 3 of the research to evaluate the tool. To ensure the reliability of the evaluation criteria, additional literature review will also be conducted to identify the evaluation criteria of the tool in the context of all stakeholders e.g. Business owners, project managers, design managers and system analyst. The research will use a quantitative approach to evaluate the effectiveness of the tool in improving change impact analysis during all phases of the system development lifecycle (SDLC). From the outcome of this evaluation, a set of practices will also be proposed on how to design and conduct change impact analysis based on Business Architecture Capabilities. The research objectives as described above are broken down per phase in Table I below. This paper will only show the results of the research completed for phase 1.

The study contributes to the body of knowledge on enterprise architecture and provides a solution to challenges faced during dependency analysis across distributed software architecture repositories. The impact analysis technique that will be used in this case study is Function Point Analysis. There are few advantages to using this technique. For example, it can be estimated earlier in the life cycle since it is only necessary to have the requisites functional requirements document, which explains the user functions expected. Estimations can therefore completed by non-experts of the system [41]. More details are described in section *F. Function Point Analysis.*

III. LITERATURE REVIEW

The aim of this section is to identify from literature, how change impact analysis can be improved for disconnected software repositories. The first section will describe change impact analysis and the difficulties of undertaking software estimation with disconnected software design models. The second section, will describe how disconnected views of enterprise architecture and software design models can be bridged using a business architecture capability approach. Thereafter the role of architecture modelling approaches and of Unified Modelling Language (UML) [71] to present software solutions will be discussed. The last section will cover the semantic integration and presentation of disconnected software design models. This will form the basis for understanding the requirements of a software toolset that will support team members during the process of change impact analysis and provide them with an end to end traceability view of the software estimation process based on Business Architecture Capabilities.

IV. Enterprise Architecture Management

Several enterprise architecture management frameworks have been developed to guide the enterprise architect and the solution architect in managing the application landscape of enterprise systems. For example, the Enterprise Architecture Management Pattern Catalog by academics [20], The Open Group Architectural Framework (TOGAF) [70] by a standardization body Open Management Group (OMG, 2009) and the Ministry of Defence Architectural Framework (MoDAF) [11] for the UK Ministry of Defence, provide several support for systems engineering and network enabled capabilities of enterprise systems. For the purpose of this study, the analogy of TOGAF will be used in showing the building blocks of an enterprise system.

A. Enterprise Architecture Evolvability

The management of change is also deeply embedded in an organization’s operational processes. A set of software evolution laws were defined by [49] which form the foundation of the work of others like [14] who did propose Architecture Evolvability Analysis Method (AREA) and a Software Evolvability Model. This can be used to solve the practical problems of managing software evolution. It is a challenging task for software providers to meet the needs of software clients if the requirements are changing frequently which then also do have a change effect on the current view of the software architecture [57]. More than one project could have similar requirements that do need a dependency analysis view between systems. The attributes of a software architecture system which causes the effect of software evolution on software architecture either have strategic value or decline in value [14] and [15]. The lower the cost of change but higher the benefit, the higher the trust investors do have in their investment in technology [64]. Therefore the need to determine the cost to implement change on a software system because of business requirements is important.

Change impact analysis is a process or method that is used to determine the cost and impact of a business requirement change on a software system [19] and [27]. Previous research proposed that impact analysis can be performed during the build, test and fix phases of software development lifecycle [68]. To provide quick impact analysis estimates the analyst should be able to visualize the evolution of the design models [48]. In large systems where software design repositories are spread across more than one department and repositories it becomes quickly difficult to comprehend the effect that a change request has on software systems. Therefore, there is a need for tools to assist the software analyst in completing that task [2]. To reduce the cost and duration to complete change impact analysis during software evolution, the effect of change is documented together with the software architecture model so that the complexity of understanding can be minimized. This makes provision for other analysts to re-use the knowledge that is captured. The more information can be accessed by the software provider doing change impact analysis, the more accurate the software estimate can be [78]. To have a unified view of distributed software architecture repositories one needs a single repository [21] that can be can be established using ontology based approach [51], [69] and [28]. Once this is established, such a system holds numerous possibilities to allow for the reasoning about properties of resulting unified models during change impact analysis [44].

To understand the requirements for building a change impact analysis software tool to support the software analyst during the systems development life-cycle, one firstly needs to understand the background of how enterprise architecture management relates to the modeling of software architecture. EAM is the management of IT assets so that the IT landscape is aligned with the business strategy [34]. This ensure that the correct decision making can be made regarding which IT assets are built to enable business to achieve maximum benefits from IT solutions that are scalable and directed towards future directions of the business [1]. Business solutions that are aligned with business objectives are described in terms of business capabilities [6].

B. *Business Capabilities*

Enterprise Architecture Business Capabilities can be represented in two different ways [6]; one is strategic modeling and the other functional modeling. From an Enterprise Architecture Management (EAM) point of view the EAM toolset focuses on the strategic modeling to produce a model of the enterprise architecture which identifies business challenges, opportunities and demands [6]. Functional modeling focuses on a model that will show all the business components and its realized application component that will be implemented.

A Business Capability defines the assets, people, processes and technology [72] to deliver the desired outcomes that supports the business strategy [45], [61] and [65]. The relationship between them are described as follows…. The EA solution stack are broken down in four building blocks, Business Architecture, Application Architecture, Information Architecture and Technology architecture [70]. As part of the EA solution stack, Business Architecture should define Capabilities which do create value streams driven by business processes. Enterprise architecture describes the Business capabilities which do provide a value stream to business. Each value stream is enabled by business processes. Business processes are driven and serviced by SOA business information services. To align business strategy and IT architecture, the concept of Business Components were introduced by [6]. The concept of Business capabilities centric extension provides a mechanism so that a business component provides a business capability and consumes a business service which harnesses information services to accomplish the business activities.

C. *Business Components*

This was to ensure that business could envision a business capability model or map that shows how a business's service provides features asked for, meets the demands from a strategic goal viewpoint and the performance metrics of the resources that are linked to that business service [25], [50] and [65]. A Business Component defined as an EA building block maps onto one or more Business Services [6]. A Business Service harnesses applications to provide business functions and information to the enterprise. To uncover the software architecture design rationale of business functions can greatly increase the understanding of the software architecture. The need to be able to evaluate the hidden architecture rationale between disconnected software repositories was proposed [49].

D. *Challenges in the Integration of EAM Models*

To have a unified understanding of all the business components that constitutes the business capabilities that were defined and modeled in disconnected software model repositories, one needs a single repository [21] that can be can be established using ontology based approach [51] and [69]. The impact on the quality on software architecture [14] and the challenges to integrate EAM Tools [54] with supported information have been identified in literature [34]. Some of them are, "Model transformation for the exchange of EA information necessary due to missing interfaces and standards, Not enough return on investment due to large initial investment efforts, Collection of information not relevant or too fine-grained for decision" [36, p. 35]. EAM tools do have their own propriety format to store the data but of these data structures are too broad or too fine grained. In order to integrate information from an EAM tool into another tool like a modeling tool and to show the architecture information related to a system design, the format of the data from EAM has to be consistent as well as the format of the data coming back into the EAM tool.

E. *EAM Modeling Languages*

Currently Unified Modelling language (UML) [71] is the de facto standard proposed by [12] to the Object Management Group. There are different types of diagrams in the UML Standard. These can basically divided into two groups, namely behaviour diagrams and structural diagrams. Behavioural diagrams show the behaviour of actors (Humans or Systems) towards the new proposed system.

Structural Diagrams show the structure of the solution and how the behaviour should be implemented [66]. For example, component diagrams are abstract representation of the business services and application components that were defined in the enterprise architecture. On the component diagrams, the software architecture components are presented with links to show the associations or dependencies between each other. Models created within MDE approach raise the level of abstraction [55] from a requirement and software system point of view [55]. To understand how these disconnected models are related to each other, a common understanding of semantic representation of models is required.

F. EAM Software Sizing

Various software estimation techniques have been investigated [50] and [76]. There are two categories of software sizing methods namely (1) Parametric methods and (2) Non-Parametric methods. Parametric methods are those using algorithms to calculate the size based upon geometry or characteristics of the products and processes, functional sizing techniques and expert systems using rules and historically based data. Non parametric methods are expert judgment which is based upon personal knowledge and experience. Various issues are limiting the accuracy of cost estimation [63] for example: Required knowledge, information and data are unavailable; a costly estimation database is required to support cost estimation according to product attributes, required similar business processes or similar products to base estimation on historical data; the estimation process requires support of knowledgeable experts; estimation processes are considered tedious; and incomplete business requirements causes estimation to be inaccurate.

One cannot assume that if a business requirement changes that looks similar to others will incur similar software cost. False analogies can occur because it is easy to perform wrong software estimation based on a similarly project. Such similar requirements could differ in critical ways [58]. Analogy based software technique can only be accomplished if the correct configurations and parameters are set [46]. It is important that a software provider and software client agree upon the method and know the shortcomings of the software sizing method used.

G.Function Point Analysis

In this research study the Function Point Analysis method [3] will be used. A benefit of using function points count method is to avoid the necessity of having to know the programming language and other technical differences in the implementation of the IT systems to do an estimate [39]. The function points are determined by using the user functions as described per functional requirement. Each user function is used as input in an effort estimation model, along with the data definitions per user function. Another advantage is that function point count can be calculated by non-technical members of the development team because the estimation is based upon user functions which are user inputs and user outputs upon the system under scope [39].

The FPA measures functional requirements as follows:

- The business ***transactions*** (e.g. *Enquiry, External Output, External Input used per transaction*) that the user can perform using the software
- The business ***data*** (e.g., *Internal Logical File or External Interface File, In memory data structure p*hysical file that is used by the application) that the software can store and access.

Each component are analyzed and then grouped into application boundaries. For each application boundary all the user software functions, called transactions within the [39] manual, are determined. FPA estimation uses these user functions as input to determine the estimation.

H.Function Point Count Method of Calculations

A summarized version of the FPC method as stipulated in the IPFUG manual [41] will be described in this paragraph. In FPA a software function is a transaction which is executed on a data set. The functions that are executed by the user are defined as user functions. These user functions are external input (EI), external query (EQ) and external output (EO). Each of the user functions can act upon internal data, called internal logical file (ILF), or external data, called external logical file (ELF). The complexity is calculated based on the number of File Types Referenced (FTR) multiplied by the data elements (DET) utilized for that particular transaction by the application component within a specified boundary. An external input (EI) is an elementary process that processes data that comes from external the application boundary. An external output (EO) is an elementary process within the application that sends data external to the application boundary. An external inquiry (EQ) is an elementary process that request data from outside the application boundary and sends data external to the application boundary. An ILF is a logical user-identifiable group of related data maintained within the boundary of the application. An external interface file (EIF) is a logically user identifiable group of data referenced or used by the application, but maintained within the boundary of another application [41].

Several other versions of the Function Point Analysis techniques have been proposed to measure a function point count for systems that cannot be counted according the normal function point count specification. COSMIC method was tested and analyzed for SOA [67] and proved to be very accurate. For SOA a new adjusted value adjustment factor is proposed to take into consideration the different complexity layers of SOA [53].

V. Research Methodology – Phase 1

A. Conducting the Literature review

A literature review was undertaken by doing a database-driven search using IEEE Xplore, ACM, AIS Electronic Library (AISel), Springer. The search was conducted from 1st February 2014 till October 2014. Journals and Conferences specifically in the Enterprise Architecture, Enterprise Systems Modeling and Software Engineering were consulted first. Relevant articles were analyzed and cross references were checked for deeper analysis to provide a good coverage of scholarly and practice-oriented publications. We see AISel, ACM and Springer focus mainly in scholarly publications and the IEEE Xplore contents to be more focused on practice.

We first started using the main terms of the topic “business capability” and “architecture capability”. Within this initial search we further searched for articles relating the above concepts with “business architecture”, “enterprise systems”, “enterprise architecture” , “software architecture”, “software engineering”, “enterprise integration”, “enterprise interoperability”, “service oriented architecture (SOA)” in all the databases. Further searches done by drilling down and filter in conjunction by using additional key words “management”, “modeling”, “estimation”, “change impact analysis” and “function point analysis” in all the databases. From the searches 48 articles were chosen that were coded using the above mentioned key words.

B. Thematic Analysis – Phase 1

This process of coding was used to build concepts and categories. Coding was also completed by matching this with codes in literature which was tagged against a paragraph or a chapter. Concept definitions become more exact and differentiations get more precise when the interviews were coded to match those collected for the observations. The key words used in the database searches were the input to complete the open coding using top-down analysis of concepts that were collected in the observations and the interviews. All of the abstract concepts are representations of events, objects, actions or interactions to allow the grouping of similar information to better understand the data.

C. The Field Study Approach (Phase 1)

C.1 Research Paradigm

Design science is fundamentally a problem solving paradigm. It seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, management, and use of information systems can be effectively and efficiently accomplished [38]. Design science approach iteratively changes the state-of-the-world through the introduction of novel artifacts [56] and [73]. The methodology is developmental. The axiology is to have value control and value creation as the outcome of the research (reference missing).

A participatory field study has been conducted as part of phase 1 of this research based upon the assumption that an objective social reality exists and can be observed and reported accurately (reference missing). This allowed the researcher to gain firsthand experience of the problem in the organizational context in which the people, events and processes exist (reference missing). It allowed the researcher to ask “how” the EAM processes occur, how the people they spent time with interact with the EAM Tools to achieve their goals and how the events in completing tasks occur. The field study has set the design project's direction and discovered unmet user needs which will be discussed in the findings section.

C.2 Field Study Description

The research took place at a South African Bank in the Financial Services sector. The organization specializes in banking products for retail clients and corporate clients. The bank is one of SA’s four largest banking groups by assets and deposits. They are a JSE Top 40 company with their ordinary shares listed on the JSE since 1969. Their market capitalisation was R107bn at 31 December 2013. They do have their own IT Group Technology division (NGT) whose purpose is to provide IT development and support services towards all of the organization. This involves business analysis and software development in various software systems, from Internet Banking systems, Mobile Banking systems to legacy systems on the Mainframe running operational services. NGT provides technology consulting which includes software product development and enterprise architecture. NGT is a centralised technology unit with responsibility for all components of the group’s technology processing, development and systems support. The group’s IT systems, databases, technology infrastructure, software development and IT project/programme management are centrally managed to provide economies of scale and facilitate a cohesive group wide service-oriented architecture (SOA) technology strategy.

In 2013, Group Technology, express the need to have an integrated view and a unified understanding of all the business components were defined in PlanningIT - an EAM Planning Tool, which presents the enterprise architecture view of the IT landscape, with Rational Software Architect - a software design modelling tool that present the IT landscape in the detail level. This is to have a dependency analysis view of all the business components that constitutes the IT landscape so that management in EA could envision a business capability map that shows how can IT solution provides features asked for by business, meets the demands from a strategic goal. The researcher started discussing his intentions with senior stakeholders (Group Technology leadership) in January 2014. This then leads to an initial exploratory activity, formal interviews and observations. This exploratory activity started with conversations with two initial participants, one in Enterprise Architecture department and another in System Development where the researcher discussed areas of concern within EAM. The necessary of the research were confirmed by the participants and the necessary permission to conduct research in this area within Group Technology was given.

In phase 1, the participatory field study has been conducted within the IT department of an organization in the financial services sector. Data was collected through a combination of participant observation, interviewing, as well as document and artefact analysis. The researcher acted as Participant observer whereby he fully participated in the behaviour activity.

The research evaluated and observed the tools that were used to manage EA Capabilities from a management perspective and a design team perspective. It also looks at integrates of data between different design repositories to enable management to have a Business capability view of architecture. During the pre-execution project planning of project demands the high level model is completed in a tool called "PlanningIT". PlanningIT is an Enterprise Architecture Management tool that assists management to plan the alignment of business and IT and to sustain the fulfillment of planned IT Assets that enable a company towards achieving its strategic objectives. A solution architect will analyze a set of requirements per project stream and then assign those Business Capabilities and IT Capabilities that will be impacted. A very high level of Business capability names are exported in Excel format so that the list can be used as input in the Technical Solution Outline document which will describe the technical solution in per IT system impacted. Currently this is model get exported via a graphic file (.png format) and do get published in Microsoft PowerPoint. It is then sent as is towards the System Development teams for input to draw up a High Level Component Model (HLCM) within Rational Software Architect (RSA) using Unified Modeling Language (UML). The HLCM are used as a base to determine the change impact analysis for another tool called Function Point Workbench. Within this third party tool each architectural component are then redrawn and depicted as in the HLCM. The UML model within RSA is then further analyzed based upon the data functions per impacted component. The impact on each is determined using Function Point Analysis (FPA) techniques which are described in the "Function Point Analysis Technique" earlier in the literature review section of this paper. The measurement on each logical IT system component is based upon transactions and data functions which determine the complexity. The field study investigate if there are any tools manual or automatically that do give management a function point count per capability of the all impacted components that have been identified in the HLCM within RSA. It also observes the how entities in one toolset are related to the same entity description defined in another model with the same meaning.

C.3 Sampling (Both for Field Study and Respondents)

The case was selected based upon the parties involved in early software estimation are Senior Manager of System Development teams, Enterprise Architects, Lead Architects, SOA Specialists and System Analysts. The participants, or those that are within the problematical situation, are selected on the basis that the information obtained would be as complete, balanced and unbiased as the situations might allow. Participants have been selected based on the years of work experience in IT, the contribution they can make towards the research and willingness to participate in the research.

C.4 Ensuring Validity & Reliability

The outcome of the design science research of phase 1 has been presented effectively to both technology-oriented and management-oriented audiences. The validity and reliability of the findings that prescribe the need for a new meta-model and proposed high level model integration design based upon literature was established by evaluation results with evidence from interviews and observations.

D. Data Collection

During the case study, interviews and observations were held of key specialists working in the Enterprise Architecture and Software Development teams responsible for doing change impact analysis. The participants were in management as well as in technical roles. Details of each role are listed in Table 2. Each interview was scribed and notes were kept during observations and follow up telephonic discussions. Each interview questions were asking so as to confirm the literature review. By thematically analyzing the interviews and the observations it clarifies the purpose of the interviews and the literature review concepts that was explored. The interviews and observations were verifying and checking facts by confirming the outcome of the analysis with the interviewees.

The following observations were scribed during interactions with EA staff while determining dependency analysis of Business Capabilities were as follows: i) the EA department cannot determine dependency analysis of a Business Capability between projects demanding the same features across multiple projects ii) The EA model view of EA assets are not directly traceable within solution designs as it is a manual analysis process iii) There is no mechanisms currently for Group Technology to provide accurate FPC based upon business capability perspective.

VI. FINDINGS - PHASE 1

Gaps were identified in literature regarding EAM change impact analyses were further confirmed by the observations and the interviews during the field study. Themes (relative to these gaps) identified from the data analysis of both literature and empirical data are shown in Table 3. The themes (relative to these gaps) identified from the empirical data are that (1) there are currently no integration between EAM and Design Repositories, (2) the current FPC methodology and tools are insufficient for SOA estimation, (3) the current toolsets do not support FPC during modeling, (4) the change impact analysis is time consuming and costly and (5) accuracy of documentation affects the accuracy of FPC.

Given that the themes identified from the literature are in line with the empirical data as shown in Table 3, they will form the basis for requirements for the software toolset to be implemented.

A. Gaps Relative to Business Capability Dependency Analysis

To capture the dependency relationship between capability, resources, requirements, business strategy and measured cost per capability per resource, one needs an integrated modeling framework [23], [26] and a meta-model foundation to work from [40]. This ensures that there is alignment between business and IT strategies when modeling the impact of the IT systems. When IT systems are modeled according the above mentioned structures then the same alignment and traceability are carried through from Business Strategy to Enterprise Architecture towards System Design. Traceability Analysis in system design have been shown useful and beneficial in reducing time and cost while doing change impact analysis using UML refinements

[18]. When using a modeling pattern and a conceptual meta-model to link all the models, the traceability information are consistent, having the same presentation, structure and meaning [18].

The thematic analysis of the empirical data showed (See Table 3), that dependency analyses across projects are usually not accurate. The Capability theme occurred 32 times and this inaccuracy was confirmed by all participants. A sample quote is provided below:

"Group Technology cannot provide accurate dependency analysis across projects and across the enterprise based upon business capability".

A company called NGT does use an EAM planning tool, called "PlanningIT" which are based upon the TOGAF content meta-model as basis to provide a structure to present the solution architecture. With this in mind the TOGAF meta-model were investigated to analyze what are the gaps in the current model to present information across according a Business Capability perspective.

The structure of the enterprise architecture building blocks when using TOGAF is depicted by the Architectural Content Framework (ACF). The content meta-model shows how all the other elements of enterprise architecture are related to one another. In a SOA environment, the Business Services provide enterprise business functionality. Data Entities are presented by enterprise conceptual models which provides for a consistent consolidated view of business entities. An IS Service is realized through an Application Component which is modeled as part of the solution architecture model and detail design models [70].

In the TOGAF content model provision is made for the fact that a capability is fulfilled by an objective of the organization. The actors on the capability within the origination use business services to realize the objectives that have to be met. A business process enables the capability to execute the expected activities and outcome. These entities that enable the capabilities, namely process, business service and the lower level system components namely application architecture components, are measurable [5], [30], [32] and [75]. The TOGAF model can be further extended by providing additional meta-entities that describes the definition of capabilities as a measurable entity. A measureable entity is an object that is to be characterized by measuring its attributes. These attributes are measurable physical or abstract properties of these entities. Each attribute is the abstract property of what is measured. In the case of using a measurement method, the method will determine how these attributes are used to calculate the measurement. From a capability point of view the measure will be the sum of all measurements of all the entities that enable the capability to fulfill its objective. Currently in practice and according the theory in the TOGAF content meta-model, the measure of capabilities and the resources are not linked together to give the sum of all impacted business services, architecture components and processes. EA ontology meta-model (see Figure 2 and Figure 3 in the Appendix) was proposed to have a uniform consistent presentation of modeling resources and capabilities [4]. The research will use this as a basis to integrate EAM models with system design models which is presented in the unified modeling language (UML).

The Business Capabilities Centric Extension model [6] proposes a Business Component view for TOGAF but does not assign the Capability directly to resources which are assets, processes, people and services as in a proposed ontological model [4]. Close observation to the quality of the TOGAF ontological model [33] concludes that the quality of meta-models can be confirmed by an ontological approach. The study by [4] further confirms that there should be no disconnection between capabilities and the resources.

We propose a meta-model (Figure 2 and Figure 3 in the Appendix) where the TOGAF capability entity is extended so that the capability concept is directly related to the business processes, actor (people), business service and architecture components by using an "isEnabledBy" property instead of a Business Component entity. A capability can therefore be assigned to the actual resources that will enable it to bring the value to what the business expects. Once a Business Capability is modeled according this meta-model structure, it makes it possible to trace dependency and measure of business capabilities which can be enabled by technical capabilities of business services and business processes as proposed by [72]. How models, those that are in different formats and structure, can utilize this meta-model to overcome the interoperability between the different modeling toolsets that are used to build these models will be discussed in the next section.

B. Gaps Relative to Enterprise Architecture Model Interoperability

A definition of Enterprise interoperability was given by [22, p1] as "the ability to (1) communicate and exchange information; (2) use the information exchanged; (3) access to functionality of a third system.".

It is therefore the ability of one tool to seamlessly exchange information from another system and also be able to utilize that information within itself. In this field study it means therefore that the EAM Tool need to be able to exchange information with the Design modeling tool and use the information for its benefit.

During the interviews and observations it was clearly identified that such interoperability between the tools does not exist. The thematic analysis does show that there were 10 occurrences of the same theme across the qualitative data which do confirm that this is a problem.

Participant - SOA Specialist:

"End to End model view of EA assets not directly traceable within solution designs" *(Participants I-3 to I-7 agree)*

To overcome the enterprise interoperability and enterprise knowledge sharing between users of Enterprise Modeling Tools a common visual based language was proposed by [59] to as an Interlingua between Enterprise Modeling Tools. A common exchange format is needed to describe independently of mappings to and from existing enterprise modeling languages that are used [22], [23] and [59].

Entity mapping and the integration of heterogeneous repositories will help to prevent inconsistency between design models [37] and [62]. To integrate disconnected models with different structure and meaning towards each other, a meta-model transformation-based approach (See Figure 1) can be followed [23] and [35]. This will add value to improve software measurement and cost [43].

Disconnected models can only be merged if the models in each side (See Figure 1) are in a format that the transformation process understands. The format definition of the source and target are defined and meanings are defined in the transformation rules. Entities with the same name or meaning can then be linked together to form one big network of entities [29]. Reusing existing information saves efforts of manual mapping of entities between two different meta-models. Once these mappings are done knowledge can be shared [59].

C. Gaps Relative to Business Capability Estimation

According the participants I-3 and I-2(see Table 2), the accuracy of software estimation depends on the accuracy of the requirements and the design rationale. During the field study, it was found that Group Technology could not provide accurate capability impact analysis estimation across many projects. Due to the dependency analysis issue described in the previous paragraph the estimation was also not accurate because not all the impacted components were known at estimation time.

Participants - Lead Architect and Enterprise Architect:

"Managing and report on capability cost and dependencies across projects per FPC not possible. Group Technology cannot provide accurate FPC based upon business capability"

Participant - SOA Specialist:

"Group Technology cannot provide accurate FPC for Service Oriented Architecture and Enterprise Data Modelling projects" (Participants I-3 to I-7 agrees).

Participant - All:

"Accuracy of FPC depends upon the accuracy of the requirements and the design model information".

Literature also concurs that accurate change impact analysis depends on the accuracy of information regarding the enterprise systems complexity analysis - how capabilities are enabled by business components; the evolvability of models by [15] and [48] – how the traceability, capturing and documentation between artifacts [28] and [48] can provide information to determine impact analysis of requirements.

D. Gaps in Capability Toolset

Function Point Count is used by business to measure productivity and to estimate development and support effort estimation. Function Point measures software functionality from the user's perspective [3]. This is usually based on requirements descriptions which are presented and modeled in UML [8]. Changes to requirements mean the UML model changes. There is a need to have consistency with the requirements represented in the models so that accurate FPC can be conducted [7].

Participant – System Analyst:

"Function point Count Tool cumbersome not show actual realization of application architecture complexity" (I4, I6-I8)

Participant – Senior Manager:

"FPC analysis is time consuming. We have to look at methods to semi automate the method and process. Recapture of design models double the time during pre-execution" (I1, I3-i7)

Currently function point count estimations are only counting the component level and not on capability level. This mean that all the FPCs' for all the architecture components, that are linked to a Business Capability as defined in the HLCM, will be added up to give the sum total of a FPC for that Business Capability. This will now enable management to report a function point count sizing effort on a Business Capability level.

VII. CONCLUSION

The stage 1 of the Research Methodology for this research has been completed by completed all the above mentioned steps shown in Table I. This study by the literature review, the interviews and observations what the main issues and gaps are in change impact analysis of business architecture capabilities. It was shown that there is no linkage in the TOGAF meta-model between business architecture capability and the resources as proposed by [4]. A new proposed relationship "IsEnabledBy" was added between capability and the different type of resources which are all the assets that are owned or controlled by an organization. Resources are described in the EA stack by using the building blocks namely Business Architecture, Application Architecture, Information Architecture and Technology Architecture. Other resources are also assigned like the people (organization), the processes and the business services which support the business. The relationship between organization responsibilities was also added as "owns". Each of the architecture components that do enable a capability were also classified as a "measureable entity" according [32]. The attributes that determine the impact and cost were added in the perspective of function point analysis. The meta-model make provision to be able to add any attribute that determine the measure. It was shown that current meta- models make no provision to determine a function point count per capability. Currently it is manually calculated across many projects which do tie up with the EAM toolset view of sizing within the organization as confirmed in the interviews. Hence, integration between EAM and the design models will be proposed which will provide a more accurate view of the enterprise services and enterprise data complexity during FPA. The purpose of the study will then be to proof that this hypothesis holds true, "Business Architecture Capabilities are measurable by using the sum of the impacted measurements of the resources that enable the capability." Evaluation of the eeffectiveness, efficiency and usefulness of the artifact with the organization criteria for the toolset will be based upon comparisons against research objectives. Evaluation techniques will include Ex-post analysis based on documented use cases and previous function count completed at the software component level, Laboratory experiment: eye tracking to analyse the design of the integration data of the toolset in detail, Expert interviews with different stakeholders (technicians, management), eye tracking to analyse plausibility and obstacles and Field study of initially defined use cases and architecture capabilities and comparison with historically documented architecture capabilities and use cases. Further research should be to develop a method of application to illustrate the use of the capability meta-model for the development of solutions integrating EA models. Also to develop a toolset, to support and demonstrate the use of the capability meta-model, for the integration of EA models. Define and develop generic specific relationships that can be used to associate the individual stand-alone Artefact for EA in general to show a solution to the problem space of dependency analysis across distributed software architecture repositories.

APPENDIX

Table I. Research methodology and research objectives matrix

Research Methodology Step	Phase Outcome	Research Objectives	Data Collection and Analysis Method
Phase 1 - Awareness of the Problem [38]	There is a need for a toolset to enable Business Capability design specifications and models that are spread over more than one repository to be unified so that change impact analysis and dependency analysis can be determined.	[R-OBJ 1]: To conduct a literature analysis to determine the current status of research and practice related to modelling of IS Capability concepts, practices, methodologies, frameworks and related issues.	Literature Analysis, Field Study, Interviews & Observations, Meta Design.
Phase 1- Suggestion [38]	Proposal of possible meta-model and high level design components that can constitute a pattern for Enterprise Architecture to integrate models giving a capability perspective.	[R-OBJ 2.1]: To develop a meta-model that can constitute a pattern for Enterprise Architecture to integrate heterogeneous models to give a Capability perspective. [R-OBJ 2.2]: To identify the essential components of a conceptual model that can be used to create Capability perspective model solutions for EA.	Literature Analysis, Qualitative Analysis and Constructive Research Method
Phase 2 – Development [56]	Design and implement a software toolset that integrates an enterprise architecture planning models and detailed software solution UML design models.	[R-OBJ 3.1]: To develop a method of application to illustrate the use of the Capability meta-model for the development of solutions integrating EA models. [R-OBJ 3.2]: To develop a toolset to illustrate the use of the Capability meta-model for integration EA models.	Literature Analysis, Constructive Research Method.
Phase 3- Evaluation [56]	Evaluation of the tool according the research objectives and requirements.	[R-OBJ 4]: To develop generic specific relationships that can be used to associate the individual stand-alone Artefact for EA in general.	Field Work + Interviews + Observation
Phase 4- Communicate [56]	Abstraction and reflection consisting of knowledge contribution on the design theories for creating integrated capability perspectives for EA.		Report Write-Up

Table II. Semi-structured interviewees

Interview	Job Title	Department	Tasks
I-1	Senior Managers	System Design	Manage cost estimations of all projects towards business owners and project managers from a System Development perspective.
I-2	Enterprise Architect	Enterprise Architecture	Manage cost estimations of all projects towards business owners and project managers from an EAM perspective.
I-3	Lead Architect	Enterprise Architecture	Manage enterprise architecture management toolsets
I-4	SOA Specialist 1	Enterprise Services	Design SOA services
I-5	SOA Specialist 2	Enterprise Services	Design SOA services
I-6	Enterprise Data Architect	Enterprise Architecture	Design Enterprise data models for services
I-7	Senior System Analyst	System Design	Design application architecture in UML and do function point count analysis
I-8	Intermediate System Analyst	System Design	Design application architecture in UML and do function point count analysis
I-9	Function Point Count Specialist	Estimation and Tooling	Verify function point per projects received from design team

Table III. Summary of themes from interviews and observations

Literature Review Concept	*Code/Theme Analysis*	*Thematic Occurrences*	*Literature Review*
Capability Dependency Analysis	*Capability*	*32*	*[19], [26], [27], [28], [30], [48], [52], [61], [72], [77], [78]*
EAM Change Impact Analysis	*Capability*	*18*	
Model Integration	*Interoperability*	*10*	*[21], [22], [23], [37], [47], [62]*
SOA Complexity Analysis	*Estimation*	*10*	*[7], [8], [25], [28], [48]*
Documentation	*Estimation*	*16*	
Capture and Determining of FPC	*Toolset*	*10*	*[2], [17] [18], [24], [28], [30], [41], [69], [78]*
Traceability towards EA	*Toolset*	*8*	

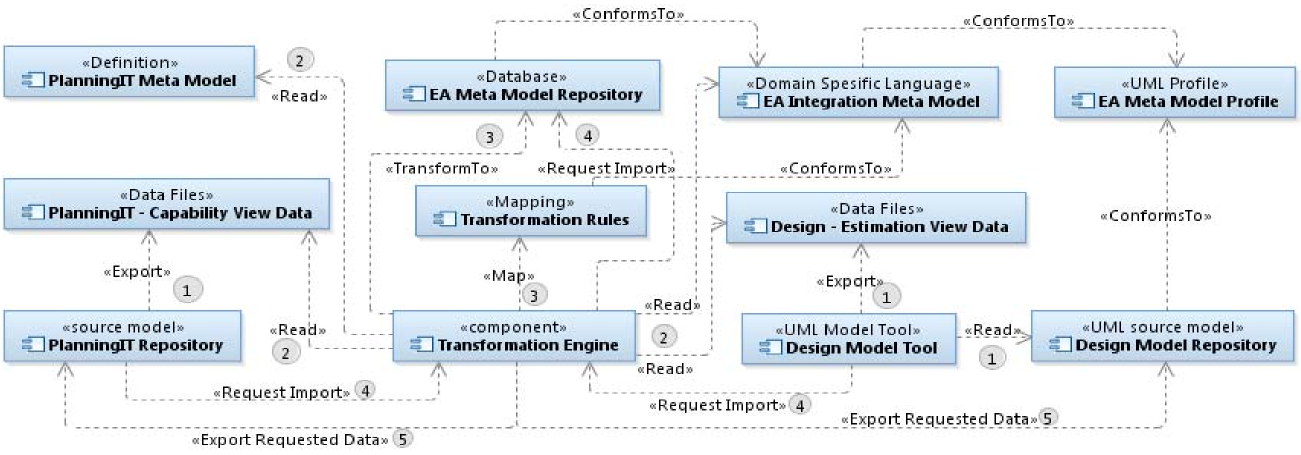
«Definition» PlanningIT Meta Model
«Database» EA Meta Model Repository
«Domain Spesific Language» EA Integration Meta Model
«UML Profile» EA Meta Model Profile
«Data Files» PlanningIT - Capability View Data
«Mapping» Transformation Rules
«Data Files» Design - Estimation View Data
«source model» PlanningIT Repository
«component» Transformation Engine
«UML Model Tool» Design Model Tool
«UML source model» Design Model Repository
«ConformsTo»
«ConformsTo»
«ConformsTo»
«ConformsTo»
«Read» 2
«TransformTo» 3
«Request Import» 4
«Export» 1
«Map» 3
«Read» 2
«Read»
«Read» 2
«Export» 1
«Read» 1
«Request Import» 4
«Request Import» 4
«Export Requested Data» 5
«Export Requested Data» 5

Figure 1 – EA model integration and transformation

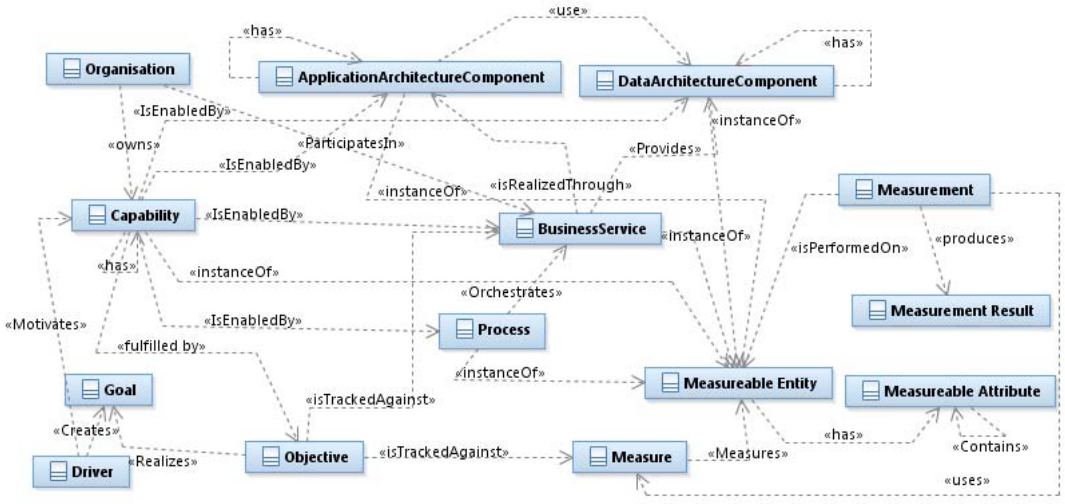

Figure 2 – An EA capability meta-model

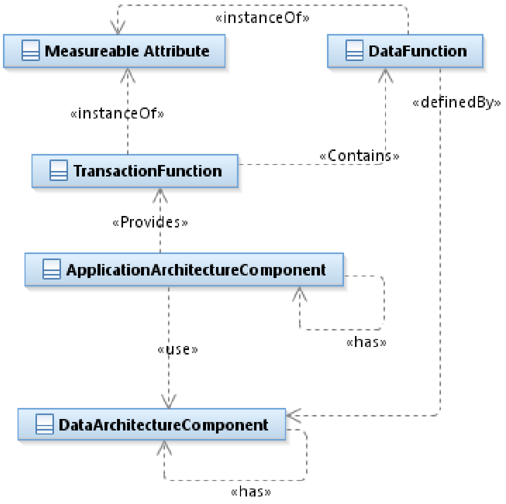

Figure 3 – An EA Capability Measureable Attributes Function Count Model

REFERENCES

[1] Ahlemann, F., Stettiner, E., Messerschmidt, M., & Legner, C. (2012). *Strategic enterprise architecture management: challenges, best practices, and future developments*. Springer.

[2] Ahmad, N., Wynn, D. C., & Clarkson, P. J. (2013). Change impact on a product and its redesign process: a tool for knowledge capture and reuse. Research in Engineering Design, 24(3), 219-244.

[3] Albrecht, A. J. (1994). Function point analysis. *Encyclopedia of Software Engineering*, *1*, 518-524.

[4] Azevedo, C. L., Iacob, M. E., Almeida, J. P. A., van Sinderen, M., Pires, L. F., & Guizzardi, G. (2013, September). An ontology-based well-founded proposal for modeling resources and capabilities in ArchiMate. In *Enterprise Distributed Object Computing Conference (EDOC), 2013 17th IEEE International* (pp. 39-48). IEEE.

[5] Barcellos, M. P., Falbo, R. D. A., & Dal Moro, R. (2010, July). A Well-Founded Software Measurement Ontology. In *Proceedings of the 2010 conference on Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS 2010)* (pp. 213-226). IOS Press.

[6] Barroero, T., Motta, G., & Pignatelli, G. (2010). Business capabilities centric enterprise architecture. In *Enterprise architecture, integration and interoperability* (pp. 32-43). Springer Berlin Heidelberg.

[7] Batista, V. A., Peixoto, D. C., Borges, E. P., Pádua, W., Resende, R. F., & Pádua, C. I. P. (2011). ReMoFP: a tool for counting function points from UML requirement models. *Advances in Software Engineering*, *2011*, 1.

[8] Batista, V. A., Peixoto, D. C., Pádua, W., & Pádua, C. I. P. (2012). Using UML stereotypes to support the requirement engineering: a case study. In *Computational Science and Its Applications–ICCSA 2012* (pp. 51-66). Springer Berlin Heidelberg.

[9] Bennett, K. (1996). Software evolution: past, present and future. Information and software technology, 38(11), 673-680.

[10] Berenbach, B. The evaluation of large, complex UML analysis and design models. Proceedings of the 26th International Conference on Software Engineering, Edinburgh, United Kingdom, 23 – 28 May 2004 (pp. 232-241). IEEE Computer Society.

[11] Biggs, B. (2005). Ministry of Defence Architectural Framework (MoDAF).

[12] Booch, G., Rumbaugh, J., & Jacobson, I. (1997). The Unified Modeling Language For Object Oriented Development, Documentation Set Version 1.0.

[13] Bredemeyer, D., Malan, R., Krishnan, R., & Lafrenz, A. (2003). Enterprise Architecture as Business Capabilities Architecture. *Bredemeyer Consulting*.

[14] Breivold, H. P., Crnkovic, I., & Larsson, M. (2012). Software architecture evolution through evolvability analysis. *Journal of Systems and Software*, *85*(11), 2574-2592.

[15] Breivold, H. P., & Crnkovic, I. (2009, August). Analysis of software evolvability in quality models. In *Software Engineering and Advanced Applications, 2009. SEAA'09. 35th Euromicro Conference on* (pp. 279-282). IEEE.

[16] Briand, L. C., Labiche, Y., & O'sullivan, L. (2003) Impact analysis and change management of UML models. Proceedings of the 19th International Conference on Software Maintenance (ICSM), Amsterdam, The Netherlands, 22-26 September 2003 (pp. 256-265). IEEE.

[17] Briand, L. C., Labiche, Y., O'Sullivan, L., & Sówka, M. M. (2006). Automated impact analysis of UML models. Journal of Systems and Software, 79(3), 339-352.

[18] Briand, L. C., Labiche, Y., & Yue, T. (2009). Automated traceability analysis for UML model refinements. Information and Software Technology, 51(2), 512-527.

[19] Bohner, S. A. (1996, November). Impact analysis in the software change process: a year 2000 perspective. In *2013 IEEE International Conference on Software Maintenance* (pp. 42-42). IEEE Computer Society.

[20] Buckl, S., Ernst, A. M., Lankes, J., Matthes, F., & Schweda, C. M. (2008, September). Enterprise architecture management patterns--exemplifying the approach. In Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE (pp. 393-402). IEEE.

[21] Chechik, M., Nejati, S., & Sabetzadeh, M. (2012). A relationship-based approach to model integration. Innovations in Systems and Software Engineering, 8(1), 3-18.

[22] Chen, D., & Daclin, N. Framework for Enterprise Interoperability. In Interoperability for Enterprise Software and Applciations: Proceedings of the Workshops and the Doctoral Symposium of the Second IFAC.IFIP I-ESA International Conference: EI2N, WSI, IS-TSPQ 2006 (pp. 77-88). ISTE.

[23] Chen, D., Doumeingts, G., & Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. Computers in industry, 59(7), 647-659.

[24] Chen, J. C., & Huang, S. J. (2009). An empirical analysis of the impact of software development problem factors on software maintainability. Journal of Systems and Software, 82(6), 981-992.

[25] Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., & Rackham, G. (2005). Impact of service orientation at the business level. IBM Systems Journal, 44(4), 653-668.

[26] Cuenca, L., Ortiz, A., & Boza, A. (2010). Business and IS/IT strategic alignment framework. In *Emerging Trends in Technological Innovation* (pp. 24-31). Springer Berlin Heidelberg.

[27] de Boer, F. S., Bonsangue, M. M., Groenewegen, L. P. J., Stam, A. W., Stevens, S., & Van Der Torre, L. (2005, August). Change impact analysis of enterprise architectures. In *Information Reuse and Integration, Conf, 2005. IRI-2005 IEEE International Conference on.* (pp. 177-181). IEEE.

[28] de Graaf, K. A., Tang, A., Liang, P., & van Vliet, H. (2012, August). Ontology-based software architecture documentation. In Proceedings of 2012 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), Helsinki, Finland, 20-24 August 2012 (pp. 121-130). IEEE.

[29] Del Fabro, M. D., & Valduriez, P. (2007, March). Semi-automatic model integration using matching transformations and weaving models. In *Proceedings of the 2007 ACM symposium on Applied computing* (pp. 963-970). ACM.

[30] Feng, T., & Maletic, J. I. Applying dynamic change impact analysis in component-based architecture design. Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, (SNPD 2006). Las Vegas, NV. 19-20 June 2006 (pp. 43-48). IEEE.

[31] Fraternali, P., Tisi, M., & Bongio, A. (2006, October). Automating function point analysis with model driven development. In *Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research* (p. 18). IBM Corp..

[32] García, F., Ruiz, F., Calero, C., Bertoa, M. F., Vallecillo, A., Mora, B., & Piattini, M. (2009). Effective use of ontologies in software measurement. *The Knowledge Engineering Review*, *24*(01), 23-40.

[33] Gerber, A., Kotzé, P., & Van der Merwe, A. (2010). Towards the formalisation of the TOGAF Content Metamodel using ontologies.

[34] Haki, M. K., Legner, C., & Ahlemann, F. (2012). Beyond EA Frameworks: Towards an Understanding of the Adoption of Enterprise Architecture Management.

[35] Hammoudi, S., Janvier, J., & Lopes, D. (2005). Mapping Versus Transformation in MDA: Generating Transformation Definition from Mapping Specification. *Foundational Ontologies*, 33.

[36] Hauder, M., Matthes, F., & Roth, S. (2012). Challenges for automated enterprise architecture documentation. In *Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation* (pp. 21-39). Springer Berlin Heidelberg.

[37] Hawkins, J. L., Khusial, D., Lyons, K. A., McAllister, M. J., McKegney, R., McKenna, M. D., & Slonim, J. (2014). *U.S. Patent No. 8,707,260*. Washington, DC: U.S. Patent and Trademark Office.

[38] Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quaterly, 28(1), 75-105.

[39] Hevner, A., & Chatterjee. S. (2010). Design science research in information systems: theory and practice. Vol. 22. Springer, 2010.

[40] Iacob, M. E., Quartel, D., & Jonkers, H. (2012, September). Capturing business strategy and value in enterprise architecture to support portfolio valuation. In *Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International* (pp. 11-20). IEEE.

[41] IFPUG, F. (2000). International Function Point Users Group (IFPUG) Function Point Counting Practices Manual.

[42] Jonkers, H., Lankhorst, M. M., ter Doest, H. W., Arbab, F., Bosma, H., & Wieringa, R. J. (2006). Enterprise architecture: Management tool and blueprint for the organisation. *Information Systems Frontiers*, *8*(2), 63-66.

[43] Kasunic, M. (2001). *Measuring systems interoperability: Challenges and opportunities*. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

[44] Khalifelu, Z. A., & Gharehchopogh, F. S. (2012). Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation. Procedia Technology, 1, 65-71.

[45] Kim, G., Shin, B., Kim, K. K., & Lee, H. G. (2011). IT capabilities, process-oriented dynamic capabilities, and firm financial performance. *Journal of the Association for Information Systems*, *12*(7), 487-517.

[46] Kocaguneli, E., Menzies, T., Bener, A., & Keung, J. W. (2013). Exploiting the essential assumptions of analogy-based effort estimation. *Software Engineering, IEEE Transactions on*, *38*(2), 425-438.

[47] Kolovos, D. S., Paige, R. F., & Polack, F. A. On-demand merging of traceability links with models. In Proceedings of the Second European Conference on Model Driven Architecture Foundations and Applications, Bilbao, Spain, 10-13 July 2006 (pp. 21-29).

[48] Lanza, M. (2001, September). The evolution matrix: Recovering software evolution using software visualization techniques. In Proceedings of the 4th international workshop on principles of software evolution, Vienna University of Technology, Austria, 10-11 September 2001 (pp. 37-42). ACM Press. New York.

[49] Lehman, M. M. (1980). Programs, life cycles, and laws of software evolution. Proceedings of the IEEE, 68(9), 1960-1076.

[50] Leung, H., & Fan, Z. (2002). Software cost estimation. *Handbook of Software Engineering, Hong Kong Polytechnic University*.

[51] López, C., Inostroza, P., Cysneiros, L. M., & Astudillo, H. (2009). Visualization and comparison of architecture rationale with semantic web technologies. *Journal of Systems and Software*, *82*(8), 1198-1210.

[52] Luthria, H., Aurum, A., Low, G. C., & Rabhi, F. A. (2010). Aligning Service Requirements with Business Strategy: A Proposed Stakeholder Value Model for SOA. In *Information Systems Development* (pp. 149-156). Springer US.

[53] Mahmood, K., Ilahi, M. M., Ahmad, B., & Ahmad, S. (2012). Empirical Analysis of Function Points in Service Oriented Architecture (SOA) Applications. *Industrial Engineering Letters*, *2*(1), 6-12.

[54] Matthes, F., Buckl, S., Leitel, J., & Schweda, C. M. (2008). *Enterprise architecture management tool survey 2008*. Techn. Univ. München.

[55] Mellor, S. J. (Ed.). (2004). *MDA distilled: principles of model-driven architecture*. Addison-Wesley Professional.

[56] Ostrowski, L., & Helfert, M. (2012). Reference model in design science research to gather and model information.

[57] Nehaniv, C. L., & Wernick, P. (2007). Introduction to software evolvability. In *Proceedings of* 9th International Workshop on Principles of Software Evolution (IWPSE 2007), in conjunction with the 6th ESEC/FSE joint meeting, Dubrovnik, Croatia, 3-4 September 2007. ACM 2007 ISBN 978-1-59593-722-3.

[58] Pandey, P. (2013). Analysis of the Techniques for Software Cost Estimation. In *Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on* (pp. 16-19). IEEE.

[59] Panetto, H., Berio, G., Benali, K., Boudjlida, N., & Petit, M. (2004). A unified enterprise modelling language for enhanced interoperability of enterprise models. In 11th IFAC INCOM2004 Symposium, April 5th-7th, Bahia, Brazil (pp. 1-12).

[60] Peffers, K., Tuunanen, T., Rothenberger, M. A., &Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of management information systems, 24(3), 45-77.

[61] Peppard, J., & Ward, J. (2004). Beyond strategic information systems: Towards an IS capability. The Journal of Strategic Information Systems, 13(2), 167-194.

[62] Piedra, N., Chicaiza, J. A., López, J., & Tovar, E. (2014). An Architecture based on Linked Data technologies for the Integration and reuse of OER in MOOCs Context. *Open Praxis*, *6*(2), 171-187.

[63] Ramasubbu, N., & Balan, R. K. (2012, June). Overcoming the challenges in cost estimation for distributed software projects. In *Software Engineering (ICSE), 2012 34th International Conference on* (pp. 91-101). IEEE.

[64] Remenyi, D., Money, A., & Bannister, F. (2007). The effective measurement and management of ICT costs and benefits. Elsevier.

[65] Ross, J. W., Weill, P., & Robertson, D. C. (2006). Enterprise architecture as strategy: Creating a foundation for business execution. Harvard Business Press.

[66] Rumbaugh, J., Jacobson, I., & Booch, G. (2004). Unified Modeling Language Reference Manual, The. Pearson Higher Education.

[67] Santillo, L. (2007). Seizing and sizing SOA applications with COSMIC function points. *Proceedings of SMEF*.

[68] Scanniello, G., Gravino, C., Genero, M., Cruz-Lemus, J. A., & Tortora, G. (2014). On the impact of UML analysis models on source-code comprehensibility and modifiability. ACM Transactions on Software Engineering and Methodology (TOSEM), 23(2), 13.

[69] Tang, A., Liang, P., Clerc, V., & van Vliet, H. (2011). Supporting co-evolving architectural requirements and design through traceability and reasoning. Relating Software Requirements and Architectures. Springer, Heidelberg.

[70] The Open Group (2009). TOGAF 9: The Open Group Architecture Framework (TOGAF). Document Number: G091. http://www.opengroup.org/architecture/togaf9-doc/arch/

[71] U. R. T Force, (1999). OMG UML Specification.

[72] Ulrich, W., & Rosen, M. (2011). The Business Capability Map: The" Rosetta Stone" of Business/IT Alignment. Cutter Consortium, Enterprise Architecture, 24(4).

[73] Vaishnavi, V. K., & Kuechler Jr, W. (2007). *Design science research methods and patterns: innovating information and communication technology*. CRC Press.

[74] Wagner, C. (2004). Enterprise strategy management systems: current and next generation. *The Journal of Strategic Information Systems*, *13*(2), 105-128.

[75] Wijayasiriwardhane, T., & Lai, R. (2010). Component point: a system-level size measure for component-based software systems. *Journal of Systems and Software*, *83*(12), 2456-2470.

[76] Wilkie, F. G., McChesney, I. R., Morrow, P., Tuxworth, C., & Lester, N. G. (2011). The value of software sizing. *Information and Software Technology*,*53*(11), 1236-1249.

[77] Zhang, H., Li, J., Zhu, L., Jeffery, R., Liu, Y., Wang, Q., & Li, M. (2014). Investigating dependencies in software requirements for change propagation analysis. Information and Software Technology, 56(1), 40-53.

[78] Zhao, J., Yang, H., Xiang, L., & Xu, B. (2002). Change impact analysis to support architectural evolution. Journal of software maintenance and evolution: research and practice, 14(5), 317-333