COLLABORATED ARCHITECTURE FRAMEWORK FOR COMPOSITION UML 2.0 IN ZACHMAN FRAMEWORK

Hermawan¹, Fika Hastarista²
¹,²Jurusan Teknik Informatika, Fakultas Teknik, Universitas Trunojoyo Madura, Indonesia (69162)

E-mail: hermawan@trunojoyo.ac.id¹, fika@trunojoyo.ac.id²

Abstract. Zachman Framework (ZF) is the framework of enterprise architecture that most widely adopted in the Enterprise Information System (EIS) development. In this study, has been developed Colaborated Architecture Framework (CAF) to collaborate ZF with Unified Modeling Language (UML) 2.0 modeling. The CAF provides the composition of ZF matrix that each cell is consist of the Model Driven architecture (MDA) from the various UML models and many Software Requirement Specification (SRS) documents. Implementation of this modeling is used to develops Enterprise Resource Planning (ERP). Because ERP have a coverage of applications in large numbers and complexity relations, it is necessary to use Agile Model Driven Design (AMDD) approach as an advanced method to transforms MDA into components of application modules with efficiently and accurately. Finally, through the using of the CAF, give good achievement in fullfillment the needs from all stakeholders that are involved in the overall process stage of Rational Unified Process (RUP), and also obtaining a high satisfaction to fullfiled the functionality features of the ERP software in PT. Iglas (Persero) Gresik.

Keywords: Agile Model Driven Design (AMDD), Colaborated Architecture Framework (CAF), Model Driven Architecture (MDA), Unified Modelling Language (UML), Zachman Framework (ZF).

1. Introduction
Previously, various tools and techniques are provided for information system (IS) development, that are specific to an aspect or many aspects relation, such as the using of the basic flowchart shapes, ERD modeling, Data Flow Diagram, or Object Oriented modelling. However, in accordance with the demands of the information technology (IT) development requirements today, where so many stakeholders involved with many different perspectives and objectives that would influence system, so IT developers require more integrated architecture blue print to fullfilled overall Rational Unified Process (RUP) stages in IT projects.
Enterprise Architecture Framework (EAF) is a tool that can be used for developing a broad range of different architectures, where set of tools for EAF is supported with list of recommended on standardization within models and documents to build fully block of IT conceptual blueprint. Benefit of the using an EAF will speed up and simplify architecture development, ensure more complete coverage of the designed solution [1].

The ZF is one of the most widely adopted EAF in the Enterprise Information System (EIS) development [2]. The ZF provides a concept to produce a systematic taxonomy to declares a relationship without replacing anything in the real world and changing the tools, techniques and methodologies, but otherwise give collaboration of various viewpoints according to many perspectives and abstraction for architecture modelling [3].

ZF same as others EAF such TOGAF, IAF, FEAF, DoDAF within the primary principal consists of conceptual modeling language, that provides modelling platform and analysis list of the EA included strategy, process and dimensions of the application, as well as support various aspects of the structure and behavior of software artifacts [4]. In generally each EAF has different focus in providing architectural solutions. By reviewing two of the most popular EAF framework that are ZF and TOGAF, ZF more focused on the constructing architecture views in multi perspective of the EA, while TOGAF as view of process to describe the RUP stages to constructing the EA [5].

However in RUP stages, many EA practitioners prefer to using hybrid architecture approach to combine various EAF by taking some of the advantages of each frameworks [5]. Such as Integrated Architecture Framework (IAF) which provides a best practices approach more in order to integrated the various frameworks that have been existing. Where the advantages of the IAF lies in values and flexibilities within combination with other EA such Togaf ADM and ZF[6].

IAF has focused more on the processes to “Why, What, and How” as best practices to give transition solving from business components into application components, but less provided scalable architectural view then ZF. While ZF didn’t defined specific modelling like as support object oriented modelling where still be a mature paradigm to analyzing and developing software components today.

For this reason, in this study we propose Colaborated Architecture Framework (CAF) that is having focus as the best practice for composing structured models and documents to defines the software requirement specification (SRS) blue print, that based on object-oriented modeling approach use the UML 2.0, which has more scalable architecture view in “What, How, Where, Who, When, Why” adopting ZF to fullfill overall RUP stages in SDLC.

2. Methodology
2.1 Collaborations of Zachman Framework and UML 2.0

The CAF composing UML within the ZF 6x6 cell of matrix dimensions, in accordance with the six perspective of Scope (planner's view), Enterprise model (business owner's view), Model of fundamental concepts (designer's view), Technology model (builder's view), Detailed representations (sub-contractor's view), Functioning system. Which provide solutions for the six dimensional abstraction of english interrogatives that are consist of: What (Data), How (Function), Where (Network), Who (People), When (Time), Why (Motivation). The using of UML to support ZF architecture is the systematic composition to be developed [7] which makes itself a basis for evaluating, establishing, and customizing other enterprise architectural frameworks, methods, and tools [8].

Within the ERP that we developed for PT. Iglas (Persero) Gresik-Indonesia consists of 7 modules: Financial-Accounting (FA), Customer Relationship Management (CRM) - Distribution, Material-Production, Procurement, Human Resource Management, Inventory, and asset management [9].
Which for sample implementation case we focus to describing CAF for one domain module that is the FA, as a core of ERP modules like as shown in Table 1.

Table 1. Composition CAF of the UML models and SRS documents in the Zachman Framework [3]

| Perspektive / Abstraction | What (Things) | How (Process) | Where (Location) | Who (People) | When (Time) | Why (Motivation) |
|---------------------------|---------------|---------------|------------------|--------------|-------------|-----------------|
| Planner/ Contextual (Scope) | (P1A1) Structure diagram of The organization units and they responsibilities in the FA department | (P1A2) Table of value chain from business processes within the FA department | (P1A3) Map location from the FA department in P.T. Iglas (Persero) Gresik | (P1A4) Structure diagram of managers who are responsible to plan, execute, and control business processes within the FA | (P1A5) Workbase list of business process activities | (P1A6) Defines Strategies and Goals of vision and mission in the Iglas FA department |
| Owner/ Conceptual (Business Model) | (P2A1) Conceptual Data Model of the FA business processes | (P2A2) Business Process Model Notation (BPMN) of FA | (P2A3) Business Logistics Diagram of FA units | (P2A4) Use Case Diagram of the FA business processes | (P2A5) Operational Schedule of business activities in the FA | (P2A6) Defines Goals of each FA business processes |
| Designer/ Logical (System Model) | (P3A1) Class Diagram for all objects in the FA domain | (P3A2) Component Diagram of application component | (P3A3) Sequence Diagram for application component | (P3A4) Component dependency matrix from one application component to others | (P3A5) RUP schedule of Model Driven Architecture | (P3A6) Defines logical design of software artifacts |
| Builder/ Physical (Technology Model) | (P4A1) Physical Data Model of database | (P4A2) Deployment Diagram of installation package of modules | (P4A3) Network Diagram of server farm and computer access | (P4A4) Interface Design for each functionality application | (P4A5) RUP schedule of Model Driven Development | (P4A6) Defines physical design of software artifacts |
| Detailed Representation (Sub-Contractor) | (P5A1) Structure and details of Django Framework models of each application components. | (P5A2) Source code descriptions of Model View Template from the Django web framework | (P5A3) List of the networks configuration for all networks component include: router, switch, wifi, and cabling. | (P5A4) Table of RACI matrix as a policy to configure security of access rights for each user who manages the software applications | (P5A5) SDL schedule of installation and maintenance | (P5A6) Defines installation and maintenance guaranty contract |
| Function Enterprise | (P6A1) Integration link between FA domain and other ERP domains | (P6A2) Package diagram of component services relations for the FA domain and other domains | (P6A3) List of the global network configurations include internet and network security | (P6A4) Activity diagram to manage activities to operate ERP services. | (P6A5) Time schedule for monitoring and controlling ERP services. | (P6A6) Defines System Operational Procedure (SOP) to operate overall ERP application services. |

---

1 All artifact accordance within the Iglas ERP blueprint
Not many previous studies that laid the blueprint mapping EAF for ERP because generally it does by many enterprise corporates in big research collaboration. In one individual study, Molnar outlines a wide-range theoretical background for analyzing the re-engineering and re-organization of ERP systems, and describes EIS relationship with EAF including ZF [10], but this study was still have struggling on the theory and strategy paradigm without any more best practice approach to defined architecture constructing.

That means, the CAF modeling can be a solution in the architecture constructing and then can be used as a choice in the best practice framework for the implementation of EAF in the development of EIS including the ERP system.

2.2. Model Driven Architecture
The CAF, it should define no-overlap determination models for architectures modeling because every cell has different perspective or objective abstraction, so it has special specifications for different problem domains [11]. To construct composition UML models in the ZF we use important models that are most influencing architecture as MDA where is shown in Table 2.

| Zachman Framework Cell | MDA Specification |
|------------------------|--------------------|
| Business Model (Owner/Conceptual), People (Who) (P2A2) | Workflow modeling in Business Process Modeling Notation (BPMN) to collaboration departments and units which have responsibility to manage business process activity |
| Business Model (Owner/Conceptual), People (Who) (P2A2) | Usecase Diagram show user requirements within business process activities |
| System Model (Designer/Logical), Data (What) (P3A1) | Class Diagram modeling to define encapsulation of data attributes, methods and they relations to create components of software. |
| System Model (Designer/Logical), Function (How) (P3A2) | Component Diagram illustrates the message to exchange data from one component to other components by request method that is exposed |
| System Model (Designer/Logical), Location (Where) (P3A3) | Sequence Diagram illustrates the interaction of methods in integration an application service |
| Technology Model (Builder/Physical), Function (How) (P4A2) | Deployment Diagram illustrates the deployment of software components into hardware components |
| Integration Model (Function Enterprise) (How) (P5A2) | Package diagram for encapsulation all components in a single domain and illustrates collaboration link with other domains. |
| Integration Model (Function Enterprise) (Who) (P5A4) | Activity diagram illustrates step by step user activities to manage application. |

MDA provides a way to define the structured view that allows a model to be constructed, viewed, developed, and manipulated at analysis and design time before it is coded [8].

2.3. Agile Model Driven Development
Furthermore, through the AMDD for supporting transition the modelling artifacts to the software artifacts [12], the application development can be done quickly and accurately meets to the feature requirements. So the coverage needs of EIS that are consists of persistent data, a lot of data, access data concurrently, a lot of user interface screens, and integrate with other enterprise applications are
achieved [13]. The main reference models for AMDD that we have executed to develop software components, they are: BPMN as a Business Service, Class Diagram as a Component Entity Service, and Sequence Diagram as a Component Utility Service [14].

From one sample scenario in the FA module, we analyze business process in financial transaction activities, where a register of financial business process is the General Ledger (GL) through the Journal Entry service. We set the Journal Entry as a core of business service to receive data from all transactions which is affecting the financial business processes. The Journal Entry in the General Ledger is the final register transaction of overall data from the whole ERP system, as shown on Figure 1. BPMN for this process is placed in P2A2 cell.

![BPMN Diagram](image)

**Figure 1.** BPMN as business process in GL

From each activity in business services, we have composed into class diagram from each component entity services, like as shown in Figure 2 where class diagram for the Journal Entry in the GL is placed in P3A1 cell. From UML Class diagram, the model design is directly driven developing of component software, because class diagram is able to defines the data attributes and methods in the business service requirement, as data and operational service in application.

For AMDD programming strategy, we recommend the use of three-tier software architecture framework, which generally supported domain specific language in domain driven design programming concept and component-based plugins [15]. This strategy is having benefits: reduced communication across the layers, design by contract is sufficient, part of application components can be tested separately [16]. Among the development frameworks that could be an option, they are: Play for Java, Rails for Ruby, Grails for Groovy and Django for Python. Where Django framework as the technology of our choice due to excess AMDD in reliability, speed, and the components plugin are easily installed and modified.
Class diagrams became references to create functionally component application, within translated they attributes and methods in a *model class*, while for integration many functionally methods as a component utility service, we use sequence diagram model to design it. For the further step, sequence diagram as addition methods within *scratch script* inside a *model class*. Like as shown in Figure 3, component utility service to create report in the journal entry that is composed with other services.

By scaffolding template engine in the *django-admin*, class model generating application like as shown in Figure 4, where it is having functionally *Crud-Retrieve-Update-Delete* (CRUD) as a basic information system operation and would can testing in a few seconds time. This way will give generic solution for Test-Driven Development directly in the application to be launching.
Within functionally application for CRUD operation on data transaction, it works like as shown in Figure 5.

3. Result and Analyze
In the software development process that is carried out for six months by 2 analysts and 7 developers, we are working from drafting the model architecture until to implement the application programming able to produces EIS within 122 features of functionally applications in the seven modules of ERP, with details in Table 3.
Table 3. The numbers of ERP modules apps

| Module                        | Numbers |
|-------------------------------|---------|
| Financial Accounting          | 30      |
| Asset Management              | 18      |
| CRM – Distribution            | 18      |
| Material – Production         | 16      |
| Procurement Management        | 17      |
| Human Resource Management     | 13      |
| Inventory Management          | 11      |

Furthermore within comparison with the features of existing ERP applications such as Open ERP 7.0, the EIS features that have been developed for ERP PT. Iglas have some advantages in functionality, because the application development is done in Taylor made, but otherwise with minimize scratch code in programming component of software so that is easy to evaluate and reconstruct, and also reducing complexity to help better understand on the evolution of enterprise business processes and their requirements [16].

Such as shown in Table 4 features of Financial-Accounting ERP modules, which have consist of four business processes that is implemented by 30 functionally applications.

Table 4. Subjective evaluation for ERP PT. Iglas by they IT managers

| Business Process | Activity                                | *Score | Open ERP |
|------------------|-----------------------------------------|--------|----------|
| Account Receivable | Delivery of account invoice             | 80     | √        |
|                  | Payment Sales                          | 90     | √        |
|                  | Delivery of sales return account        | 80     | √        |
|                  | Delivery of assets sale account         | 80     | √        |
|                  | Payment of asset sales                  | 80     | X        |
|                  | Delivery of asset sales payment account | 90     | X        |
| Account Payable  | Delivery of purchase request account    | 80     | √        |
|                  | Expenditures payment                    | 90     | √        |
|                  | Delivery Expenditures payment account   | 80     | √        |
|                  | Payroll approval                        | 90     | √        |
|                  | Delivery payroll account                | 80     | X        |
| Cash/Bank        | Create the cash master                  | 90     | X        |
|                  | Create the bank master                  | 90     | √        |
|                  | Create the exchange master              | 90     | √        |
|                  | Create the tax master                   | 90     | √        |
|                  | Statement of Cash                       | 95     | √        |
|                  | Statement of Bank                       | 95     | √        |
|                  | Displacement Cash / Bank                | 90     | √        |
|                  | Capital investment                      | 85     | X        |
Business Process | Activity | *Score | Open ERP |
--- | --- | --- | --- |
Creating the Master Account | 90 | ✓ |
Creating the Master Fiscal Year | 90 | ✓ |
Creating the Master Period | 90 | ✓ |
Creating the Master Journal | 90 | ✓ |
The Jurnal Entry | 95 | ✓ |
The Jurnal Adjustment Entry | 85 | ✓ |
The Asset Depreciation Journal Entry | 85 | X |
GL Posting | 80 | ✓ |
Subledger reporting | 80 | ✓ |
Balance reporting | 80 | ✓ |
Loss/Profit reporting | 80 | ✓ |

4. Conclusion
The results of this study concluded, CAF that is composed with Zachman framework and UML fully supported MDA, and also through implementation of the most advanced method and technology of AMDD meets with the stages of the RUP process quickly and accurately, and also simple to learn and adjustment to establish the complexity requirements of EIS that is proofed to develop ERP application. Because today still a bit of study on the scope of software engineering to arrange EAF for the EIS development within detail UML modelling determination, so CAF can be implemented as the best practice reference to determining the artifacts models and documents that supports all works in the RUP which meets with the standard model and technology today.

5. Related Work
In the next study can be developed pattern of software design[17] and for advance using Service Oriented Architecture (SOA)[18] for transformation MDA to AMDD, which provides better best practice in enterprise application development methods.

6. Reference
[1] Rachel Harrison 2009, TOGAF™ Version 9 Foundation Study Guide, (The Open Group : Van Haren Publishing) chapter 2 p 16
[2] Carla Marques Pereira, Pedro Sousa, A method to define an Enterprise Architecture using the Zachman Framework, ACM symposium on Applied computing, Pages1366-1371, New York 2004.
[3] J. F. Sowa, J. A. Zachman, Extending and formalizing the framework for information systems architecture, IBM 1992
[4] Eric Yu, Markus Strohmaier, Xiaoxue Deng Exploring Intentional Modeling and Analysis for Enterprise Architecture, EDOCW 06, 10th IEEE International 2006
[5] Brian H. Cameron, Eric McMillan, Analyzing the Current Trends in Enterprise Architecture Frameworks, Journal of Enterprise Architecture pp 60-71, Association of Enterprise Architects 2013
[6] J. van ’t Wout et al. 2010, The Integrated Architecture Framework Explained, (Capgemini SA: Springer) chapter 4 p 161-200
[7] Nikolaidou M, et al., Employing Zachman Enterprise Architecture Framework to Systematically Perform Model-Based System Engineering Activities, System Sciences, pp 1 - 10, HICSS ’09. 42nd Hawaii International Conference 2009
[8] A. Fatolahi and F. Shams, An investigation into applyingUML to the Zachman Framework. Information Systems Frontiers, 8(2):133–143, 2006
[9] Dimitris N. Chorafas 2001, *Integrating ERP, CRM, Supply Chain Management, and Smart Material* (Florida:Aurbach)

[10] Bálint Molnár, The Country-specific Organizational and Information Architecture of ERP Systems at Globalised Enterprises, *BUSINESS SYSTEMS RESEARCH* Vol.2, No.2, pp.1-56, 2011

[11] Dragan Milicev 2009, *Model-Driven Development with Executable UML*, (Indiana: Wiley Publishing) chapter 3 p 27

[12] Scott W. Ambler 2004, et al., *Agile Model-Driven Development With UML 2.0*, (Cambridge : Cambridge University Press)

[13] Martin Fowler 2002, et al., *Patterns of Enterprise Application Architecture*, (Boston:Addison Wesley)

[14] Hermawan, Rianarto Sarno, Developing Distributed System with Service Resource Oriented Architecture, *TELKOMNIKA*, Vol.10, No.2, pp. 389-399, June 2012

[15] Johan den Haan, *DSL development: 7 recommendations for Domain Specific Language design based on Domain-Driven Design*, http://www.theenterprisearchitect.eu/blog/2009/05/06/dsl-development-7-recommendations-for-domain-specific-language-design-based-on-domain-driven-design/, published may 6 2009

[16] Mike Cohen 2010, *Succeeding With Agile*, (Boston: Addison Wesley)

[17] Frank Buschmann 2007, et al., *Pattern Oriented Software Architecture*, Volume 5, (New Jersey:John Wiley & Sons)

[18] Thomas Erl 2007, *SOA: Principles of Service Design* (New Jersey:Prentice Hall)