

Using Organization Goal-Oriented Requirements Engineering Result to Develop Sales Information System

Fransiskus Adikara, Bayu Hendradjaya, Benhard Sitohang

Abstract— Goal-oriented requirements engineering methods can use organization goals to elicit information system requirements. This paper will demonstrate how to use the result of organization goal-oriented requirements engineering process in sales information system development project. The research method will use Extreme Programming method to develop sales information system in a distribution company. OGORE approach will conduct in the planning phase to get system requirements and the result will be use in design phase using UML diagram. The testing phase will evaluate whether the system functions can meet requirements (goal) KPIs expectation.

Index Terms— information system design, goal-oriented requirement engineering, software engineering, organization goal

I. INTRODUCTION

INFORMATION System nowadays have been use to support organization to achieve business goals and increase the organization competitive advantage[1][2]. However, the information system still faces several problems, especially in requirements engineering and planning phases[3]. One of the requirement engineering techniques is Goal Oriented Requirements Engineering that refers to the use of goals for requirements elicitation, elaboration, organization, specification, analysis, negotiation, documentation and evolution[4].

In our previous paper, we proposes an extension of GORE approach that uses the organization goals (the overall objectives, purpose and general mission of an organization) to elicit system requirements[5]. In others paper, we also proposed to refine requirements with Case-Based Reasoning method to get the high quality requirements that are based on the reuse of requirements[6]. By doing requirements engineering based on organization goals, the system was expected to have functions to prioritize the needs of the stakeholders or organization, rather than based on the needs

of the users or activities on the operational level.

The system designed and developed in this paper is built upon the results of the Organization Goal-Oriented Requirements Engineering (OGORE) process. This paper will demonstrate how the system requirements captured during the requirements engineering phase is carried on to the design and development process, to then be tested if the results are able to fulfil the KPIs set. Results of the OGORE process, which has been modelled after the Goal Tree Model, will be translated using the Unified Modelling Language (UML) diagram[7] in the system design process.

In section two of this paper, we start with the overview of related works, and in section three, we explain our research methods. In section four, we demonstrate the system planning, design, coding, and we illustrate the result of the developed system. We then discuss the implementation of the result of the system development in section five, and finally, we conclude our research in section six.

II. RELATED WORKS

In our previous paper [5], we proposed an extension of GORE approach that used the organization goals (the overall objectives, purpose and general mission of an organization) to elicit system requirements, so the system functions and the resulting requirements can be more qualified and relevant with organizations goals. We also proposed to adapt Case-Based Reasoning (CBR)[8] method in requirements engineering process and combine it with AGORA method[9] to refine and analyze requirements in our previous work[6]. The combination was intended to get the high quality requirements that are based on the reuse of requirements to improve the requirements' quality, so that the requirements are focused more on the need of the organization, rather than on others requirements.

Our proposed concept from previous research[5] is explained in these steps (Figure 1). The first step is to define the existing vision, mission, and the organization goals. Requirements engineers with high level stakeholders (Board of Directors) discussed and identified which organization goals that expected to be achieved by the information system development. Furthermore, the second step we defined the KPIs of each goal as target to be achieved. These KPIs were also used as control tools to reduce user requirement risks caused by the user's personal interests[5]. These two steps are organization goal-oriented elicitation process.

Manuscript received January 10, 2017; revised February 02, 2017

Fransiskus Adikara is with the Informatics of Esa Unggul University, Jl. Teruasn Arjuna, Jakarta, Indonesia (phone: +6287888007037; e-mail: fransiskus.adikara@esaunggul.ac.id).

Bayu Hendradjaya is with the School of Electrical Engineering and Informatics of Institute Technology Bandung, Jl. Ganesha, Bandung, Indonesia (e-mail: bayu@informatika.org).

Benhard Sitohang is with the School of Electrical Engineering and Informatics of Institute Technology Bandung, Jl. Ganesha, Bandung, Indonesia (e-mail: benhard@informatika.org)

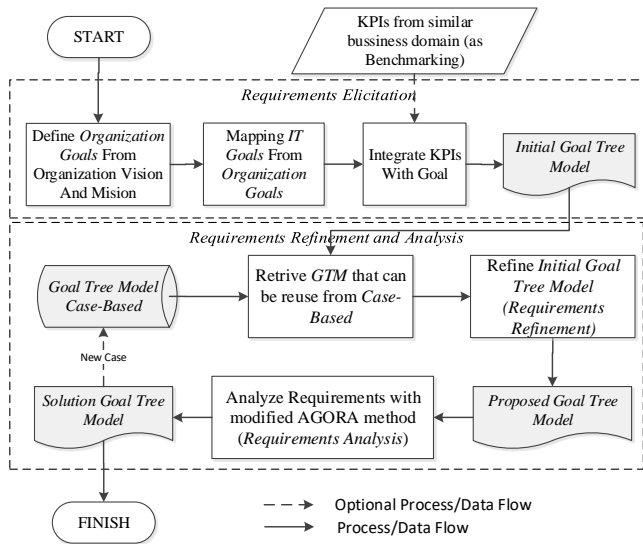


Fig. 1. Organization Goal-Oriented Requirements Engineering Method Flowchart Diagram.

After elicitation process, the Initial Goal Tree Model, KPIs (Key Performance Indicators) and business domains used as initial information for requirements refinement and analysis process. The third step is to refine requirements by processed elicitation result in Case-Based Reasoning methods that we proposed in previous research [6].

Unified Modelling Language (UML) is a modelling standard for software built using object-oriented programming techniques. UML emerged because of the need for a visual modelling system to specify, illustrate, build, and document software systems[10]. We use UML modelling to design the system based on the requirements elicitation process. In previous research [11], this approach was used to show and fill the gap between the requirements engineering phase and the design phase.

III. RESEARCH METHOD

The following are explanations of the research methods:

- 1) Data was collected from the OGORE process, in the form of Goal Tree Model.
- 2) Goal Tree Model was then summarized into the needs of the system along with its KPIs.
- 3) Development of the whole system using the Extreme Programming (XP) method[12] by doing:
 - a) Planning: collecting and eliciting the system requirements through OGORE process.
 - b) Design: converting the Goal Tree Model into

business processes scenario into a UML diagram.

- c) Coding: execution of the coding phase based on the UML diagram which has been established using Delphi application development platform, and the using SQL server as database engine.
- d) Testing of the developed programming code.

IV. SALES INFORMATION SYSTEM DEVELOPMENT

Based on Extreme Programming (XP) method[12], as previously explained, we started the development of sales information system, as will be explained in the next section.

A. Planning Process

We used a sales information system project in Distribution Company as a research object to demonstrate our approach in OGORE. The distribution company was growing and the sales processes were becoming more complex. The main business of Distribution Company is to distribute their fabric items to their retail channels. This company's vision is to become the most professional and a nation-wide fabric distribution company in Indonesia. Meanwhile, this company's mission is to distribute quality fabric to all areas in Indonesia with professional, reliable, and trustworthy services. From the company's vision and mission, the company describes the organization's goals as follows:

- 1) To increase purchase products control.
- 2) To increase sales products control.
- 3) To improve sales distribution channels.
- 4) To improve time delivery and items quality.

From these organization goals, requirements engineers and Board of Director prioritize number 2 goal to be support by sales information system. Based on this goal, we extracted system requirements using OGORE methods as explain in previous section. The OGORE method result is Solution Goal Tree Model (GTM) that modeling in Figure 2.

B. Design Process

In designing a new system, the author used UML diagrams consisting of use case diagram, class diagram, activity diagram, sequence diagram, and other diagrams that we provided in the full research report. Before we made a use case, the OGORE in form of Solution GTM must be mapped to use case component. The mapping result show in Table 1. The relationship between actors and use cases analysed in the user requirements table that modelled based on Table 2 and shown in Figure 3.

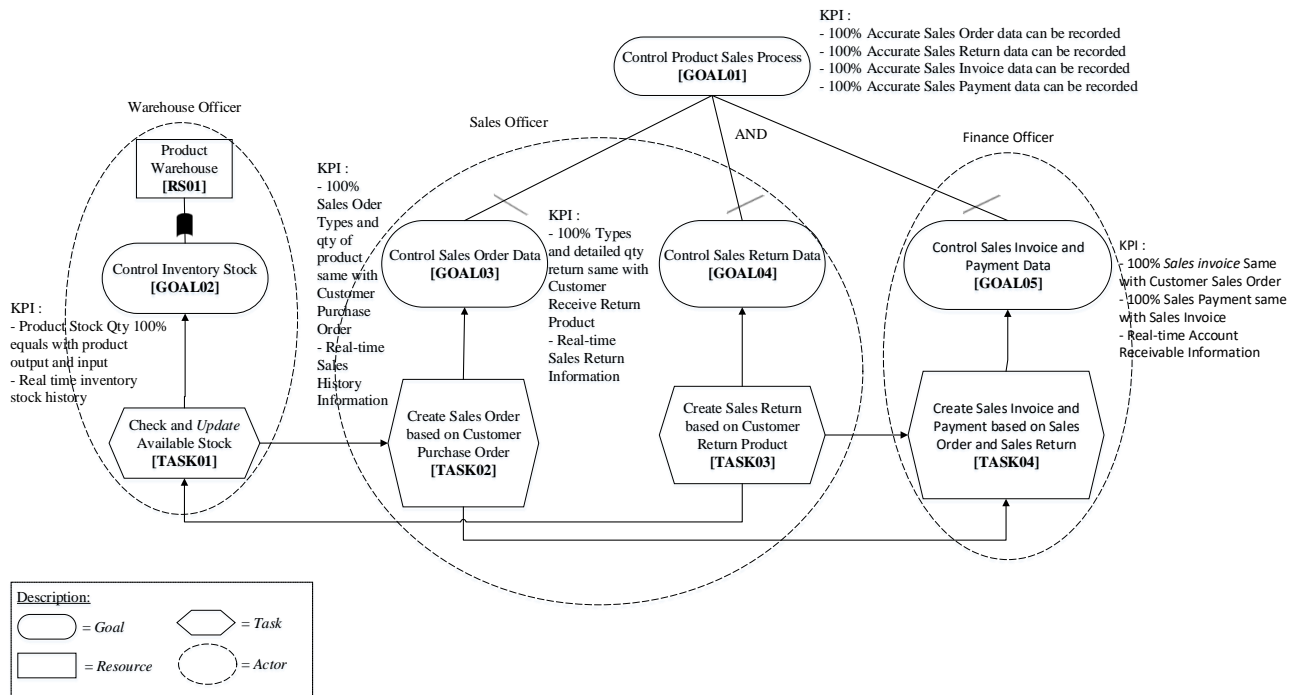


Fig. 2. Result of OGORE Method in form of Solution Goal Tree Model (GTM).

Table 1. GTM Mapping to Use Case Diagram Component

No.	Goals (Requirements)	KPIs	Resource	Task	Use Case	Actor
1.	Control Product Sales Process [GOAL01]	<ul style="list-style-type: none"> - 100% Accurate Sales Order data can be recorded - 100% Accurate Sales Return data can be recorded - 100% Accurate Sales Invoice data can be recorded - 100% Accurate Sales Payment data can be recorded 	-	(Task 2-4 derived from this Goal)	-	-
2.	Control Inventory Stock [GOAL02]	<ul style="list-style-type: none"> - Product Stock Qty 100% equals with product output and input - Real time inventory stock history 	Products Warehouse [RS01]	Check and Update Available Stock [TASK01]	<ul style="list-style-type: none"> - Create INV [USE01] - Create INV out [USE02] - Create INV in [USE03] - View Inventory Movement Report [USE04] 	<ul style="list-style-type: none"> - Warehouse Officer [ACT01] - Sales Officer [ACT02]
3.	Control Sales Order Data [GOAL03]	<ul style="list-style-type: none"> - 100% Sales Oder Types and qty of product same with Customer Purchase Order - Real-time Sales History Information 	-	Create Sales Order based on Customer Purchase Order [TASK02]	<ul style="list-style-type: none"> - Create SOR [USE05] - Create SO [USE06] - View Sales Report [USE07] 	Sales Officer [ACT02]
4.	Control Sales Return Data [GOAL04]	<ul style="list-style-type: none"> - 100% Types and detailed qty return same with Customer Receive Return Product - Real-time Sales Return Information 	-	Create Sales Return based on Customer Return Product [TASK03]	<ul style="list-style-type: none"> - Create SO Return [USE08] 	Sales Officer [ACT02]
5	Control Sales Invoice and Payment Data [GOAL05]	<ul style="list-style-type: none"> - 100% Sales invoice Same with Customer Sales Order - 100% Sales Payment same with Sales Invoice - Real-time Account Receivable Information 	-	Create Sales Invoice and Payment based on Sales Order and Sales Return [TASK04]	<ul style="list-style-type: none"> - Create Invoice [USE09] - Create Payment [USE10] - View Transaction Movement Report [USE11] 	Finance Officer [ACT03]

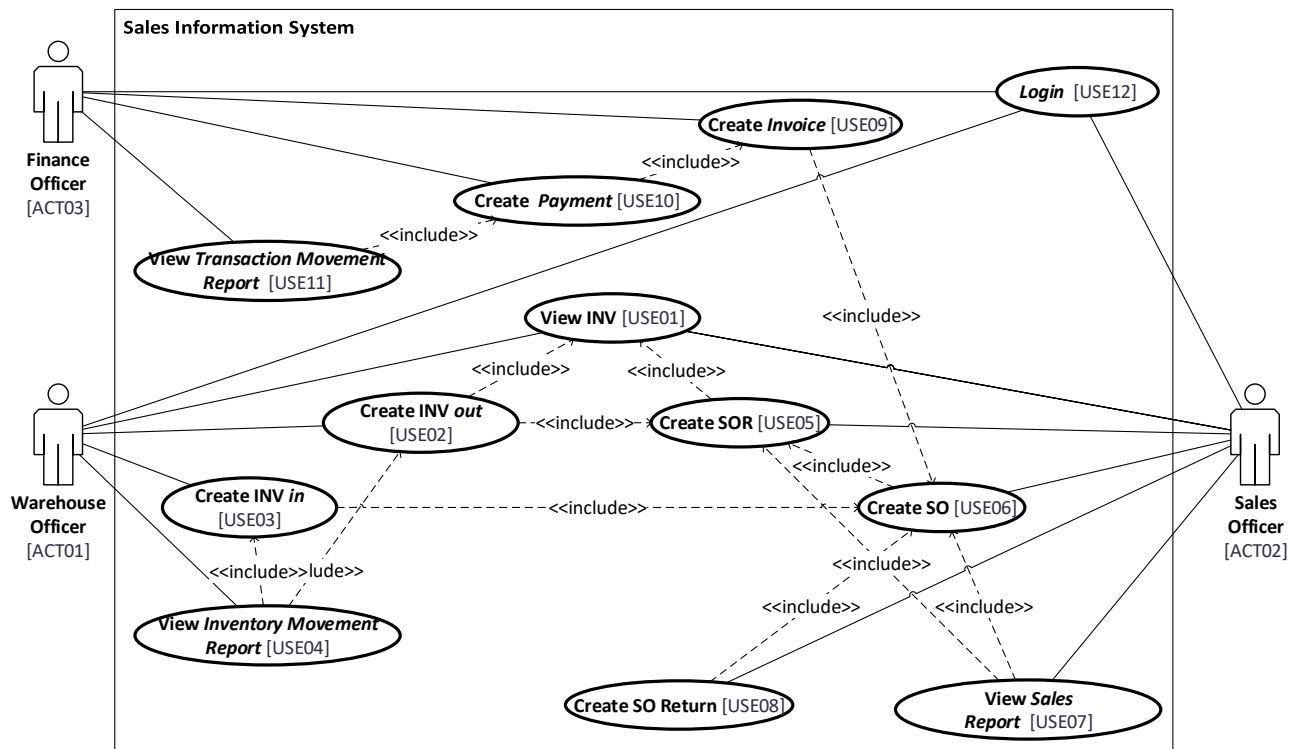


Fig. 3. Sales Information System Use-Case Diagram.

Table 2 Use Case Diagram Mapping Description

No.	Use Case	Actor	Description
1.	Login [USE12]	Sales Officer, Warehouse Officer, Finance Officer	Actor can access system with Login activity
2.	View INV [USE01]	Sales Officer, Warehouse Officer	Actor can view inventory list availability
3.	Create INV out [USE02]	Warehouse Officer	Actor can create product list that will be out from warehouse based on Sales Order Requisition (SOR)
4.	Create INV in [USE03]	Warehouse Officer	Actor can receive product with maximal quantity based on confirmed SO Return quantity
5.	View Inventory Movement Report [USE04]	Warehouse Officer	Actor can view Inventory Movement Report
6.	Create SOR [USE05]	Sales Officer	Actor can create Sales Order Requisition (SOR)
7.	Create SO [USE06]	Sales Officer	Actor can proceed SOR that status has been completed become Sales Order (SO)
8.	View Sales Report [USE07]	Sales Officer	Actor can view Sales Report
9.	Create SO Return [USE08]	Sales Officer	Actor can confirm return products from customer and create Sales Order Return
10.	Create Invoice [USE09]	Finance Officer	Actor can make Invoice
11.	Create Payment [USE10]	Finance Officer	Actor can make Payment
12.	View Transaction Movement Report [USE11]	Finance Officer	Actor can view Transaction Movement Report

C. Coding Process

The coding process was determining the Programming Language Standard. The coding phase in this research used Delphi application development programming language and SQL server for database management.

D. Testing Process

Test unit was determined to identify the achievement of system closure based on KPIs in requirements engineering process. The following are the points executed in the testing:

- 1) Sales Officer:
 - a) Login trial;
 - b) View Inventory trial;
 - c) Create SOR trial;
- 2) Warehouse Officer:
 - a) Login trial;
 - b) View Inventory trial;
 - c) Create INV out trial;
 - d) Create INV in trial;
 - e) View Inventory Movement Report;
- 3) Finance Officer:
 - a) Login trial;
 - b) Create Invoice trial;
 - c) Create Payment trial;
 - d) View Transaction Movement Report;

- d) Create SO trial;
- e) Create SO Return trial;
- f) View Sales Report trial;

V. RESULT AND DISCUSSION

As a result, all system functions can be execute and display in the form of the following reports: Sales Report containing the number sales order and sales return from customers, Transaction Movement Report that containing the summary of customers invoice and payment process, and Inventory Movement Report that describe detailed inventory transaction input and output. Example of the system is shown in Figure 4.

Fig. 4. Create SO Form by Sales Officer.

All system function in testing unit can be running and generating the reports. Based on the reports generated by the system, we can conclude that KPIs and expected functions have been fulfilled by the system, in line with the organization goal. The system is able to ensure that 100% Accurate Sales Order data can be recorded, 100% Accurate Sales Return data can be recorded, 100% Accurate Sales Invoice data can be recorded, and 100% Accurate Sales Payment data can be recorded. By achieving this, the new information system can support the achievement of the second organization goal, which is to increase sales products control.

Users who are used to the process of simple data changes using the old software would need to adapt to the new system, because for each product related transaction automatically calculate inventory movement, invoice, and payment calculation. In product sales return process, aside from being logged, all changes also alter the number of product sold and actual number of products return in real time. Therefore, there shouldn't be any foul play between sales officer and warehouse officer to accept products without any history for that particular procedure.

Because these system functions are derived from the organizational needs, issues related to user needs and issues at risk of serving personal interests in the operations can be eliminated and not processed as long as the system functions are able to meet their own needs. An example of bad user habits in old software is wrong price input in sales order. Commonly the sales officer still can change product price although the SO already become Invoice. The effect of this bad habit, the SO total will not same with Invoice total, the report will be unbalance and there will be chance foul plays between sales officer and finance officer. In the new software requirements process, the system function will strictly forbidden for sales officer change the product price after the SO have been become invoice.

VI. CONCLUSION

In this paper, we have demonstrated how the system requirements can be elicited from organization goals that were extracted into information system functions. We also showed in this paper how to define and transform GTM as OGOR process result into use case to develop information system. The developers can more easily design the system because the focus of the requirements is now more on the organizational goals, and the requirements have been refined from the best practice solution in similar business domains. The information system design process can minimize custom processes based on user requests if we can fulfill the organization needs with this OGOR approach.

With this requirements engineering approach and the information system development process, the company can increase the product sales control. The company will be able to manage the sales return transaction and reduce the risk by eliminating all improper sales return procedure like changing product return quantity.

This sales information system can still be further developed. Some suggestions that can be done for system improvement are: using leveling authorization to change product price, improvement in the system for direct linkage to bookkeeping, and lastly, requirement engineering process that takes into account the non-functional requirements such as calculation of system capability when running on a separate network.

REFERENCES

- [1] G. D. Bhatt and V. Grover, "Types of Information Technology Capabilities and Their Role in Competitive Advantage: An Empirical Study," *J. Manag. Inf. Syst.*, vol. 22, no. 2, pp. 253–277, 2005.
- [2] B. Ives and G. P. Learmonth, "The information system as a competitive weapon," *Communications of the ACM*, vol. 27, no. 12, pp. 1193–1201, 1984.
- [3] F. Adikara, B. Sitohang, and B. Hendradjaya, "Goal-Oriented Requirements Engineering: State of the Art And Beyond," in *The 2nd International Conference on Information Technology and Business Application*, 2013.
- [4] E. Letier and A. Van Lamsweerde, "Deriving operational software specifications from system goals," *ACM SIGSOFT Softw. Eng. Notes*, vol. 27, no. 6, p. 119, 2002.
- [5] F. Adikara, B. Hendradjaya, and B. Sitohang, "A New Proposal for The Integration of Key Performance Indicators to Requirements Elicitation Process Originating from Organization Goals," in *International Conference on Data and Software Engineering*, 2014.
- [6] F. Adikara, B. Hendradjaya, and B. Sitohang, "Requirements Refinements and Analysis with Case-Based Reasoning Techniques to Reuse The Requirements," in *The 5th International Conference on Electrical Engineering and Informatics 2015*, 2015, pp. 460–465.
- [7] A. Dennis, B. H. Wixom, and R. M. Roth, *System Analysis and Design: 5th Edition*. 2012.
- [8] M. U. Ahmed, S. Begum, E. Olsson, N. Xiong, and P. Funk, "Case-based reasoning for medical and industrial decision support systems," *Stud. Comput. Intell.*, vol. 305, pp. 7–52, 2010.
- [9] H. Kaiya, H. Horai, and M. Saeki, *AGORA: attributed goal-oriented requirements analysis method*. Ieee, 2002, pp. 13–22.
- [10] Alan Dennis; Barbara Haley Wixom; Roberta M. Roth, *Systems Analysis and Design: An Object-Oriented Approach with UML*, 5th Editio. Wiley, 2015.
- [11] A. Van Lamsweerde, *Goal-oriented requirements engineering: from system objectives to UML models to precise software specifications*, vol. 6. IEEE, 2003, pp. 744–745.
- [12] L. Lindstrom and R. Jeffries, "Extreme Programming and Agile Software Development Methodologies," *Inf. Syst. Manag.*, vol. 21, no. 3, pp. 41–52, 2004.