A Standard Approach To Find Out Multiple View Points To Describe An Architecture Of Social Systems - Designing Better Payment Architecture To Solve Claim-Payment Failures Of Japan's Insurance Companies -

Seiko Shirasaka
Keio University, Graduate School of System Design and Management
4-1-1 Hiyoshi, Kohoku-ku, Yokohama 223-8526, Japan
shirasaka@z3.keio.jp

Abstract. According to the definition of "System" in various standards and handbooks, a system does not necessarily have hardware-software components. This means that a social system without hardware-software elements is a kind of system. However, systems engineering techniques are not directly applied to define a solution of a non-technical social system's problems. In this paper, we propose a standard approach to find out multiple view points to describe architecture, because the view points are one of the most important aspects of architectural design activities. And also we show one example that we applied our proposed approach to non-technical social system: insurance system. It proved to be positive in applying architectural design technique with appropriate view points to design social system architecture.

Introduction

According to INCOSE Systems Engineering Handbook, the definition of system is:
"An interacting combination of elements to accomplish a defined objective. These include hardware, software, firmware, people, information, techniques, facilities, services, and other support elements."
This means that a system does not necessarily include technical elements (hardware, software, firmware and etc.). Many social systems such as currency system, political system, financial system does not include technical elements. However, there is few examples that systems engineering techniques are directly applied to social system design.
This paper will develop a standard approach to find out multiple view points to describe social system architecture and apply architectural design methodology with the multiple view points to solve a failure of social system. And we will verify our proposed approach by designing solution architecture for the claim-payment failure of Japan's insurance company.

The Current Multiple View Points for Architectural Design

Standard. Architectural design is one of the key activities in systems engineering. System architecture is usually described in several view points because a description from one view point is not enough to describe it correctly. Some of the systems engineering standards such as IEEE 1220 and ANSI/EIA632 specify the view points. For example IEEE1220 specifies three view

points to describe system architecture. Those are operational view, functional view and physical view. ANSI/EIA632 specifies two view points: logical view and physical view. We can just follow these standards as far as they are appropriate to be applied.

Architecture Framework. There are several architecture frameworks and they also specify multiple view points to describe system architecture. One of the most famous frameworks is DoD architecture framework (DoDAF). DoDAF specifies three view points: operational view, system view and technical view. These views are specific to technical systems. That means that DoDAF can be used only for technical system. Federal enterprise architecture framework (FEAF) specifies four view points: business view, data view, application view and technology view. These views are clearly specific to information systems. That means that FEAF can be used only for information system. The Zachman's framework specifies six views: scope view, business view, system view, technology view, detailed representation view and functioning enterprise view. These views are not specific to information systems but to technical systems. There are other frameworks such as The Open Group Architecture Framework (TOGAF) and Ministry of Defence Architecture Framework (MODAF). Their view points are also limited to technical systems architecture description. These architecture frameworks are widely used for architectural design of technical systems.

Our Approach To Find Out View Points

Current View Points. The view points which are used in systems engineering standards and architecture frameworks are not appropriate to non-technical system. Some of social systems are non-technical system. This means that we have to find out appropriate view points for the description of non-technical social system architecture. However, it is not easy to find out appropriate view points. We study the current view point specified in the standards and the frameworks. And we found out that many of them have similar structure of view points. That is "Layered" structure. The three view points in IEEE1220 have layered structure. Its top view is the operational view. The operations are supported and realised by the functions. And the functions are supported and realised by the physical things. Also the operations utilise the functions. And the functions utilise the physical things. (Fig.1) The three view points in DoDAF have layered structure, too. The operations are supported and realised by the systems. And the systems are supported and realised by the technology. The four view points in FEAF are same.

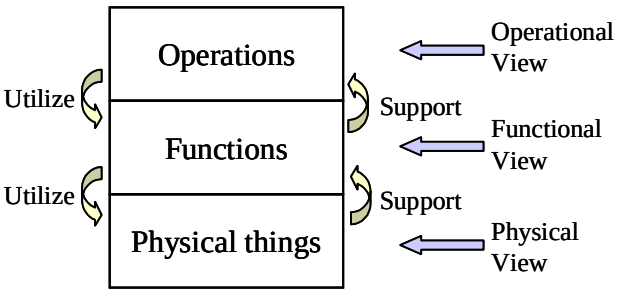

Figure 1. Relation of Multiple Views in IEEE1220

However, some of the frameworks have non-layered view points. The Zachman's frameworks have the scope view. The scope is not supported and realized by business. The scope is a kind of an attribute of business which is being dealing with. That means that the relation between the scope and the business is not the vertical relation but the horizontal relation. (Fig.2)

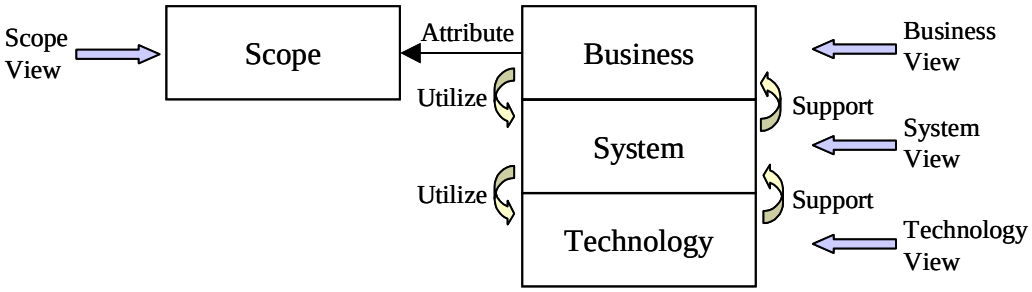

Figure 2. Relation of Multiple Views in the Zachman's Framework

Our Approach. Our proposed approach to find out multiple view points is mainly to follow the layered concept. First put the goal as top view. And then identify "What supports and realizes the above view?" By repeating it, we can develop the layered multiple views as the key vertical view points. And if some supplemental views are required, they can be put as horizontal view points. (Figure 3)

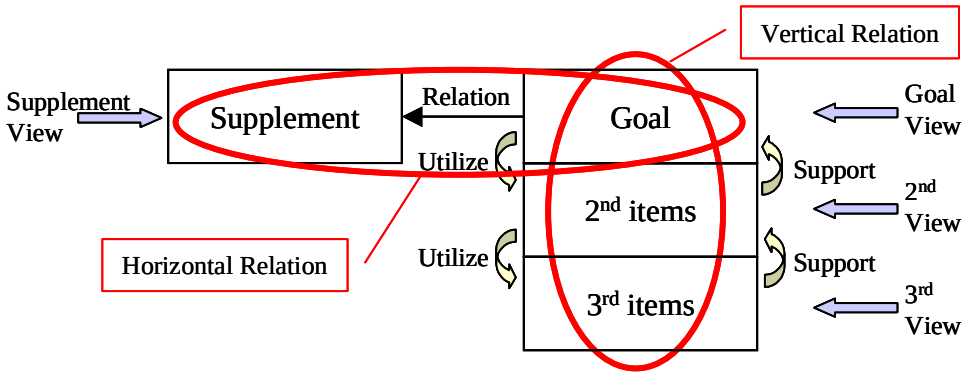

Figure 3. Vertical/Horizontal Structures of Multiple Views

One view can be supported and realised by multiple views (like a pyramid). (Figure 4) And one view can support and realize multiple views (like a reverse pyramid), too. (Figure 5)

By utilizing vertical and horizontal view development concept, we can find out multiple view points which are required to perform system architectural design.

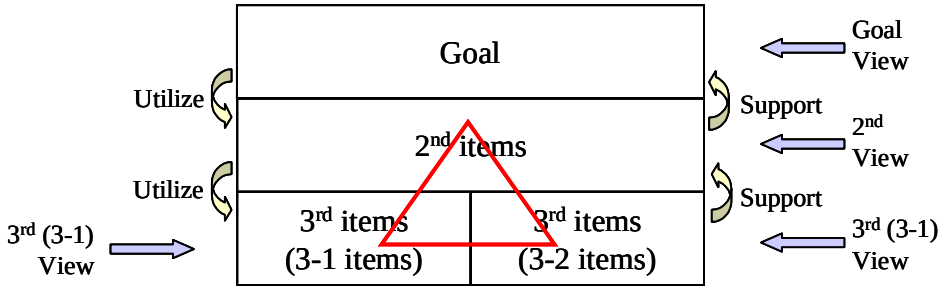

Figure 4. Pyramid Type Structures of Multiple Views

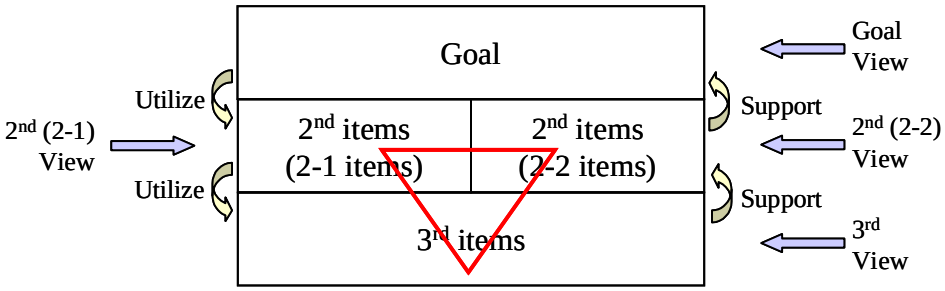

Figure 5. Reverse Pyramid Type Structures of Multiple Views.

Architectural Design To Solve Claim-Payment Failures Of Japan's Insurance Companies

The Problem. Between 2005-2008, most Japanese life insurance companies were found to have failed to pay proper insurance claims to policyholders. Such failures are judged as a clear violation of the insurance business law of Japan and show serious deficiencies of internal control and governance in insurance companies. The financial authorities of Japan implemented full scale inspections and took administrative action in those cases. According to Japan's financial services supervision authorities' successive findings, such non-payments or payment leakages reached to JPY 144.3 billion (roughly $ 1.4 billion). This problem became a big social issue in Japan. More detail explanation is described in the paper, "Claim-Payment Failures Of Japan's Insurance Companies And Designing Better Payment Architecture: Finding A Standard Solution To Socio-Critical Systems By Applying the System Engineering Vee Model Approach (First Report)" presented in INOCSE symposium 2009.

View Points. We applied our layered view point approach to design the architecture to solve the problem. The top view is a "goal view". The goal is that an insurance company correctly pays proper insurance claims to policyholders. This goal is supported and realised by activities which are conducted by several stakeholders. We select "Activity View" as the second level view points. The activities are conducted by a person who might have a role to do it in an organization. What makes him or her to do it? A compelling force may make a person to conduct it. Usually rules are used as a driver of compelling force. However, he or she might try to find out the way in which he or she escapes from it if he or she doesn't want to do it. It means that the compelling force is not sufficient to support and realise the activities. The system has to motivate the person to do it on their own initiative. The activities are supported and realised by compelling force and motivation. We call them as "Activity Promotion View". The activity promotion view is consist of rule view and motivation views. Even if the person is forced and motivated to conduct the activities, he or she could make a mistake. To detect the unintentional mistakes, the system should have a mechanism to confirm that the activities are really finished or not. We name it as "Activity Confirmation View." We may need an infrastructure to support and realise the efficient execution of the activities and the activity confirmation. We call it as "Activity and Activity Confirmation Support View." (Figure 6)

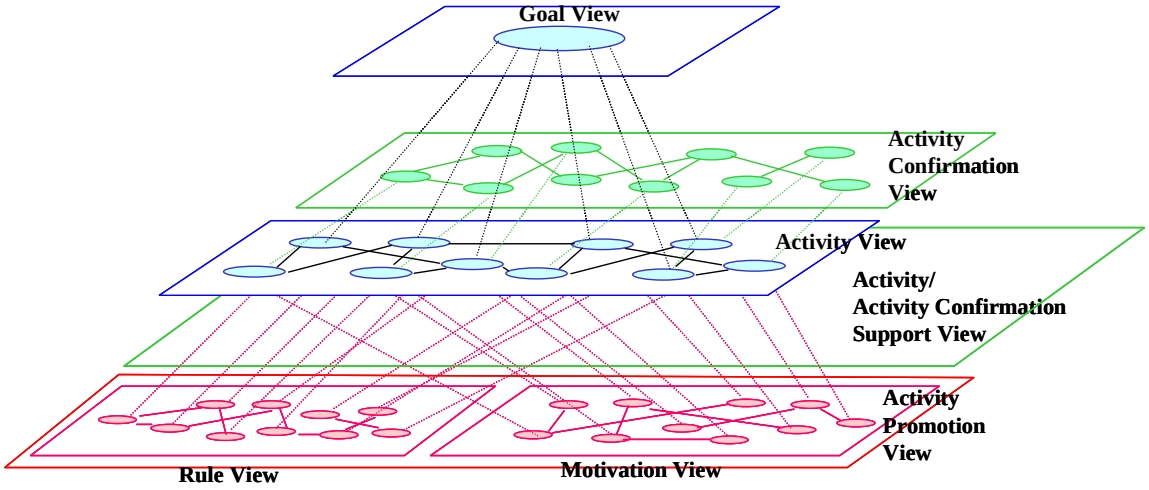

Figure 6. View Structure of the Solution of Japan's Claim Payment Failure

The activity view has a vertical relation with the goal view. The activity view, rule view and the motivation view make a pyramid type structure. They have a pyramid type vertical relation. The activity confirmation view is a supplemental view of the activity view. They have a horizontal relation. The activity view, the activity confirmation view and the activity/activity confirmation support view have a reverse pyramid type structure. They have a reverse pyramid type vertical relation.

Goal View. The goal view describes the goal state which solves the problem. It means that the policyholders correctly receive claim-payment.

Activity View. The activity view describes the structure of the activities that realise the goal. We applied our layered view point approach to design the architecture to solve the problem. The activities are divided into the sub-activities and the sequences of the activities are identified. And the person who conducts a sub-activity is allocated. These steps are repeated until one sub-activity is conducted by one person. These steps are very similar to functional design and functional allocation to a subsystem. Figure 7 shows the structure of sub-activities which are divided from the goal. Figure 8 shows the sequences of sub-activities and their allocation. Figure 9 shows the allocated results.

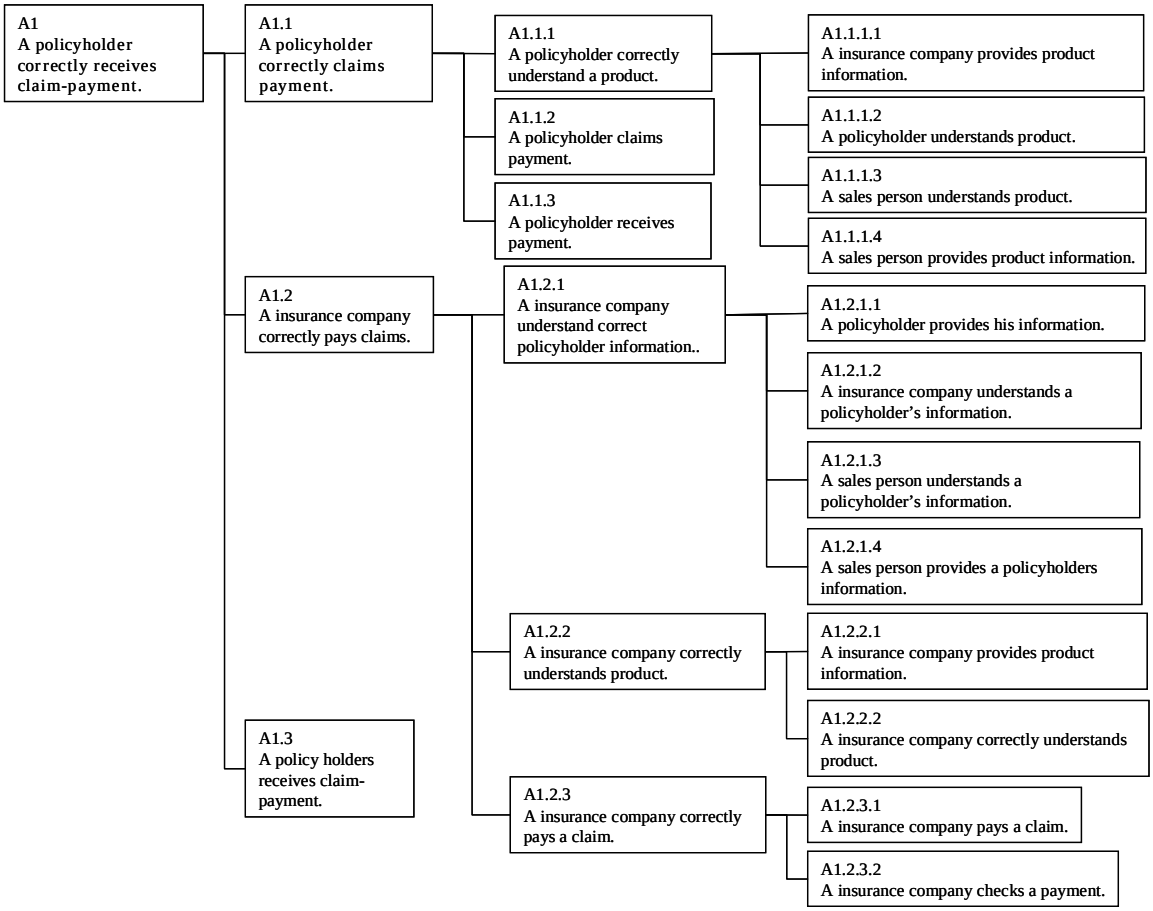

Figure 7. Structure Of Sub-activities

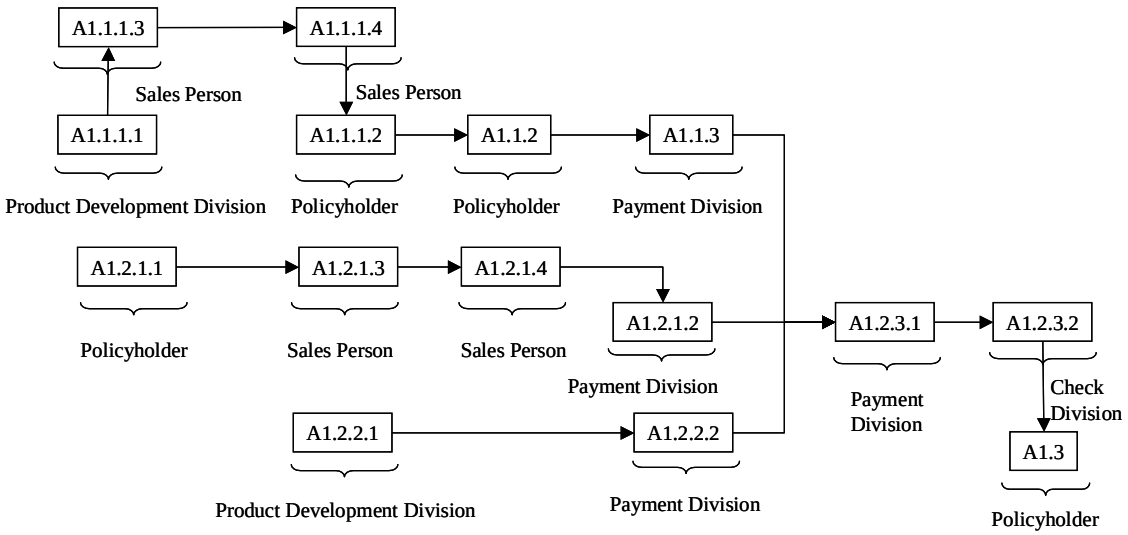

Figure 8. Sequences Of Sub-activities and The Allocation

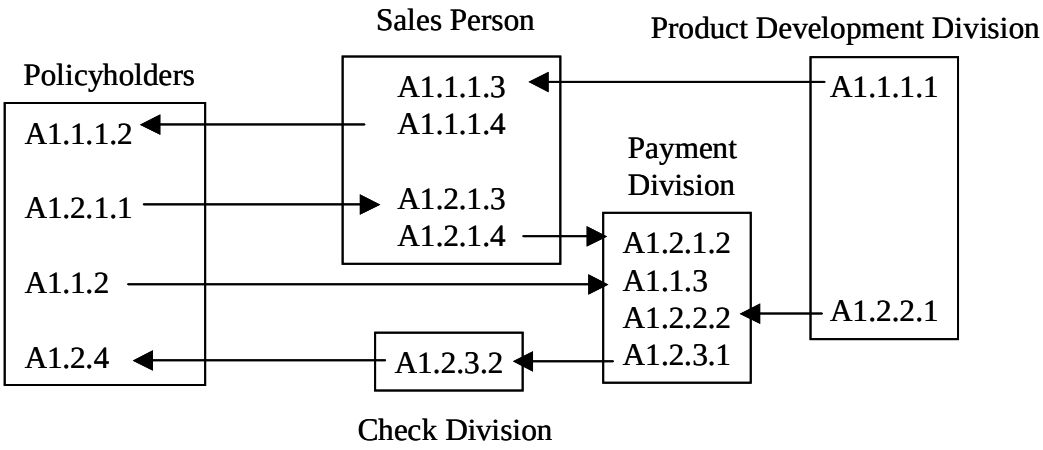

Figure 9. The allocated Sub-Activities

Rule View. The rule view is a part of the activity promotion view. The activity promotion view is the architecture that makes a person to conduct an activity. The rule forces a person to conduct an activity even if he or she doesn't want to do it. At the moment, we categorize rules into two parts. One is internal rules. The other is external rules. The rules in an insurance company are categorized as the internal rules. And the rules related to policyholders are categorized as the external rules. (Figure 10) Currently we identified where we need rules. In near future, we would like to define what kind of a rule is appropriate at each identified places. Some places require very strict rules like a law or a rule with penalty. Others may require just weak rules like guideline.

In the real world, one of the improvement orders by the FSA is to make the written document that includes contents of contract, risk information and confirmation of the agreement. This rule is same as "External-1" rule in Fig.10.

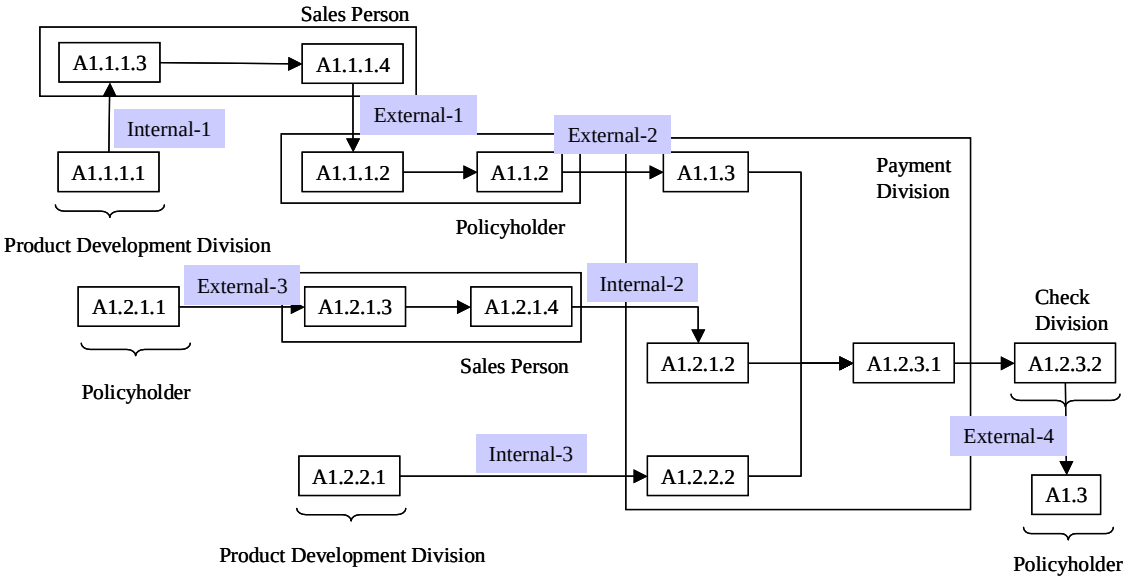

Figure 10. The Rule View

Motivation View. The rule view is a part of the activity promotion view. Even if the person is forced to conduct the activities, he or she might find a way not to do it as far as he or she is not

motivated. The motivation view shows the architecture that the person is motivated to conduct the activities. All activities which are identified at the activity view require something to motivate the person. The places where we need motivation are identified. (Figure 11)

In the real world, one of the improvement orders by the FSA is to simplify the insurance product by the reduction of options. This is a kind of “Motivation 1,2, 11 and 12” because people can explain the insurance products more easily.

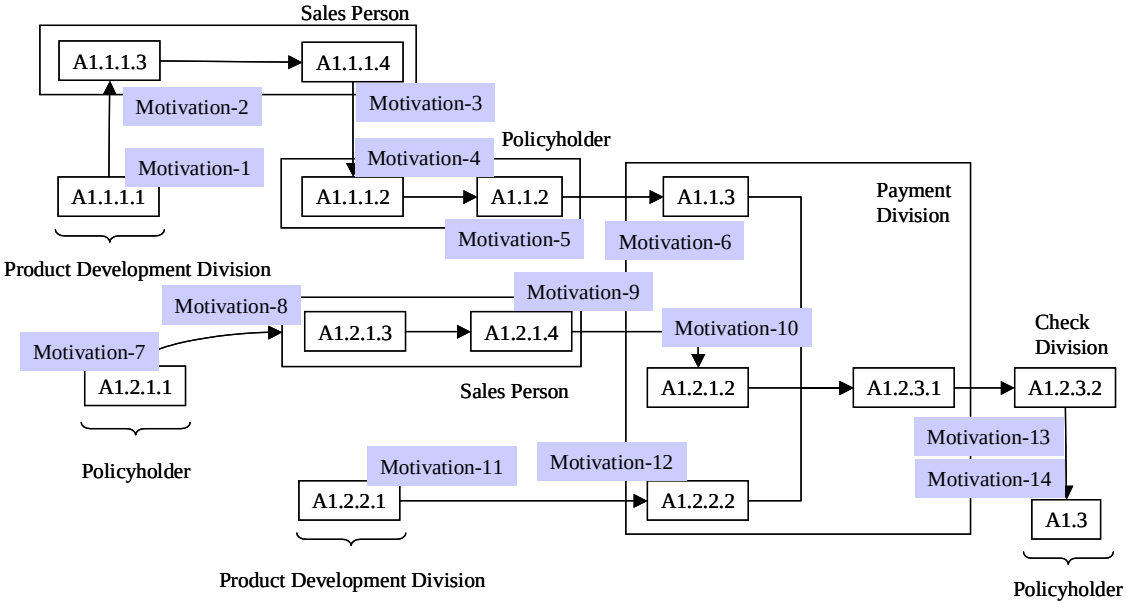

Figure 11. The Motivation View

Activity Confirmation View. Even if a person is motivated and he or she willing to conduct activities, he or she may make a mistake. To cover such kind of unintentional failure, we add the activity confirmation view. This view shows how the failure is covered. All activities have responses not only to the person who conduct the activity but also to a process inspector who checks whether a process is conducted or not. (Figure 12)

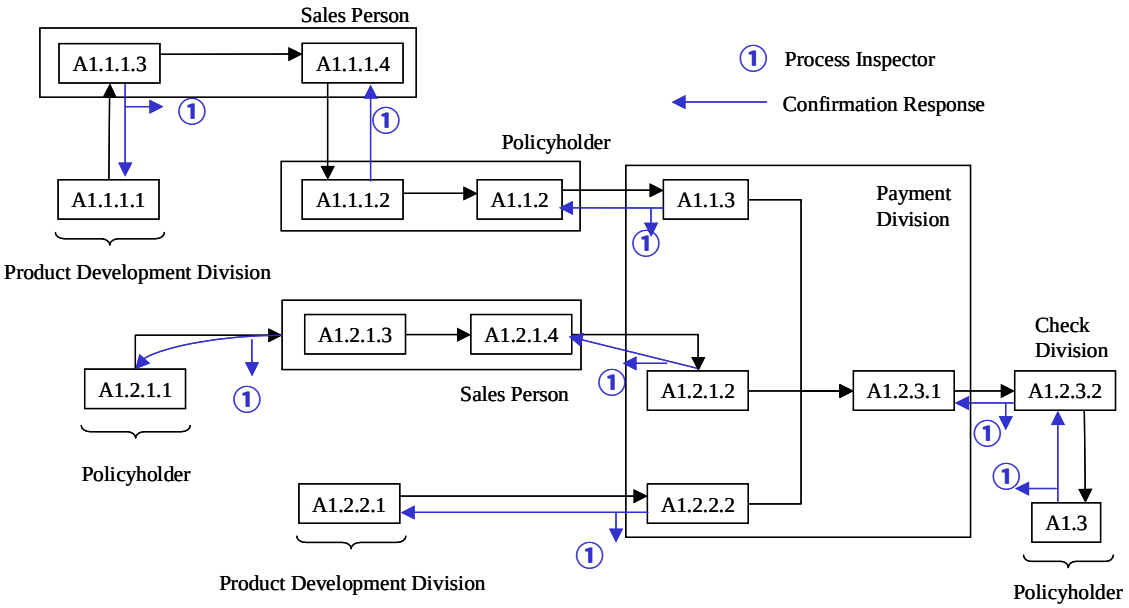

Figure 12. The Activity Confirmation View

Activity/Activity Confirmation Support View. The activity/activity confirmation support view shows the architecture of the infrastructure that supports the activity and the activity confirmation are correctly and efficiently conducted. We define two sub views in this view. One is data view. The other one is infrastructure view. If we start the development of this infrastructure, we may use the framework such as FEAF, the Zachman Framework.

Data View. The data view is a part of the activity/activity confirmation support view. It shows what kind of data is exchanged. (Figure 13)

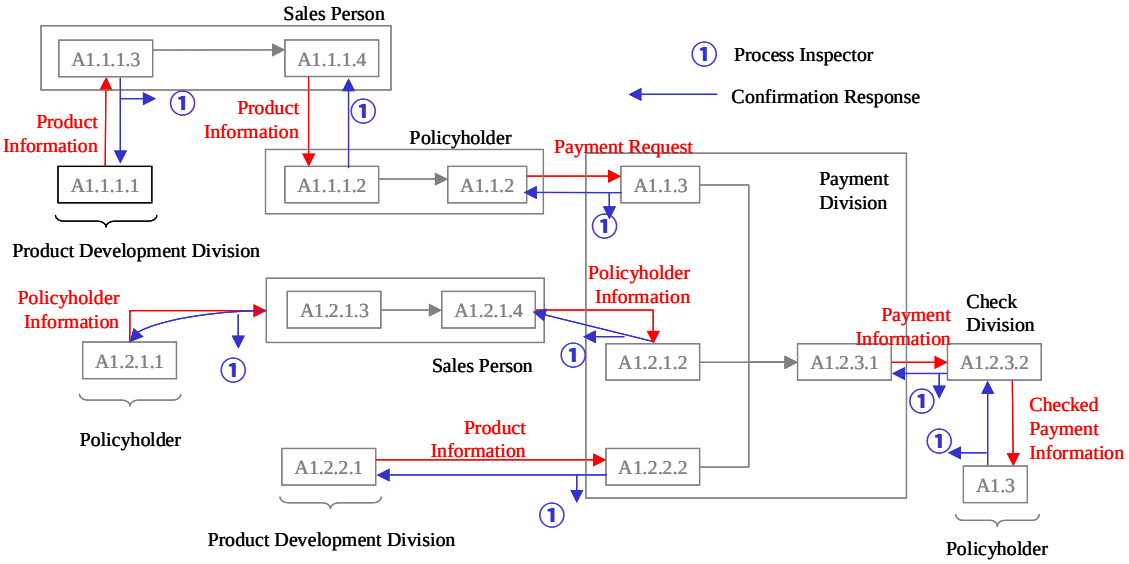

Figure 13. Data Views

Infrastructure View. The infrastructure view is a part of the activity/activity confirmation support view, too. It shows the infrastructure that can support the data exchange to conduct the

activities and the activity confirmation. (Figure 14)

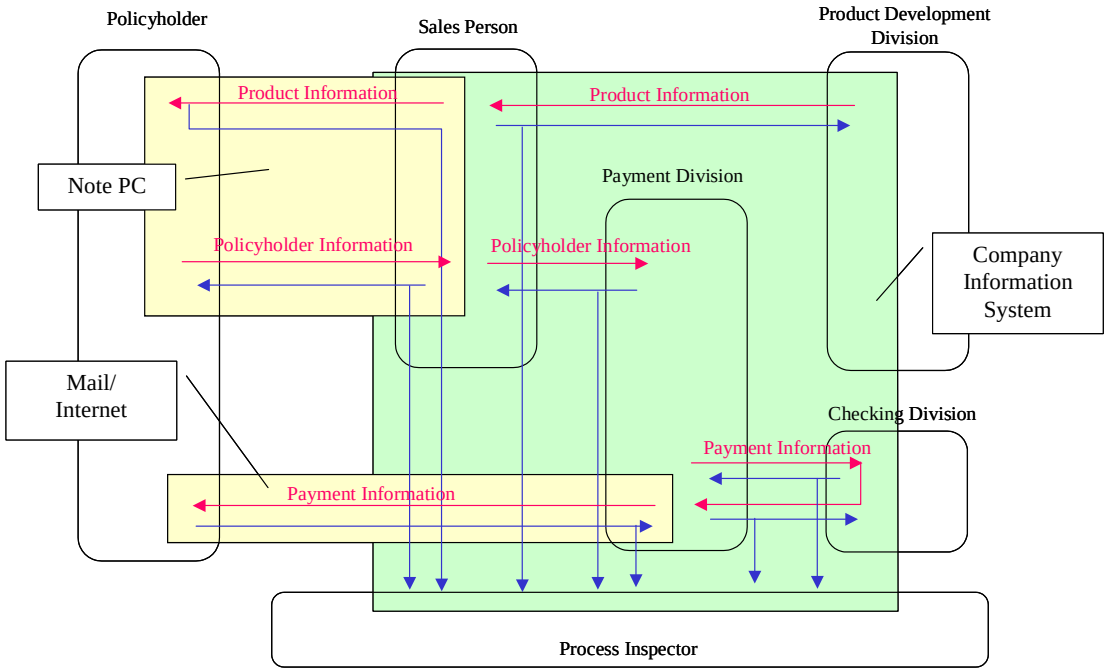

Figure 14. The Infrastructure View

Difference between Our Architectural Design And Administrative Response To The Claim-Payment Failures Of Japan's Insurance Companies

Many of the real administrative responses are identified in our architectural design. This means that FSA's responses can be designed by our architectural design. However, some of our design is not covered by FSA's response. They may have existed without any improvement. We need more detail analysis on these parts. And also some of FSA's response is not covered by our architectural design. Especially the improvements of the governance of insurance companies are not covered. This means that we might lack a view to cover it. Even if we have some discrepancy between our architectural design and administrative response, our proposed approach are still effective.

Conclusions

We propose a standard approach to find out multiple view points to describe architecture. And we apply our proposed approach to a non-technical system architectural design. We show our architectural design to solve the claim-payment failures of Japan's insurance companies. And the comparison of our results and the real administrative response shows that our approach can lead effective results even if we need some detail analysis to evaluate our design precisely.

References

INCOSE 2006 systems engineering handbook - a guide for system life cycle processes and activities, Version 3. ed. Cecilia Haskins.

Yasui, T, 2009 Claim-Payment Failures Of Japan's Insurance Companies And Designing Better Payment Architecture: Finding A Standard Solution To Socio-Critical Systems By Applying the System Engineering Vee Model Approach (First Report), INCOSE Symposium 2009

U.S. Department of Defense. 2007. DoD Architecture Framework v.1.5

U.S. Department of Defense. 1997. Federal Enterprise Architecture Framework v.1.1

Rechtin, E. and Maier, M. W.1996. The Art of Systems Architecting. CRC Press.

Biography

Seiko Shirasaka is Visiting Associate Professor of Graduate School of System Design Management, Keio University. In 1994 he earned a Master's degree in Astronautics from University of Tokyo and immediately joined in Mitsubishi Electric Corporation. Since then he worked for several space system development projects as a systems engineer.