A PATTERN LANGUAGE FOR HANDOVERS

KEI ITO, Waseda University
HIRONORI WASHIZAKI, Global Software Engineering Laboratory, Waseda University
JOSEPH W.YODER, The Refactory, Inc.,
YOSHIAKI FUKAZAWA, Global Software Engineering Laboratory, Waseda University

In 2007 issues with handovers became apparent in Japan, when many people from the Baby Boomer Generation retired simultaneously. Most business people are familiar with the concept of a handover. Although effective handovers are crucial for seamless business operations during personnel changes, the preferable elements for a handover are ambiguous and not well researched. This motivated us to examine this topic. We held a workshop on handovers to assess actual problems due to undesirable handovers. The results were used to elucidate three handover anti-patterns, which identify concrete problems of handovers. To mitigate the problems, we found handover solution patterns by analyzing three anti-patterns. In this paper, we introduce three solution patterns: *Spread of knowledge*, *Handover in a different room* and *Firewall for the handover*. The predecessor prepares to handover by *Spreading the knowledg*e about the system to the third parties in the preparation phase. This lets the predecessor ensure the important knowledge to be remained in their organization. The predecessor and the successor *move to a different room* during their handover. This helps to the predecessor and the successor concentrating on the handover. *Firewall for handover* is used by the third party to protect the predecessor from the job responsibilities which still need to be performed during the handover. The third party stands in for the predecessor to support the handover. Herein, we outline the handover solution patterns more detail.

Categories and Subject Descriptors: • **Social and professional topics~Project and people management** • **Software and its engineering~Maintaining software** • **Software and its engineering~Software verification and validation**

General Terms: Pattern Language

Additional Key Words and Phrases: Handover, Pattern Language, People Management, Maintaining software

ACM Reference Format:

Ito, K., Washizaki, H, Yoder, J.W., and Fukazawa, Y. 2016. Pattern Language for Handovers. HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. 23 (October 2016), 12 pages.

1. INTRODUCTION

The life cycle of large software systems inevitably includes personnel changes. A key factor in seamless personnel changes is to successfully transition (handover) responsibilities as people leave the company or transition to another area within the company. In fact, the handover of software is one of the most critical stages in the software life cycle [V90]. Despite its importance, handover problems are not well studied [AM10a] and many undesirable handovers occur in business daily [KHY16]. One study mentioned that insufficient knowledge of the successor is one of the main problems in a handover [AM10a], while another found that the information sharing process contains complex problems [GE08]. Although concrete solutions for handover problems, would be invaluable to businesses, such solutions have yet to be investigated.

Previously, we identified handover problems using anti-patterns. We presented three handover anti-patterns at Asian PLoP 2016 (*Unsupported to review*, *Background is unclear*, and *Necessary knowledge is omitted*) [KHY16], which were confirmed by questionnaires sent to businesses on 23 April 2016. By analyzing the anti-patterns, their origins became clear. We noted that one of the main causes of an undesirable handover is the failure of a predecessor to communicate clearly with his or her successor. From this analysis, we noted many handover solution patterns. Handover anti-patterns and handover solution patterns are related to each other. To better understand handover issues, it is useful to understand both handover anti-patterns and handover solution patterns.

Pattern languages are used to provide desirable solutions based upon known best practices. They help guide a solution given the context and problem domain. Although the concept of pattern languages has primarily been applied to architecture and software fields, it has been applied to various fields such as cooking [YSAAT16], and child development [ATTET16]. Herein it is used to present desirable handover solutions. Our patterns are written as a mixture of the Alexandrian Pattern and Fearless Change Pattern forms [JR16] [MR] [MR15]. The context, problems, solutions, and forces are written with stories. We start out each pattern with a quote, followed by a brief description of the pattern. We then use a scenario to describe the context, problem, forces, and solution. This storytelling form, which is similar to that presented in the Fearless Change patterns can be easily understood. To describe concrete stories, we assume a fictitious company, Waseda Co., Ltd.

These patterns are intended for anyone involved in a handover process. The scenarios herein assume a system department, but we expect that even those without a system's background may find the paper useful since a handover is a common business concept. Concrete scenarios are used to help the reader understand the use and interactions of the patterns. For this paper, the scenario we used to illustrate the patterns is when someone transfers within a company. However, these patterns should also be applicable to other contexts such as retirement and job changes. Also note that there is an assumption that there is time to do the handover with the person transferring within the company such as a few weeks or a few months.

The rest of the paper is organized as follows. Section 2 defines our concept of a handover using an activity diagram, which expresses the actors and each actor's activity. Section 3 introduces concrete handover scenes as an example for a fictitious company as an example to be used in our handover patterns. Section 4 outlines the pattern language for handovers using a pattern map and presents three handover solution patterns in detail. Section 5 introduces a case study related to the pattern language proposed in this paper. Finally, Section 6 concludes this paper and outlines potential future works.

2. CONCEPT OF A HANDOVER

A handover is a process of transforming responsibilities and knowledge from the predecessor to the successor [AM10a] [AM10b]. Because handovers occur frequently within most organizations, they are a persistent problem in business. Some handovers are straightforward and simple, while other become very complex. In general, a handover has three phases (preparation, during handover, and after handover) and up to three actors (predecessor, successor, and preferably a third party). The predecessor, successor, and third party are all employees within the organization. The best scenario occurs when sufficient time is allotted for an incremental handover with a lot of mentoring. In this case, it is possible to extend the handover over a longer period and might only involve the predecessor and the successor.

Oftentimes you have to do the handover in a much shorter amount of time. In this paper we will examine these shorter handover situations. In these situations, the third party becomes an important role. The third party is someone who is concerned and responsible for the handover, for example the manager of the project. For each phase, each actor completes specific tasks.

Handover activities can be decomposed into the following nine steps (see Fig 1)

(1) Predecessor's preparation activity
(2) Successor's preparation activity
(3) Third party's preparation activity
(4) Predecessor's during handover activity
(5) Successor's during handover activity
(6) Third party's during handover activity
(7) Predecessor's after handover activity
(8) Successor's after handover activity
(9) Third party's after handover activity

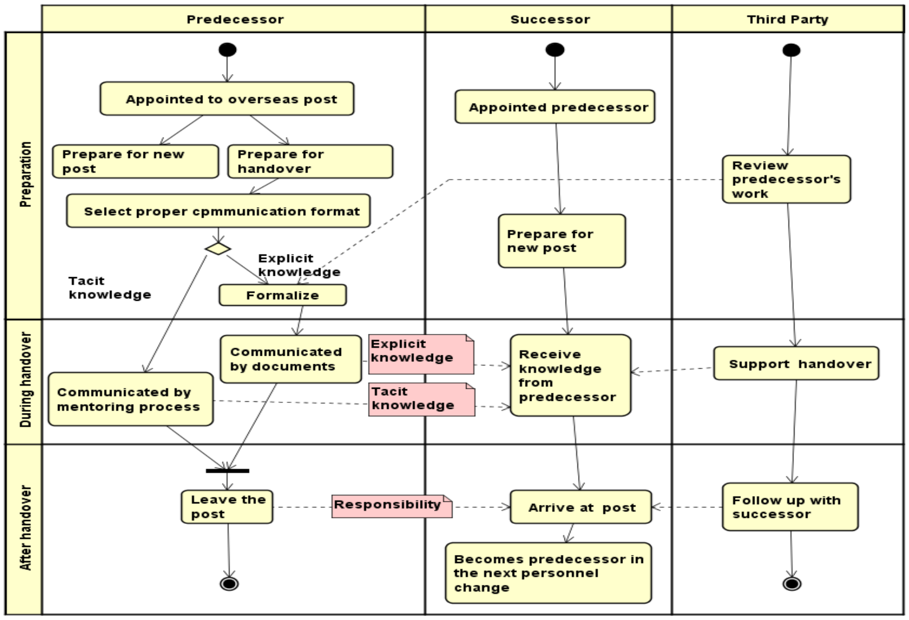


Figure 1. Handover activity diagram

In the preparation phase, the predecessor chooses the knowledge to be shared and formalizes explicit knowledge into the proper form to be communicated. The successor prepares for the new post (e.g., by studying or receiving formal training to understand the knowledge to be communicated by the predecessor). Note that, selected knowledge consists of explicit knowledge and tacit knowledge. The explicit knowledge can be formalized and documented in some formalized fashion.

Tacit knowledge contains formless elements like experiences and ideas, which are best shared through a mentoring process (i.e., pairing up and sharing the knowledge while the successor performs the tasks). Some explicit knowledge can be difficult to formalize because it is complex and requires a lot of time to characterize. In this case, it is best to share this explicit knowledge through a mentoring process. It is important for the predecessor to select an effective form of communication for the handover process.

Ideally a third party reviews any formalized knowledge documented by the predecessor. During the handover period, the exchange of knowledge between the predecessor and the successor occurs. The predecessor communicates the knowledge, which the successor receives and comprehends. The third party assists by supporting the predecessor during the handover of knowledge to the successor. After finishing the handover process, the successor assumes the predecessor’s responsibilities. Since the successor might be unfamiliar with the job, the third party follows up regularly with the successor answering any questions they may have.

3. SCENE

In this section, we outline the concrete scene we used to describe our patterns. Each pattern has a scenario to explain its contexts, forces, problems, and consequences. A concrete scenario helps make the patterns more understandable. In addition, a concrete scenario can be used to evaluate the performance [JRA14].

3.1 Example Scenario

Hiro works at Waseda Co., Ltd., which is one of the largest companies in Japan. Its business focuses on system auditing and consulting. Hiro belongs to the internal system department and is responsible for the development, operation, and maintenance of internal systems. He has been in charge of the financial system for a long time. However, he is being transferred overseas. Kei, a mid-level employee, has been appointed as Hiro’s successor. Before leaving, Hiro has to handover knowledge of the financial system to Kei.

The financial system was developed in Java about ten years ago. It is one of the biggest internal systems in Waseda Co., Ltd. Hiro has been in charge of the system since its inception. The system outputs the settlement of accounts for external auditors and the government for tax purposes. An accountant in charge inputs accounting data. The system must be revised when the tax law and other requirements change. Additionally, scheduled maintenance also causes downtime of the system.

3.2 Characters

In the following examples, the handover consists of the activities of three actors (predecessor, successor, and third party), but there are four characters: predecessor (Hiro), successor (Kei), and the third party role is shared by two people (Joe and Yoshiaki). Each character plays a unique role.

Hiro, who is a veteran worker in charge of the financial system for a long time, is being transferred overseas. Although Hiro is staying within the organization, he is moving to a different department, and possibly in another location. He will not be directly involved with the financial system after the handover, but he can answer questions from the successor. Kei is a mid-level employee, who has been appointed as Hiro’s replacement. She has been an employee of Waseda Co., Ltd for a couple of years, but in a different department with Hiro. Joe is a veteran worker and a colleague of Hiro. He also works for the internal system department but is in charge of a different system. Yoshiaki is a responsible person in the internal system department. He is the project manager of the financial system and Hiro's boss.

4. PATTERN LANGUAGE FOR HANDOVERS

In this section, we introduce pattern language for handovers. The handover solution patterns are used in combination to deal with handover anti-patterns. There are six combinations: substantial handover, reliable handover, agile handover, follow up handover, traceable handover, and stable handover. Each combination deals with specific handover problems. For instance, the substantial handover combination deals with a low quality handover by enhancing communication between the predecessor and the successor. It prevents unexpected omitted knowledge and background information. By using the combination, we can avoid two anti-patterns, *Necessary knowledge is omitted* and *Background is unclear*. Figure 4 shows the structure of each combination and what problem the patterns deals with.

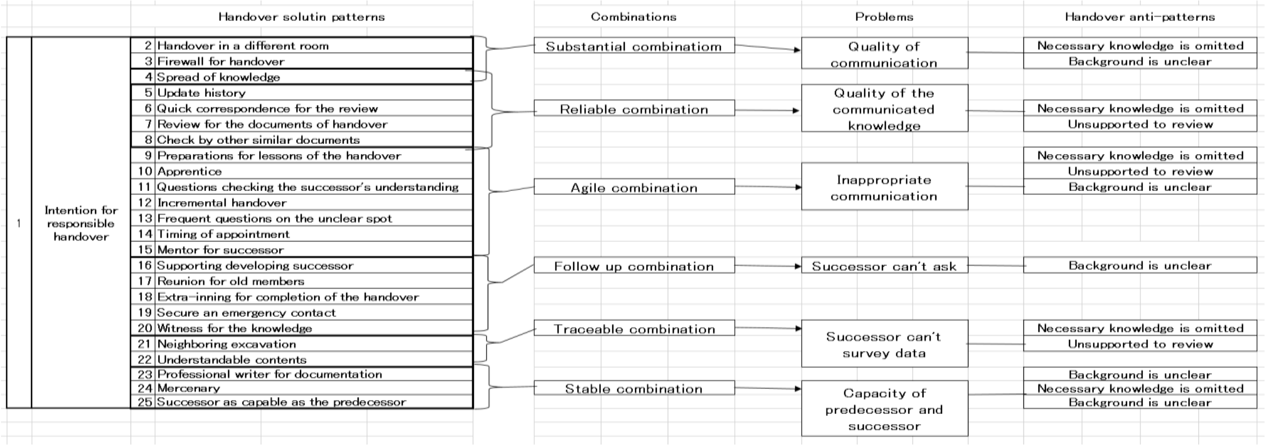
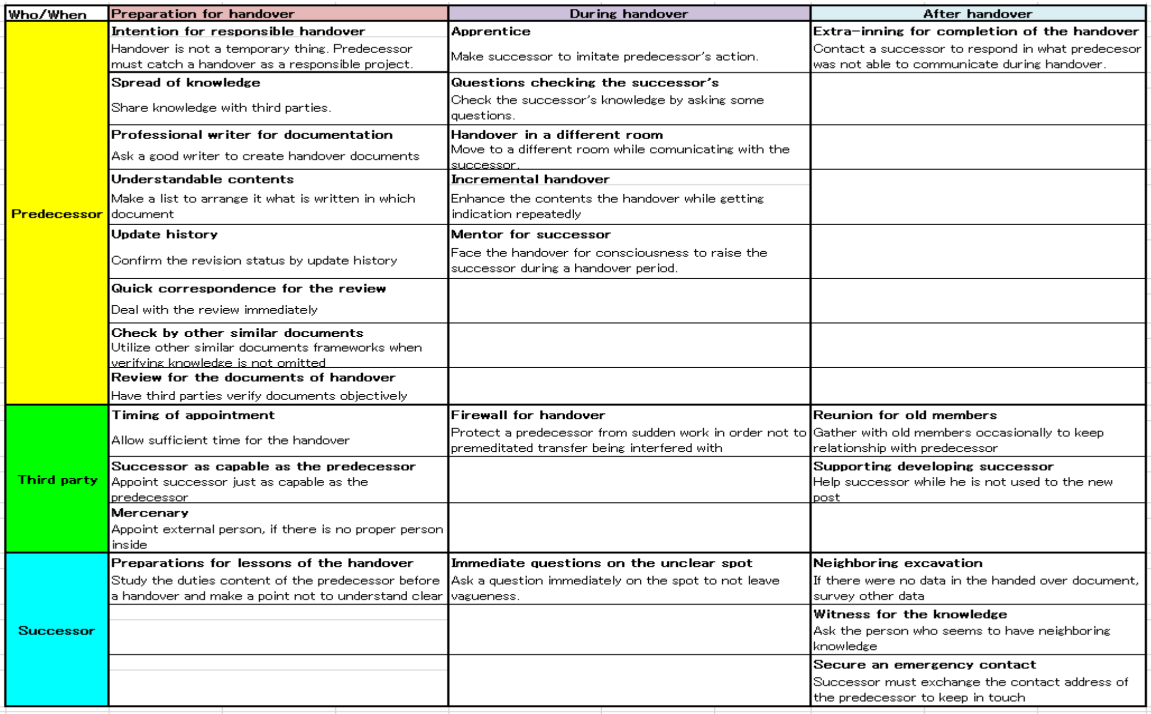

Handover solutin patterns
Combinations
Problems
Handover anti−patterns
1
Intention for responsible handover
2 Handover in a different room
3 Firewall for handover
4 Spread of knowledge
5 Update history
6 Quick correspondence for the review
7 Review for the documents of handover
8 Check by other similar documents
9 Preparations for lessons of the handover
10 Apprentice
11 Questions checking the successor's understanding
12 Incremental handover
13 Frequent questions on the unclear spot
14 Timing of appointment
15 Mentor for successor
16 Supporting developing successor
17 Reunion for old members
18 Extra−inning for completion of the handover
19 Secure an emergency contact
20 Witness for the knowledge
21 Neighboring excavation
22 Understandable contents
23 Professional writer for documentation
24 Mercenary
25 Successor as capable as the predecessor
Substantial combinatiom
Reliable combination
Agile combination
Follow up combination
Traceable combination
Stable combination
Quality of communication
Quality of the communicated knowledge
Inappropriate communication
Successor can't ask
Successor can't survey data
Capacity of predecessor and successor
Necessary knowledge is omitted
Background is unclear
Necessary knowledge is omitted
Unsupported to review
Necessary knowledge is omitted
Unsupported to review
Background is unclear
Background is unclear
Necessary knowledge is omitted
Unsupported to review
Background is unclear
Necessary knowledge is omitted
Background is unclear

Figure 4. Pattern map of A Pattern Language for Handovers

The substantial handover combination enhances the quality of communication between the predecessor and the successor, realizing a quality handover. The reliable handover combination focuses on preventing inaccurate knowledge from being handed over. It enhances the quality of shared knowledge and helps makes it more reliable. The agile handover combination enables the predecessor and the successor to communicate appropriately. The follow up handover combination helps the successor gain support from the predecessor and the third party. The traceable handover combination provides a survey for the successor as it assumes that some knowledge is unintentionally omitted. The stable handover combination aims to ensure a reasonable handover.

Table 1. List of Handover solution patterns



Who/When	Preparation for handover	During handover	After handover
Predecessor	**Intention for responsible handover** Handover is not a temporary thing. Predecessor must catch a handover as a responsible project.	**Apprentice** Make successor to imitate predecessor's action.	**Extra−inning for completion of the handover** Contact a successor to respond in what predecesor was not able to communicate during handover.
	Spread of knowledge Share knowledge with third parties.	**Questions checking the successor's** Check the successor's knowledge by asking some questions.	
	Professional writer for documentation Ask a good writer to create handover documents	**Handover in a different room** Move to a different room while comunicating with the successor.	
	Understandable contents Make a list to arrange it what is written in which document	**Incremental handover** Enhance the contents the handover while getting indication repeatedly	
	Update history Confirm the revision status by update history	**Mentor for successor** Face the handover for consciousness to raise the successor during a handover period.	
	Quick correspondence for the review Deal with the review immediately		
	Check by other similar documents Utilize other similar documents frameworks when verifying knowledge is not omitted		
	Review for the documents of handover Have third parties verify documents objectively		
Third party	**Timing of appointment** Allow sufficient time for the handover	**Firewall for handover** Protect a predecessor from sudden work in order not to premeditated transfer being interfered with	**Reunion for old members** Gather with old members occasionally to keep relationship with predecessor
	Successor as capable as the predecessor Appoint successor just as capable as the predecessor		**Supporting developing successor** Help successor while he is not used to the new post
	Mercenary Appoint external person, if there is no proper person inside		
Successor	**Preparations for lessons of the handover** Study the duties content of the predecessor before a handover and make a point not to understand clear	**Immediate questions on the unclear spot** Ask a question immediately on the spot to not leave vagueness.	**Neighboring excavation** If there were no data in the handed over document, survey other data
			Witness for the knowledge Ask the person who seems to have neighboring knowledge
			Secure an emergency contact Successor must exchange the contact address of the predecessor to keep in touch

There are 25 handover solution patterns and are distributed over the following 9 sections: Predecessor's preparation activity, Successor's preparation activity, Third party preparation activity, Predecessor's during handover activity, Successor's during handover activity, Third party during handover activity, Predecessor's after handover activity, Successor's after handover activity, and Third party after handover activity (Table1). Each pattern belongs to a specific section.

We found these patterns by first analyzing 3 anti-patterns previously discussed followed by studying handover issues and talking with businesses that have dealt with handover problems. The procedure of identifying the patterns is described in Figure 5.

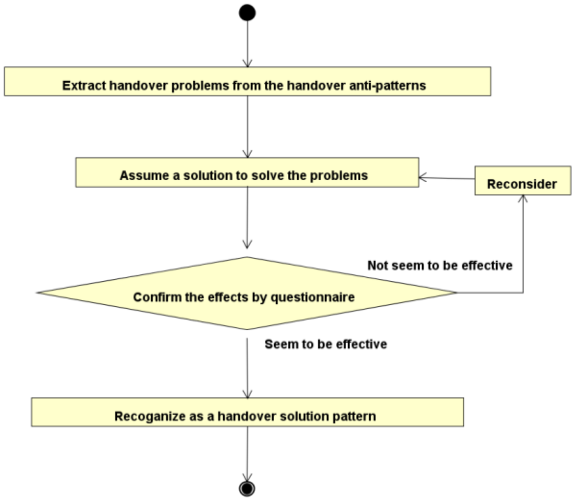

Extract handover problems from the handover anti-patterns
Assume a solution to solve the problems
Reconsider
Not seem to be effective
Confirm the effects by questionnaire
Seem to be effective
Recoganize as a handover solution pattern

Figure 5. Procedure of identifying the patterns

We originally held a handover workshop, where we collected data on handover experiences. Most participant recollected failures, which helped us find our three handover anti-patterns. These were subsequently used to identify concrete handover problems. By analyzing the anti-patterns, we noted root causes and outlined them as problems and solutions. To validate the patterns, we created questionnaires and targeted businesses to validate the handover patterns. In the questionnaires, we explained each pattern's success story and asked the respondents to indicate whether they have had similar experience. This questionnaire focused on the three patterns presented in this paper (*Spread of knowledge, Handover in a different room*、 and *Firewall for handover*). Our questionnaire confirmed that these patterns can be effective for handover operations (see Section 5). We plan on validating the other 22 patterns through various questionnaires in industrial settings in the near future.

4.1 Spread of knowledge

“We need not only 1st hand, but also 2nd and 3rd hands”— Katsuhide Nomura (Famous Japanese baseball coach)
In the preparation phase, the predecessor prepares to handover the system by sharing basic knowledge about the system to third parties. This helps the predecessor prepare and to ensure important details are covered (Fig. 6).

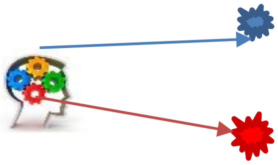


Figure 6. Spread of Knowledge

Hiro has been in charge of the financial system for a long time, but he will soon be transferred overseas. He does not have enough time for handover. His successor will be Kei, who has never worked on the financial system. Since Hiro is the only expert of the system and no one can review his documentation for the handover, and help the successor.

***

How can Hiro best support the handover process in order to transfer enough knowledge to Kei so she can take over the financial system?

Compared with Hiro, Kei is less experienced and knowledgeable. Kei might need some help after the transfer. There are some people like Yoshiaki and Joe, who understand parts of the financial system and are also responsible to make sure the system is maintained properly. Hiro has no reservations communicating his knowledge about the financial system to Joe and Yoshiaki because there is a horizontal atmosphere in the organization and he has a long trusting relationship with both.

Therefore:
The predecessor, Hiro, starts to organize his knowledge about the financial system and begins to share core knowledge about the system to responsible third parties before the handover such as Joe and Yoshiaki.

As he prepares for the handover, Hiro starts by spreading knowledge about the financial system to Joe and Yoshiaki among their other tasks. Hiro defines the roles of Joe and Yoshiaki as actual operations and future visions, respectively. Both become more knowledgeable about the business side and the technical side of the financial system. Although neither is an expert, together they become fairly knowledgeable about the system and are capable of reviewing Hiro’s documents. Before the transfer, Hiro communicates the role of Joe and Yoshiaki to Kei. After the transfer, Kei can ask Joe for help about the technical side, and Yoshiaki about the business side.

***

The *Spread of Knowledge* can sometimes be done informally like a daily chat. However, this *Spread of Knowledge* will often not contain detailed information. It is important that the predecessor shares knowledge with more than one person. Third parties have their own tasks and it is impossible to become an expert of the system. Note that *Spread of Knowledge* is not the spread of responsibility. The only responsibe person for the system is the successor Kei.

The predecessor should choose a third party who has at least a basic understanding of the system and responsibilities. In this scenario, Joe and Yoshiaki are in the same section with Hiro, and they have some responsibility for the financial system. During this *Spread of Knowledge*, the third party gains a better understanding of the system and can apply this knowledge in their everyday work. The predecessor needs to make sure that each third party's knowledge is clear so that they will be able to share it with the successor.

The *Spread of knowledge* can also help to find reviewers of the handover documentation. The *Group Validation* [JN05] and *Creator-Reviewer* [Li00] assist with the *Spread of knowledge* because they recognize the need for a third party. Thus, *Spread of knowledge* is a necessary condition for: *Review for the documents of handover*, *Handover in a different room*, *Firewall for handover*, *Supporting developing successor*, and *Witness for the knowledge*. *Spread of Knowledge* is effective for dealing with many of the problems noted in the *Handover anti-patterns* [KHY16]. Yoshiaki, the third party, might use a *Mercenary* or *Successor as capable as the predecessor* to avoid the shortage of the successor.

4.2 Handover in a different room

“No time for baseball players to suffer in human relations. Everything should be resolved by chasing a ball, swinging a bat” — Shigeo Nagashima, (Famous Japanese baseball coach)

Handover in a different room is when the predecessor uses a separate room to limit interruptions during the handover process. The predecessor relies on the third party’s gained knowledge about the system to allow them to concentrate on the handover with fewer interruptions. （Fig 7).

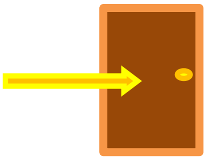


Figure 7. Handover in a different room

As soon as Hiro accepts an overseas transfer, he prepares by initiating *Spread of knowledge* to Yoshiaki and Joe. Hiro then begins the handover with Kei. However, the handover is often interrupted by phone calls and people coming to Hiro's desk to ask simple questions about the financial system. They cannot concentrate on the handover.

***

How can Hiro and Kei minimize distractions so they can focus on the handover process?

Kei is not as knowledgeable and experienced as Hiro about the financial system. Hiro needs to make sure that he communicates sufficient core knowledge about the financial system to Kei without overwhelming her.

Because Hiro knows the system well, he can answer quickly. The normal path to ask for help is to call Hiro’s desk for any issues related to the financial system. However, it can take a lot of time during the day to respond to all of these requests although Hiro has very limited time to prepare for the handover. There is no strict rule that bans him from leaving his office, Hiro can use a vacant meeting room in the internal system department office.

Therefore:

To concentrate on communicating the predecessor’s knowledge to the successor without interruptions, Hiro and Kei use an available meeting room during the handover process.

Hiro temporarily moves into the vacant meeting room to concentrate on the handover with Kei. Before the move, Hiro asks Joe and Yoshiaki to act for him while he is away from his desk. Joe and Yoshiaki substitute for him between their jobs answering his phone and any questions that may arise. Because Hiro and Kei can concentrate on the handover without interruptions, the handover can be substantial and more successful.

***

This pattern requires the *Spread of knowledge* because the third party cannot act for the predecessor without the necessary knowledge. Thus, the predecessor must carry out the *Spread of knowledge* properly before *Moving to a different room*. Even if they moved to the room, other people may come to the room to ask a question. To minimize this, Joe and Yoshiaki become a *Firewall for the handover*. Because third parties have other responsibilities, this pattern has limitations. To avoid these problems, handovers must be completed in a timely fashion. Otherwise, the third party might become overwhelmed and unable to perform all of their normal tasks. *Day Care* [JN05] enhances the performance by concentrating on the system development, while the professional is taking care of a new member. *Handover in a different room* is similar to *Day Care* because it enhances the handover quality by allowing the predecessor and successor to concentrate on the handover, while third parties temporarily act for the predecessor.

Hanover in a different room aims to allow for better communications between the predecessor and the successor because it limits interruptions. Not enough communication is one of the main causes for the *Background to be unclear* [KHY16] and *Necessary knowledge to be omitted* [KHY16], Thus, *Handover in a different room* can be an effective countermeasure to help avoid these anti-patterns.

4.3 Firewall for the handover

“It was toilsome game, but everybody helped me. “— Hiroki Kuroda, (Famous Japanese baseball pitcher)

The third party uses *Firewall for handover* to protect the predecessor and the successor during the handover. The predecessor has many job responsibilities which still need to be performed during the handover. This can often interfere with the handover. The third party stands in for the predecessor to support the predecessor during the handover (Fig 8).

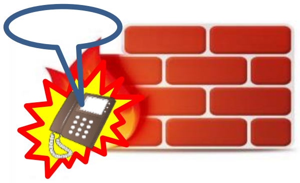


Figure 8. Firewall for the handover

As soon as Hiro accepts the overseas transfer, he begins his handover. Since knowledge about the financial system is very important, Joe and Yoshiaki want Hiro to allot a sufficient time for the handover. But Hiro is very busy and has many interruptions, preventing him from allotting the sufficient time for the handover.

***

Hiro’s regular tasks involving the financial system often kep him very busy and do not allow much time for the handover. How can Hiro find the necessary time to ensure a good handover to Kei?

Compared with Hiro, Kei has less experienced and knowledgeable. If Hiro cannot find the time to communicate sufficient core knowledge about the financial system to her, she cannot support the financial system. There is very little time left before Hiro’s transfer. Joe and Yoshiaki need to help create an environment where Hiro can effectively communicate his knowledge about the financial system to Kei.

Because there is a sense of unity in the internal system section, Joe and Yoshiaki feel that it is natural to help with Hiro’s handover. Hiro has many responsibilities with the financial system, which often results in answering many calls or questions that arise during normal operations of the system. It is important that the financial system keeps operating and people have their questions answered.

Therefore:
Joe and Yoshiaki help Hiro by acting as a firewall for Hiro during the handover. They can do this by assisting with some of the normal operations, including answering regular inquires.

Joe and Yoshiaki start by asking Hiro how they can help him. Hiro thanks them and forwards calls and inquiries about the financial system to Joe and Yoshiaki. Joe and Yoshiaki have learned some important information about the financial system when Hiro executed the *Spread of knowledge.* Therefore, they are able to help Hiro by answering his phone and responding to questions, especially the simple ones, allowing Hiro to spend most of his time communicating knowledge to Kei. After Hiro's transfer, Kei is accomplished in the role of Hiro’s successor.

***

Spread of knowledge is a necessary condition for *Firewall for the handover*. The *Handover in a different room* and *Firewall for the handover* are often used in combination. However, third parties must be careful not to be overly assertive. Remember that third parties are not experts. When third party cannot adequately address a situation, they must judge the urgency of the situation. If a situation is urgent, the third party must interrupt the handover by asking the predecessor for help.

Since the third party has his own work, becoming a *Firewall for the handover* can be problematic because this requires balancing helping with the handover while still performing his normal duties. The firewall person works to be as flexible as possible as he assume the roles of *Firewalls* [JN05] and *Gatekeeper* [JN05]. A cohesive team is crucial for success. The third party should assume that the predecessor’s job is temporarily part of his or her job. *Breaking Down Barriers* [JRH14] contributes to the performance of the *Firewall for handover*. This pattern is an effective countermeasure against the *Background is unclear* [KHY16] and *Necessary knowledge is omitted* [KHY16] anti-patterns.

5. CASE STUDY

To examine the effectiveness of these patterns, we sent a questionnaire to five people within four different companies. The questionnaire targeted businesspersons working for the system department more than three years. This questionnaire is the beginning of a case study to help validate our patterns and elucidate other possible scenarios and patterns. Table 2 shows the result of the questionnaire.

Table 2. Result of questionnaire

	Effectively used	Ineffectively used	Not used
Spread of knowledge	3	0	2
Handover in a different room	4	0	1
Firewall for the handover	3	0	2

According to the table, the patterns proposed in this paper seem to be effective. To better illustrate this, we introduce one of our experiments as a case study of these patterns. At a company we interviewed one of the responders, which we will call A. He shared his experience of using "*Spread of knowledge*" and "*Handover in a different room*" during the handover process. Below is a summary of what we learned from the interview.

5.1 Scene

A works for one of the biggest computing companies in Japan, which develops and sells database management systems. Because A will be retiring in the near future, it is important that he handover his responsibilities. A is in charge of the development for part of this system, which consists of multiple modules that are developed using the process described in Fig. 9Each module has an owner (developer) who is responsible for its development of modules. A is one of the module owners and has developed various modules. The module owner is the only person who can edit his modules. During the handover, A used the "*Spread of the knowledge*" and "*Handover in a different room*" patterns, which he felt were very effective during the handover process.

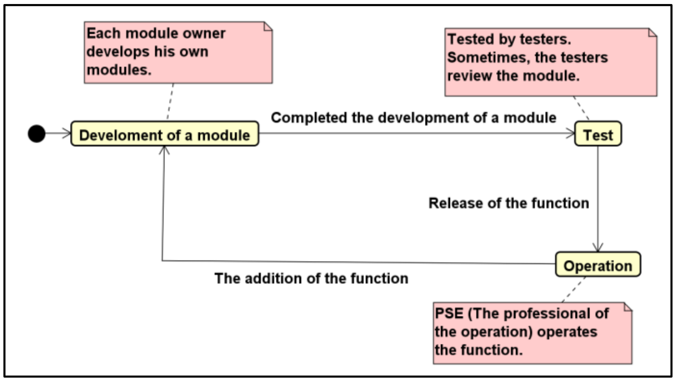

Each module owner develops his own modules.
Tested by testers. Sometimes, the testers review the module.
Develoment of a module
Completed the development of a module
Test
Release of the function
Operation
The addition of the function
PSE (The professional of the operation) operates the function.

Figure 9. Process of the development

5.2 Experience with Spread of knowledge

Many people, like testers and developers, are involved in the development of the system. A thinks it is natural to share knowledge with many of your coworkers. Also, the company where A works encourages people to share knowledge. There are two parties that are very interested in the work A is responsible for and have a vested interest in a successful handover.

The first party is those responsible for maintaining the system and for adding new features. When A needs to add new features, he must understand the knowledge about the previous release. However, sometimes A cannot communicate immediately with the people who were responsible for defining previous features. Therefore, the developers are required to write plain specifications and code comments so that anyone can understand the knowledge at a later date. This is a kind of *Spread of knowledge*.

The second party is other module owners. There are other module owners (developers) that are working on various modules that are somehow related with A's modules. For example other modules may use features from A's modules or need to integrate with A's modules. Therefore, it is important that A shares knowledge about his modules with other related module owners. By applying the *Spread of knowledge*, many colleagues can review A's work, reducing issues and helping his successor better understand the modules that he will be responsible for after the handover. The company's *Spread of knowledge* policy has assisted in successful handovers.

5.3 Experience with Handover in a different room

During the handover, A moved to a different room when sharing knowledge with his successors and third parties. The main reason A moved to a different room was to avoid bothering other people since the handover may be noisy. However, an added benefit was that A could concentrate on communicating without interruptions.

6. CONCLUSION AND FUTURE WORK

In this paper, we outlined 25 handover solution patterns as patlets, and described three of them in detail: *Spread of knowledge*, *Handover in a different room,* and *Firewall for handover*. We also expressed the relation between handover anti-patterns and handover solution patterns by a pattern map and define them as a pattern language for handover. To better understand these patterns in relation to handover issues, we used concrete scenarios. Our questionnaire presented to four companies revealed that the three patterns introduced in Table 2 were common practices seen by these companies. However, the remaining 22 patterns have yet to be formalized and verified. In addition, other unidentified patterns may exist. In the future, we plan to formalize and verify the effectiveness of the remaining 22 patterns and continue mining the language for additional patterns.

There are three future works for a pattern language for handover. First, our pattern language defines the third party as an indirect person for the handover, but in reality, there are various kinds of indirect persons for handovers. We need to define the roles of these other persons. Second, we outlined the handover solution pattern sequences in text only. However, it would be great to see visualization of the pattern sequences. We will consider about the visualization of the pattern sequences. Third, our pattern language does not address important questions as part of the handover process, "What is the conversation and what should be formally documented?"

ACKKNOWLEDGEMENT

We thank all the respondents of the questionnaire. We would also like to thank our shepherd Lise Hvatum for her valuable feedback during the PLoP shepherding process.

REFERENCES

[V90] Thomas Volleman. 1990. "Transitioning From Development to Maintenance," *in Proc Software Maintenance, 1990, Proceedings., Conference on 1990,* San Diego, USA, 1990., 189-199

[AM10a] Ahmad Salman Khan., Mira Kajki-Mattsson. 2010. "Core handover Problems," *in Proc 11th International Conference on Product Focused Software 2010 (PROFES'10)*, Limerick, Ireland, 2010., 135-139

[KHY16] Kei Ito., Hironori Washizaki., Yoshiaki Fukazawa. 2016. "Handover anti-patterns," *in Proc 5th Asian Conference on Pattern Language of Programs 2016 (Asian PLoP 2016),* Taipei, Taiwan, 2016.

[GE08] T. Grandon Gill., Eli Cohen. 2008. "Research themes in complex Informing", Informing Science: the International Journal of an Emerging Transdiscipline (2008, Vol. 11) 1)., pp. 147-164

[YSAAT16] Yuma Akado., Shiori Shibata., Ayaka Yoshikawa., Akimitsu Sano., Takashi Iba. 2016. "Cooking Patterns: A Pattern Language for Everyday Cooking," *in Proc 5th Asian Conference on Pattern Language of Programs 2016 (Asian PLoP 2016)*, Taipei, Taiwan, 2016.

[ATTET16] Alice Sasabe., Taichi Isaku.,Tomoki Kaneko., Emi Kubonaga., Takashi Iba. 2016. "Parenting Patterns: A Pattern Language for Growing with your Child," *in Proc 5th Asian Conference on Pattern Language of Programs 2016 (Asian PLoP 2016)*, Taipei, Taiwan, 2016.

[JR16] Joseph W. Yoder., Rebecca Wirfs-Brock. 2016. AsianPLoP® Pattern Writing Bootcamp, http://pl.csie.ntut.edu.tw/asianplop2016/files/AsianPLoP2016BootcampPatternWriting.pdf

[MR] Manns, M. L., and Rising, L. Fearless Change: Patterns for Introducing New Ideas. Addison-Wesley Professional, 2004.

[MR15] Manns, M. L., and Rising, L. More Fearless Change: Strategies for Making your Ideas Happen. Addison-Wesley Professional, 2015.

[AM10b] Ahmad Salman Khan., Mira Kajki-Mattsson. 2010. "Taxonomy of Handover Activities," *in Proc 11th International Conference on Product Focused Software 2010 (PROFES'10),* Limerick, Ireland, 2010., 131-134

[JRA14] Joseph W. Yoder., Rebecca Wirfs-Brock., Ademar Aguiar. 2014. “QA to AQ Patterns about transitioning from Quality Assurance to Agile Quality,” in Proc 3rd Asian Conference on Pattern Language of Programs 2014 (Asian PLoP 2014), Tokyo, Japan, 2014.

[JN05] James O. Coplien., Neil B. Harrison. 2005. Organizational Patterns of Agile Software Development. Pearson Prentice Hall

[Li00] Linda Rising. 2000. The Pattern Almanac. Addison-Wesley Longman Publishing

[JRH14] Joseph W. Yoder., Rebecca Wirfs-Brock., Hironori Washizaki. 2014. “QA to AQ Part Three Shifting from Quality Assurance to Agile,” *in Proc 10th Latin American conference on Pattern Language of Programs 2014 (SugarLoaf PLoP 2014)*, Brazil, Ilha Bela, 2014.

APPENDIX

We described the status before and after using the patterns by class diagram and object diagram to express the results of the patterns. By using UML, readers can easily understand how the pattern works and what problems they solve. Herein, we introduce three figures describing *“Spread of knowledge”*, *”Handiver in a different room”*, and *”Firewall for the handover”*.

Spread of knowledge

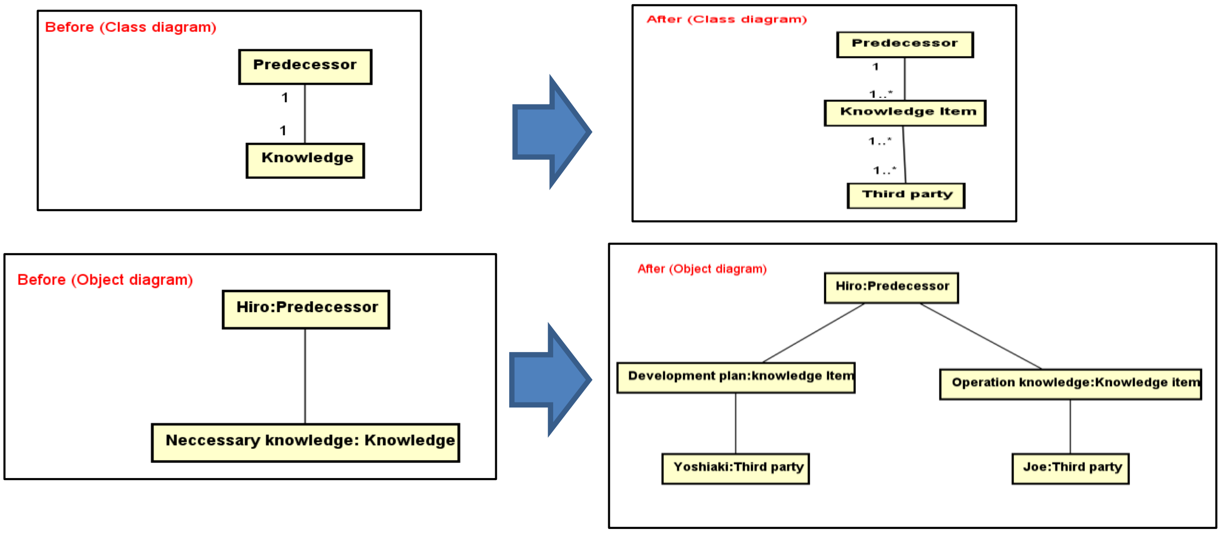


Figure 10. Spread of knowledge

Before using patterns, “Knowledge” and “Predecessor” are one to one relationship, and only the predecessor has “Knowledge”. However, "Knowledge" is subdivided as "Knowledge item" after the pattern is used, and these are shared by several third parties. The reason for subdividing “Knowledge” is that it is inappropriate for third parties to have all the knowledge. Because there is a danger that the location of responsibility will be ambiguous.The successor should be the only one who having all the knowledge. By using this pattern, you can take a backup of the knowledge that the predecessor has, aftter the personel changes.

Handover in a different room

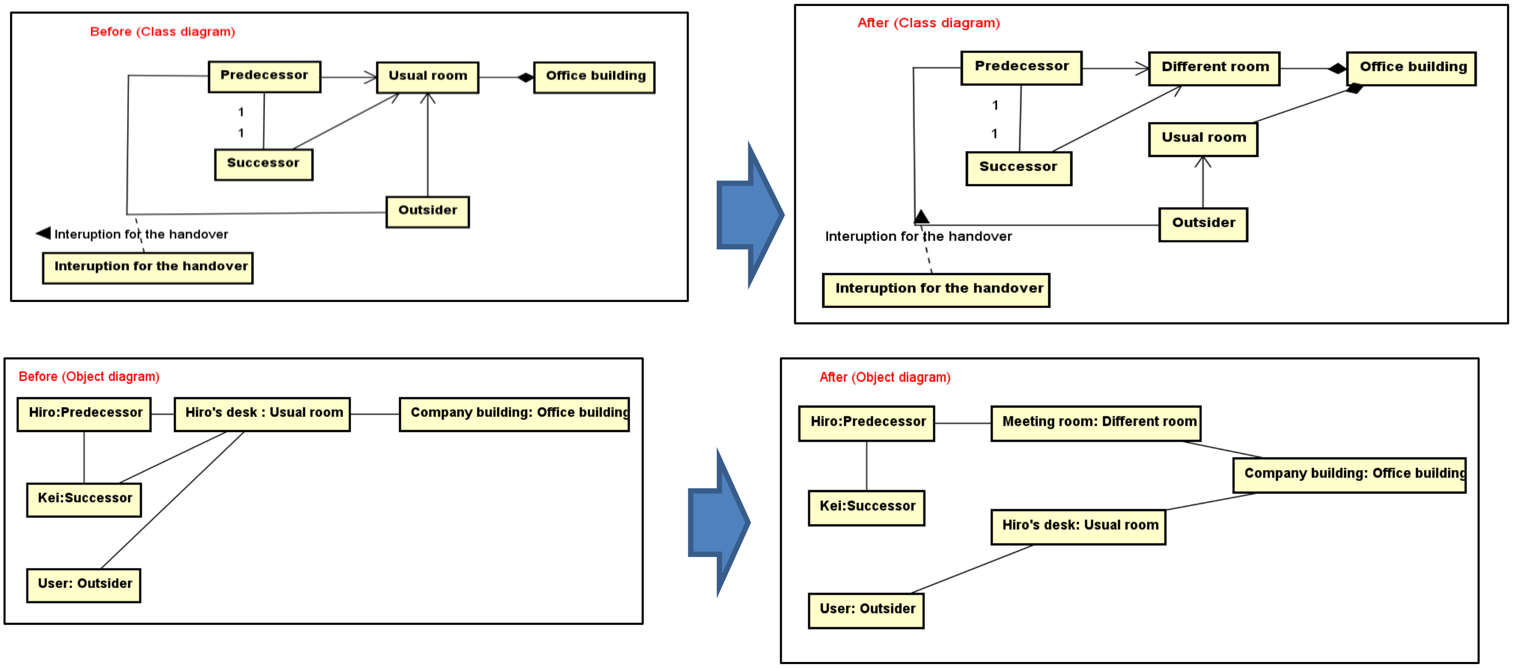


Figure 11. Handover in a different room

Before using the pattern, the predecessor and successor are handing over at "Usual room" (in the object diagram"Hiro's dek"). At the usual room, outsiders might come to ask some questions to the predecessor, and the handover is interrupted by them. Therefore, the predecessor and successor move to a “Different room” ("Meeting room" in the object diagram) and hand over there. Then outsiders will not be able to find them in the “Usual room” and the handover will not be interrupted.

Firewall for the handover

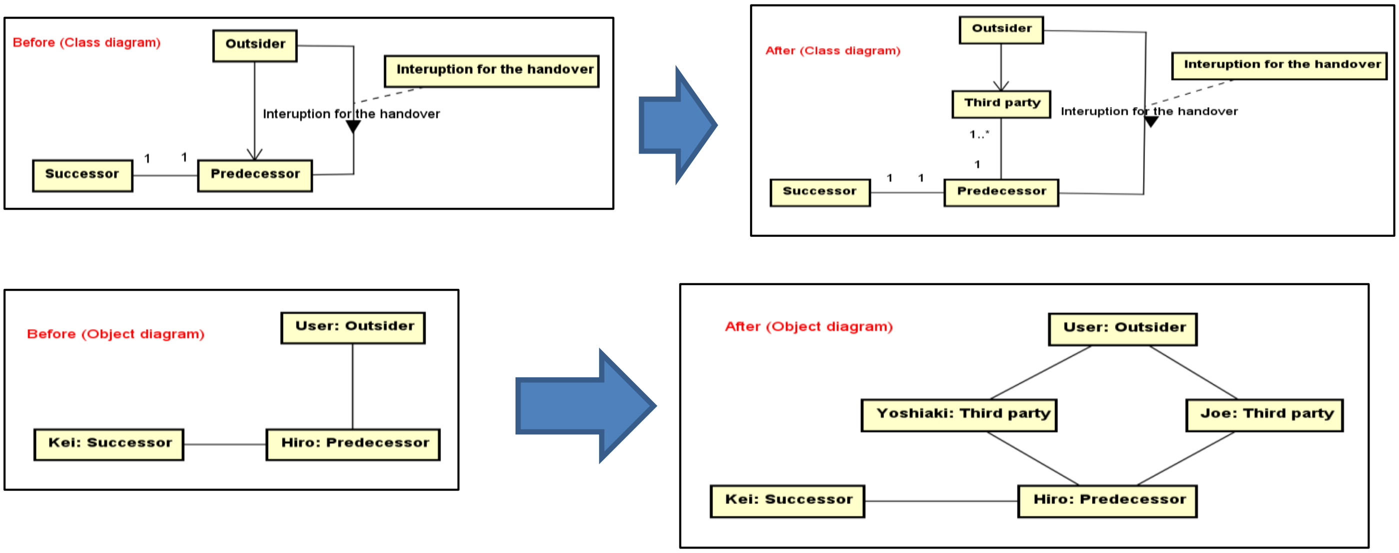


Figure 12. Firewall for handover

Before using patterns, outsiders are in direct contact with the predecessor. However, by relaying the communication between outsiders and predecessors by third parties, outsiders can not directly contact their predecessor. Thanks to the third party being a relay, we can avoid interruption of unnecessary handover by outsiders. In other words, the third party has become the *firewall of the handover*.