

The Role of Analysis Patterns in Systems Analysis

Anna Bobkowska, Jakub Grabowski
Gdansk University of Technology, Poland
Contact e-mail: annab@eti.pg.gda.pl

Abstract

Analysis patterns seem to be a promising approach supporting software analysis. However, the usefulness of this approach has not been proved. This paper presents a 'big picture' of issues related to application of analysis patterns in software development process from practitioner's perspective. It examines several aspects of analysis patterns and their application in software development. Then, it provides a method of system analysis with analysis patterns. And finally, it presents a case study of analysis with analysis patterns for a system supporting a small real medical centre.

1. Introduction

Analysis patterns are conceptual structures which represent patterns for object-oriented analysis. The first book about analysis patterns was written by Martin Fowler [F97], who defines pattern as an idea that has been useful in one practical context and will probably be useful in others. For example, no matter what we plan, we consider some actions, resources, parties, time periods and locations. Analysis patterns capture the experience of analysts and provide reusable fragments of object-oriented analysis models. It is believed that application of analysis patterns can improve quality of analysis and reduce time and cost of system analysis. Fowler's book contains patterns for the areas of accountability, observations and measurements, inventory and accounting, planning and trading. They were developed on the basis of author's experience in industrial projects. Another source of analysis patterns are the proceedings of PLoP workshops [P]. With the reference to the more general idea of 'pattern', the meaning of analysis pattern has been enriched by the notion of 'template' - a universal structure which is valid regardless of the domain. Several analysis patterns can be found in the proceedings of the PLoP workshops, e.g. [F00, FY99, FY00, FY01, HF04a, S00].

Analysis patterns are promising, but not a very popular practice in software development nowadays. They are not supported by general purpose UML tools although some of the UML tools contain design patterns. In the area of application of analysis patterns it is worth to mention the following pieces of work: a proposition of template for analysis patterns descriptions made to facilitate their application [H01]; an observation that analysis patterns can be applied by specialization when one has abstract patterns or by analogy when one transports models from another project [FY00b]; an attempt to integrate analysis patterns with MDA approach [FN05] and the idea of 'stable analysis patterns' (which were designed to satisfy criteria of traceability and generality) together with a method for applying them in opposition to applying analysis patterns by analogy [HF04b]. These papers suggest ways of applying analysis patterns, but they do not pose questions related to effectiveness of this application. Thus, more systematic research related to application of analysis patterns is needed.

Copyright retain by authors.

Permission granted to Hillside Europe for inclusion in the CEUR archive of conference proceedings and for Hillside Europe website.

The goal of this paper is to investigate the application of analysis patterns. But we do not attempt to find a possible application of analysis patterns. We take the perspective of analysts who is interested in the following: How analysis patterns can really support the process of analysis and writing documentation of analysis phase? What are their strengths and their weaknesses? What is their role in analysis? Which systematic process of system analysis with analysis patterns could be applied?

These questions are motivated by the fact that software projects usually have no resources for waste on application of 'yet another' technique. All activities must have their justification in increase of quality or efficiency. Analysis is a very important phase in software development process and it can benefit from analysis patterns.

This paper provides a framework for dealing with very detailed aspects of analysis patterns application. It constitutes a 'big picture' of all identified important issues which should be taken into account as the context in more detailed research.

The target audience are researchers and practitioners who have got already an idea of analysis patterns, but they would like to explore them in more details including the role of analysis patterns in analysis, their strengths and weaknesses, the process of their application and the effectiveness and efficiency aspects of their use. This paper may be interested for professionals with a skeptical attitude towards patterns and analysis patterns in particular. They might find out conditions which must be met in order to succeed with patterns.

The paper is structured as follows. Second section examines analysis patterns for their features and possibilities of application in system analysis. Section three takes a practitioner's perspective and presents a method of systematic application of analysis patterns in system analysis. Section four presents a case study of applying analysis patterns in analysis of a small medical centre. Section five presents conclusions.

The structure does not entirely represent the order of performing research. In the course of work many iterations took place and results of some pieces of work described later in this paper were the input to aspects described earlier. The case study was done (by Jakub Grabowski) in the parallel to theoretical investigations (by Anna Bobkowska). The study described in section 2.7 was made earlier and, in fact, it has inspired posing some of the questions. The proposition of the process described in section 3 is the answer to the problems identified in section 2. The authors hope that use of this process allows to eliminate some problems with application of analysis patterns.

2. Analysis Patterns under Examination

While searching for the role of analysis patterns in software development, we discuss the following aspects:

- Analysis patterns from the perspective of analysis phase;
- Different aspects of the notion of analysis patterns;
- Generality of analysis patterns with respect to domain;
- Fit between the context of discovery and the context of application;
- Quality of analysis patterns;
- Efficiency of the application of analysis patterns;
- Perception of analysis patterns by MSc students.

2.1. Analysis Patterns from the Perspective of Analysis Phase

Analysis is a creative part of software development process. It is concerned with creating the vision of software product, exploring details of business processes, learning about domain and checking whether the proposed solution is really what is needed in the customer's enterprise. The fundamental assumption of analysis is that different companies have different needs even if they operate in the same domain. Some analytical activities can be made according to some guidelines, e.g. describing existing business processes, but others require visionary skills, e.g. creating innovative systems or re-organizing the business process. Analysts must also combine different points of view and discover tacit knowledge. Thus the process of analysis cannot be replaced by automatic insertions of the analysis patterns. However, it can be supported by providing some knowledge about domain, vocabulary and solutions. Analysis patterns are one of many sources of knowledge about systems solutions.

Analysis patterns cover only a few aspects which are important for analysis in software development. With the increasing size of software systems and the increasing role of economical aspects and business modeling when developing systems, modern approaches to system analysis (IIBA [iiba], RUP [RUP]) focus more and more attention on relationship between business and system, as well as the relationship between system deployment and economical factors in software projects.

A proven area of software technology related to systems analysis is requirements engineering. In some cases, requirements engineering techniques are more useful than system analysis, e.g. testing similar systems, ethnography, surveys, communication with customers and future users of the system. Potentially analysis patterns can facilitate communication of analysts and customers by delivering to analyst the knowledge about the domain and vocabulary of customers. However, when customers have different mental representations than these described by analysis patterns, they can even make the analysis more difficult.

One of the reasons of a small popularity of analysis patterns might be the fact that not always in software development a systematic object-oriented analysis is made. Analysis patterns are strictly related to object-oriented analysis. Furthermore, they are a more advanced technique and they are even more seldom in use. In practice, analysts focus mainly on the scope of the system and their functional requirements, fit in context and fit to business. These aspects are seldom covered by analysis patterns. Different approaches to object-oriented analysis can be found in literature, e.g. there are differences between OMT [OMT], Fowler [F97] and Rational Unified Process [RUP]. Some approaches focus more on business while others on key classes of the system. There is no standard of object-oriented analysis since UML [UML] defines only the language and not the application. Additionally, there are some problems with analysis patterns themselves. Fowler doesn't use UML diagrams and structured descriptions. While asking the representatives of software companies or even universities whether they know and use analysis patterns the answer is usually 'no'. The issue of whether analysis patterns are not used because of the lack of knowledge about them, or because of uselessness of the idea of analysis patterns, or maybe because of low maturity of analysis patterns nowadays, requires more investigation.

2.2. Notion of Analysis Pattern

The notion of analysis pattern combines the notion of analyst experience and the notion of a kind of universal template of solution which can be reused in several projects. However, experience from one project does not necessarily mean the template for all other projects.

According to the rules of induction, if one wants to have a general solution one should work out independently several specific cases and make their generalization. Analysis patterns presented by Fowler were developed on the basis of a few industrial projects and most of analysis patterns presented at PLoP workshops were developed in academic environment and their usefulness in real industrial projects is uncertain.

There is a diversity of analysis patterns. Factors, other than their origin, include:

- Different scope of problem - analysis patterns range from a single class, e.g. quantity or time stamp patterns, to fragments of solutions which contain about 20 classes;
- Level of maturity - some patterns were very precisely elaborated while others represent just a 'first approach' to analysis with a list of aspects they do not cover;
- Level of abstraction - some patterns were generalized on the basis of several solutions and present an abstract pattern while others are just description of someone's experience and de facto are fragments of concrete projects.

The diversity of patterns has impact on differences of their potential application.

Additionally, patterns might provide terminology at a higher level of abstraction and resulting benefits for communication similar to the application of design patterns. With design patterns the developers can use just one word 'bridge' or 'mediator' for explaining the entire fragment of solution. However, in order to fulfill that goal analysis patterns should be general and they should have unique and representative names.

2.3. Generality of Domain and Progress in Systems Development

Generality of domain influences the commonality of potential applications of analysis patterns. Some patterns represent domains which are popular and others deal with very specific domains. For example, planning is popular in almost every field of activity while the application of trading is limited to banks and advanced exchange of goods.

Some domains can be characterized by a closer similarity among the cases whilst others are different from case to case, e.g. accounting in a given country is standardized and all accountants must do it similar way while Customer Relation Management (CRM) can be based on a diversity of possible strategies and it can be done in several ways.

Some domains are independent of the context whilst others are dependent on it, e.g. local legal regulations, local customs, local values, etc.

Software systems evolve in time. They become larger and they cover wider and wider scope of activities. The progress in technology and the state-of-the-art of developing systems means that patterns made decades ago might not be useful nowadays. They need to be updated.

2.4. Fit between Context of Discovery and Context of Application

For better understanding of the application of analysis patterns, it is important to distinguish a context of discovery and a context of application. The context of discovery represents the circumstances in which analysis patterns were discovered. The areas include: country, company, projects, methodology, mentality of the authors, time, etc. The context of application represents the circumstances in which analysis patterns are applied. It might happen that an analysis pattern is to be applied in a country in which different regulations or customs obey, or in a project with different focus, needs or methodology. The bigger gap between these two contexts, the lower probability that patterns will fit to the problem at hand.

The analysis patterns need customizations during their application. Let's consider an example of a small shop and a supermarket. Although both of them sell goods, some analysis patterns made in the context of supermarket might be useless in the small shop because of different problems and processes. Another example can be planning. It seems to be a very general analysis pattern. However, plans in different domains cover specific artifacts, e.g. planning software project (additionally to what planning pattern contains) requires phases, iterations and distinction to project-level and iteration-level plans.

2.5. Quality of Analysis Patterns

It is believed that analysis patterns will allow to increase quality of the analysis. The argument is they are verified in successful applications. However, in order to satisfy the goal of increasing the quality of analysis, the quality of analysis patterns must be high. Analysis patterns support elaborating more detailed solutions in the scope covered by the pattern. However there are also the following risks:

- solution contains redundant model while a simpler model is needed;
- the fragments of model which are not covered by the pattern are skipped although they are needed;
- analysts-beginners cannot evaluate quality of the applied patterns and they believe they have good solutions while actually they have poor models.

An interesting and fundamental issue is the meaning of 'quality of the analysis patterns'. Assuming that the goal we want to achieve is ease of searching for useful patterns, ease of their application in terms of customizations and traceability as well as increase in quality of solution, a provisional list of the quality characteristics includes:

- precision of the content - analysis patterns which contain more details are considered as more useful;
- generality regarding domain - more general analysis patterns can find application in more projects;
- level of abstraction - more abstract patterns will fit to more problems, but on the other hand they will require more customizations;
- structure of description - structured description facilitates applications;
- aspects of description - some descriptions of analysis patterns do not cover functionality which is an essential aspect of software analysis; the patterns which cover all essential aspects will be more useful.

Apart from these aspects of quality which are application-specific, analysis patterns should satisfy the criteria of object-oriented analysis models, such as completeness in a given scope, correctness, consistency as well as clear element names on the diagram, clear descriptions, clear layout of diagram etc.

2.6. Efficiency of Analysis Patterns Application

Statements about analysis patterns efficiency claim it is possible to achieve reduction of time and cost of software development with analysis patterns. They usually use the idea of inserting some fragments of solutions instead of making system analysis from scratch.

The idea of savings assumes the most optimistic case: high quality patterns, easy access to patterns, fit of the patterns to problem at hand and easy integration of patterns with other models. In reality, the following risks and their consequences might appear:

- when customizations of patterns are necessary - the savings of time might not be so large;
- when the context of application is very different from the context of discovery - the analysis patterns might not fit at all or their application is not efficient;
- when wrong patterns have been chosen for application - the analysis patterns might not fit at all or their application is not efficient;
- when there are no patterns to support a given part of the solution - the time is wasted for searching for analysis patterns and traditional analysis must be made anyway;
- when the quality of patterns is poor - a poor solution is achieved and analysis must be redone;
- a hidden use of resources is the work on integration of analysis patterns with other models.

In order to avoid the problems described above, one should evaluate usefulness of the patterns before applying them.

When searching for a reference model to estimate savings one encounters a large set of variables. The results of object-oriented analysis depend on analyst's personality, skills and experience, domain and novelty of project, methodology in use and its fit to the analyst style. Most of them can't be controlled. The only variable which can be universal is the methodology (micro-process). A different methodology should be used when making analysis without analysis patterns, and a different one is required to make analysis with analysis patterns.

2.7. Perception of Analysis Patterns by MSc Students

A voice in discussion about analysis patterns can be the perception of analysis patterns by MSc students in specialization of Software Engineering. We make an exercise related to application of analysis patterns which covers the following tasks:

- describe a group of analysis patterns including Fowler's patterns as well as selected PLoP patterns,
- find for them as many applications as possible,
- try to apply them in some cases by customizing the patterns for this application,
- provide comments on their application.

The results of the exercise are presented and discussed on the forum of the group.

About sixty students took part in this exercise. The following results were achieved. The students perform the descriptions very well. The applications they find depend more on imagination of students than on the group of patterns. The analysis with the customization of models seldom is done well. The comments depend on which group of patterns they were working with and what were their achievements.

We have collected the following comments and opinions. The students appreciate industrial origin of the Fowler's patterns. They consider them as advanced, complicated, poorly described (unstructured description of text with diagrams in other notation than UML), sometimes too specific for a universal pattern. Fowler's patterns are considered as potentially useful description of how someone have done the analysis. The patterns from PLoP proceedings are more controversial. Some students consider them useful in practice or useful for learning as academic examples (according to their context of discovery), but others have totally negative attitude. They argue they are useless because they present obvious and simplified models and that they sometimes contain defects. These patterns are considered as easy to understand, well-structured, immature, good for a start but not sufficient for solving real industrial problems. We suspect that the reason of the critics is the difference between

expectations and the content of patterns or preferences of models made in different modeling style, e.g. regarding the use of association class related to association class. Students with industrial experience claim that the analysis patterns sometimes do not cover all essential aspects. For example, in case of stock pattern - it presents only a basic structure for goods in the stock while it does not consider documents, nor calculation of the value of goods in the store, nor different kinds of sub-stores; or in the area of negotiations - there is no representation for offers nor indicators of negotiations which must exist in Negotiation Support Systems.

To conclude, analysis patterns appear as a valuable description of analyst experience with the potential to be used in other projects, but in current state of the art they shouldn't be treated as general, universal templates. Their application cannot replace analysis. Students usually have had an industrial experience and the level of knowledge about analysis they will have when leaving university. Thus, the results generalize well to software developers who have graduated from university but do not have experience with application of analysis patterns. But their opinions about analysis patterns mean that if they do not consider analysis patterns as a mature technology they will not use it in practice.

3. System Analysis with Analysis Patterns

In this section, we are looking for a realistic methodology how to make the best use of analysis patterns in system analysis. After the examination of the analysis patterns, the fundamental assumption of the approach is that they might be helpful in system analysis but the choice and integration of analysis patterns will not replace the process of analyzing software requirements.

3.1. Macro-process for Reuse of Analysis Patterns

Analysis patterns can be considered as reusable fragments of solutions and general rules of reuse paradigm can be applied. Reuse paradigm defines two processes:

- development 'for reuse', which produces reusable assets and stores them in a repository;
- development 'with reuse', which develops applications with the use of reusable assets; it contains also other activities e.g. acquiring or verification of requirements.

By analogy, system analysis with analysis patterns should also have these two processes. They are presented in Fig. 1. The process 'for reuse' contains creation and collection of analysis patterns. It can be realized in a company by creating specific patterns or by acquiring (and customizing) the patterns which have been published. The outcome of this process is a repository with structured descriptions of analysis patterns. The process 'with reuse' contains software development which includes system analysis with analysis patterns.

One of the tasks made during the application of analysis patterns is searching for relevant patterns. The repository of analysis patterns plays the key role in this activity. In order to assess efficient application of analysis patterns the repository should contain appropriate number of right patterns and the patterns should be described with a template which facilitates searching. By analogy to other reuse techniques one can expect much better improvement in quality with the use of company-made analysis patterns than with general ones.

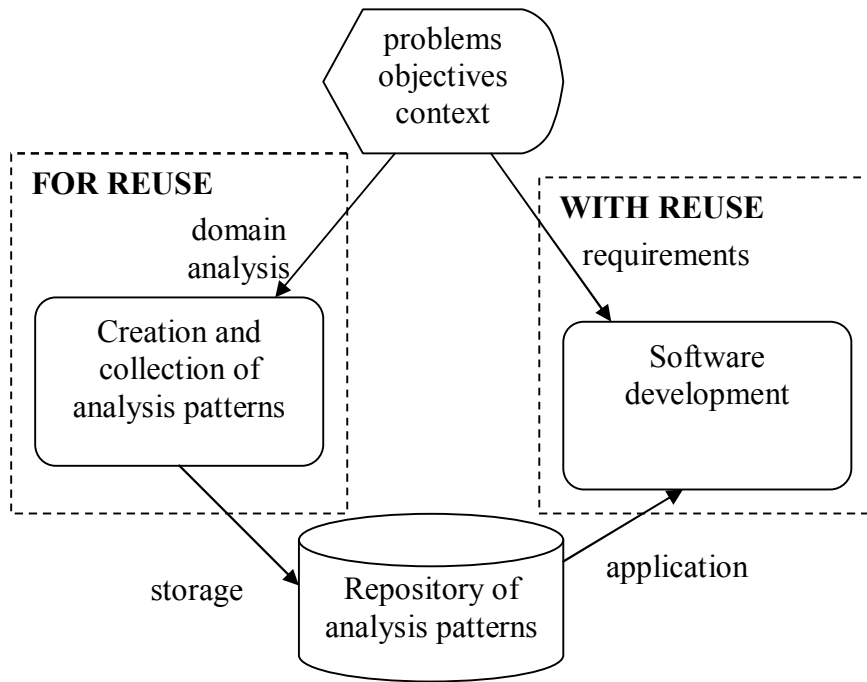


Fig. 1. Analysis patterns in the perspective of development 'for' and 'with' reuse.

3.2 Micro-process of Analysis with Analysis Patterns

The process of system analysis with analysis patterns which is a part of the process "with reuse" is presented in Fig. 2 and activities are described in Table 1. The main role which performs all the activities is analyst. Sometimes the analyst must communicate with the customer, especially when preparing the vision and validating the model. The product of this process is documentation of the analysis which contains both textual specifications and models of the system.

Some people prefer using analysis patterns after they make models. In this case, analysis patterns are useful for extensions, improvements and verification of the model. When referring to the process presented in Fig. 2., draft models made during the task 'identify areas and make draft models' are more detailed and 'customization to the problem' is specialized to extensions and improvements with analysis patterns.

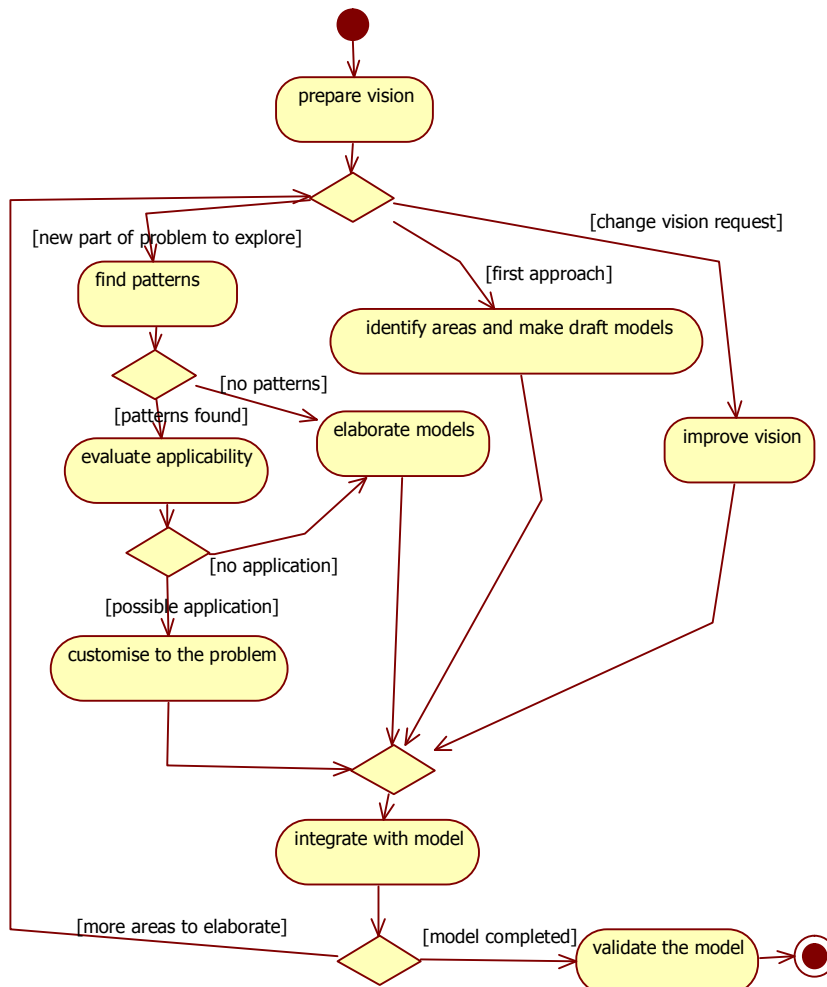


Fig. 2. Activity diagram with a sketch of system analysis with analysis patterns

Table 1. Description of activities in the process of system analysis with analysis patterns.

ACTIVITY	DESCRIPTION
prepare vision	the activity of making initial analysis in order to determine scope of the system in context as well as objectives, priorities, users and the scope of functionality
identify areas and make draft models	the activity of defining key areas, problems and classes before acquiring the patterns; with this activity it should be possible to avoid the risk of skipping the areas uncovered by the patterns, it might contain use case modeling as well
improve vision	the activity performed when analysis leads to the discovery of a better solution than one stated in vision or the analysis causes inconsistency with vision, in this case the vision should be modified
find patterns	the activity of searching for the patterns which could be useful in the analysis of the area under exploration
evaluate applicability	the activity of evaluating the patterns which have been found for their quality, fit to project and possibility of application
customize to the problem	a group of activities which make the patterns fit to the problem at hand; it can include specialization, analogy, simplification, enhancements, replacements of some element with others etc.
elaborate model	the activity performed when no patterns can be applied in a given area, in this case traditional methods of analysis should be used
integrate with model	the activity of integrating the elaborated part of problem with the fragments which have been developed before
validate the model	the activity of checking whether the model expresses what has been expected, e.g. with the participation of a customer

3.3 Specific Context of Use of Analysis Patterns

Analysis patterns are one of the sources of knowledge about the system under analysis. They will not replace the activities of meeting customers and other stakeholders, reading their documents, business process modeling, observations of their work and making decisions about the objectives of the system under development and its business value. Thus, a question for another piece of research is how analysis patterns can be integrated with other methods of system analysis and requirements engineering.

So far, we have presented a proposition of system analysis with analysis patterns. But in several contexts they can gain a new application. In the context of Model Driven Architecture (MDA), the application of patterns is one of the types of transformations. Although MDA deals mainly with transformations of Platform Independent Models (PIM) to Platform Specific Models (PSM) which correspond to design level, analysis patterns could be used in refinements of Computation Independent Models (CIM). A more important role can be played by analysis patterns in Model Driven Software Development (MDSD) approach. It assumes that the models play the central role in software development and in more mature forms that all organizational knowledge is represented in models. In a company where several similar systems are developed, some analysis patterns can be applied in their development. In this case, many company 'home-made' analysis patterns can be created and applied both in UML or domain specific languages (DSL).

4. Case Study

The goal of this case study was to demonstrate the application of analysis patterns. We were interested in the following aspects:

- benefits and problems related to the use of analysis patterns,
- the impact of analysis patterns on the process of analysis,
- efficiency-related aspects.

The analysis with analysis patterns has been performed for a small real medical centre. It was a non-commercial project made as a part of Master Thesis. The choice of the organization was motivated by the fact that there are many medical analysis patterns. Additionally, we were in contact with a doctor employed in this medical centre which allowed for realistic business model of this organization as well as for acquisition and verification of actual requirements for the system.

4.1. Description of the Medical Centre

The medical centre employs about six doctors, a few nurses and a receptionists. It is located in a one-storey building which contains five consulting rooms, a registration office, a waiting room and a few small storage rooms. Doctors have their weekly schedule which contains information on their working hours and planned holidays. Patients can register at the registration office for an appointment on a given day. Medical records are kept in a large paper archive placed in the registration office. Each day a receptionist checks out (looking into doctors' registers) which patients have appointments with a given doctor and carries their medical records to doctor's consulting room before the beginning of work. At the end of the working day, a receptionist takes all the medical records back from doctors' rooms to the archive in the registration office.

4.2. Scope of Analysis

The analysis phase lasted for over a month and involved over a dozen of meetings with the doctor and the receptionist employed in the medical centre. The full version of this case study takes over a hundred of pages. The main products of the analysis are:

- Vision document - which describes the details of organization and focuses on the key areas of the analyzed system. These are: electronic medical records, patient registration for the appointment with doctor, stock and resources management, orders and shipments, and contracts for rendering medical services by external health care organizations;
- Use cases - three diagrams with five actors and forty-three use cases with appropriate descriptions;
- Static model - a class diagram with forty-nine classes and their descriptions.

4.3. Application of Analysis Patterns

Four analysis patterns has been applied in the case study. These are: the analysis pattern for reservation and use of reusable entities [FY99], the patterns for observations and measurements [F97], the analysis patterns for the order and shipment of a product [FY00] and the analysis pattern for inventories [F00]. Analysis patterns application can be summarized by the following metrics. Twenty-five classes of the full class diagram (49 classes) were adopted from analysis patterns. There were nine major modifications made to the original patterns' classes. The modifications included specialization, removal of some original classes because they were redundant and inadequate for the model, changes in class or attribute labels as well as addition of a few new attributes.

As an example, we present customization of the analysis pattern for reservation and use of reusable entities. The original class diagram of this pattern contains the following classes:

- Client - with attributes of name, address, phone_No;
- Entity_Type - with attribute Type;
- Entity - with attributes entity_id and entity_layout;
- Availability - with attributes capacity and number_avail;
- Reservation - an association class between Client and Entity_Type with attributes start_date/time, end_date/time and confirmation_No;
- UseRecord - an assocoation class between Client and Entity with attributes contract_No, act_start_date/time, plan_end_date/time, act_end_date/time and payment_type.

Fig. 3 presents a fragment of the customized class diagram. The following modifications has been made during the customization of the pattern:

- The original class “Reservation” was used by analogy and resulted in a class “Registration”. It was also simplified since reservation is made for a given time and date whilst registration requires only a given date in this medical centre.
- “Doctor” corresponds to the “Entity” from the original pattern diagram. This analogy may seem quite odd, but in fact it fits well in this application. Patient literally 'uses' not the doctor but the advantage of doctor's services for a given period of time.
- “Visit” is an equivalent of the original class “UseRecord”, because it confirms the fact that the patient was examined by the doctor on a specific day and that the examination lasted for a certain period of time. Visit class contains the results of the medical examination (as a part of that patient's medical record). Every visit should have a corresponding registration.

- The “Patient” refines the “Client” class. Patient reserves a particular entity, what means that he registers for an appointment with a given doctor for a given day and this fact finds its confirmation in a “Registration” written to the doctor's register of visits. Every Patient is a beneficent of doctor's services in the same way as client uses a certain entity.
- “Doctor's register of visits” represents the general “Availability” class from the original pattern diagram. It contains all registrations to given doctor and can be used to determine his availability on a certain day.
- Class “Entity_Type” was considered unnecessary ('doctor's specialty' was modeled with 'type') and it was removed from the model.

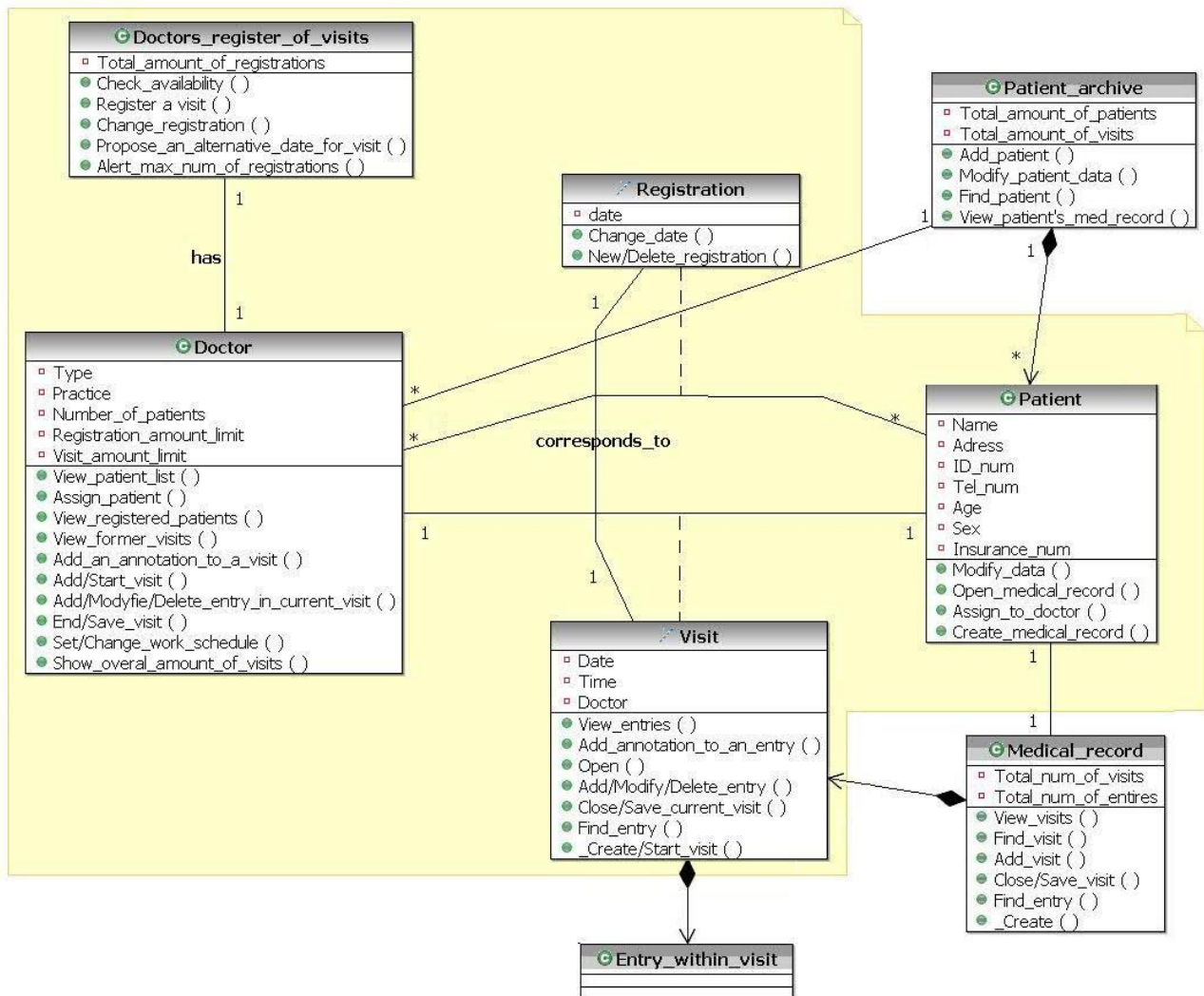


Fig. 3. Fragment of customized class diagram

4.4. Reflections on the Application

This case study was made independently from the work on the method described in section 3. Thus, this method was not used but, in fact, all its activities took place. First, a rich and exhaustive vision of the system was prepared. It precisely describes the scope of the system in the context. The objectives of the system were formulated and priorities for them were assigned. The vision also included the initial requirements for the system and a brief description of its users and their characteristics. Then, use case model was developed. The task of 'identifying areas and making draft models' was limited to defining the areas for

elaboration. Some draft models were created at the stage of evaluating analysis patterns applicability. Finding patterns activity was performed during the earlier part of work on the diploma. The search for useful patterns for the area of exploration was not a necessary task since the author prepared earlier a catalog of analysis patterns and he had a good knowledge of patterns. A brief evaluation of the acquired pattern applicability was performed. The main task at this stage was to determine whether they fit to project and whether their application was possible and reasonable. The context of discovery was compared with the situation and needs of the medical centre. This task allowed to estimate the amount of modifications required to fit the patterns to the project. It provided justification for their use in this project. The activity of customizing the patterns took place at the beginning of static model creation, just after the use case diagram was made. Some modifications of the patterns were made to fit them to the projects requirements. They mainly included simplification (removing unnecessary elements) and specialization (changing the names of a few classes and attributes). In general, this activity was performed surprisingly smoothly and it did not bring larger difficulties. The patterns made foundations for the model and the base for further elaboration and enhancements which involved traditional methods of analysis. After the analysis was finished the model was validated. This process brought a positive result which means that the model expressed what was expected. The major difference with respect to the proposed method was the focus on use cases after preparing the vision and before applying analysis patterns.

The acquisition of analysis patterns started early, at the stage of developing use cases and it has influenced to some extent the meetings with the representatives of the medical centre. Analysis patterns were a kind of reference points for the discussions. They directed the questions which were asked while searching for elements of patterns and concepts in the organization's description. The fact that patterns' concepts may sometimes differ from those of the analyzed organization had to be taken into account to avoid possible misunderstandings.

In this application of analysis patterns we have faced the problems related to the differences between the context of discovery and the context of application. Each of the patterns used in the analytical process had to be adapted to the analyzed environment. In order to make this task, the essential similarities and differences between the mentioned context had to be identified. Then, necessary adjustments and modifications in the model were made.

Application of analysis patterns has brought the following benefits. The greatest advantage was a feeling of shortening of the time spend to understand the rules, principles and mechanisms in the domain. This approach simplified the whole process as the patterns brought not only a general skeleton structure but also many ready elements useful for covering some areas of the medical centre. The last important advantage of analysis patterns is that they introduce flexibility into the model. It may affect not only the resulting analysis models, but also the design and implementation phase in the software development project.

One of the main disadvantages of using analysis patterns were some certain difficulties in adapting them to the set of gathered requirements. Definitely, the main reason was their high level of generality and sometimes a lack of good examples of their application. Unfortunately, the feature of generality can not be completely eliminated because its removal can make the patterns stop fulfilling their major function i.e. being templates of solutions for common, recurring problems. That is the reason why the system analysts, who intends to use analysis patterns, should be aware that they will have to spent a considerable amount of time on making all the necessary modifications and extensions. In some cases, the time spent on seeking patterns and then adapting them to match system requirements will equalize or even exceed the time needed to perform the whole analysis without using the patterns at all. This

matter might get one to wonder if it is reasonable to make use of patterns and benefit from the unquestionable flexibility they introduce.

We would like to finish this section with a 'post-mortem' analysis. The model appeared to have some defects. For example, it doesn't contain doctor's first name and surname. When making traditional analysis the doctor's name probably would be the first attribute we would add to the class of doctor. As it was used here by analogy to 'entity' and entities do not usually have names but identifiers, this attribute is missing. Another example of mistake is 'entry_within_visit' which does not have attributes although it should have had. This is a new class and the author forgot to elaborate it. Yet another example of mistake is multiplicity of 'visit' which should not be 1 on both sides. These mistakes prove that application of analysis patterns does not necessarily improve the quality of the entire solution. Analysis must be made with or without analysis patterns. After making this case study, we were informed that more analysis patterns for medical applications were published [SF04, SFL05]. We did not ignore them intentionally but we did not find them in the large archives of the PLOP workshops. This observation confirms the important role of repositories of analysis patterns.

This case study generalizes well to the situation in industry when inexperienced analyst performs analysis with general-purpose analysis patterns. With the increasing experience analysts could learn how to identify and avoid the problems. The use of the method from section 3 could facilitate effective application. In case of using repository of company-made analysis patterns the effectiveness and efficiency are expected to increase significantly.

5. Conclusions

In order to clarify the role of analysis patterns in software development we have examined several aspects of analysis patterns and their application. Their strengths result from the idea of reusable fragments of models, from the experience of analysts and from the verification of the solution in real software projects. Their weaknesses are related to problems with generality, abstraction, precision and description of analysis patterns. During the application of analysis patterns, analysts should take into account the difference between the context of discovery and the context of application. Analysis patterns can facilitate system analysis but it is unlikely they eliminate the need of using other techniques related to system analysis and requirements engineering. They can support creating object-oriented class diagrams and help in understanding the domain. They do not support uncovering specifics of the system, business modeling and, in most cases, analysis of system's functionality.

A systematic process of system analysis with analysis patterns has been proposed. It suggests elaboration of the vision of system under development and then using the available and useful patterns in elaboration of the areas of the problem. In order to avoid unnecessary problems each analysis pattern should be evaluated before using in the project and then customized and integrated with the remainder of solution. The entire model should be validated. It is worth to mention that intention of using of analysis patterns should result in changes of the process. In order to eliminate the risk of leaving uncovered the parts which are not elaborated by the patterns, identification of the 'areas for elaboration' at the beginning of the process is suggested. This process was developed with the consideration of realistic industrial assumptions. The acceptance by practitioners requires verification in further studies.

The case study has shown that it is possible to apply analysis patterns in system analysis. It has also indicated for some benefits and problems of this application. The main benefits are: help in discovering details about the area, flexibility of the model, and possibility of using the

knowledge about the domain in communication with customers. The main problems are: lack of coverage of the functional perspective by analysis patterns, redundant models or lacking details caused by the patterns use, time spent on customizations, misunderstandings when patterns correspond to different mental representations than these owned by customers.

From this perspective we can additionally formulate some requirements for analysis patterns. In order to increase their usefulness, they should have structured descriptions which would facilitate searching for needed analysis patterns, they should cover functionality, e.g. use cases, they should have unique names, and they should be verified and improved to be characterized by really good quality. Catalogues of analysis patterns would facilitate the process of searching for needed patterns.

Some of the discoveries can be valid also for other kinds of patterns. For example, the difference between the context of discovery and the context of application seems to be general enough to exist also in other kinds of patterns. The aspects which were used to examine differences between patterns might be useful for examining other kinds of patterns as well. Finally, the method of analysis patterns application with a small change of 'vision of the system' to 'goal one wants to achieve' and a change of 'model' to 'solution' seems to be applicable also for other kinds of patterns. However, these are only suppositions which should be validated by the designers of other kinds of patterns.

Acknowledgements

Authors would like to thank Eduardo Fernandez for his comments on this paper during the process of shepherding at EuroPLOP. We would like to express appreciation to his work on analysis patterns despite of some critical statements about analysis patterns in the paper.

References:

- [F00] Fernandez E.B., Stock Manager: An Analysis Pattern for Inventories, *Proceedings of PloP 2000*
- [FY00] Fernandez, E. B., Yuan, X., Brey, S. Analysis Patterns for the Order and Shipment of a Product, *Proceedings of PLoP 2000*.
- [FY99] Fernandez E.B., Yuan X. An analysis pattern for reservation and Use of Reusable Entities, *Proceedings of PloP 1999*.
- [FY01] Fernandez E.B., Yuan X. An analysis pattern for the repair of an entity, *Proceedings of PloP 2001*
- [FY00b] E.B. Fernandez and X. Yuan, Semantic analysis patterns, Procs. of 19th Int. Conf. on Conceptual Modeling, ER2000, 183-195.
- [FN05] Filkorn R., Navrat P., An approach for integrating analysis patterns and feature diagrams into Model Driven Architecture, LNCS 2281, 2005.
- [F97] Fowler M., Analysis Patterns: Reusable Object Models, Addison-Wesley, 1997.
- [HF04a] Hamza H.S., Fayad M.E., The Negotiation Analysis Pattern, *Proceedings of PloP 2004*
- [HF04b] Hamza H., Fayad M.E., Applying Analysis Patterns Through Analogy: Problems and Solutions: in Journal of Object Technology, 3,4 April 2004
- [H01] Hahsler M., Software Engineering with Analysis Patterns, Technical Report of WU-Wien, 2001
- [IIBA] International Institute of Business Analysis, <http://www.theiiba.org/am/>
- [RUP] IBM Rational Unified Process, Academic Initiative Repository.
- [UML] OMG Unified Modeling Language, v 2.1.1., www.uml.org.
- [P] Pattern Languages of Programs (PLoP), <http://hillside.net/plop/>

- [OMT] Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W., "Object-oriented modeling and design.", Prentice Hall Int., 1991.
- [S00] Sesera L., A Recurring Fulfilments Analysis Pattern, *Proceedings of PloP 2000*.
- [SF04] Sorgente T., Fernandez E.B., "Analysis patterns for patient treatment, *Proceedings of PLoP 2004*.
- [SFL05] Sorgente T., Fernandez E.B., Larrondo-Petrie M.M., "The SOAP pattern for medical charts", in *Proceedings of the 12th Pattern Languages of Programs Conference (PLoP2005)*.