Experiments on Pattern Language-Based Modeling

Rosana T. V. Braga[1]
Fernao S. R. Germano
Paulo Cesar Masiero[2]
{rtvb,fernao,masiero}@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo – Campus de São Carlos

Abstract

The GRN pattern language can be used to guide the analysis of applications in the business resource management domain. The result is an analysis model for the application, showing the patterns applied to model it and allowing the use of a tool to automatically obtain the final application code. In this paper we describe a series of experiments conducted to evaluate the quality of the analysis models produced using the GRN pattern language compared to the analysis models produced by using an ad-hoc approach. Three experiments were performed with different kinds of subjects: undergraduate students, graduate students, and information technology professionals. The results have shown that the analysis models produced using the GRN pattern language have less errors than those produced using an ad-hoc system analysis approach.

Keywords: *System analysis, Software reuse, pattern languages.*

1 Introduction

Software reuse aims at promoting the reuse of artifacts in higher abstraction levels, as for example analysis artifacts. Recent studies in software reuse point to the relevance of patterns in software development, mainly during the design phase, where proven solutions can be used to solve common problems found [14]. Software patterns try to capture the experience acquired during software development and synthesize it in a problem/solution form [11]. Other researches have been conducted claiming that, more than isolated patterns, pattern languages provide a way to organize the knowledge about a specific domain into specific patterns that can be systematically applied in the development of systems for the same domain [20, 6]. So the patterns of a pattern language represent the temporal sequence of decisions that lead to the complete design of an application, becoming a method to guide the development process [6].

We argue that, if there is a pattern language for a specific domain, applications can be modelled using it, producing results that are more complete and contain less errors if compared to the models produced using conventional approaches such as, for example,

[1]Financial support from FAPESP Process n. 98/13588-4.
[2]Financial support from FAPESP/CNPq.

object-oriented methods supported by UML [17]. We are also motivated by the fact that a framework can be built based on the pattern language, so that the final applications can be more easily built by mapping the patterns applied to model the application into the framework. Another motivation is the existence of a tool that automatically does this mapping. The framework construction and instantiation, as well as the tool construction and use, are out of the scope of this paper and are shown in previous works [1, 4, 5, 3].

In this paper we describe three experiments performed to evaluate the usefulness of the GRN pattern language to model applications in the business resource management domain, in terms of number of errors done and time spent in modeling. The paper is organized as follows. Section 2 shows some related works concerning software reuse during the analysis phase. Section 3 presents the GRN pattern language. Section 4 describes the three experiments conducted and their results. Section 5 presents the concluding remarks.

2 Related Work

There are several conventional approaches for object-oriented analysis, such as OMT [18], Fusion [8], and RUP [15]. They offer well-defined steps through which an application is modelled and subsequently designed and implemented. However, in all of them the modeling is done from scratch, i.e., the system behavior is discovered in an incremental way, through several iteration cycles. The advantages of using such approaches is that any type of system can be modelled, regardless of its domain, complexity or size.

Domain analysis techniques have been proposed in the last two decades aiming at identifying and organizing knowledge about some class of problems [16, 19]. Domain analysis can result in one or more work products to be used in future developments, depending on the technique used, as for example reusable requirements, context models, domain models, feature models, data models, reusable components, taxonomic classification of terms, etc. However, many existing domain analysis approaches focus on establishing domain requirements aiming at producing reusable software, such as frameworks, software components, and application generators, with little interest in reusing analysis artifacts.

Coad [7], Fowler [10], and Hay [12] present numerous analysis patterns that can be used as building blocks to compose complete business models. These patterns can be used in a independent way, i.e., they can be combined in many different ways. However, they are not organized in pattern languages, although this could be done, forming pattern languages for several distinct domains. Coad establishes some strategies for using his patterns, aiming to produce complete applications. Fowler and Hay present some examples of how to use several patterns in a specific system. However, no previous work exists, to the authors knowledge, comparing system analysis done from scratch to system analysis based on patterns.

3 The GRN pattern language

The GRN pattern language [2] (*Gestão de Recursos de Negócios*, in Portuguese) was built based on the experience acquired during development of systems for business resource management. Business resources are assets or services managed by specific applications, as for example videotapes, products or physician time. Business resource management applications include those for rental, trade or maintenance of assets or services.

GRN has fifteen patterns that guide the developer during the analysis of systems for this domain. Figure 1 shows the relationships among these patterns. The first three patterns concern the identification, quantification and storage of the business resource. The next seven patterns deal with several types of management that can be done with these resources, as for example, rental, reservation, trade, quotation, and maintenance. The last five patterns treat details that are common to the seven types of transactions, as for example payment and commissions. All GRN patterns have a structure diagram that uses UML notation. So, each pattern has participant classes, each of them with attributes, methods and operations. Besides, a pattern can have alternative solutions depending on the specific context in which it is applied. So, pattern variants are used to denote each possible solution to the same problem.

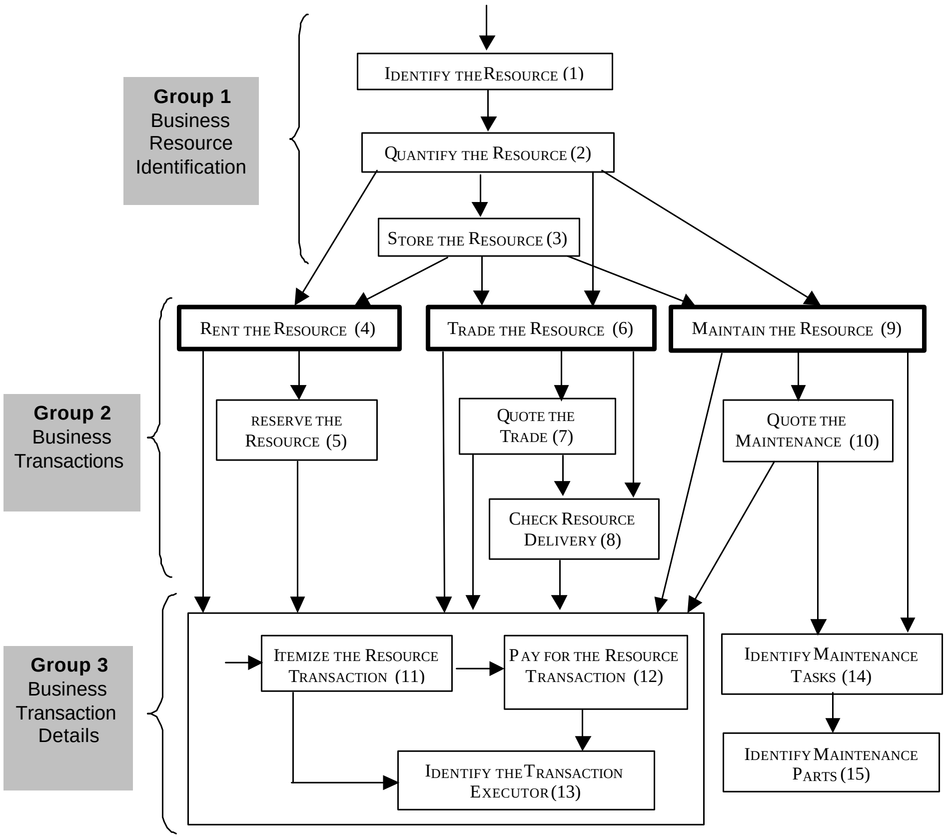

Figure 1: GRN Pattern Language: relationship among patterns

4 The experiments

The three experiments, described below using the structure proposed by Wholin et al. [21], consisted of the analysis of two different systems, although they have a similar difficulty level [3]. The requirements of the systems to be modelled were supplied in advance to the subjects. The first was a hotel management system and the second a car rental system.

[3] the difficulty level has considered the number of requirements and the number of classes of the final system

The task assigned to the subjects was to model both systems in two phases, each of which using a different approach. The first approach, named "*ad hoc* approach", consisted of modeling a system using only their own knowledge about object orientation, supported by the UML notation [17] and by an *ad hoc* process for modeling. The second approach, named "GRN approach", consisted of modeling a system using the GRN pattern language and its corresponding process for system analysis. This process is explained elsewhere [5] and is not shown here due to space constraints. The goal was to compare the time spent and the number of errors made using both approaches.

4.1 E-GRN-1: First Experiment with the GRN pattern language

4.1.1 Experiment Definition

Object of Study: GRN Pattern Language.

Purpose: Evaluate GRN use to model business resource management systems.

Quality Focus: System modeling easiness.

Perspective: System developers in the GRN domain.

Context: The experiment was performed by thirty-five undergraduate students as subjects, divided in twelve groups. The students were provided with the GRN pattern language and the corresponding process to use it. They had a thirteen-hour previous training. The experiment was conducted in August, 2001 and the students were in the eighth semester of the Computer Science Course at ICMC-University of São Paulo.

4.1.2 Planning

Context Selection: The experiment was conducted independently by several groups. There was no communication among them, but they had freedom to determine their own schedule to perform the tasks, as long as they took notes of the exact time spent in a special spreadsheet. The experiments were executed by the students out of the classroom and with no supervision. The students had no previous knowledge of the GRN pattern language. The problem used in the experiment was real, although small. So, the study is valid in a specific Software Engineering domain context.

Hypothesis Formulation

Null hypothesis: There are two hypothesis: NH1 – the time spent to model the system using the GRN approach is similar to or greater than the time spent using the *ad hoc* approach; and NH2 – the number of errors performed in the system modeling using the GRN approach is similar to or greater than the number of errors using the *ad hoc* approach.

Alternative Hypothesis: There are two hypothesis: AH1 – the time spent to model the system using the GRN approach is smaller than the time spent using the *ad hoc* approach; and AH2 – the number of errors performed in the system modeling using the GRN approach is smaller than the number of errors using the *ad hoc* approach.

Variables Selection:

Independent Variables: *Methodology used*: In the first phase, the *ad hoc* approach was used, with half the groups modeling a hotel system and the other half modeling a car rental system. In the second phase, the GRN approach was used, swapping the target system among the groups (groups who have modelled the hotel system now modelled the car rental system and vice-versa); *Students experience and interest area*: the students had previous experience with object-oriented analysis and interest in different Computer Science areas.

Dependent Variables: *total time* spent to model the example system and *number of errors* found in the resulting models, which were corrected based on a basic solution. The errors found were annotated using the classification schema shown in Table 1. Beginning with a maximum grade (10), the correction criteria was to discount points using a grade scale, for example, less 0.25 points for 1 to 5 attribute errors, less 0.5 points for 6 to 10 attribute errors, and so forth. We consider that if the time spent in the modeling is greater and if more errors are found, then the development costs are increased.

Table 1: Types of error considered in the GRN experiments

Id Code	Error Description
#cla	Number of classes created in the model. The student may forget to include a class or may add unnecessary classes. If the student omits a class, then we also count the attributes, methods, relationships, and cardinalities omitted consequently.
#atr	Attributes of each class. The student may forget to include some attributes, may include redundant or unnecessary attributes, or may include attributes in the wrong class.
#met	Methods and operations of each class. The student may forget to include some methods/attributes, may include redundant or unnecessary methods, or may include them in the wrong class.
#rel	Relationship among classes may have been forgotten, may have been placed in the wrong place, or may have an incorrect type (for example, aggregation instead of specialization). If the student omits a relationship, then we also count the cardinalities omitted in consequence.
#car	Cardinality of the relationships. The student may have omitted the cardinality of may have informed a wrong cardinality.

Subjects selection: The subjects were chosen based on convenience (the closest and most convenient people were selected as subjects), as they were students of an undergraduate course. The division of the thirty-five students in twelve groups was not at random, as the students have set up the groups at their will. The distribution of the hotel and car rental system among the groups was at random in the first phase and inverted in the second phase. The students had no choice to decide whether or not to participate, i.e., the experiment was part of a mandatory project in the course. So, it is not possible to guarantee that the results would be the same if the participants were others, for example, professionals assigned in a *ad hoc* way or volunteers.

Experiment Design: Table 2 shows the groups and systems division in the two experiment phases.

Instrumentation: The material supplied to the subjects to perform the experiment was composed of: guidelines to perform the experiment; requirements documents of both systems (hotel and car rental); GRN pattern language document; process for using

Table 2: E-GRN-1 Design (H=Hotel System and CR=Car Rental System)

Group	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Phase 1: Modeling with UML	H	CR	H	CR	H	CR	H	CR	H	CR	H	CR
Phase 2: Modeling with GRN	CR	H	CR	H	CR	H	CR	H	CR	H	CR	H

GRN; Spreadsheets to collect the experiment data (time spent, difficulties found, students categorization, etc).

Validity Evaluation: *Conclusion Validity*: the experiment was elaborated so that the results proved or not the hypothesis, without influence of other external factors. *Internal Validity*: the experiment can be replicated with the same participants and subjects. The lack of communication and competition among participants favors this replication. *External Validity*: it is possible that the results cannot be generalized out of the scope of this study, because the choice was not at random and does not reflect the market conditions. *Construction Validity*: an evaluation survey was applied to the participants, through which it was possible to know their previous experience in the study object domain. So, it was possible to measure the influence of these factors in the results. The fact that students were developing a project that is part of their course, and thus is worth a grade, could be a threat, since they could try to manipulate the information about time spent to improve their grades. However, this fact was not considered, because the students were warned about the importance of the accuracy of this information for the experiment and it was guaranteed to them that the time spent would not influence the final grade.

4.1.3 Operation

Preparation: after making contact with the participants, the necessary instrumentation was arranged: the GRN pattern language was made available on the Web, the training was prepared with slides about the content, and the questionnaires were printed. It was guaranteed to participants that they would remain anonymous.

Participants: The profiles of the participating students are shown in Tables 3 and 4, regarding interest area and experience in the target domain, respectively.

Table 3: Interest area of E-GRN-1 participants

Interest area	**Number of students**	**Percentage**
Networks/Distributed Systems	4	11%
Databases	6	17%
Artificial Intelligence	6	17%
Software Engineering/Information Systems	3	9%
Computer Graphics	6	17%
Hipermedia	9	26%
Hardware	1	3%
Total of students	**35**	**100%**

Execution: The experiment was conducted in two phases: the first was preceded by a one-hour training about UML, as the students already had knowledge about object-oriented development. The tasks were delivered to the students and consisted of modeling the system using the *ad hoc* process. They had to deliver the system

Table 4: Students Experience – E-GRN-1

Experience in the target domain	Number of students	Percentage
No experience	5	14%
Developed projects during undergraduate courses using structured analysis	9	26%
Developed projects during undergraduate courses using using object-oriented analysis	20	57%
Developed, professionally, up to 3 projects in this domain	1	3%
Total of students	**35**	**100%**

class diagram, containing the classes (with attributes, methods and operations) and the relationships among classes (with corresponding cardinalities). The participants received special forms to be filled in during the experiment. They have delivered the results of the first phase after two weeks. Then the second phase has taken place. A twelve-hour training was done about the GRN pattern language. The task assigned to them was to model the other system (inverted in relation to the first phase) aided by GRN. The group had to deliver the system class diagram, similarly to the first phase, but they should annotate, for each participant class, the role played by it in the applied pattern. Each group received a form to take notes of time spent and difficulties found during the second phase. The due date was also set to two weeks.

Data Validation: The forms distributed to participants were checked to guarantee that they were correctly filled in. Several questions were asked to participants to ensure that they have followed the suggested recommendations. All of them participated on the experiment in a responsible manner, so none of them was discarded.

4.1.4 Analysis and Interpretation

Table 5 shows the results obtained in the first phase of E-GRN-1, which consisted of the system analysis using the *ad hoc* approach. Table 6 shows the results obtained in the second phase of E-GRN-1, which consisted of the system analysis using GRN. They show the time spent (in persons/hour), errors found (following the criteria shown in Table 1) and the final grade obtained by the several groups. The final grade was computed without considering the time spent, but only the number of errors done.

Comparing the averages presented in Table 5, we observe that the average grade of the odd groups is approximately the same as that of the even groups. This helps us to confirm (although it does not prove) that the two systems have the same difficulty level, as the students profile is rather similar and the groups were homogeneously divided. Comparing the overall averages of Tables 5 and 6, we notice an increase both in the groups final grade and in the hours spent in the modeling. On the other hand, we notice a decrease in the global number of errors made.

Figures 2 and 3 graphically illustrate some information contained in Tables 5 and 6. In particular, in these figures it was considered the time (in persons/hour) spent to model the system and the number of errors done, which constitutes the alternative hypothesis AH1 and AH2.

The statistical analysis of the results was done using two types of tests: the Wilcoxon post-sum non-parametric test for comparing two independent samples and the Mann Whitney non-parametric test for comparing two independent samples [13, 9]. The ex-

Table 5: E-GRN-1 Results – *Ad hoc* Approach

Group	Persons/hour	#cla	#atr	#met	#rel	#car	Total errors	Grade
Hotel System								
1	3,00	1,00	26,00	15,00	10,00	19,00	71	3,75
3	15,00	1,00	20,00	7,00	9,00	16,00	53	4,25
5	3,50	0,00	14,00	2,00	0,00	0,00	16	8,75
7	13,00	1,00	20,00	6,00	8,00	13,00	48	5,75
9	3,80	1,00	17,00	19,00	5,00	26,00	68	4,25
11	7,50	0,00	15,00	15,00	4,00	10,00	44	6,50
Average	**7,63**	**0,67**	**18,67**	**10,67**	**6,00**	**14,00**	**50**	**5,54**
Car Rental System								
2	6,00	1,00	14,00	3,00	4,00	8,00	30	7,50
4	3,00	1,00	7,00	2,00	2,00	8,00	20	8,25
6	4,00	3,00	29,00	17,00	12,00	25,00	86	2,00
8	4,50	1,00	21,00	6,00	6,00	14,00	48	5,75
10	5,50	1,00	14,00	4,00	6,00	10,00	35	6,75
12	4,00	13,00	24,00	19,00	9,00	20,00	85	1,00
Average	**4,50**	**3,33**	**18,17**	**8,50**	**6,50**	**14,17**	**50,67**	**5,21**
Overall Average	**6,07**	**2,00**	**18,42**	**9,58**	**6,25**	**14,08**	**50,33**	**5,38**

Table 6: E-GRN-1 Results – GRN Approach

Group	Persons/hour	#cla	#atr	#met	#rel	#car	Total errors	Grade
Car Rental System								
1	6,00	3,00	14,00	8,00	4,00	26,00	55	5,00
3	12,00	0,00	11,00	10,00	1,00	3,00	25	7,50
5	3,80	1,00	8,00	2,00	2,00	3,00	16	8,50
7	12,00	2,00	19,00	4,00	4,00	26,00	55	5,50
9	10,00	1,00	19,00	5,00	4,00	8,00	37	6,75
11	3,00	3,00	18,00	11,00	3,00	8,00	43	5,00
Average	**7,80**	**1,67**	**14,83**	**6,67**	**3,00**	**12,33**	**38,5**	**6,38**
Hotel System								
2	7,50	1,00	17,00	5,00	2,00	8,00	33	7,00
4	12,00	2,00	6,00	2,00	0,00	1,00	11	8,50
6	9,00	2,00	12,00	0,00	4,00	6,00	24	7,50
8	12,00	1,00	10,00	1,00	0,00	1,00	13	8,75
10	6,70	0,00	10,00	2,00	4,00	6,00	22	8,25
12	5,50	1,00	8,00	11,00	4,00	2,00	26	7,00
Average	**8,78**	**1,17**	**10,50**	**3,50**	**2,33**	**4,00**	**21,5**	**7,83**
Overall Average	**8,29**	**1,42**	**12,67**	**5,08**	**2,67**	**8,17**	**30**	**7,10**

periment was planned so as to obtain two independent samples: Set 1 (Hotel System), consists of performing the *Ad hoc* approach by the odd groups *versus* the GRN approach by the even groups; and Set 2 (Car Rental System), consists of performing the *Ad hoc* approach by the even groups *versus* the GRN approach by the odd groups.

We have considered as statistically meaningful the comparisons whose `p-value` is less than 0,05. Table 7 shows the results obtained, corresponding to the four comparisons shown in Figures 2 and 3. The tests were inconclusive for the "number of persons/hour" criteria in both systems (AH1) and for the number of errors done in the car rental system (AH2). For the number of errors done in the hotel system, the tests were conclusive, i.e., we can say that the students using the GRN approach to model the hotel system have done less errors than the students that used the *ad hoc* approach in the analysis of this same system (AH2 hypothesis).

Besides the quantitative analysis of the results shown above, a qualitative analysis could be done based on the testimony of two of the twelve groups of students that performed the experiment. According to one of them, the system analysis was easier and more complete using the GRN approach, although the time spent has been greater, due

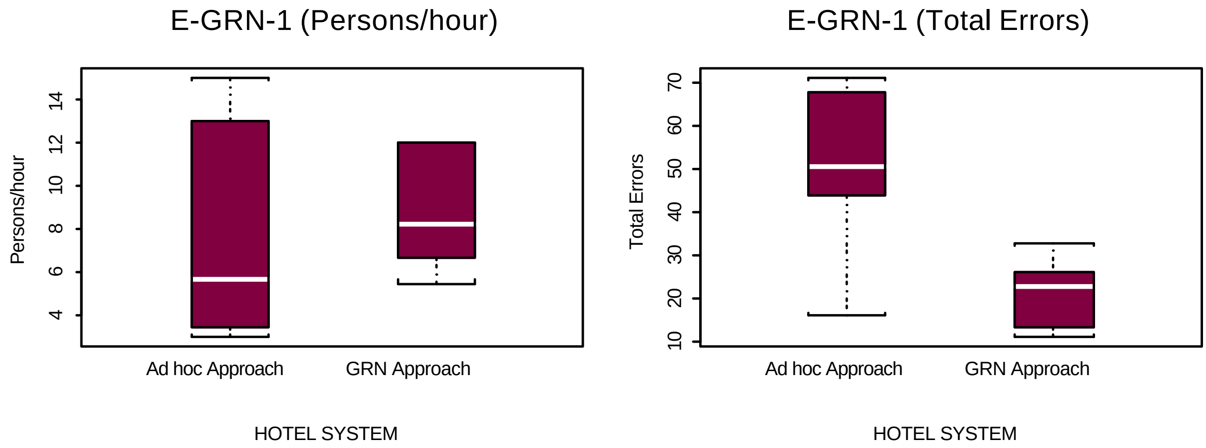

Figure 2: Graphical representation of E-GRN-1 results in the two approaches for the Hotel System

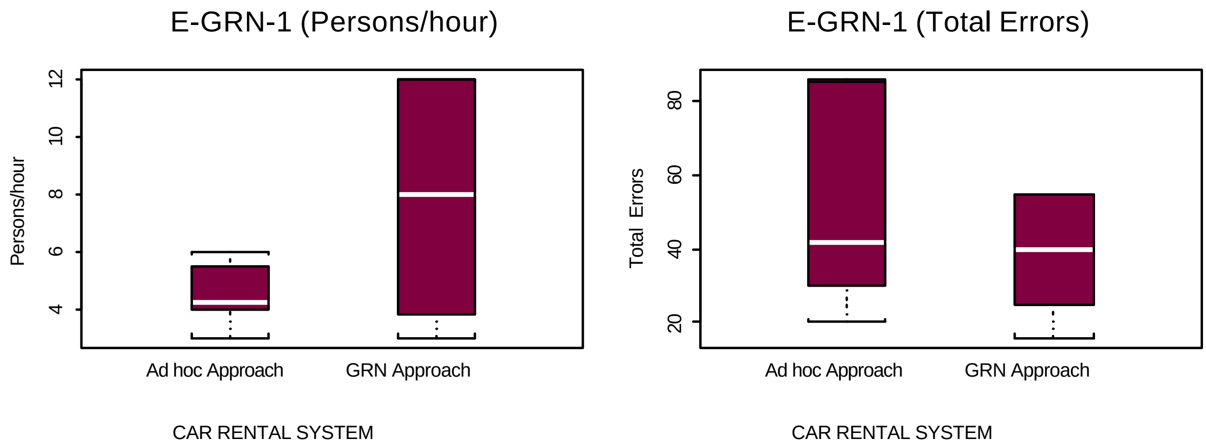

Figure 3: Graphical representation of E-GRN-1 results in the two approaches for the Car Rental System

Table 7: E-GRN-1 Statistical Results

Data sets	**Criteria**	**P-value – Wilcoxon**	**P-value – Mann Whitney**	**Result**
Set 1 (Hotel System)	Persons-hour	0,4821	0,5887	inconclusive
	Total errors	0,0306	0,0261	conclusive
Set 2 (Car Rental System)	Persons-hour	0,2946	0,3095	inconclusive
	Total errors	0,6884	0,6991	inconclusive

to the lack of experience in using GRN. The other group said that the analysis using GRN implied in a meaningful increase in the number of details to be taken care of, causing a greater number of hours spent in modeling. At the same time, this has given a positive feeling to the group and they think that the model would be faulty if they had used only the *ad-hoc* approach, i.e., they would probably have forgotten some important details during the analysis.

These two testimonies reinforce the results obtained, justifying the increased time spent and the greater final grade. The increased time spent does not necessarily imply decrease productivity, as a better grade reflects quality improvement in the model obtained, due to the smaller number and type of errors done. Thus, although in a longer time, we obtain a model that is closer to the final model to be designed and implemented. Yet, according to some students testimony, this time can be improved as they gain experience in the use of GRN.

4.2 E-GRN-2: Second Experiment with the GRN pattern language

4.2.1 Experiment Definition

Object of Study, Purpose, Quality Focus, and Perspective: Equal to E-GRN-1.

Context: the experiment was performed by nine graduate students as subjects. They have received the GRN pattern language, a process for using it, and a previous six-hour training. The experiment was done in October, 2001 and the students were attending the course "Topics in Software Engineering" at the Computer Science Department of the Federal University of São Carlos.

4.2.2 Planning

Context Selection and Hypothesis Definition: similar to E-GRN-1.

Variables Selection:

Independent Variables: *Methodology used*: In the first phase, the *ad hoc* approach was used to model the hotel system by all the students, while in the second phase the GRN approach was used to model the car rental system, also by all the students.

Dependent variables: similar to E-GRN-1.

Subjects selection: The students were chosen similarly to E-GRN-1. However, the hotel system was randomly chosen to be modelled using the *ad hoc* process, and the car rental system to be modelled using the GRN approach.

Experiment Design: Table 8 shows the division of groups and systems in the two experiment phases.

Table 8: E-GRN-2 Design (H=Hotel System and CR=Car Rental System)

Student	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Phase 1: Modeling with UML	H	H	H	H	H	H	H	H	H
Phase 2: Modeling with GRN	CR	CR	CR	CR	CR	CR	CR	CR	CR

Instrumentation and Validity Evaluation: similar to E-GRN-1.

4.2.3 Experiment Operation

Preparation similar to E-GRN-1.

Participants: Six of the nine participating students are interested in the area of Software Engineering and three in databases. Concerning their practical experience in the information systems domain, eight of them have developed projects during undergraduate or graduate courses (four of which using structured analysis and four using object-oriented analysis), and one of them has developed some systems professionally.

Execution: similar to E-GRN-1, except that students worked individually in this experiment, rather than in groups.

Data Validation: similar to E-GRN-1.

4.2.4 Analysis and Interpretation

Table 9 shows the results obtained in the first phase of E-GRN-2 (using the *ad hoc* approach) and Table 10 shows the results obtained in the second phase (using the GRN approach). A straight observation of these results indicates that the time spent to model the system was slightly greater in the GRN approach, as opposed to the number of errors, which was smaller in the GRN approach. We do not show the results graphically, due to lack of space.

Table 9: E-GRN-2 results – *Ad hoc* approach

Students	**Persons/hour**	**#cla**	**#atr**	**#met**	**#rel**	**#car**	**Total errors**	**Grade**
Hotel System								
1	6,40	3,00	12,00	2,00	6,00	11,00	34	5,75
2	3,00	0,00	14,00	5,00	3,00	4,00	26	7,75
3	2,50	0,00	24,00	7,00	11,00	10,00	52	5,50
4	1,00	2,00	23,00	7,00	7,00	13,00	52	5,00
5	3,00	1,00	5,00	4,00	2,00	3,00	15	8,50
6	3,00	3,00	30,00	7,00	8,00	14,00	62	4,50
7	3,00	0,00	12,00	0,00	2,00	4,00	18	8,50
8	3,50	0,00	3,00	0,00	1,00	6,00	10	9,00
9	2,20	2,00	16,00	4,00	5,00	6,00	33	6,50
Average	**3,07**	**1,22**	**15,44**	**4,00**	**5,00**	**7,89**	**33,56**	**6,78**

Table 10: E-GRN-2 results – GRN approach

Student	**Persons/hour**	**#cla**	**#atr**	**#met**	**#rel**	**#car**	**Total errors**	**Grade**
Car Rental System								
1	6,00	0,00	18,00	0,00	2,00	4,00	24	8,00
2	3,70	2,00	13,00	2,00	4,00	5,00	26	7,25
3	5,00	3,00	10,00	3,00	6,00	4,00	26	7,00
4	2,00	1,00	12,00	5,00	4,00	4,00	26	7,50
5	3,50	1,00	7,00	2,00	2,00	2,00	14	8,50
6	3,00	3,00	11,00	4,00	2,00	4,00	24	7,00
7	2,00	2,00	8,00	7,00	5,00	4,00	26	7,25
8	3,30	2,00	21,00	8,00	3,00	6,00	40	6,00
9	3,70	1,00	11,00	10,00	2,00	4,00	28	7,25
Average	**3,58**	**1,67**	**12,33**	**4,56**	**3,33**	**4,11**	**26**	**7,31**

The statistical analysis of the results was done using the Wilcoxon non-parametric test (paired) for comparing two co-related samples [13, 9], as this experiment was planed to allow only the comparison between the hotel system and the car rental system, performed by the same students. The results were inconclusive for both persons-hour (p-value = 0,2349) and number of errors done (p-value = 0,2591).

Some students testimonies can help the results qualitative analysis. A student said that the GRN helps modeling aspects for which they do not have enough domain knowledge. As an example, he mentioned that the PAY FOR THE RESOURCE TRANSACTION pattern helped him to model payment aspects, for which he had not enough experience. Like in E-GRN-1, in this experiment we also had students with difficulties in applying GRN for the first time, but they also said that it would be more productive in subsequent applications. This difficulty can be due to the short time dedicated to training, which was about six hours, compared to the E-GRN-1 thirteen-hour. Several errors were done by students when applying GRN patterns, causing the inclusion of undesired classes or the omission of important classes to model the application. So, a lesson learned from this experiment is that we have to guarantee that participants have understood well the GRN patterns and how to use them (for example, at least an exercise of modeling a system using GRN must be included in the training). Another student commented that he felt assurance when modeling using GRN, mainly when defining classes, attributes and relationships. Actually, several students that had difficulty to model relationships among classes in the first phase of the experiment had a reduction of more than 50% in the number of this type of errors in the second phase.

4.3 E-GRN-3: Third Experiment with the GRN pattern language

4.3.1 Experiment Definition

Object of Study, Purpose, Quality Focus, and Perspective: Equal to E-GRN-1.

Context: the experiment was performed in November, 2001 by twenty information technology (IT) professionals, divided in 6 groups. They have received the GRN pattern language, a process for using it, and a previous three-hour training. The time spent with training was reduced due to the circumstances under which the experiment was done: the students had only one day available (about 8 hours in total). The students participating in the experiment were doing a specialization course in "Information Technology" at State University of São Paulo (UNESP) in Presidente Prudente. It is important to notice that the students were already working as IT professionals, as can be seen in Table 12.

4.3.2 Planning

Context Selection, Hypothesis Definition and Variables Selection: similar to E-GRN-1.

Subjects Selection: similar to E-GRN-1, but there were twenty students, divided in six groups of three or four students each.

Experiment Design: Table 11 shows the division among groups and systems in the two experiment phases.

Table 11: E-GRN-3 Design (H=Hotel System and CR=Car Rental System)

Group	**1**	**2**	**3**	**4**	**5**	**6**
Phase 1: Modeling with UML	H	CR	H	CR	H	CR
Phase 2: Modeling with GRN	CR	H	CR	H	CR	H

Instrumentation and Validity Evaluation: similar to E-GRN-1.

4.3.3 Experiment Operation

Preparation: similar to E-GRN-1.

Participants: Although twenty students had initially participated in the experiment, two groups (G3 and G4) abandoned the experiment in the second phase, so it was necessary to discard them. Among the thirteen remaining participants, seven had interest in the area of software engineering/information system. Concerning their practical experience, five of them had concrete experience in the development of systems in the GRN domain. The professional activity performed by the participants can be seen in Table 12.

Table 12: Students Jobs (E-GRN-3)

Present Job	**G1**	**G2**	**G5**	**G6**	**Total**	**%**
Network Administrator	1				1	8%
System Analyst	1	1			2	15%
Professor	2	1	3		6	46%
Programmer				2	2	15%
Manager / Chief		1			1	8%
Director Assistant				1	1	8%
Total	**4**	**3**	**3**	**3**	**13**	**100%**

Execution: similar to E-GRN-1, except regarding the groups organization, as each group had three to four people, and regarding the time to deliver the results: the first phase was done in the morning, with an one-hour training about UML followed by the groups meeting to model the system; the class diagram was delivered before lunch; the training for the second phase was done in the same day, in the afternoon; and the second task was assigned to the students to be delivered in two weeks.

Data Validation: Two groups did not deliver the second part of the project (modeling using the GRN approach). So, these groups were discarded. The evaluation forms distributed to the remaining participants were checked to see if they were correctly filled in. Some questions were done to ensure that they had followed the recommendations.

4.3.4 Analysis and Interpretation

Table 13 shows the results obtained in the first phase of E-GRN-3 (*ad hoc* approach), while Table 14 shows the results obtained in the second phase of E-GRN-3 (GRN approach). A simple observation of these results also indicates a greater average time in the second phase and a smaller average number of errors. The statistical analysis was not done due to the smaller number of data sets in this experiment.

Table 13: E-GRN-3 Results – *Ad hoc* approach

Group	Persons/hour	#cla	#atr	#met	#rel	#car	Total errors	Grade
Hotel System								
1	7,00	1,00	21,00	11,00	3,00	4,00	40	6,00
5	6,00	1,00	13,00	15,00	6,00	16,00	51	4,75
Average	**6,50**	**1,00**	**17,00**	**13,00**	**4,50**	**10,00**	**45,5**	**5,38**
Car Rental System								
2	5,70	1,00	4,00	2,00	2,00	7,00	16	8,50
6	8,00	1,00	17,00	11,00	5,00	8,00	42	5,75
Average	**6,85**	**1,00**	**10,50**	**6,50**	**3,50**	**7,50**	**29**	**7,13**
Overall Average	**6,68**	**1,00**	**13,75**	**9,75**	**4,00**	**8,75**	**37,25**	**6,25**

Table 14: E-GRN-3 results – GRN approach

Group	Persons/hour	#cla	#atr	#met	#rel	#car	Total errors	Grade
Car Rental System								
1	8,00	1,00	8,00	9,00	3,00	5,00	26	7,50
5	6,00	2,00	9,00	3,00	4,00	8,00	26	7,75
Average	**7,00**	**1,50**	**8,50**	**6,00**	**3,50**	**6,50**	**26**	**7,63**
Hotel System								
2	8,40	1,00	17,00	0,00	2,00	3,00	23	8,25
6	10,50	9,00	28,00	6,00	10,00	9,00	62	3,50
Average	**9,45**	**5,00**	**22,50**	**3,00**	**6,00**	**6,00**	**42,5**	**5,88**
Overall Average	**8,23**	**3,25**	**15,50**	**4,50**	**4,75**	**6,25**	**34,25**	**6,75**

Comparing this experiment to E-GRN-1 and E-GRN-2, we notice that the GRN approach had less advantages and more difficulties. Although the modeling time per person had been approximately the same in the three experiments, the number of errors done by students was apparently greater in E-GRN-3. Again, as explained in the evaluation of E-GRN-2, this difficulty of the students can be attributed to the short time dedicated to training (about three hours).

Another issue that can be discussed here is that the usefulness of using GRN is influenced by the experience of its users. There seems to be a trend that GRN is more useful to unexperienced developers than to those that already have some practice in the domain. This can be justified by the fact that experienced developers, even unconsciously, have in their mind some informal patterns of solutions that they have used before, i.e., the patterns are structures that they have learned to use intuitively and they are part of their personal experience. So, they prefer to do the modeling directly, instead of using patterns with which they are not familiar. They need time and additional training to adapt to these formal new patterns. A result that helps enforcing this hypothesis is that there are more wrong classes in the results obtained with the GRN approach, both in E-GRN-2 and E-GRN-3, which have more experienced participants. By analyzing the resulting models it is possible to see that the students have included classes that were not part of the requirements just because they were part of the patterns or they have omitted some classes because they have applied the wrong pattern.

5 Concluding remarks

The experiments indicated the usefulness of the GRN pattern language to model applications in the business resource management domain. There is an apparent tendency that the use of pattern languages is more helpful to less experienced developers, for whom the number of errors done using the GRN approach was significantly smaller than using the *ad hoc* approach. The other hypothesis established in the experiments, which concerned the time spent to model a system using the different approaches, was not confirmed, as the participants spent more time using the GRN approach. However, this can be attributed to the short time dedicated to the participants training. As the three experiments were conducted almost in parallel, we only detected this problem after concluding them. To better evaluate the hypothesis, future experiments should be conducted, extending the training to guarantee that participants know the pattern language satisfactorily, thus measuring productivity for trained analysts, rather than first-time use analysts.

Several problems were identified in the GRN pattern language during the experiments. This allowed the modification of GRN patterns to improve them, easing their use in future experiments. Besides, the goal is to make GRN self-contained, so that it can be used with no training. This conforms to a desirable characteristic of patterns: the solution has to be documented in such a way that developers, other than the pattern author, are able to understand and apply it by their own. This is why, in the pattern Conferences, papers are discussed in more than one-hour sessions, in which the pattern author has to remain quiet for almost all the time. In this way, he or she can observe other persons view about the proposed pattern, and can modify the points that are obscure to readers.

Associated to GRN there is a framework, named GREN, which was developed to support the implementation of applications modelled using GRN. There is also a tool, named GREN-Wizard [3], to support the GREN instantiation. It was designed so that framework users need only to know GRN in order to obtain their specific applications. So, applications modelled using GRN can be automatically implemented using the GREN-Wizard. Both GREN and GREN-Wizard were subjects of other previous works [1, 3, 4, 5] and are out of the scope of this paper. E-GRN-1 was followed by another experiment, in which the students used GREN and GREN-Wizard to implement the hotel system. The development using GREN-Wizard took approximately half an hour, while the manual instantiation using the GREN white-box version took approximately 10 hours.

References

[1] R. T. V. Braga. *A Process for Construction and Instantiation of Frameworks Based on a Domain-Specific Pattern Language (in Portuguese)*. PhD thesis, ICMC-USP, Sao Carlos – SP, February 2003.

[2] R. T. V. Braga, F. S. R. Germano, and P. C. Masiero. A pattern language for business resource management. In *6th Pattern Languages of Programs Conference (PLoP'99)*, Monticello – IL, USA, 1999.

[3] R. T. V. Braga and P. C. Masiero. GREN-Wizard: a tool to instantiate the GREN framework. In *Tools Session at 16th Brazilian Symposium on Software Engineering (SBES 2002)*, pages 408–413, Gramado-RS, October 2002.

[4] R. T. V. Braga and P. C. Masiero. A process for framework construction based on a pattern language. In *Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC)*, pages 615–620, IEEE Computer Society, Oxford-England, September 2002.

[5] R. T. V. Braga and P. C. Masiero. The role of pattern languages in the instantiation of object-oriented frameworks. *Lecture Notes on Computer Science*, 2426-Advances in Object-Oriented Information Systems:122–131, sep 2002.

[6] D. Brugali, G. Menga, and A. Aarsten. *A Case Study for Flexible Manufacuring Systems*, pages 85–99. Domain-Specific Application Frameworks: Frameworks Experience by Industry, M. Fayad, R. Johnson, –John Willey and Sons, 2000.

[7] P. Coad, D. North, and M. Mayfield. *Object Models: Strategies, Patterns and Applications*. Yourdon Press, 2 edition, 1997.

[8] D. Coleman et al. *Object Oriented Development – the Fusion Method*. Prentice Hall, 1994.

[9] W. J. Conover. *Practical Nonparametric Statistics – 2a. ed.* Wiley, New York, 1980.

[10] M. Fowler. *Analysis Patterns*. Addison-Wesley, 1997.

[11] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1994.

[12] D. Hay. *Data Model Patterns – Conventions of Thought*. Dorset House Publishing, New York - NY, USA, 1996.

[13] M. Hollander and D. A. Wolfe. *Nonparametric Statistical Methods*. John Wiley, New York, USA, 1973.

[14] R. E. Johnson. Documenting frameworks using patterns. In *OOPSLA '92*, pages 63–76, 1992.

[15] P. Kruchten. *The Rational Unified Process: An Introduction*. Addison-Wesley, 2000.

[16] R. Prieto-Diaz and G. Arango. *Domain Analysis and Software System Modeling*. IEEE Computer Science Press Tutorial, 1991.

[17] C. Rational. Unified Modeling Language, 2000. Disponvel na URL: `http://www.rational.com/uml/references`.

[18] J. Rumbaugh et al. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.

[19] W. Schafer, R. Prieto-Diaz, and M. Matsumoto. *Software Reusability*. Ellis Horwood, 1994.

[20] H. A. Schmid. Systematic framework design by generalization. *Communications of the ACM*, 40(10):48–51, 1997.

[21] C. Wholin, P. Runeson, M. Hst, M. Ohlsson, B. Regnell, and A. Wessln. *Experimentation in Software Engineering - An introduction*. Kluwer Academic Publishers, 2000.