Evaluating the Effect of Composite States on the Understandability of UML Statechart Diagrams

José A. Cruz-Lemus[1], Marcela Genero[1], M. Esperanza Manso[2] and Mario Piattini[1]

[1]ALARCOS Research Group, Department of Computer Science
University of Castilla – La Mancha
Paseo de la Universidad, 4 13071 Ciudad Real (Spain)
{JoseAntonio.Cruz, Marcela.Genero, Mario.Piattini}@uclm.es
[2]GIRO Research Group, Department of Computer Science
University of Valladolid
Campus Miguel Delibes, E.T.I.C. 47011 Valladolid (Spain)
manso@infor.uva.es

Abstract. UML statechart diagrams have become an important technique for describing the dynamic behavior of a software system. They are also a significant element of OO design, especially in code generation frameworks such as Model Driven Architecture (MDA). In previous works we have defined a set of metrics for evaluating structural properties of UML statechart diagrams and have validated them as early understandability indicators, through a family of controlled experiments. Those experiments have also revealed that the number of composite states had, apparently, no influence on the understandability of the diagrams. This fact seemed a bit suspicious to us and we decided to go a step further. So in this work we present a controlled experiment and a replication, focusing on the effect of composite states on the understandability of UML statechart diagrams. The results of the experiment confirm, to some extent, our intuition that the use of composite states improves the understandability of the diagrams, so long as the subjects of the experiment have had some previous experience in using them. There are educational implications here, as our results justify giving extra emphasis to the use of composite states in UML statechart diagrams in Software Engineering courses.

1. Introduction

Modeling is at the core of many disciplines, but it is especially important in engineering because it facilitates the communication and construction of complex things from smaller parts [14]. Models help us understand a complex problem and its potential solutions through abstraction. It seems obvious, therefore that software systems, which are often among the most complex of all engineering systems, can benefit greatly from using models and modeling techniques [12]. Over the last three decades, the abstraction level has not only risen from implementation over design to analysis; there is also a recent interest in code generation frameworks such as the Model Driven Architecture (MDA) [9] proposed by the Object Management Group (OMG). To the extent that code generation is used, it seems likely that factors which

L. Briand and C. Williams (Eds.): MoDELS 2005, LNCS 3713, pp. 113-125, 2005.

influence evolvability on the implementation level, such as the naming of variables and a badly structured program code, will become less relevant. Hence, in this context, the evolvability of information systems would be more and more determined by that of the models [15].

Linked to the idea of models which are capable of evolution, UML statechart diagrams have become an important technique for describing of the dynamic aspects of a software system and are also an important element of OO design documents [4].

According to [12], in order to be useful and effective, an engineering model must possess, to a sufficient degree, the following five key characteristics: abstraction, understandability, accuracy, predictiveness and inexpensiveness .

The motivation for this research comes from the fact that in a previous work [3] we have studied the relationship between many of the constructs of the UML statechart diagrams and the effect that they have on the understandability of the diagrams themselves. To do so, we had previously defined and validated, both theoretically and empirically, a set of metrics [2] for evaluating the structural properties of UML statechart diagrams, based on UML v.1.4 [8]. But in all these works we have found that the effect of composite states on the understandability of the UML statechart diagrams was unclear. A composite state is a state that contains other states within it. When the behavior of a class is quite complicated, using composite states may be useful, as we can join those simple states that are part of a larger common one. Intuitively, grouping into a composite state those that are highly related could help to improve the understandability of a diagram.

In this work we will focus on the evaluation of the effect that a construct of the UML meta-model [8] has on one of the afore-mentioned .characteristics. More specifically, we will evaluate the effect that composite states have on the understandability of UML statechart diagrams, which are of the most commonly used diagrams when modeling using UML and which are part of the main UML diagrams set established in [5].

In order to clarify these impressions, we have designed and performed a controlled experiment and a replication so as to evaluate whether the use of composite states really does improve the understandability of the diagrams, as may be thought intuitively. In this work we will present the experimental process and the conclusion that has been reached after the performance of the experiment.

In section 2, we define our research question and formulate the work hypotheses. Later, we test these hypotheses in the experiment and its replication as reported in section 3. In section 4 we discuss the validity threats to our experiments. Finally, section 5 sets out the conclusions reached and the future work that is planned.

2. Research Question and Hypotheses

As the main goal of the current work is to ascertain if the use of composite states can make the UML statechart diagrams easier to understand, our research question can be stated as:

> *Does the use of composite states improve the understandability of UML statechart diagrams?*

Based on previous experiments [3] and on our intuition and experience working with UML statechart diagrams, we think that the answer to this question should be a ‘yes’, especially when the person that is trying to understand the UML statechart diagram is used to working with this modeling language and this kind of diagram.

In order to evaluate our research question, we carried out a controlled experiment and a replication. In these experiments, we considered the efficiency of the subjects in understanding the diagrams, i.e. the relationship between how accurately they solve the required tasks and how quickly they do this. The *understandability efficiency* was defined as correct answers given by the subjects divided by the time spent on answering the questions related to an UML statechart diagram. This was used to evaluate the property we have previously mentioned: the efficiency of the subjects.

On the basis of our research question we formulated the following experimental hypotheses:

- H_0: the use of composite states does not improve the understandability efficiency of an UML statechart diagram.
- H_1: the use of composite states improves the understandability efficiency of an UML statechart diagram.

3. Experimental Process

In this section, we describe a controlled experiment and a replication that we carried out for testing the hypotheses stated in the previous section. All the experimental process is based on the guidelines outlined in [16].

3.1. First Experiment

This experiment took place at the University of Murcia (Spain) in February 2005. Its main features are the following:

Subjects. 55 Computer Science students from the University of Murcia participated in this experiment.

The tasks to be performed did not require high levels of industrial experience, so experiments with students could be considered as appropriate [1, 6]. Moreover, students are the next generation of people entering this profession , so they are close to the population under study [7]. Besides, working with students implies a set of advantages [15], such as the fact that the prior knowledge of the students is rather homogeneous. The availability of a large number of subjects is another plus point.

All the subjects were in the fourth year of Computer Science and had received a complete Software Engineering course in which they had studied modeling techniques, including UML. They also received a short training session before the performance of the experiment, in which the main constructs of UML statechart diagrams were commented on and where two examples of the tasks to be performed by them were explained by the conductor of the experiment. So we consider that the level of experience they brought to the experiment was acceptable.

Experimental design. We selected a factorial with interaction confounded. Our dependent variable was the understandability of UML statechart diagrams and we would measure this through the previously introduced measure *understandability efficiency*. Our independent variables were the Universe of Discourse (UoD) to which the diagrams were related and the use or not of composite states (CS) in the diagram.

We used two different Universes of Discourses (UoD's): an ATM machine and a phone call. For each of them, we presented two different diagrams, conceptually identical. One of the diagrams included composite state(s) and the other did not.

As each subject would receive two diagrams, one with and another without composite states, and each of them related to a different UoD, we obtained two different groups as shown in Table 1. The diagrams of each group were given to the subjects in different orders. For instance, in group A, the subjects first had to solve the tasks related to an ATM machine without composite states and, after that, those related to a phone call with composite states or exactly the same tasks for the same diagrams but in an inverse order (phone call with composite states and then ATM machine without composite states).

Table 1. Overview of the experimental design

	Universe of Discourse	
	ATM machine	**Phone call**
Without composite states	Group A	Group B
With composite states	Group B	Group A

Group A was performed by 28 subjects and group B 27 subjects.

Experimental task. As commented previously, we used two different UoD's, one modeled the behavior of an ATM machine and the other the behavior of a phone call. These UoD's were quite usual and not exceptional at all, so that there was no need for extra effort in understanding the diagrams.

Each diagram had a test which contained 6 questions which were conceptually similar and set out in the same order. In fact, in both diagrams of each UoD, the questions were the same. The questions inquired about what state would be reached after the triggering of some events which were in a given state. Another question asked which state would be reached after a certain sequence of events and guard conditions. There was a final inquiry as to what sequence was the minimum possible for going from one given state to another. The subjects had to note down the times at which they started and finished answering the questions, as well as providing the answers to the questions themselves.

An example of the experimental material given to the subjects can be found in Appendix A, at the end of the present work.

Experimental procedure. The experiment started with a twenty-five-minute introductory session in which the conductor briefly explained the behavior of the elements of an UML statechart diagram. After that, the materials for the experiment were randomly distributed to the subjects.

In order to increase the motivation and interest of the subjects, they were explained that the exercises that they were going to perform could be similar to those that would find in their exam at the end of the term.

At this point two examples in shortened version were performed by the supervisor, who explained the correct answer to each question and the way of noting down the starting and finishing times properly.

Data analysis and interpretation[1]. First we carried out an analysis of the descriptive statistics of the data. We obtained the results shown in the box-plot of figure 1 and eliminated the extreme and atypical data, obtaining the results displayed in Table 2. In this table, we show the descriptive statistics of the valid data for the diagrams that used composite states and of those that did not.

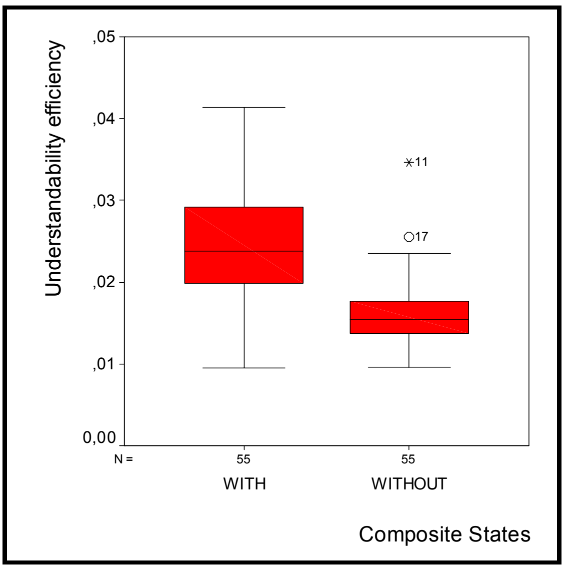

Fig. 1. Box-plot of the data from the first experiment

Table 2. Descriptive statistics of the understandability efficiency (first experiment)

CS	N	Mean	S.E.	Min.	Max.	Skew.	Kurtosis
With	55	0.024165	0.007447	0.00947	0.04138	0.1659	-0.5494
Without	51	0.015269	0.002809	0.00962	0.02151	0.0721	-0.3164

[1] All the data analysis was carried out by means of SPSS [13]

Table 2 shows that these subjects, who were quite familiar with the use of UML statechart diagrams, obtained much better results for efficiency when working with those diagrams that used composite states.

After this, we decided to perform an ANOVA, because this type of analysis allows us to analyze the interaction between the independent variables under study when the measurement of the dependent variable is repeated [10].

The results of the ANOVA which was performed for the understandability efficiency are shown in Table 3. The last column of Table 3 represents the level of significance, which will allow us to reject or accept the hypothesis we have formulated.

Table 3. ANOVA results for understandability efficiency in the first experiment

Source	Sum of Squares	df	Mean Squared	F	Significance level
Subject (Group)	2.137E-03	51	4.190E-05	2.378	0.001
Error	8.632E-04	49	1.762E-05		
UoD	3.555E-04	1	3.555E-04	20.182	0.000
Error	8.632E-04	49	1.762E-05		
CS	1.711E-03	1	1.711E-03	97.133	0.000
Error	8.632E-04	49	1.762E-05		
Group	8.334E-05	1	2.778E-05	0.675	0.572
Error	2.154E-03	52.301	4.119E-05		
Interaction	4.108E-02	1	4.108E-02	1003.084	0.000
Error	2.160E-03	52.732	4.096E-05		

In each row of the table we have the different factors to be taken into account:

– The interaction between the subject and the group of diagrams that he/she has performed.
– The UoD of the diagrams.
– The use of composite states.
– The group of diagrams that the subject has performed (see Table 1).
– The interaction of the factors.

We can observe that there exist several factors whose significance level is below 0.05; hence these affect the understandability efficiency. We do not study the effect of the interaction of factors nor the Group factor as the significance level for this is 0.572 (over 0.05).

We are especially interested in the CS factor, which indicates if a diagram uses this kind of constructor or not. In this case, its value is below 0.05, which implies that the use of composite states affects the understandability efficiency.

In figure 2, we can also observe the profile plot of the data, which indicates that independently of the UoD, using composite states in the diagrams makes the understandability efficiency increase.

Combining the results obtained in Table 2. and figure 2, we can reject the hypothesis H_0, which asserted that the use of composite states did not improve the understandability efficiency of an UML statechart diagram.

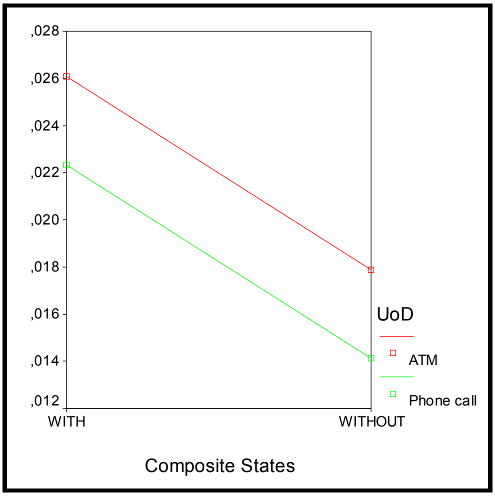

Fig. 2. Understandability efficiency profile plot from the first experiment

3.2. Experiment Replication

This replication took place at the University of Alicante (Spain) in March 2005. As most of its features are similar to those we have commented on before for the first experiment, we will go over only the differences between them:

- In this case the subjects were 178 Computer Science students from the University of Alicante.
- In order to increase the interest and motivation of the subjects, they would be granted with some extra points in the exam at the end of the term. Anyway, they participated voluntarily and some of the students decided not to perform the experiment.
- The skill of the subjects using UML for modeling, especially UML statechart diagrams, was much lower in this replication, as most of them had only a few months of experience, and they had not worked with some UML meta-model constructs (e.g. composite states) yet. They received the same training session as in the original experiment before performing the replication, but even with this, their experience level was much lower, compared to the first group of subjects.
- Due to space limitations in the classrooms where the replication took place, the subjects were divided into two groups of 92 and 86 subjects respectively and they performed the experiment at a different time. To be more specific, the second group finished one hour later, but there was no interaction between the subjects of both groups.
- The materials for the experiment were given out randomly to the subjects and a half of them (89 subjects) performed each possible option (A and B).

Data analysis and interpretation. Again, our first step consisted of an analysis of the descriptive statistics of the data. We obtained the results shown in the box-plot of figure 3. In this case also, we eliminated the extreme and atypical data and obtained the results shown in Table 4.

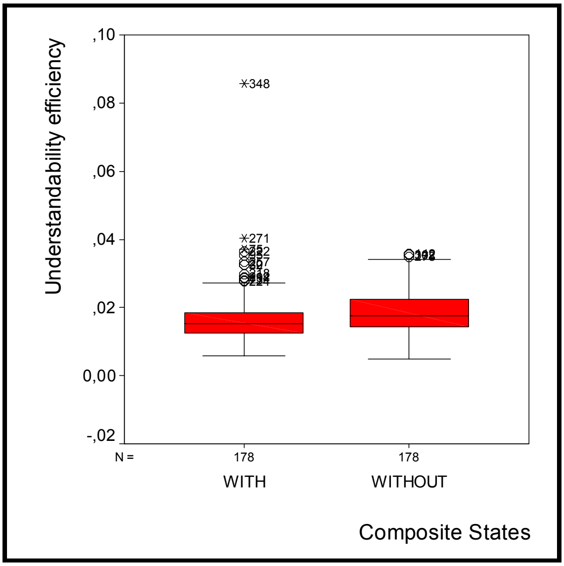

Fig. 3. Box-plot of the replication data

Table 4. Descriptive statistics of the understandability efficiency (replication)

CS	N	Mean	S.E.	Min.	Max.	Skew.	Kurtosis
With	160	0.014956	0.003720	0.00580	0.02449	0.3205	-0.1812
Without	173	0.018106	0.005440	0.00496	0.03109	0.3649	-0.3192

In this case, the results were better for the diagrams which did not use composite states. The lack of experience of the subjects working with this kind of UML diagram was a key factor in obtaining these results. Anyway, although the subjects had scarcely worked with composite states, the difference in the mean values are much smaller than in the case of the first experiment, where the diagrams that used composite states were much more efficiently understood than the others.

In the replication, we also applied an ANOVA and obtained the results shown in Table 5.

Again, we do not study the effect of the interaction of sources nor the Group factor, as the significance level for this is 0.129 and the test power was 0.451. In this case, the value of the factor CS is also below 0.05, as happened in the experiment. So, again in the replication, the results show that using composite states in UML statechart diagrams affects their understandability efficiency. In this case the effect is negative and makes the understandability decrease, but as we have remarked before, this effect is a consequence of the lack of experience that the subjects had.

Table 5. ANOVA results for understandability efficiency in the replication

Source	Sum of Squares	df	Mean squared	F	Significance level
Subject (Group)	3.759E-03	176	2.136E-05	1.649	0.001
Error	1.981E-03	153	1.295E-05		
UoD	1.606E-03	1	1.606E-03	124.044	0.000
Error	1.981E-03	153	1.295E-05		
CS	6.283E-04	1	6.283E-04	48.519	0.000
Error	1.981E-03	153	1.295E-05		
Group	4.827E-05	1	4.827E-05	2.326	0.129
Error	3.994E-03	192.445	2.075E-05		
Interaction	8.737E-02	1	8.737E-02	4210.117	0.000
Error	3.994E-03	192.445	2.075E-05		

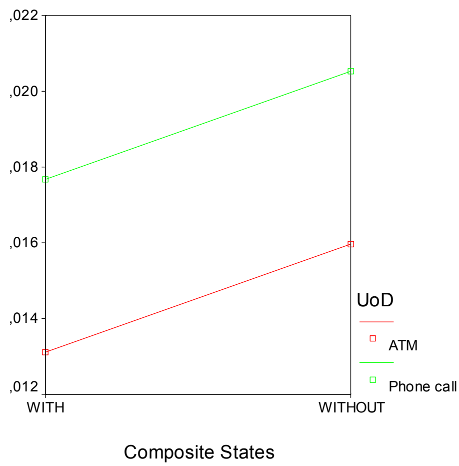

Fig. 4. Understandability efficiency profile plot from the replication

4. Threats to Validity

We must keep in mind a number of validity issues that are typically related to experiments of this type.

First, the subjects were not professional modelers. Obviously, we would expect much better results if the subjects were more experienced. However, the limited difficulty of the tasks and the different UoD's make the students become suitable experimental subjects, as they are much easier to work with than some others.. Nevertheless, further replications of these experiments using people already working in this profession would be really interesting.

Secondly, the diagrams that have been used represent relatively simple models and it is possible that if real-projects data were used, we could obtain different results.

In order to alleviate possible effects of learning and fatigue, we counterbalanced the order in which treatment combinations were given to the subjects; furthermore, the subjects were assigned at random to each possible treatment order sequence. To minimize plagiarism, the experiment conductor encouraged an honest performance of the experiment and was present in the room throughout.

Finally, in order to decrease a possible 'session effect', in the replication the subjects were randomly assigned to the session in which they performed the experimental tasks.

5. Conclusions and Future Work

The appearance of the MDA, and hence the emphasis to be put on the models, has favored that UML statechart diagrams have become an important technique in the describing of the dynamic aspects of a software system.

In previous works [3] we have studied the relationship between many of the constructs of the UML statechart diagrams and the effect that they have on the understandability of the diagrams, based on a set of metrics that we had previously defined and validated [2]. In these works we had found that the effect of the composite states on the understandability of the UML statechart diagrams was not clear. So we designed and performed a controlled experiment and a replication in order to evaluate this effect. The experiment and its replication were carried out by students of two different Spanish Universities. The results obtained show that the use of composite states improves the understandability efficiency of UML statechart diagrams if the subjects have a certain level of experience in working with this kind of UML diagrams. Thus, we can conclude that using composite states when modeling the behavior of systems through UML statechart diagrams makes them more understandable.

These findings give greater justification than ever for putting special emphasis on the use of composite states when teaching UML statechart diagrams in Software Engineering courses.

In spite of these encouraging findings, we considered them to be preliminary. Further validation is needed, to be performed with experienced practitioners, as well as by taking data from real projects. When we have obtained conclusive results about the effect of composite states on the understandability of UML statechart diagrams, we will investigate the optimal nesting level within the composite states.

It could also be interesting testing the hypotheses again but using other experimental design in which the effect of interaction is not confounded, in order to obtain more knowledge about it.

Once UML 2 [11] is adopted as standard by the OMG we will study the metamodel corresponding to the statechart diagrams, in order to find out if the findings presented in the present work are also valid for this version of the language. In addition, we will investigate whether our proposed metrics [2] could be used as maintainability indicators of UML statechart diagrams.

Acknowledgements

This research is part of the MESSENGER project (PCC-03-003-1) financed by ‘Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha (Spain)’, the CALIPO project supported by ‘Dirección General de Investigación del Ministerio de Ciencia y Tecnología (Spain)’ (TIC2003-07804-C05-03) and the DIESEL project (TIN2004-03145), financed by ‘MEC-FEDER’.

The authors would like give their sincere thanks to Professor Ambrosio Toval from the University of Murcia and Professor Cristina Cachero from the University of Alicante for allowing us to perform the above experiments with their students.

References

1. Basili, V., Shull, F. and Lanubile, F.: Building Knowledge through Families of Experiments. IEEE Transactions on Software Engineering, Vol. 25(1999) 456-473
2. Cruz-Lemus, J. A., Genero, M. and Piattini, M.: Metrics for UML Statechart Diagrams. In: Metrics for Software Conceptual Models. Genero, Piattini and Calero (eds.), Imperial College Press, UK (2005)
3. Cruz-Lemus, J. A., Maes, A., Genero, M., Poels, G. and Piattini, M.: Analyzing Data Extracted from a Family of Experiments for Evaluating UML Statechart Diagrams Understandability. Research Working Paper, University of Ghent (to appear) (2005)
4. Denger, C. and Ciolkowski, M.: High Quality Statecharts through Tailored. Perspective-Based Inspections. Proc. of 29th EUROMICRO Conference "New Waves in System Architecture". Belek, Turkey. (2003) 316-325
5. Erickson, J. and Siau, K.: Theoretical and Practical Complexity of UML. Proc. of 10th Americas Conference on Information Systems. New York, USA. (2004) 1669-1674
6. Höst, M., Regnell, B. and Wohlin, C.: Using Students as Subjects - a Comparative Study of Students & Professionals in Lead-Time Impact Assessment. Proc. of 4th Conference on Empirical Assessment & Evaluation in Software Engineering (EASE 2000). Keele, UK. (2000) 201-214
7. Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., El-Emam, K. and Rosenberg, J.: Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Transactions on Software Engineering, 28 Vol. 8. (2002) 721-734
8. Object Management Group: UML Revision Task Force. OMG Unified Modeling Language Specification, v.1.4. document formal/01-09-67. (2001)
9. Object Management Group: MDA - The OMG Model Driven Architecture. (2002)
10. Reynoso, L., Genero, M. and Piattini, M.: Measuring OCL Expressions: An approach based on Cognitive Techniques. In: Metrics for Software Conceptual Models, Genero, Piattini and Calero (eds.), Imperial College Press, UK. (2005)
11. Rumbaugh, J., Jacobson, I. and Booch, G.: The Unified Modeling Language Reference Manual, Second Edition. Addison-Wesley. (2005)
12. Selic, B.: The Pragmatics of Model-Driven Development. IEEE Software, 20 Vol. 5. (2003) 19-25.
13. SPSS: SPSS 11.5, Syntax Reference Guide, Chicago, USA, SPSS Inc. (2002)
14. Thomas, D.: MDA: Revenge of the Modelers or UML Utopia? IEEE Software, 21 Vol. 3. (2004) 15-17
15. Verelst, J.: The Influence of the Level of Abstraction on the Evolvability of Conceptual Models of Information Systems. Proc. of 3rd International Symposium on Empirical Software Engineering (ISESE 2004). Redondo Beach, USA. (2004) 17-26.

16. Wohlin, C., Runeson, P., Hast, M., Ohlsson, M.C., Regnell, B. and Wesslen, A.: Experimentation in Software Engineering: an Introduction. Kluwer Academic Publisher. (2000)

Appendix A. An Example of the Experimental Material

In this appendix we show part of the experimental material handed out to the subjects in the experiments. These two diagrams model a phone call; the first one (figure 5) uses composite states and the second (figure 6) does not. The complete original (in Spanish) material can be found at http://alarcos.inf-cr.uclm.es/

The following text sets out the questions that had to be solved by the experimental subjects. In this study, the questions were the same for both diagrams.

QUESTIONNAIRE (PHONE CALL DIAGRAM)

CHECK TIME (HH:MM:SS): __ : __ : __

Please solve the following questions related to the diagram shown on the following page. This diagram models the **behavior of a phone call:**

1. If we are in the state DIALING and the event *Dial digit* occurs, which state do we reach?
2. If we are in the state OBTAINING LINE and the event *Time exhausted* occurs, which state do we reach?
3. Starting in the state DIALING, which state do we reach if the following sequence of events occurs?
   Number dialed [Number valid]
   On-line
   Destination answers
4. Starting in the state CONNECTED, which state do we reach if the following sequence of events occurs?
   New call
   On-line
   Hang up

5. Write down the minimum sequence of events and conditions needed, to go from the state DIALING to the state DISCONNECTED:

6. Write down the minimum sequence of events and conditions needed, to go from the state CONNECTING to the state BUSY:

CHECK TIME (HH:MM:SS): __ : __ : __

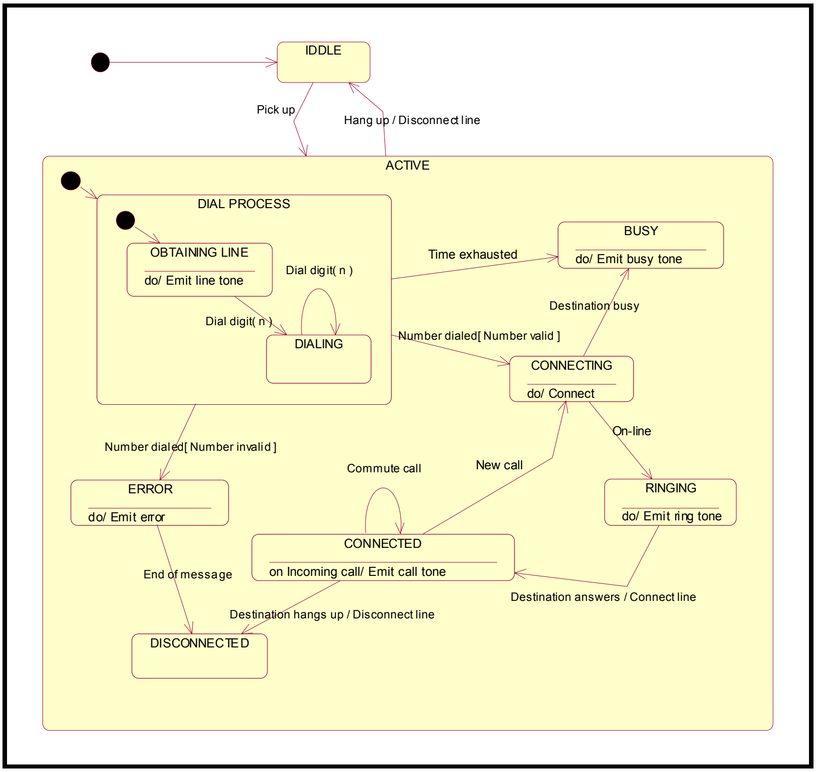

Fig. 5. Example of diagram with composite states (phone call)

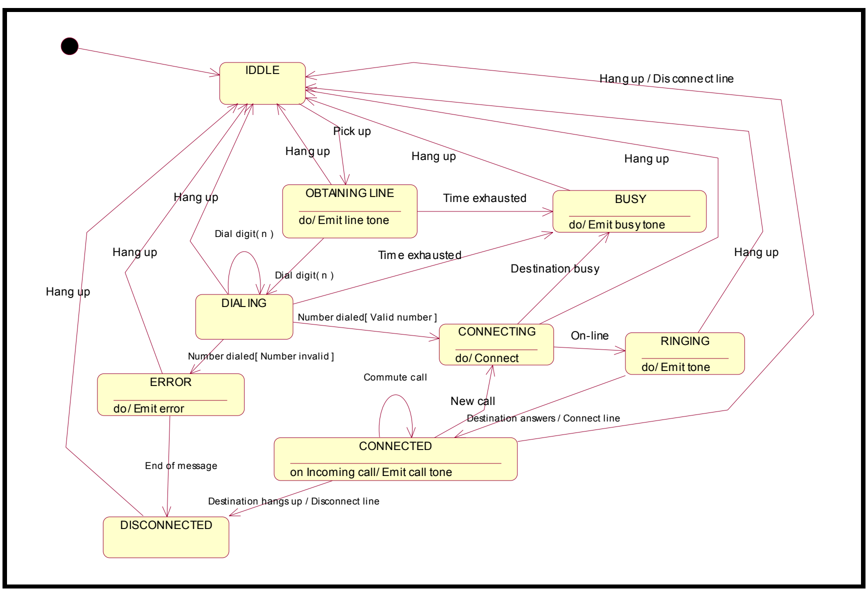

Fig. 6. Example of diagram without composite states (phone call)