Code merging analysis for different procedure types

Sanja Candrlic, Mile Pavlic, Martina Asenbrener

***Abstract*—**This paper presents the results of a research on code merging in a specific development environment. The research data was collected during a timeframe of 5 years of software development and maintenance. Four different software products were monitored. During that time 49.572 procedure mergings occurred and 498 of them were recognized as conflicts. A merge conflict occurs when two programmers make changes at the same time on the same source code block (procedure or program module, for example) and then try to merge their changes into a central server repository. In this paper merge conflicts, source code merging and their relationship with procedure types were analyzed. The collected data is organized according to the procedure type. The research goal was to establish if there is a connection between merge conflicts and procedure types and to analyze the differences in code merging for different procedure types.

***Keywords*—**Merge conflict, procedure type, team of programmers, team software development.

I. Introduction

NEW software development models that appeared during the nineties presume parallel software development activities, with a goal of shortening software development time [1]. Until then, activities in the software development cycle where conducted in a sequence (waterfall model). New software development models led to an even greater need for parallel software development activities and for team software development. Complex business tasks require an integral software package that covers all of business activities. It became impossible for a single programmer to develop such software – if working alone he/she will need more time than a business organization is willing or able to provide. Business markets strive for fast and efficient development of high quality software and computer firms need to readjust. New software developer teams arise and each team member tends to specialize for a certain activity or an activity group within the software development cycle.

Parallel activities conducted by many programmers in a software development team require a well-planned task distribution and create a necessity for painstaking planning and management activities [2]. Software merging is recognized as an essential aspect of maintenance and evolution of large-scale software systems [3]. Web application software development increased the need for version control, not only for source code, but even for other types of files and modules, such as applets, images, graphics [4].

The need for communication in the development team is constant. Different types of communication gaps between team members, not only between programmers are described in [5]. In global software development the lack of communication may cause many problems and special attention should be given to communication support and team management [6].

Even a successful distribution of tasks among team members doesn’t ensure a neat merging of independently written source code that includes separate task solutions. Sharing of program code is easier if developers use a tool that supports team software development. Even with a team software development tool that automates software version merging, situations that can’t be solved automatically do happen [7]. These situations are called merge conflicts. A merge conflict appears when two programmers try to merge the same block of a source code that has been independently modified by each programmer.

Merge conflicts can’t be solved automatically. The process of finding a proper solution involves programmers but their time is precious and always missing. The goal is to bring the number of conflicts and manual interventions to a minimum.

We may assume that conflict emergence is influenced by different parameters [8]. The goal of this research was to determine and compare the frequency of conflict emergence for different procedure types. This data could be used as a basis for making various decisions for the leader of a software development team, such as what procedures types require special treatment in order to avoid the emergence of conflicts and possible loss of program code.

In this paper mergings of program code and merge conflicts are analyzed. The research took into consideration the data about software development and maintenance in a given development environment of a software company [9]. This paper presents the results of a research on team software development in a specific development environment in which a set of development tools are used: C, Clarion code generator, Pl/Sql and various products such as a database synchronization tool (“Synchro”), Oracle and Clarion drivers (modified to meet specific company needs), a configuration management tool

Sanja Candrlic is with Department of Informatics, University of Rijeka, Rijeka, 51000, Croatia (phone: +385-51-584-706; fax: +385-51-584-749; e-mail: sanjac@inf.uniri.hr).

Mile Pavlic is with Department of Informatics, University of Rijeka, Rijeka, 51000, Croatia (e-mail: mile.pavlic@ris.hr).

Martina Asenbrener is with Department of Informatics, University of Rijeka, Rijeka, 51000, Croatia (e-mail: masenbrener@inf.uniri.hr).

("Builder") and a thin client solution ("Thin@"). The research data was collected based on activities of a team of developers during a timeframe of 5 years of software development and maintenance. Four different software products were monitored. During that time 49.572 procedure mergings occurred and 498 of them were recognized as conflicts. Based on that data, the percentage of conflicts in all mergings is 1%. This may seem like a small percentage, but every possible source code loss caused by merge conflict should be avoided.

This paper extends our previous research [10]. We expect that the results of this research will help project leaders in making decisions about engaging more experienced programmers on more delicate procedure type assignments.

II. Development Environment

A software product consists of interconnected program units. A program unit can include smaller parts. The data model (entity-relationship diagram) of a software structure shown in a Fig. 1 was built by using the entity-relationship method [11], [12].

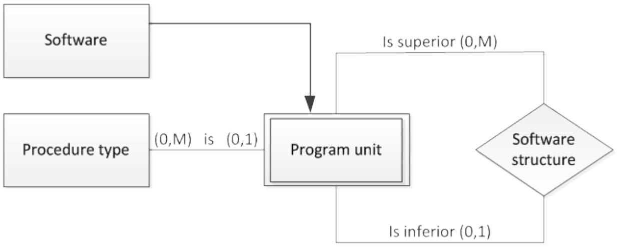

Fig. 1 Model of the software structure

If we observe any program unit, it may include even smaller program units. The structure of a software product may be presented by a hierarchical relationship between program units (Fig. 1). A program unit on the lowest level doesn't include smaller parts. Therefore, the cardinality of the relationship "is superior" is (0,M). For each part of the program structure it is known that it belongs to a precisely one superior part (with the exception of the top program unit that doesn't belong to any) and the cardinality of the relationship "is inferior" is (0,1).

The number of the levels of program units is not determined, but in this research we analyzed software products that have three levels. The three levels of the software structure are: application, program module and procedure.

An application or business software is a program with a specific business purpose and it is used in a specific business area to solve a business task by using data processing techniques. A program module is an independent software part that is connected to other software parts [13]. It may be a constituent part of the whole and include smaller parts. A procedure is a logical software part of that performs different operations.

Regardless of the development environment, it is always possible to define three levels of program units. The software products analyzed in this research are developed in a specific development environment which uses a 4th generation program languages with an integrated development interface. The data dictionary in this environment presents a knowledge database in which programmers enter the system description and the environment generates the program support for the given information system. The software developed in the described environment consists of a data dictionary and a number of program modules. The program modules are almost independent. They are connected with a single program module that keeps them together and external procedures that can be called from any module. These are the possible procedure types: browse, form, menu, report, window and source.

Browse (or grid) procedure type is a type used for data browsing. Form procedure type enables users to perform an action over data. Menu procedure type enables user to browse software possibilities and to choose an option or option group that offers new options or submenus. Window procedure is a procedure of general type that can be used for any purpose, for example it can be used as a procedure for a window that offers a user to enter a group of parameters needed for a specific report as input data. Report is a procedure type for data extraction, data processing and screen presentation, often with the possibility of printing. Source procedure is a type of procedure used to save source code for some batch process. The result of this procedure type usually can't be seen as a screen, but it is rather used to keep source code needed to perform some action in the software.

This research paper used the data from a tool for team software development support called Builder [14]. Builder was developed for their own purposes by a Croatian firm and it is not a commercial software [15]. It is successfully used to support software development teams since 2002, and it is used by around 30 developers on more than 50 software products of different sizes. The developers are situated in two different locations with flexible working hours. That makes their direct communication sometimes hard or even impossible. In these circumstances a tool that supports team software development should make daily tasks more efficient.

Although many different tools that support team software development can be found on the market, Builder is used because it is adopted to this specific software development environment and to the development processes of the company. It works on the basis of optimistic locking [16] and recognizes the changes on the procedure level.

III. Data Analysis

Business applications whose merging data was analyzed in this research are: IMIS [17], AO [18], PRIS [19] and FAROS [20]. A set of development tools was used to develop screens and procedures and the Oracle database was used to save the data. The development of information systems is based upon the MIRIS methodology for information system design and development [21].

The size and the structure of business applications were analyzed. The data collected is shown in Table I. Based on that data it is possible to compare the complexity of given applications.

Table I Complexity of business applications

Criteria of complexity	Business application			
	AO	IMIS	PRIS	FAROS
Nr of prog. modules	3	61	18	24
Nr of procedures	123	2.225	1.065	1.309
Nr of browse proc	24	688	266	344
Nr of form proc	26	556	222	343
Nr of window proc	33	353	346	247
Nr of report proc	34	536	200	242
Nr of source proc	5	83	28	132
Nr of menu proc	1	9	3	1

The data from Table I is in its relative ratio shown graphically in Fig. 2.

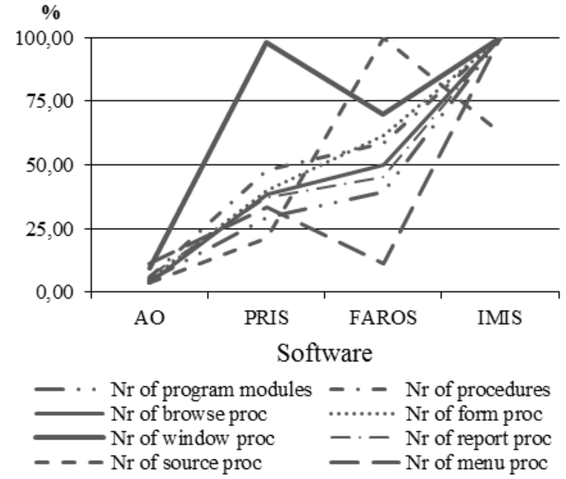

Fig. 2 Software complexity comparation

One may see a correlation between most of the chosen criteria. This graph may lead to the assumption that for information system complexity estimation it can be enough to choose a single criteria, but that is yet to be determined (by testing a larger number of applications and criteria) and it's not a part of this research. Based on all chosen criteria, AO software is the less complex. IMIS is the system with the highest complexity based on almost all criteria, with the exception of the number of source type procedures.

The remaining data shows two more exceptions in preservation of linearity: number of menu and window procedures of PRIS information system compared to FAROS information system. We may also conclude that the ratio between the number of different type procedures is not constant a value – some information systems may include larger number of procedures of some type and smaller number of procedures of some other type. Yet, we may conclude that for the given information systems the numbers of browse, form and report procedure types follow the complexity of first two criteria: number of program modules and number of all procedures. It is possible to perform a future research that would analyze the number of procedures for each procedure type and the reasons behind that specific value.

IV. Results

In this chapter the data about mergings for every procedure type is analyzed. In Table II the data about the mergings, merge conflicts and number of months taken into analysis for each business application is presented. The research covered 49.572 mergings and 498 conflicts.

Table II Range of research

Criteria	Business application			
	AO	IMIS	PRIS	FAROS
Period (months)	44	48	34	56
Nr of mergings	1.574	16.202	9.415	33.010
Nr of conflicts	14	90	84	386

The goal of this analysis was to determine the possible difference between the data about mergings and conflicts for different types of procedures and to determine the deviation between the average and the data measured for each business application and procedure type. Merge conflicts, their frequency of emergence in parallel mergings and in all mergings were analyzed.

For every business application and every procedure type (PT) the following data was measured: the number of parallel mergings for the given procedure type (Nr PM), the number of all mergings for the given procedure type (Nr M) and the number of conflicts (Nr C). The data is shown in Table III.

Table III The share of conflicts in mergings

Nr of conflicts in Nr of mergings						
PT	Nr C	NrPM	NrM	NrPM/ NrM (%)	NrC/ NrPM (%)	NrC/ NrM (%)
BRW	173	4.749	15.073	31,51	3,64	1,15
FRM	130	3.830	10.775	35,55	3,39	1,21
WIN	85	2.269	8.961	25,32	3,75	0,95
REP	64	2.502	10.409	24,04	2,56	0,61
SRC	26	962	3.211	29,96	2,70	0,81
MEN	20	1.044	1.143	91,34	1,92	1,75
∑	498	15.356	49.572	-	-	-
Av.	-	-	-	30,98	3,24	1

Let us check how much work is done in parallel for each procedure type. Based on the data in Table III in column Number of parallel mergings (Nr PM) and Number of all mergings (Nr M), the share of parallel mergings in all mergings was calculated (Nr PM/ Nr M) for each procedure type. According to the data calculated, for every procedure type parallel work was performed equally, with the exception of the menu procedure type. The described data is shown graphically in Fig. 3.

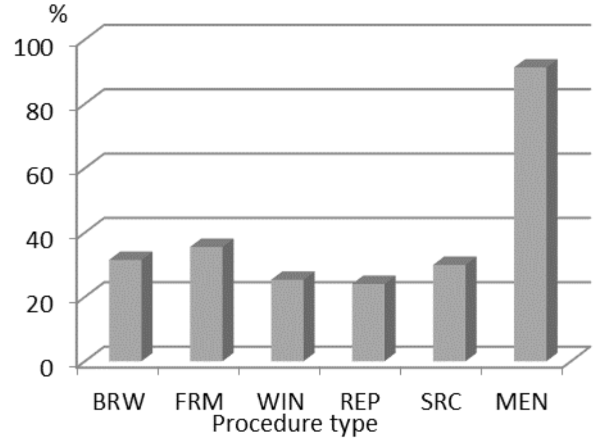

Fig. 3 Share of parallel mergings in all mergings

Mergings of the menu procedure type were even 91% done in parallel, while for the other procedure types the percentage of parallel mergings in all mergings was between 24% and 36%. Regardless of the procedure type, the percentage of parallel mergings (15.356) in all the mergings (49.572) is 31%. That means that every third merging could have created a merge conflict. As we can see from this data, in these teams a lot of work was done in parallel.

Based on the data in Table III in the column Number of conflicts (Nr C), Number of parallel mergings (Nr PM) and number of mergings (Nr M), the share of conflicts in parallel mergings (Nr C/ Nr PM) and the share of conflicts in all mergings (Nr C /Nr M) were calculated for every procedure type. The share of conflicts in parallel mergings (Nr C/ Nr PM) for every procedure type is graphically shown in Fig. 4. If we compare the data we can see that the share of conflicts in parallel mergings for menu procedures is lower than for other procedure types.

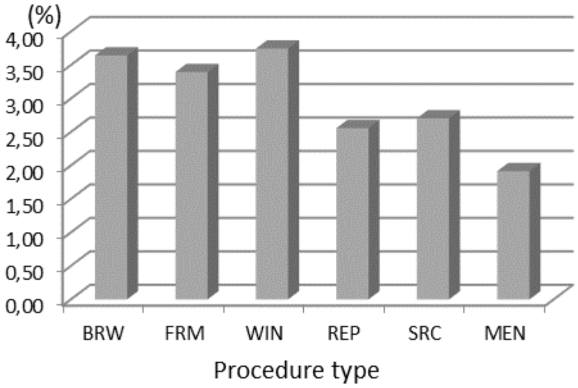

Fig. 4 Share of conflicts in parallel mergings

Let us observe conflicts and mergings regardless of procedure type. The sum of all conflicts is 498 and the sum of all parallel mergings is 15.356. Based on that data, the percentage of conflicts in parallel mergings is 3,24%. The percentage of conflicts in all mergings performed (49.572) is 1%.

Specific application development environments may influence the average values and bring to a significant deviation. In order to check if that is the case, in Tables IV–IX the data is shown for every business application and every procedure type separately.

In Table IV the data on conflicts (Nr C), parallel mergings (Nr PM) and all mergings of browse procedures (Nr M) is shown. In the column Number of parallel procedures (Nr PP) the number of browse procedures on which parallel work was performed is shown. In the column Number of procedures (Nr P) the number of all procedures of browse type is shown for each business application.

The sum of conflicts on browse procedures is 173 and the sum of parallel mergings on browse procedures is 4.749. Based on that data, the average share of conflicts in parallel mergings of browse procedures is 3,64%. The average share of conflicts (Nr C = 173) in all mergings of browse procedures (Nr M = 15.073) is 1,15%.

Table IV Merge conflicts for browse procedures

Procedure type: BROWSE							
App	Nr PP	NrP	Nr PM	NrM	Nr C	NrC/ NrPM (%)	NrC/ NrM (%)
AO	13	24	51	313	0	0,00	0,00
IMIS	280	688	1.027	3.946	21	2,04	0,53
PRIS	82	266	374	2.303	35	9,36	1,52
FAROS	639	344	3.297	8.511	117	3,55	1,37
∑	1.014	1.322	4.749	15.07 3	173	-	-

A significant deviation from the average value is noticeable in Table IV for the PRIS business application in the column Share of conflicts in parallel mergings of browse procedure (Nr C / Nr PM). The data shows that the share of conflicts in parallel merging of browse procedures is relatively high: more than 9% of parallel mergings of browse procedures generated conflicts.

There is a significant difference even for other types of procedures between the values for PRIS business application and the values for other three remaining business applications (Tables V – IX): the data about PRIS conflicts are at least two times higher than the average.

The data about form procedure mergings and conflicts are shown in Table V.

The average percentage of conflicts (Nr C = 130) in parallel mergings of form procedures (Nr PM = 3.830) is 3,39%. The average percentage of conflicts Nr C = 130) in number of all mergings of form procedures (Nr M =10.775) is 1,21%.

Table V Merge conflicts for form procedures

Procedure type: FORM							
App	Nr PP	NrP	Nr PM	NrM	NrC	NrC/ NrPM (%)	NrC/ NrM (%)
AO	32	26	304	496	9	2,96	1,81
IMIS	139	556	496	2.260	16	3,23	0,71
PRIS	24	222	85	959	8	9,41	0,83
FAROS	669	343	2.945	7.060	97	3,29	1,37
∑	864	1.147	3.830	10.775	130	-	-

According to the data about window procedure merging (Table VI), the average percentage of conflicts (Nr C = 85) in parallel mergings of window procedures (Nr PM = 2.269) is 3,75%. The average percentage of conflict (Nr C = 85) in all mergings of window procedures (Nr M = 8.961) is 0,95%.

Table VI Merge conflicts for window procedures

Procedure type: WINDOW							
App	Nr PP	NrP	Nr PM	NrM	NrC	NrC/ Nr PM (%)	NrC/ NrM (%)
AO	11	33	62	229	1	1,61	0,44
IMIS	131	353	536	2.802	7	1,31	0,25
PRIS	51	346	242	1.722	24	9,92	1,39
FAROS	270	247	1.429	4.208	53	3,71	1,26
∑	463	979	2.269	8.961	85	-	-

As shown in Table VII, 64 conflicts on report procedures were found. The average percentage of these conflicts in parallel mergings (Nr PM = 2.502) is 2,56%, and the average percentage of conflicts in all mergings (Nr M = 10.409) is 0,62%.

Table VII Merge conflicts for report procedures

Procedure type: REPORT							
App	Nr PP	NrP	Nr PM	NrM	NrC	NrC/ Nr PM (%)	NrC/ NrM (%)
AO	16	34	81	295	2	2,47	0,68
IMIS	297	536	1.192	5.169	20	1,68	0,39
PRIS	32	200	114	1.286	6	5,26	0,47
FAROS	277	242	1.115	3.659	36	3,23	0,98
∑	622	1.012	2.502	10.409	64	-	-

The data about source procedure mergings and conflicts are shown in Table VIII.

Table VIII Merge conflicts for source procedures

Procedure type: SOURCE							
App	Nr PP	NrP	Nr PM	NrM	Nr C	NrC/ NrPM (%)	NrC/ NrM (%)
AO	0	5	0	38	0	0,00	0,00
IMIS	62	83	170	626	1	0,59	0,16
PRIS	10	28	31	230	2	6,45	0,87
FAROS	159	132	761	2.317	23	3,02	0,99
∑	231	248	962	3.211	26	-	-

According to the data in Table VIII, the average percentage of conflicts (Nr C = 26) in parallel mergings of source procedures (Nr PM = 962) is 2,7%, and the average percentage of conflicts in all mergings (Nr M = 3.211) is 0,81%.

The data about menu procedure mergings and conflicts are shown in Table IX.

Table IX Merge conflicts for menu procedures

Procedure type: MENU							
App	Nr PP	NrP	Nr PM	NrM	NrC	NrC/ NrPM (%)	NrC/ NrM (%)
AO	4	1	29	52	0	0,00	0,00
IMIS	53	9	312	356	2	0,64	0,56
PRIS	19	3	139	159	9	6,47	5,66
FAROS	55	1	564	576	9	1,60	1,56
∑	131	14	1.044	1.143	20	-	-

According to the data in Table IX there were 20 conflicts in 1.044 parallel mergings of menu procedures, and that gives 1,92%. The percentage of conflicts in all mergings of menu procedures (Nr M =1.143) is 1,75%.

V. The Difference in Merging and Conflicts for Different Procedure Types

After analyzing the data for each procedure type, in the next step of the research the difference between procedure types will be analyzed.

The first part of the analysis is based upon comparing the share of parallel mergings in all mergings for each procedure type. The significance of the difference between the share of conflicts in the number of parallel mergings for each pair of procedure types is calculated. Chi-square test is used. The data for this calculation is given in Table III in the NrPM/NrM (%) column. We assume the standard border level of significance at 0,05. In the Table X the results of the analysis are shown.

Table X Chi-square test on Number of parallel mergings (NrPM) in Number of mergings (NrM)

PRC type	BRW	FRM	WIN	REP	SRC	MEN
BRW		4,04 46,064 <0,0001	6,19 103,855 <0,0001	7,47 168,384 <0,0001	1,55 2,889 0,0892	59,83 1653,601 <0,0001
FRM	4,04 46,064 <0,0001		10,23 239,288 <0,0001	11,51 334,128 <0,0001	5,59 34,071 <0,0001	55,790 1328,332 <0,0001
WIN	6,19 103,855 <0,0001	10,23 239,288 <0,0001		1,28 4,181 0,0409	4,64 25,865 <0,0001	66,02 2001.92 <0,0001

REP	7,47 168,384 <0,0001	11,51 334,128 <0,0001	1,28 4,181 0,0409		5,92 45,035 <0,0001	67,30 2189,466 <0,0001
SRC	1,55 2,889 0,0892	5,59 34,071 <0,0001	4,64 25,865 <0,0001	5,92 45,035 <0,0001		61,38 1275,730 <0,0001
MEN	59,83 1653,601 <0,0001	55,790 1328,332 <0,0001	66,02 2001.92 <0,0001	67,30 2189,466 <0,0001	61,38 1275,730 <0,0001	

The goal of the research was to determine whether there was a significant difference between the share of parallel mergings in all mergings when each two pairs of procedure types are compared. The first value in each cell of the Table X marks the real difference between the shares (%), the second value is Chi-square statistics, and the third value is the level of significance, the p-value.

Let us see the data for BRW procedure type. The only procedure type whose share of parallel mergings in all mergings is not significantly different from the same data for BRW procedure type is SRC procedure type (p=0,0892, p-value >0,05). The difference between the share of parallel mergings in all mergings for BRW procedure type and the share of parallel mergings in all mergings for every other procedure type is significantly different (p-value <0,001, except for WIN and REP pair whose p-value is greater than 0,001 but still p< 0,05 and considered statistically significant). According to the analysis, when we compare any two procedure types, the difference between the share of conflicts in parallel mergings is almost always significant (except for the BRW-SRC pair).

The next step was to perform the analysis of the difference between the share of conflicts in all mergings for each pair of procedure types. Chi-square test was used. The data for this calculation is given in Table III in the NrC/NrM (%) column. The result of the analysis is shown in Table XI.

Table XI Chi-square test on Number of conflicts (NrC) in Number of parallel mergings (NrM)

PRC type	BRW	FRM	WIN	REP	SRC	MEN
BRW		0,06 0,147 0,7019	0,2 1,929 0,1648	0,54 18,917 <0,0001	0,34 2,531 0,1116	0,6 2,757 0,0968
FRM	0,06 0,147 0,7019		0,26 2,826 0,0927	0,6 20,362 <0,0001	0,4 3,227 0,0724	0,54 2,005 0,1568
WIN	0,2 1,929 0,1648	0,26 2,826 0,0927		0,34 6,871 0,0088	0,14 0,369 0,5435	0,8 5,548 0,0185
REP	0,54 18,917 <0,0001	0,6 20,362 <0,0001	0,34 6,871 0,0088		0,2 1,213 0,2708	1,14 17,098 <0,0001
SRC	0,34 2,531 0,1116	0,4 3,227 0,0724	0,14 0,369 0,5435	0,2 1,213 0,2708		0,94 6,253 0,0124
MEN	0,6 2,757 0,0968	0,54 2,005 0,1568	0,8 5,548 0,0185	1,14 17,098 <0,0001	0,94 6,253 0,0124	

Let us check the difference between the shares of conflicts in all mergings when comparing each two pairs of procedure types. The difference between the shares of conflicts in mergings is significant when we compare report and browse, report and form, report and window, report and menu, window and menu and window and source procedure types. For those pairs of procedure types the p-value is <0,05 and is considered statistically significant. For other pairs of procedure types the difference between the share of conflicts in mergings is not significant.

In the Fig. 5 the variable Number of parallel procedures (NrPP) for each procedure type is shown using the Box-Plot graph.

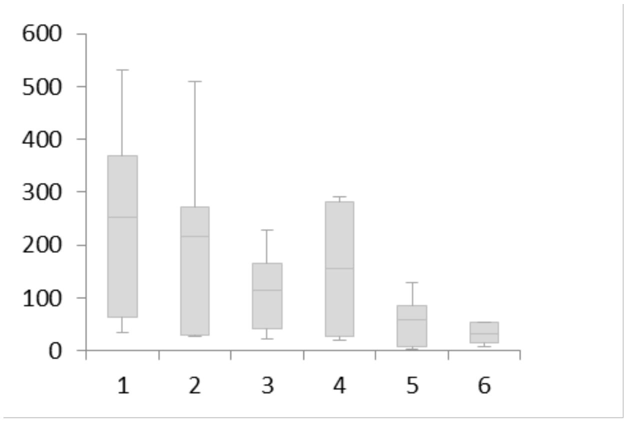

Fig. 5 Number of parallel procedures

Procedure types are positioned on the x-axis, where 1 stands for BRW, 2 for FRM, 3 for WIN, 4 for REP, 5 for SRC and 6 for MEN procedure type. When we compare the data for different procedure types we can see that the number of parallel procedures is very variable for the browse and form procedure types, while it is not so variable for the report and menu procedure types.

In Fig. 6 the variable Number of procedures (NrP) for each procedure type is shown using the Box-Plot graph and using the same notation for procedure types on x-axis.

One can see that the range of the variable Number of procedures is significantly different for the last two procedure types (SRC and MEN) than the first four, while BRW, FRM, WIN and REP have very similar range. This was expected because business applications are based upon procedures of these four procedure types.

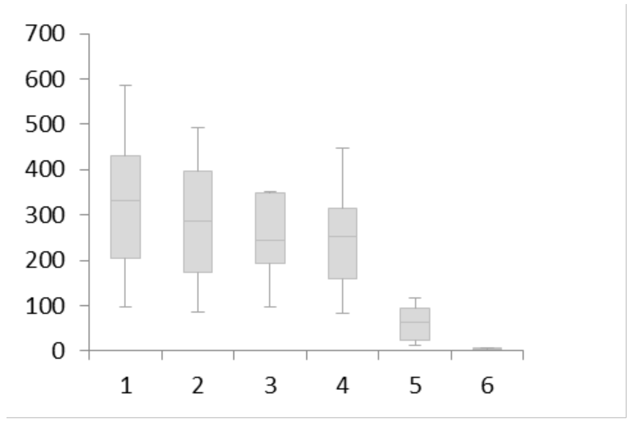

Fig. 6 Number of procedures

In the Fig.7 the variable Number of mergings (NrPM) for each procedure type using Box-Plot graph is shown using again the same notation for procedure types on x-axis. We can see that the first four procedure types (BRW, FRM, WIN and REP) have a similar range.

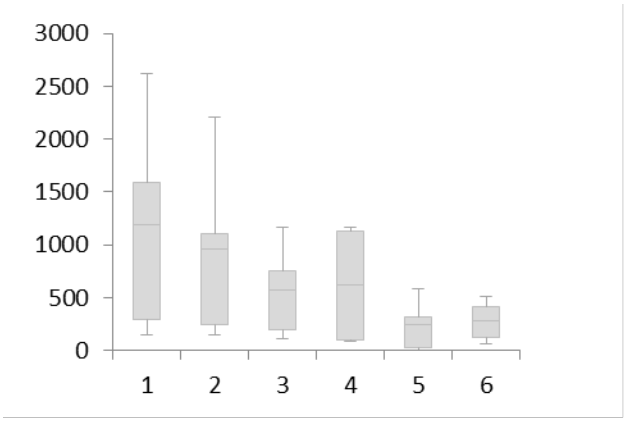

Fig. 7 Number of parallel mergings

In the Fig. 8 the variable Number of mergings is shown. Number of mergings is rather similary ranged for BRW, FRM, WIN and REP procedure types, while SRC and MEN have rather different range.

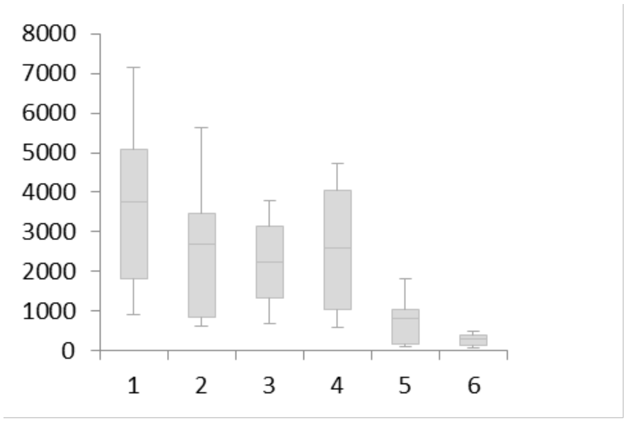

Fig. 8 Number of mergings

The range of the last variable analyzed, the number of conflicts, is shown in Fig. 9. The range of number of conflict is rather variable, as we can see in the Plot-Box graph.

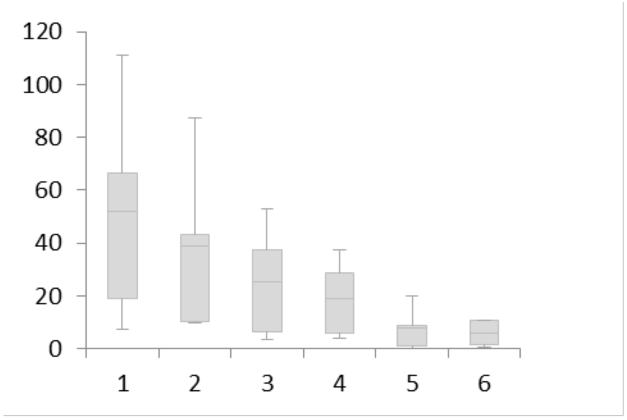

Fig. 9 Number of conflicts

After comparing the differences in the share of parallel mergings in all mergings and the differences in the share of conflicts in all mergings, we can conclude that for some procedure types the difference is rather significant. When we compare the range of values for different variables analyzed, we can see the differences as well.

VI. Conclusion

This research analyses merge conflicts and source code merging and their relationship with procedure types. The data analysis about mergings in the described development environment shows that conflicts may occur on every procedure type. Programmers' work on any procedure type should be performed cautiously to avoid loss of source code.

According to the data, parallel work on menu procedures happens more often than on other procedure types but the share of conflicts in them is rare. As shown in Table III, the percentage of parallel mergings in all merging of menu procedures is rather high (higher than 90%). Regardless of that, no significant number of conflicts in these mergings was found. Namely, menu procedures are central procedures of business software, but team members usually finish their work on them rather fast. The changes in these procedures are rarely complex and mostly include implementation of procedure calls on procedures of other types. These tasks are simple and can be quickly performed. Conflicts on menu procedures mostly emerge after complex programming assignments [22] that include implementation of many new procedure calls and other menu changes during longer periods of time. Programmers must be aware of a special status that menu procedures have in a business application and plan in advance and coordinate their changes on menu procedures.

A significant deviation from the average value of the share of conflicts in parallel mergings and the share of conflicts in all mergings is noted for the PRIS application for every procedure type. After a detailed analysis of the gathered data, we noticed that the frequency of conflicts in the beginning of the project was high. A reason for that might be specific development situations with short deadlines, frequent reassignments of developers within project, illness, overstrain caused by long stays outside of hometown and all-day and all-night working shifts during intense development periods. The question that arises is whether task distribution on this project was clear enough or the complexity of the numerous tasks made conflicts unavoidable.

After comparing the share of parallel mergings in all mergings for each two procedure types, we can conclude the difference between those shares is significant, at least for the given data. Only one pair of procedure types (browse and source procedure types) didn't have statistically significant difference for the data analyzed.

We may say that that procedure types differ significantly in the share of merging that is performed in parallel, but are relatively more similar by the share of conflicts in mergings performed.

A broader research should analyze the influence of other parameters on frequency of conflict occurrence, such as the number of developers in a team and the number of procedures in program module.

References

[1] B. Boehm, *A View of 20th and 21st Century Software Engineering*. International Conference on Software Engineering ICSE'06, May 20–28 2006, Shanghai, China, pp. 12-29.

[2] R. Kraut, L. Streeter, *Coordination in Software Development*. Communications of the ACM, Vol. 38, No. 3, 1995, pp. 69-81.

[3] T. Mens, *A State-of-the-Art Survey on Software Merging*, IEEE Transactions on Software Engineering, Vol.28, No.5, May 2002, pp. 449-462.

[4] J. Wong, M. A. M. Capretz, *An Implementation for Merging Images for Version Control*. Proceedings of the 10th WSEAS International Conference on COMPUTERS, July 13-15, 2006, Vouliagmeni, Athens, Greece, pp. 653-658.

[5] D. Kumlander, *On Software Design and Development Supporting Requirements Formulation*. Proceedings of the 10th WSEAS International Conference on COMPUTERS, July 13-15, 2006, Vouliagmeni, Athens, Greece, pp. 818-825.

[6] F. Dumitriu, D. Oprea, G. Mesnita, *Issues and strategy for agile global software development adoption*. Recent researches in Applied Economics, Proceedings of the 3rd World Multiconference on Applied Economics, Business and Development (AEBD'11), WSEAS Press, Iasi, Romania, July 1-3, 2011, pp. 37-42.

[7] R. E. Grinter, *Recomposition: Putting It All Back Together Again*. Proceedings of the 1998 ACM conference on Computer supported cooperative work, Seattle, Washington, USA, November 14-18, 1998, pp. 393–402.

[8] S. Candrlic, *Model of a system for team software development and maintenance* – doctoral dissertation, Faculty of humanities and social sciences, University of Zagreb, 2012.

[9] RIS. Project Builder. Project documentation, Rijeka, 2002.

[10] S. Candrlic, M. Pavlic, A. Jakupovic. *Analysis of the Relationship between Merge Conflicts and Procedure Types*. Proceedings of the 7th European Computing Conference (ECC'13), WSEAS Press, June 25-27, 2013, Dubrovnik, Croatia, pp. 160-166.

[11] P. P. Chen, *The Entity-Relationship Model – Towards a Unified View of Data*. ACM On Database Systems, Vol.1, No.1, 1976, pp. 9-36.

[12] M. Pavlic, *Oblikovanje baza podataka*. Rijeka: Odjel za informatiku, Sveuciliste u Rijeci, 2011. pp. 77-134.

[13] M. Kis, Englesko-hrvatski i hrvatsko-engleski informaticki rjecnik, Naklada Ljevak, Zagreb, 2002.

[14] S. Candrlic, M. Pavlic, P. Poscic, *Model of a System for Team Software Development*. ITI 2006, June 19-22, Cavtat, Croatia, pp. 117-122.

[15] RIS. Project Builder. Project documentation, Rijeka, 2002.

[16] D. Thomas, A. Hunt, *Pragmatic Version Control using CVS*. Dallas, Texas: The Pragmatic Bookshelf, 2004., pp. 1-20.

[17] RIS, Croatia osiguranje. Project IMIS. Project documentation, Rijeka, 2003.

[18] RIS, Croatia osiguranje. Project AO. Project documentation, Rijeka, 2005.

[19] RIS. Project PRIS. Project documentation, Rijeka, 2005.

[20] RIS. Project FAROS. Project documentation, Rijeka, 2009.

[21] M. Pavlic, *Informacijski sustavi*. Zagreb: Školska knjiga, 2011. pp. 286-289.

[22] S. Candrlic, *Model of a system for team software development and maintenance* – doctoral dissertation, Faculty of humanities and social sciences, University of Zagreb, 2012.