Model-Based Metrics to Estimate Maintainability

Nada Almasri(✉) and Luay Tahat

Gulf University for Science and Technology, West Mishref, Kuwait
{Almasri.n, tahat.l}@gust.edu.kw

Abstract. Software maintenance is becoming more challenging with the increased complexity of software and frequent applied changes to accommodate the rapidly changing technologies and user requirements. In this paper we provide model-based metrics to estimate the maintainability of state-based systems. The purpose of the metrics is to provide a tool that can be used by the system maintenance team to identify critical artifacts of the underlying system and to allow for better planning of the change process. The provided metrics is based on Extended Finite State Machine models (EFSM), and it provides two measures to identify critical transitions. The experimental study shows that the metrics is highly effective in spotting transitions that can cause severe propagation of a change when they are being changed, as well as transitions that are highly sensitive to changes applied to an EFSM model.

Keywords: Maintainability · EFSM · Critical transitions · Sensitive transitions

1 Introduction

The demand for large and complex software systems has been steadily increasing over time. The development and maintenance of these systems are difficult and costly due to the increased complexity of the systems. A major challenge during software maintenance is determining the consequences of applying a requested change to the system. This change may be due to a request to add a functionality, remove a functionality, or fix a bug in the system. Within this context, the system developers would want to estimate (1) if a modification is applied on one component of the system, will other components be affected by this modification? What are these affected components? What percentage of the system do they make? (2) for a stable system component which is not touched by the requested modification, what is the possibility that the modification will propagate to the component? The first set of questions focus on estimating the severity of the requested modification in terms of the number of components affected directly or indirectly by that modification, while the second question focus on estimating the sensitivity of certain components of the system to the modifications applied somewhere else in the model. Estimating the ***severity*** of a modification and the ***sensitivity*** of the system components to modifications can greatly enhance the maintainability of the system as it allows the development team to forecast the scope of the change in order to effectively plan its implementation.

M. H. Miraz et al. (Eds.): iCETiC 2018, LNICST 200, pp. 59–69, 2018.
https://doi.org/10.1007/978-3-319-95450-9_5

One way to manage the complexity of system development process is to use system models in order to reduce ambiguity, misunderstanding, and misinterpretation of system specifications [1–3]. Furthermore, models can be used for test generation [4, 5], test suite reduction [6, 7], and test case prioritization [8–13]. In this paper we use Extended Finite State Machine models which are used to model state-based systems, and we extract the maintainability metrics from these models instead of dealing with their complex underlying system.

In the context of EFSM models, the severity measure predicts the extent to which a change applied to one EFSM transition will propagate to other transitions in the model. The sensitivity measure predicts how often a particular transition under consideration will be affected by a modification applied somewhere else in the model. A transition is identified as a critical transition when a modification applied to it severely propagates to other transitions, or when it has high probability to be affected by a modification applied elsewhere in the model.

The rest of the paper is organized as follows: Sect. 2 provides an overview of state based modeling. Section 3 introduces the two measures used to identify critical transitions. In Sect. 4 presents the empirical study, while Sect. 5 outlines the related work. Finally, in Sect. 6 the conclusion and the future research are discussed.

2 Related Work

Failure mode, effects, and criticality analysis (FEMCA) is a familiar analytical technique in engineering, and particularly in fields such as aviation and automotive [14]. The technique is usually used during the design and the production of new products to estimate the safety risks and hazards. Within the safety context, the technique is mainly based on brain storming the possible failures and the expected consequences of theses failures from human safety perspective.

In the context of software engineering, how critical a modification applied to the software is, is referred to as "impact analysis". Bohner and Arnold [19] define impact analysis as "identifying the potential consequences of a change, or estimating what needs to be modified to accomplish a change". Several research papers presented code-based impact analysis techniques [15], while only a few targeted model-based impact analyses [16, 17]. Almasri [17] proposed an approach to measure the impact of a change at the model level. Their work focused on measuring the change impact for a change applied to EFSM models using model dependencies.

Generally, impact analysis techniques are used to measure the impact of a given modification. The metrics we are suggesting, however, in this paper attempts to estimate how critical EFSM transitions of a specific EFSM model are in general, for any potential change in the future.

3 State-Based Modeling with EFSM

An EFSM model M can be formally expressed as: M = (Σ, Q, Start, Exit, V, O, R) where:

Σ is the set of events, Q is the set of states, Start ∈ Q is the start state, Exit ∈ Q is the exit state, V is a finite set of variables, O is the set of actions, R is the set of transitions, where each transition T is represented by the tuple: $T = (E, C, A, Sb, Se)$ where: E ∈ Σ is an event, C is an enabling condition defined over V, A is a sequence of actions, $A = \, < a1, a2, \ldots.., aj >$, where ai ∈ O. The action may manipulate variables, read input or produce output. Sb ∈ Q is the transition's originating state, Se ∈ Q is the transition's terminating state.

A transition T in R is triggered when the system is in the originating state Sb(T), the event E(T) occurs, and the enabling condition C(T) is evaluated to TRUE. When transition T is triggered, the A(T) sequence of actions is performed and the system is transferred to the terminating state Se(T). EFSM models may be depicted as graphs where states are represented by nodes and transitions by directed edges between states.

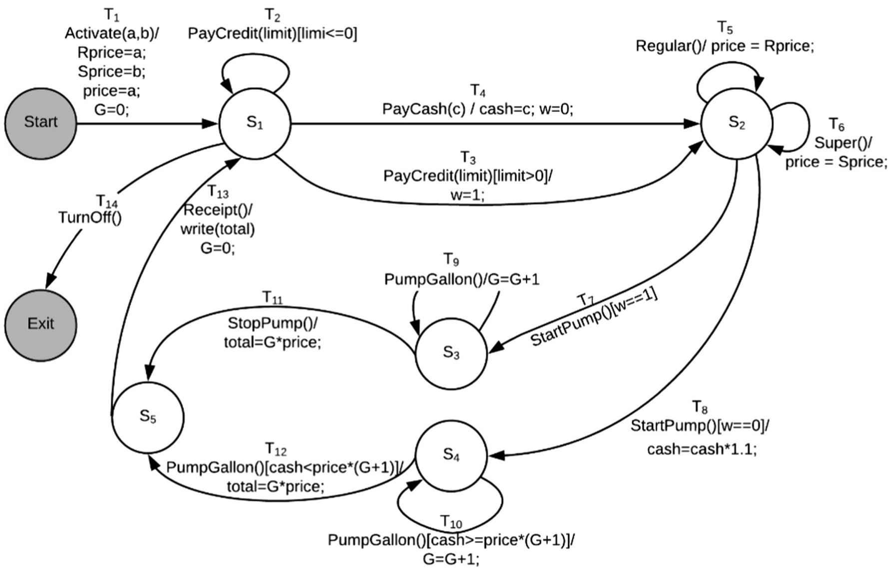

Fig. 1. Fuel pump EFSM model

Figure 1 shows an example of an EFSM model for a Fuel Pump system. According to this model, when the pump is activated, the prices for regular fuel and super fuel are initialized with a default price set as regular price. A person using the pump has the choice to pay by credit or cash. If credit payment is chosen, the credit card is validated based on the available limit. After making the payment choice, the customer gets to choose the type of fuel to pump, and the price to be paid by the user is initialized accordingly. If cash payment is chosen, the customer is rewarded with an extra 10%

once the pumping is started. When the person starts pumping fuel, the amount of gas pumped in is tracked. Finally, as the pump stops pumping gas, the total price is calculated, and a receipt is printed out. At this stage, the pump can be used by the next customer, or it can be turned off.

4 Model-Based Metrics Using Model Dependence

The purpose of the metrics is to identify critical EFSM transitions assuming that such transitions require greater attention from the development team during the maintenance and testing phases of the system development lifecycle.

For a given transition Ti in an EFSM model, if a change is requested, the transition Ti can be subject to change, or the change can be applied somewhere else in the model.

If the change is applied to Ti and it propagates to a large number of other transitions in the model, then Ti is considered as a critical transition since its change affects a large portion of the model. In this case, we call this measure *change-severity* of Ti, and it is denoted as Sv(Ti).

If, on the other hand, the change is applied somewhere else other than Ti, then Ti can still be considered critical if it has a high probability to be impacted by that change. In this case, we call this measure Ti's *sensitivity* to change, and it is denoted as Sn(Ti).

In order to quantify the propagation of a change from Ti to other transitions in the model or vice versa, we use model dependencies which exist between EFSM transitions.

4.1 Model Dependence

The metrics presented in this paper is based on data and control dependence which exist between transitions in EFSM models [13, 17]. These dependencies capture the notion of potential "interactions" between transitions in the model.

Data dependence captures the notion that one transition defines a value to a variable and another transition may potentially use this value. There exists data dependence between transitions Ti and Tk if transition Ti modifies value of variable v, transition Tk uses v, and there exists a path (transition sequence) in the model from Ti to Tk along which v is not modified. For example, there exists data dependence between transitions T1 and T6 in the model of Fig. 1 because transition T1 assigns a value to variable Rprice, transition T6 uses Rprice, and there exists a path (T1, T4, T6) from T1 to T6 along which Rprice is not modified.

Control dependence was originally defined for program's Control Flow Graph (CFG) [18]. Control dependence captures the notion that one node in the control graph may affect the execution of another node. In [1], the concept of program control dependence was extended to EFSM models. Control dependence in an EFSM exists between transitions and it captures the notion that one transition may affect traversal of another transition.

For example, transition T5 is control dependent on T4 in the model of Fig. 1 because (1) Sb(T4) does not post dominate Sb(T5) (condition 1 of control dependence definition is true) and (2) state Sb(T5) post dominates transition T4 (condition 2 is

TRUE). Note that Sb(T4) is S1 and Sb(T5) is S2. The issue of control dependence in EFSMs is discussed in more details in [2, 6, 10, 13, 17].

Data and control dependence in the model can be graphically represented by a directed graph where nodes represent model transitions and directed edges represent model data and control dependencies.

More formally, let M = (Σ, Q, Start, Exit, V, O, R) be an EFSM model and let $G = (R, E)$ be a model dependence graph of model M where:

R is a set of nodes (set of transitions).

E is a binary relation on R, $E \subseteq R \times R$, referred to a set of directed edges where: edge (Ti, Tk) $\in$ E, if there exists data or control dependence between transitions Ti and Tk.

Alternatively, the dependency between transitions can be represented as a matrix where the D, C, or B labels are used to represent data dependency, control dependency, and both data and control dependency between two transitions. Table 1 shows dependence matrix for the Fuel Pump example in Fig. 1. From the matrix we can see that transition T1, T2, T3, T4, and T14 don't depend on any other transition on the model. Other transitions have a mix of dependencies on other transitions. For example, Transition T5 has data dependency on transition T1 with respect to variable price, and it has to control dependencies on T3 and T4.

Table 1. Dependency matrix for fuel pump model

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
T1														
T2														
T3														
T4														
T5	D		C	C										
T6	D		C	C										
T7			B	B										
T8			B	B				D						
T9	D						C		D				D	
T10	D				D	D		B		D			D	
T11	D				D	D	C		D				D	
T12	D				D	D		B		D			D	
T13			C	C							D	D		
T14														

4.2 Transition's Change Severity

The impact of a change applied to an EFSM model can be measured using the approach presented in [17]. However, in this paper, our purpose is to estimate the expected severity of a change if a particular transition undergoes a change, without actually specifying what type of change the transition may experience. Knowing this information allows identifying critical transitions beforehand prior to applying any changes.

To measure Ti's severity of a change, denoted as Sv(Ti), all transitions that are control or data dependent on Ti are identified, and recursively, their dependent transitions are also identified. The set of dependent transition in this case is called the set of Affected transitions with respect to Ti. The larger this set is, the more severe the change of Ti is considered.

To formally define the set of affecting transitions, we define the relationship "affects" as follows:

Let $G = (R, E)$ be the dependence graph of the model M. A transition T in R "affects" another transition T′ in R *if and only if* there is a non-null path from T to T′ in G.

It is worth mentioning that although control and data dependence relationship is not transitive, the "effects" relationship represents the transitive closure of the dependence relationship [3]. For example, if transition T1 depends on transition T2, and transition T2 depends on transition T3, then T3 "affects" T1.

Below, is the formal definition of the set of transitions affected by a particular transition Ti.

Let $G = (R, E)$ be the dependence graph of the model M. The set of affected transitions for a transition Ti in G, denoted as AD(Ti), is the set of all transitions Tj, where Ti "affects" Tj. Formally, we define this set as:

$$AD(T_i) = \{T_j | T_j \in R, \text{ and } T_i \text{ "affects" } T_j \text{ in } G\} \tag{1}$$

Having identified the set of transitions affected by a given transitions Ti, the percent of transitions affected by Ti out of all transitions in the model represents the severity of the change applied to the transition Ti and denoted as $Sv(T_i)$. The number of the transitions in AD(Ti) is denoted as $|AD(Ti)|$, and the number of the transitions in the model M is denoted as $|R|$. More formally, the severity of a change applied to transition Ti is estimated using the following formula:

$$Sv(T_i) = |AD(Ti)|/|R| \tag{2}$$

4.3 Transition's Sensitivity to Change

To measure the sensitivity of transition Ti to a potential change applied to the model, all transitions on which Ti is either data or control dependent on are identified in a recursive manner. These set of identified transition are called Ti's *Affecting* transitions. The larger the set of affecting transitions is, the more sensitive to change the transition Ti is considered. The larger this set is, the more sensitive Ti is considered.

Below, is the formal definition of the set of transitions affecting a particular transition Ti.

Let $G = (R, E)$ be the dependence graph of the model M. The set of affecting transitions for a transition Ti in G, denoted as AG(Ti), is the set of all transitions Tj that "affects" the transition Ti. More formally:

$$AG(T_i) = \{T_j | T_j \in R, \text{ and } T_j \text{ "affects" } T_i \text{ in } G\} \tag{3}$$

Having identified the set of transitions affecting a given transtions Ti, the transition's sensitivity to change, denoted as $Sn(T_i)$, is the percent of transitions affecting Ti out of all the transitions in the EFSM model. The number of the transitions in AG(Ti) is denoted as |AG(Ti)|, and the number of the transitions in the model M is denoted as |R|. More formally, the sensitivity of a given transition Ti in an EFSM model can be calculated using the following formula:

$$Sn(Ti) = |AG(Ti)|/|R| \tag{4}$$

Table 2, shows the "affects" relationship between transitions in the fuel pump model, and for each of the transition in the model, the table displays the size of the set of affected transitions AD(T) and the size of the set of affecting transitions AG(T). Each row in the table shows what transitions are affected by a given transition Ti. For example, the first row shows that T1 is affects T5, T6, T10, T11, T12, and T13.

Table 2. "Affects" relationship in fuel pump model

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	|AG(T)|
T1															0
T2															0
T3															0
T4															0
T5	D		C	C											3
T6	D		C	C											3
T7			B	B											2
T8			B	B				D							3
T9			B	B			B		D						4
T10	D		B	B	D	D		B		D					7
T11	D		B	B	D	D	B		D						7
T12	D		B	B	D	D		B		D					7
T13	D		B	B	D	D	B	B	D	D	D	D			11
T14															0
|AD(T)|	6	0	9	9	4	4	3	4	3	3	1	1	0	0	

Table 3, demonstrates the severity Sv(T) and the sensitivity Sn(T) of each transition. From both tables we can see that transition T13 is the most sensitive to change in the fuel pump model since it is affected by 11 out of the 14 transitions in the model (Sn (T13) = 0.79). Indeed, the value 0.79 could be interpreted as 79% of the transitions in the model affects transition T13. On the other hand, T13 does not affect other

Table 3. Severity and sensitivity measures for transitions in fuel pump model

Transition	Sv(T)	Sn(T)
T1	0.43	0.00
T2	0.00	0.00
T3	0.64	0.00
T4	0.64	0.00
T5	0.29	0.21
T6	0.29	0.21
T7	0.21	0.14
T8	0.29	0.21
T9	0.21	0.29
T10	0.21	0.50
T11	0.07	0.50
T12	0.07	0.50
T13	0.00	0.79
T14	0.00	0.00

transitions in the model (Sv(T13) = 0), so its change is not expected to propagate to any other transitions (assuming that the change doesn't involve setting the value of a variable which was not previously defined at T13).

5 Exploratory Study

In this section we investigate the effectiveness of the two measures in identifying critical transitions in the fuel pump model. To do so, we write a tool which randomly generates a hundred arbitrary changes on the fuel pump model. Then for each transition, we check how many times the transition was touched by the 100 changes, and how many times it touched other transitions.

Finally, we track how many times each transition in the model was touched by the 100 changes. In addition, for each transition Ti, we track how many other transitions where touched by the change of Ti.

The results obtained after running the tool to apply 100 changes on the model are provided in Table 4. The first column of the table which is labeled as "Changed" shows how many times a change was applied on a particular transition. The second column "Touched", shows how many times a transition was touched by a change applied elsewhere in the model, and the third column labeled as "Touching" shows how many times a transition was touched by a change applied to the transition with interest. For example, for transition T1 we can see that it is changed 7 times out of the 100 changes applied to the model. For all of the 100 changes, it was never touched by a change applied to any of the other transitions in the model, while its change touched other transitions 42 times.

The results of the experiment show that the transition that was most frequently touched by a change is T13 which was touched by a change for 80 times. The transition

Table 4. Results of the exploratory study

Transition	Changed	Touched	Touching
T1	7	0	42
T2	6	0	0
T3	8	0	72
T4	9	0	81
T5	8	24	32
T6	2	24	8
T7	4	17	12
T8	6	23	24
T9	11	32	33
T10	6	46	18
T11	7	49	7
T12	12	46	12
T13	3	80	0
T14	11	0	0
SUM	100	341	341

whose change touched a large number of other transitions in the EFSM model was T4 which touched other transitions for 81 times.

The transitions that were not frequently touched by a change were T1, T2, T3, and T4. While the transitions that didn't touch other transitions in the model were: T2, T13, and T14.

These results were consistent with the severity and sensitivity measures estimated for each transition in the model. Indeed, the transitions that have high severity values, touched other transitions more frequently than transitions with lower severity values. For example, T1 (Sv = 0.43), T3 (Sv = 0.64), T4(Sv = 0.64) touched other transitions for 41, 72, and 81 times respectively. While transitions having the severity value of zero (namely T2, T13, and T14) didn't touch any other transition in the model.

Similarly, transitions that have high severity value were touched by a change more frequently than transitions with lower severity values. For example, T1, T2, T3, and T4 have a sensitivity value of zero, and during the experiment they were not touched by any change applied to other transitions in the model. While transitions T10 (Sn = 0.50), T11 (Sn = 0.50), T12(Sn = 0.50), and T13 (Sn = 0.79), were touched by a change for 46, 49, 46, and 80 times respectively.

6 Threats to Validity, Limitations, and Future Work

The major threat to validity for the presented study is the use of a single model (Fuel Pump Model) to test the effectiveness of the two measures. To handle this limitation, the study considered a large number of random changes to be applied to the model. Additionally, it is worth mentioning that the purpose of the current study is simply to illustrate the potential effectiveness of the two measures, while an extended study is planned in the future to cover a larger number of models with different sizes and different characteristics.

Another limitation of the proposed approach is the assumption that the probability of applying a change to any single transition in the model is the same for all transitions in the model. This assumption considers that all transitions in the model have approximately, comparable complexity. While this assumption can be true for some models, other probability metric should be considered for models that don't satisfy this assumption. For example, one can assume that a transition that has a complex condition composed of several sub-conditions joined with logical OR has higher probability to undergo a change compared to a transition that doesn't have any condition associated to it. Consequently, this assumption should be taken into consideration when the metrics are applied. Joining, the results obtained from the metrics with a human expert who can confirm the criticality of a transition given its complexity would generate more reliable conclusions.

7 Conclusion

In this paper we presented two model-based measures that can be very useful during the software maintenance. The severity of an EFSM transition estimates how severe a change applied to the transition can be. The scope of the severity of the change is measured in terms of the number of transitions to which the change may propagate. The propagation of the change is measured using data and control dependencies between transitions in the EFSM model. The sensitivity of an EFSM transition to a change applied to a model is also measured using model dependencies. However, when looking at the sensitivity, we investigate how often a change applied to other transitions in the model will propagate to the transition under consideration.

System development teams can use these two measures as a way to better estimate the severity of a change applied to the model, and to identify the transitions that will more frequently be affected by a change.

In future research, we will apply the measures to a larger set of models, and we will experiment with actual changes instead of random changes.

Acknowledgement. This research is sponsored by Kuwait Foundation for the Advancement of Science (KFAS) as part of project P116-18QA-01.

References

1. Korel, B., Singh, I., Tahat, L., Vaysburg, B.: Slicing of state -based models. In: IEEE International Conference on Software Maintenance, pp. 34–43 (2003)
2. Korel, B., Tahat, L.: Understanding modification in state-based system. In: Proceedings of the 12th IEEE International Conference on Program Comprehension (IWPC 2004), London, UK, pp. pp. 246–250, September 2004
3. Tahat, L., Almasri, N.: Identifying the effect of model modifications in State-Based models and systems. J. Adv. Comput. Sci. Technol. **2**(1), 9 (2013)

4. Cheng, K., Krishnakumar, A.: Automatic functional test generation using the extended finite state machine model. In: The 30th ACM/IEEE Design Automation Conference, pp. 86–91 (1993)
5. Vaysburg, B., Tahat, L., Korel, B., Bader, A.: Automating test case generation from SDL specifications. In: Proceedings of 18th International Conference on Testing Computer Software, pp. 130–139 (2001)
6. Korel, B., Tahat, L., Vaysburg, B.: Model based regression test reduction using dependence analysis. In: Proceedings of the International IEEE Conference on Software Maintenance, pp. 214–223 (2002)
7. Vaysburg, B., Tahat, L., Korel, B.: Dependence analysis in reduction of requirement based test suites. In: Proceedings of the ACM International Symposium on Software Testing and Analysis, pp. 107–111 (2002)
8. Korel, B., Tahat, L., Harman, M.: Test prioritization using system models. In: Proceedings of IEEE International Conference on Software Maintenance, Budapest, Hungary, pp. 559–568 (2005)
9. Tahat, L., Korel, B., Harman, M., Ural, H.: Regression test suite prioritization using system models. J. Softw. Testing Verification Reliab. **22**(17), 481–506 (2012)
10. Tahat, L., Korel, B., Hartman, M., Ural, H.: Regression test suite prioritization using system models. Softw. Test. Verification Reliab. J. (STRV) **27**(17), 481–506 (2012). Wiley Inter science, special edition on Model-Based Testing
11. Yoo, S., Harman, M.: Regression Testing Minimisation, Selection and Prioritisation - A Survey. Department of Computer Science, King's College London, Technical report: TR-09-09, October 2009
12. Thomas, S.W., Hemmati, H., Hassan, A.E., Blostein, D.: Static test case prioritization using topic models. Empirical Softw. Eng. **19**(1), 182–212 (2014)
13. Tahat, L., Korel, B., Koutsogiannakis, G., Almasri, N.: State-based models in regression test suite prioritization. Softw. Qual. J. **25**(3), 703–742 (2016)
14. Stamatis, D.H.: Failure mode and effect analysis: FMEA from theory to execution. ASQ Quality Press, Milwaukee (2003)
15. Li, B., Sun, X., Leung, H., Zhang, S.: A survey of code-based change impact analysis techniques. Softw. Test. Verification Reliab. **23**(8), 613–646 (2013)
16. Briand, L., Labiche, Y., O'Sullivan, L., Sowka, M.: Automated impact analysis of UML models. J. Syst. Softw. **79**, 339–352 (2006)
17. Almasri, N., Tahat, L., Korel, B.: Toward automatically quantifying the impact of a change in systems. Softw. Qual. J. **25**(3), 601–640 (2017)
18. Ferrante, K., Ottenstein, K., Warren, J.: The program dependence graph and its use in optimization. ACM Trans. Program. Lang. Syst. **9**(5), 319–349 (1987)
19. Bohner, S.A., Arnold, R.S.: Software Change Impact Analysis. IEEE Computer Soc. Press, Los Alamitos (1996)