ANALYSIS OF ENGINEERING CHANGE REQUESTS USING MARKOV CHAINS

Í. Ö. Arnarsson, E. Gustavsson, J. Malmqvist and M. Jirstrand

Abstract

Engineering change requests are important and plentiful in the product development process to enhance a product. In this paper we use Markov chains on ECRs in a large product development project and display the results in a Markov chain DSM. The DSM shows statistical probability of a transition pathway for an industrial design process and together with engineering domain knowledge we identify patterns and improvement opportunities. It turns out that 8% of ECRs are closed directly after creation, most common pathway is not followed in early statues and status iterations are seen in the DSM.

Keywords: engineering change, product development, design models, Markov chains, process analysis

1. Introduction

Product development projects often need to make an enhancement to a product. The documents used to initiate such a change process are well known under the name of Engineering Change Requests (ECRs). The motive for making engineering changes can be to improve, enhance or adapt the product to identified opportunities or issues (Pikosz and Malmqvist, 1998). ECRs are used to specify desired product changes, and keep track of the evolution of a requested change from initiation, search for solution, verification, and decision on acceptance. ECRs thus contain both product- and process related information.

The ECR process is a part of the engineering change process, corresponding to the first four states of Jarratt et al.'s (2011) generic engineering change management (ECM) process (Figure 1). (The last two states are known as the engineering change order process). Hamraz et al. (2013) review methods for ECM and identify 25 key requirements on such methods, including various components of process model building and use.

The engineering change process takes place under different phases of product development and different authors have proposed generic models. Jarratt et al. (2011), Figure 1 suggested a six state engineering change process that begins with the ECR to be raised, identification of solutions, risk assessment of solution(s), selection and approval of solution, implementation and finally a review of the change.

An ECR goes through multiple states during the process and can change ownership, meaning that ECRs can have many pathways during the resolution process of the change request. Arnarsson et al. (2017) found that engineers expressed a need to analyse these pathways and lead times to make improvements of the ECR process.

ECRs are logged and stored in databases. Single large development projects can contain up to tens of thousands of ECRs (Arnarsson et al., 2017). Due to the number of ECRs that need processing, it is crucial to both rationalize and to secure the quality of the ECR process.

The idea with this work is to use design process modelling and measurement in order to identify recurring ECR process patterns and to figure out what can be improved within the process. Such rationalization should be based on real measurements of ECR process performance. Companies are

realizing the large amounts of data is stored and that data could be uses for continuous improvements by supporting decision making in new projects (Davenport et al., 2010; Wu et al., 2014). These masses of data include ECR reports. Measuring the ECR process is not trivial, however. Different types of ECRs are routed in different pathways through the system. Hence, there is no single ECR process, but rather many. Thus, the ECR can be characterized as a stochastic process, which during its lifetime evolves between discrete ECR states (under investigation, testing, solved).

It is difficult to get an overview and understanding of such processes. However, Markov chains (Gilks et al., 1995) are probabilistic models used for modelling how discrete state processes evolve over time. Therefore, the paper explores the potential of applying Markov chains for analysing ECR processes, including the use of Markov chain matrices for stochastic process visualization.

This paper is the third part in a three-part project aimed at utilizing design analytics tools such as machine learning (methods where computers learn and find patterns and correlations in data) and big data analytics (statistical methods for large amounts of data) for analyzing and guiding product development projects. Three components of data mining and data analytics are in focus in each part of this project; i) data, ii) domain knowledge, and iii) mathematical tools such as algorithms, optimization, and statistical models (Fayyad et al., 1996). The first paper focused on data and investigated how a database containing the Engineering Change Requests (ECRs) that were issued during a product development project could be explored and visualized (Arnarsson et al., 2016).
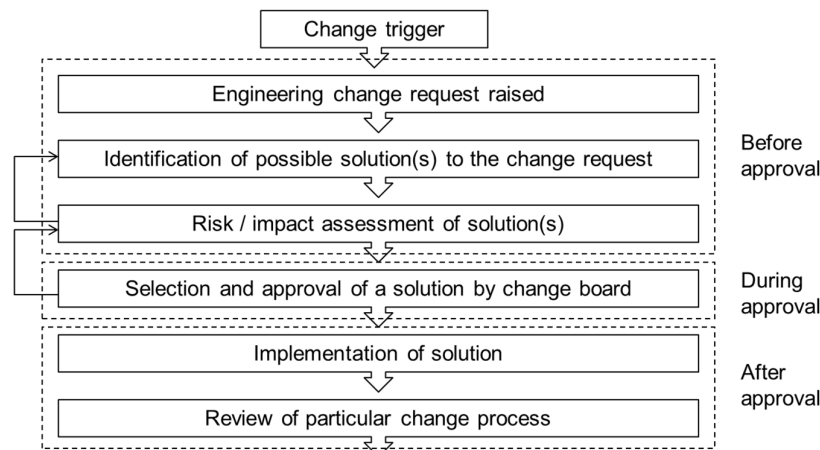
Figure 1. A generic engineering change process (Jarratt et al., 2011)
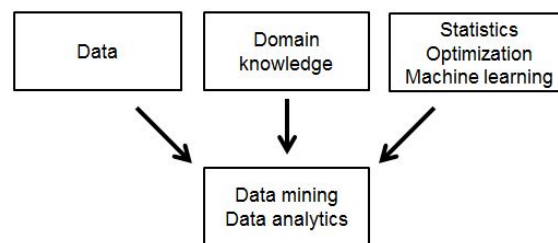
Figure 2. Schematic illustration of data mining and data analysis components

The second paper (Domain knowledge) investigated product developers' need for information about ECRs during development projects, The research identified specific questions that product developers may have regarding ECR data so analytics can be applied effectively (Arnarsson et al., 2017). The second paper thus identified a specific need to perform ECR process analysis. It was decided to demonstrate in this paper how ECR process analysis can be performed and to present the results.

To this end, this paper applies the third component (Mathematical tools) in order to perform data analytics (see Figure 2). Specifically, the following research questions are addressed:
1. *How can Markov chains be applied on ECR databases for analysis?*
2. *What patterns can we observe in the Markov chain DSM?*

DESIGN ORGANISATION AND MANAGEMENT

3. What improvements can these observations lead to?

4. What are the efforts, benefits and limitations of using Markov chain DSM for ECR process analysis?

2. Earlier work

Let us now review earlier work on design process modelling & simulation, ECR processes and Markov chain models.

The ECR process is a kind of design process. Wynn and Clarkson (2018) surveyed available design process & simulation models and illustrate the rich variety of models. Wynn and Clarkson argue that that detailed, task-based models of design processes can support design, management and improvement of "meso-level" processes such as the ECR process. Due to the complexities in processes no single model can fit all. However, design structure matrices (DSM) (Steward, 1981; Eppinger and Browning, 2012; Browning, 2016) have successfully been used to construct task-based models of design processes, including stochastic factors. Design structure matrices offer support for both qualitative and quantitative analysis of processes, e.g., visualization of processes as well as computation of process lead times.

On the other hand, Markov chain models (Norris, 1998; or Gilks et al., 1995) have many applications in real life situations where one wants to investigate and understand processes evolving between different discrete states. Markov chain models have previously been utilized for analysing product development processes (Figure 3) in, e.g., Ahmadi et al. (2001), where the authors employ Markov chains for developing procedures to minimize iterations during the development process which adversely affect development time and costs. Cho and Eppinger (2001) use Markov chains to simulate a product development process with the aim of providing better project planning and control and Dong (2002) tries to employ ideas from Markov chain models for understanding organizational interactions during product development processes.

However, earlier work on DSM and Markov chain have typically been applied to situation- or system-specific design processes, e.g., a brake design process (Smith and Eppinger, 1997). Smith and Eppinger (1997) further note that generating reliable data for a DSM is challenging, and will require additional effort for each new system-specific design process that is modelled. Using the Markov chain method to create a Markov chain model is essentially a DSM.
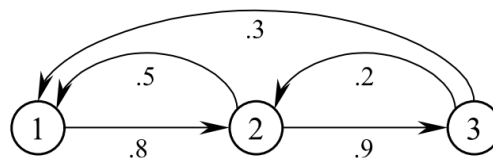
Figure 3. Illustration of three activity stage model with dependency relations between activities (Ahmadi et al., 2001)

The ECR process, by contrast, relies on a standardised process that is applied to a multitude of different systems and components. The adaptations made, e.g., with respect to estimated severity of the change, are dynamic in the sense that they are decided on by the actors during process execution. The standardisation constitutes an opportunity in this case. ECR data is stored in corporate PDM databases in standard formats: Arnarsson et al. (2016) showed how data mining and analytics tools could be applied to visualise and explore ECR databases, showing, for example, change request frequency over time in a design process.

Combining ECR databases, data mining and Markov chain matrices enables us to quickly construct DSM models of the ECR process. The aim is to gain more insights into the entire product development process. For example, by analysing ECR processes from a Markov chain model perspective, one can draw conclusions regarding how product developers are handling issues during projects, and detect similarities and differences between different product development teams. Comparisons can be done on success factors and outliers by investigating why departments differ in the process that could identify e.g. communication issues between departments, improve solving efficiency, and handling issues of ECRs. Such analyses are not industry practice today.

In conclusion, statistical ECR process DSM analyses have not been performed so far despite that the data is available. Engineers would like to identify how the resolution process of an ECR looks like in

the big picture to improve the process. There exists no previous research that specifically analyses the data from ECR states in product development projects. The literature cited above has identified questions that can aid the analysis of ECRs. This paper addresses this research gap, and aims to apply Markov chain modelling and analysis to the ECR database in product development.

3. Research approach

The study is based on a product development database containing ECRs from a real industrial commercial vehicle development project. The Markov chain models approach enabled a way to analyse large set of ECRs together that can provide new data driven insights, compared to manually analysing a few ECRs at a time.

The data within the ECRs contain states transitions that include timestamps (data and time) and have the possibility to take on more than 30 unique states. Each ECR takes up a state for different phase in the resolution process starting from "ECR created" to "ECR solved". Each ECR contains data about all the historical state transitions it has taken under the resolution process, and if an ECR has changed owner during the process.

A Markov chain is a type of stochastic process where the transition probabilities between the available states fulfil the Markov property, meaning that the probability of evolving from one state to another only depends on the current state. This implies that the process is in a sense memoryless, and does not care about the history of the process. For the case when we consider an ECR as a discrete stochastic process, this means that when an ECR is in a specific state at a specific time point, the probability of being transferred from that state to another only depends on the state of the ECR is in at that time point. Because we assume that the Markov property (memoryless property) hold.

We can describe the process by using a Markov transition matrix which describes the probabilities of evolving from one state to another (i.e., the element (x, y) in the matrix represents the probability of the ECR evolving from state x to state y. For the product development project considered here, we have information regarding when and between which states the ECRs evolved during the project. By utilizing the available information we can estimate the Markov transition matrix for all ECRs in the project. The matrix is estimated by counting the number of times ECRs has transitioned between the states and then normalizing the matrix by rows.

The Markov transition matrix can then be utilized for understanding the normal flow of ECRs, i.e., what seems to be the most common transition patterns for the ECRs. One can also detect issues in the process by, for example, finding states where many ECRs are reissued or closed prematurely. Another approach is to create Markov transition matrices for subgroups of ECR types, or for subgroups of the company organization. By utilizing the different transition matrices, one can detect differences and similarities between the subgroups, and draw conclusions.

4. Results

The data science capabilities in this research are described in a methodology for understanding the whole process and presented in Figure 4 below.

Figure 4. Flowchart of the methodology

4.1. ECR data is extracted from database, cleaned and prepared

The data for the ECRs is stored in a corpus databases. It is then extracted into Excel-files and loaded into Python for cleaning and preparation of the data. The Markov chain model was estimated from data using Python. The Markov chain model is therefore used to model the design process to analyse data from earlier projects to potentially reveal process issues and patterns for improvements in new projects.

DESIGN ORGANISATION AND MANAGEMENT

4.2. Markov chain model estimated from data and visualized in a matrix

The analysis of ECRs and their state transitions are presented in a Markov chain DSM (Figure 5). The project analysed contained 17,745 ECRs with 100,508 state transitions. Due to data privacy reasons fictional numbers have been introduced into the matrix that still reveal similar patterns as with the initial numbers. The Markov chain model is a transition probability matrix with element (x, y) in the table, where x is the row and y is the column representing the probability that an ECR transitions to state y when it is situated in state x. For example, there is a probability of 0.41 for an ECR transitioning to state 19 [ECR with solving responsible] when the ECR has state 10 [ECR distributed]. Only probabilities above 0.01 are marked in the matrix and the probabilities above 0.1 are highlighted with a grey box.

There are 34 ECR process states used in the Markov chain DSM that are organized into 8 categorizes. Each state has a numerical name and description, were the first numerical digit indicates what states belong to the same category. Categories are also colour coded in the matrix for additional visual recognition. Categories are outlined below in Table 1:

Table 1. Main categories for the process states used in the Markov chain DSM

1 ECR created	5 Verification phase
2 ECR re-issued	6 External ECR
3 Identification of possible solution	7 ECR outside of project scope
4 Verification by factory	8 ECR solved

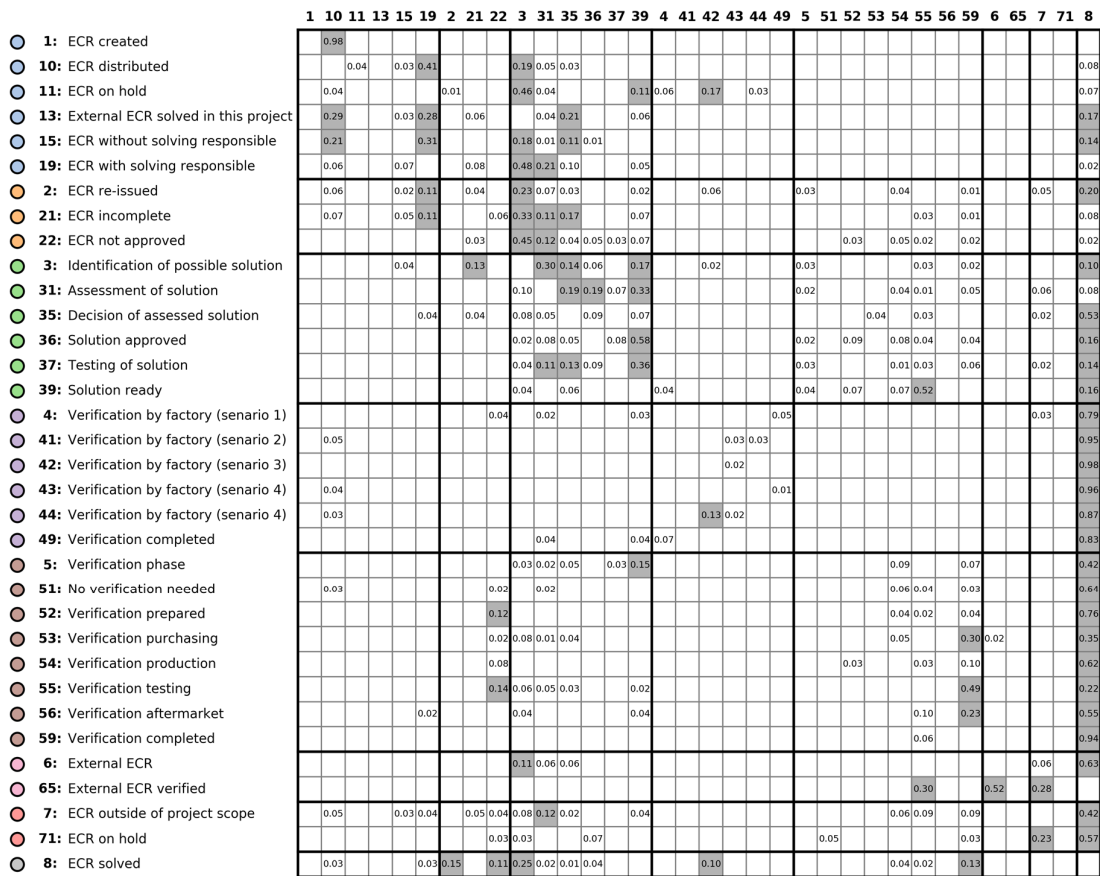| State	1	10	11	13	15	19	2	21	22	3	31	35	36	37	39	4	41	42	43	44	49	5	51	52	53	54	55	56	59	6	65	7	71	8
1: ECR created	0.98																																	
10: ECR distributed		0.04		0.03	0.41					0.19	0.05	0.03																						0.08
11: ECR on hold	0.04					0.01				0.46	0.04				0.11	0.06		0.17		0.03														0.07
13: External ECR solved in this project	0.29			0.03	0.28			0.06			0.04	0.21			0.06																			0.17
15: ECR without solving responsible	0.21				0.31					0.18	0.01	0.11	0.01																					0.14
19: ECR with solving responsible	0.06		0.07						0.08	0.48	0.21	0.10			0.05																			0.02
2: ECR re-issued	0.06		0.02	0.11				0.04		0.23	0.07	0.03			0.02			0.06				0.03				0.04		0.01				0.05		0.20
21: ECR incomplete	0.07			0.05	0.11		0.06			0.33	0.11	0.17			0.07												0.03	0.01						0.08
22: ECR not approved						0.03				0.45	0.12	0.04	0.05	0.03	0.07						0.03				0.05	0.02	0.02							0.02
3: Identification of possible solution			0.04		0.13						0.30	0.14	0.06		0.17			0.02				0.03				0.03	0.02							0.10
31: Assessment of solution										0.10	0.19	0.19	0.07		0.33							0.02				0.04	0.01	0.05			0.06			0.08
35: Decision of assessed solution				0.04				0.04		0.08	0.05			0.09	0.07									0.04			0.03				0.02			0.53
36: Solution approved										0.02	0.08	0.05		0.08	0.58							0.02			0.09	0.08	0.04	0.04						0.16
37: Testing of solution										0.04	0.11	0.13	0.09		0.36							0.03				0.01	0.03	0.06			0.02		0.14
39: Solution ready										0.04		0.06				0.04						0.04			0.07	0.07	0.52							0.16
4: Verification by factory (scenario 1)								0.04		0.02					0.03						0.05											0.03		0.79
41: Verification by factory (scenario 2)	0.05																		0.03	0.03														0.95
42: Verification by factory (scenario 3)																		0.02																0.98
43: Verification by factory (scenario 4)	0.04																			0.01														0.96
44: Verification by factory (scenario 4)	0.03																		0.13	0.02														0.87
49: Verification completed											0.04					0.04	0.07																	0.83
5: Verification phase										0.03	0.02	0.05		0.03	0.15										0.09		0.07							0.42
51: No verification needed	0.03							0.02			0.02														0.06	0.04	0.03							0.64
52: Verification prepared										0.12															0.04	0.02	0.04							0.76
53: Verification purchasing										0.02	0.08	0.01	0.04												0.05			0.30	0.02					0.35
54: Verification production										0.08														0.03		0.03		0.10						0.62
55: Verification testing										0.14	0.06	0.05	0.03		0.02												0.49							0.22
56: Verification aftermarket				0.02						0.04					0.04												0.10	0.23						0.55
59: Verification completed																											0.06							0.94
6: External ECR										0.11	0.06	0.06																				0.06		0.63
65: External ECR verified																											0.30			0.52	0.28			
7: ECR outside of project scope	0.05			0.03	0.04		0.05	0.04		0.08	0.12	0.02			0.04										0.06	0.09	0.09							0.42
71: ECR on hold							0.03	0.03						0.07										0.05			0.03						0.23	0.57
8: ECR solved	0.03				0.03	0.15				0.11	0.25	0.02	0.02	0.01	0.04					0.10					0.04	0.02	0.13							

Figure 5. The transition probability matrix (Markov chain DSM) for all ECRs in a single large product development project

4.3. Matrix analysed together with engineers (domain knowledge)

The Markov chain model was visualized in a matrix that was evaluated together with engineers who had domain knowledge about the process and could identify areas of interest in the process.

The matrix reveals a most common pathway for ECRs that can be read out by starting at state 1 on y-axis and read the highest probability for a transition on the x-axis. If we then repeat the previous process but start from the next state on the y-axis that had the highest previous probability we can then trace the most likely path ending at state 8. The pathway is therefore: 1 → 10 → 19 → 3 → 31 → 39 → 55 → 59 → 8. When the Markov chain DSM was constructed, engineers with domain knowledge were asked to draw conclusions based on their expertise that could reveal facts and patterns in the ECR process. Here below are four illustrations are demonstrated from the test of the model and example of what conclusions can be drawn from the matrix base on engineering domain knowledge.

4.3.1. Case 1: ECR created, distributed and closed. State transitions: 1 → 10 → 8

This case (Figure 6) shows when an ECR is created and after being distributed it is directly closed by transitioning into state 8. The matrix shows that 8% of all ECRs who have been created are closed directly and in bigger project with tens of thousands ECR this is a substantial amount of work that could be non-value adding.
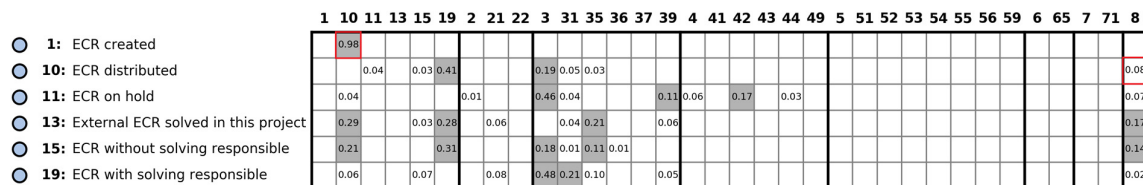
	1	10	11	13	15	19	2	21	22	3	31	35	36	37	39	4	41	42	43	44	49	5	51	52	53	54	55	56	59	6	65	7	71	8
1: ECR created	0.98																																	
10: ECR distributed		0.04		0.03	0.41					0.19	0.05	0.03																						0.08
11: ECR on hold	0.04				0.01					0.46	0.04				0.11	0.06		0.17		0.03														0.07
13: External ECR solved in this project	0.29			0.03	0.28		0.06			0.04	0.21				0.06																			0.17
15: ECR without solving responsible	0.21				0.31					0.18	0.01	0.11	0.01																					0.14
19: ECR with solving responsible	0.06			0.07			0.08			0.48	0.21	0.10			0.05																			0.02

Figure 6. Highlighting in red state transitions for case 1

The benefits with this insight is that now we can estimate the amount of ECRs that were created as a change request in the system, but instead of working with the ECRs they got closed directly. One can ask what the leading cause for this is and engineering knowledge points towards that this can be a poor request of an ECR discarded by a more knowledgably person or related to the quality of the content in the ECR. Possible improvements are to extract these ECRs and find the root cause for why the requested ECRs are turned down.

4.3.2. Case 2: "ECR with solving responsible" state skipped and transitions straight to "Identification of possible solution". State transitions: 10 → 19 → 3 vs. 10→ 3

After an ECR has been written and transitions to a correct solving responsible most of the ECRs go into a state where it is checked that they have reached the correct solving responsible. This check should take place before the investigation phase starts. This is the correct way of working which is confirmed in the matrix. What can be seen is that the next probable action when distributing an ECR is that confirmation of a solving responsible is skipped and the ECR goes straight into identification of possible solution. This process deviation implies a risk that the wrong solving responsible is assigned to an ECR (Figure 7).
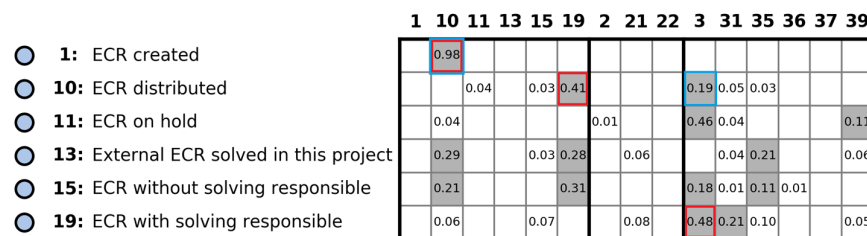
	1	10	11	13	15	19	2	21	22	3	31	35	36	37	39
1: ECR created	0.98														
10: ECR distributed			0.04		0.03	0.41				0.19	0.05	0.03			
11: ECR on hold		0.04					0.01			0.46	0.04				0.11
13: External ECR solved in this project		0.29			0.03	0.28	0.06			0.04	0.21				0.06
15: ECR without solving responsible		0.21				0.31				0.18	0.01	0.11	0.01		
19: ECR with solving responsible		0.06			0.07		0.08			0.48	0.21	0.10			0.05

Figure 7. Highlighting in red and blue state transitions for case 2

The system offers a way to identify that an ECR has a solving responsible that will be responsible for leading the ECR in the identification of a possible solution. This is fairly early in the ECR process states and it is know that most ECRs should follow the same pathway. However, there is room for improvement to make sure everyone is consistent in their work. ECRs are sometimes issued to an incorrect department who is not in the right responsible area for a change and might therefore not start working on a solution risking of prolonging the solving time. The Markov transition matrix can help identify cases were the solving responsible is not assigned to the ECR.

DESIGN ORGANISATION AND MANAGEMENT

4.3.3. Case 3: "Identification of possible solution" state skipped and transitions straight to "Assessment of solution". State transitions: 19 → 3 → 31 vs. 19 → 31

When an ECR has reached solving responsible to work on it the next transition is identification of a possible solution. The most common process is to transit the ECR into identification of a solution and note that a solution will be worked on and that take the state of announcing that work has started. The deviation here form the most common pattern is that part of the most common pathway is deviated from. Meaning that work has started on the solution, but the state of identifying a solution has been skipped (Figure 8).

	1	10	11	13	15	19	2	21	22	3	31	35	36	37	39
1: ECR created	0.98														
10: ECR distributed		0.04		0.03	0.41						0.19	0.05	0.03		
11: ECR on hold	0.04					0.01					0.46	0.04			0.11
13: External ECR solved in this project	0.29			0.03	0.28		0.06				0.04	0.21			0.06
15: ECR without solving responsible	0.21				0.31						0.18	0.01	0.11	0.01	
19: ECR with solving responsible	0.06			0.07			0.08			0.48	0.21	0.10			0.05
2: ECR re-issued	0.06			0.02	0.11		0.04				0.23	0.07	0.03		0.02
21: ECR incomplete	0.07			0.05	0.11				0.06		0.33	0.11	0.17		0.07
22: ECR not approved							0.03			0.45	0.12	0.04	0.05	0.03	0.07
3: Identification of possible solution					0.04			0.13			0.30	0.14	0.06		0.17
31: Assessment of solution											0.10	0.19	0.19	0.07	0.33
35: Decision of assessed solution					0.04		0.04				0.08	0.05		0.09	0.07
36: Solution approved											0.02	0.08	0.05	0.08	0.58
37: Testing of solution											0.04	0.11	0.13	0.09	0.36
39: Solution ready											0.04		0.06		

Figure 8. Highlighting in red and blue state transitions for case 3

Since most ECRs go through the process of identifying a possible solution and then the assessment of that solution it looks like a process issue when a solving responsible moves the ECR into an assessment of the solution without first identifying a possible solution. These two states should likely come in the order like they appear in the matrix because a possible solution should be noted before the assessment of it begins.

4.3.4. Case 4: Iterations in the ECR transition process

Iterations in the transition process are when a transition return to a previous transition Iterations can be detected in the model, look at state 10 [ECR distributed] e.g. three iterations can be seen in the ECR-created group, one in ECR re-issued group and one in the Identification of possible solution. Based on the known fact that late process changes are costly one can assume that the same applied for iterations later in the process. Examples of iterations in the transitions are noted in the list below and an illustration of three cases from ECR-created are show in Figure 9:

- ECR-created: 10→11→10, 10→15→10, 10→11→2→10.
- ECR re-issued: 10→11→2→7→10.
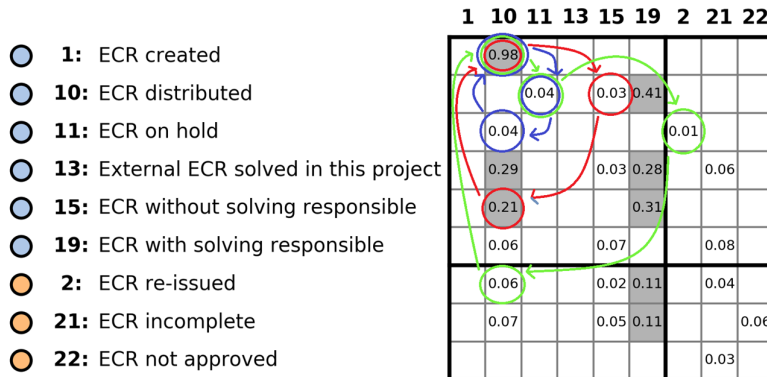- Identification of possible solution: 10→31→10.

	1	10	11	13	15	19	2	21	22
1: ECR created	0.98								
10: ECR distributed		0.04		0.03	0.41				
11: ECR on hold	0.04					0.01			
13: External ECR solved in this project	0.29			0.03	0.28		0.06		
15: ECR without solving responsible	0.21				0.31				
19: ECR with solving responsible	0.06			0.07			0.08		
2: ECR re-issued	0.06			0.02	0.11		0.04		
21: ECR incomplete	0.07			0.05	0.11				0.06
22: ECR not approved							0.03		

Figure 9. Illustration of three iterations that take place early in the process [ECR-created] around state 10 [ECR distributed]

5. Discussion

The study showed how Markov chain models can be used for modelling and analysis of real design processes, focusing on what conclusions and improvement opportunities can be obtained with such a model. The aim is to identify patterns from ECRs with a Markov chain model that are difficult to analyse on case-by-case level, but an ad hock approach like DSM can contribute to improvements in the product development process. We see that usage of Markov chain modelling can be generalized for other ECRs, since ECRs are widely used in industry and any shortcoming seen in relation to that would be on gathering the data form databases.

Let us now consider the stated research questions in relation to the findings and additional use cases for future work.

5.1. How can Markov chains be applied on ECR databases for analysis?

Markov chains and Markov transition matrices can be used for understanding the flow of ECRs through the different ECR states. By creating the Markov chain transition matrix one can use it directly to draw conclusions by noting common patterns or finding anomalies. An even more powerful utilization of the Markov chains is that it is possible to compare transition matrices for different departments in the organization and capture differences in work flow of the various departments. We have found that the matrix matches our initial expectations of gaining a holistic overview of the process and the visualization provides a way to observe patterns.

5.2. What patterns can we observe in the Markov chain DSM?

The matrix helps identifying the most common transition process for the states that an ECR takes. That insight can be beneficial and can open up a discussion if an ECR is allowed to deviate from a pathway in the process or is allowed to take up any state of choice in the process. Early in the process before the approval of a solution ECRs should follow a fairly similar pathway of creating the ECR and making sure it has a solving responsible to identify the solution. We found that the Markov chain DSM has the ability of identify those deviations. The conclusions are best drawn with the help of engineers who have domain knowledge about the different states in the process and can point out deviations and areas that are of interest for them. Another pattern is the iterations seen in the model and pointed out in the early transitions categories. More iterations can be seen throughout the model that take a bit more effort of tracking those pathways.

5.3. What improvements can these observations lead to?

We showed by using Markov chains, existing process issues in the system can be unveiled by observing transitions of the statues ECRs are assigned to. From observing the transitions newly created ECRs take, we discovered that 8% of all ECRs are directly closed after creation. Initiating an ECR takes resources and time, and if 8% of all generated ECRs are prematurely closed for this particular project that can be viewed as a waste of time and resources. This can be even more significant for larger development projects that might generate more than tens of thousands of ECRs.

In the early stages of ECRs, there is a process of distribution and assigning a solving responsible that sometimes seems to be handled in an inconsistent way. It is not mandatory to take the state of identifying a solving responsible and the probability matrix confirms that the most common pathway for ECRs is not followed for all ECRs in these early states. This inconsistency can be seen as a deviation in the process and could be made a mandatory state.

The same holds for the state after a solving responsible has been secured and the process of identifying a possible solution has started. In that case, most ECRs transition to investigation phase by noting that an identification of a possible solution is undergoing, but in this ECRs pathway that state is skipped and transitions instead into the next probable state of going straight into verification of the solution. Since this deviation is also in early states of ECRs when a solution has not been approved, ECR process should follow a similar pathway there. Regarding the iterations, the model can be used to point them out and ask the engineers why this occurs and how to possibly prevent reiteration of earlier states in efforts to improve the way of working with ECRs.

5.4. What are the efforts, benefits and limitations of using Markov chain DSM for ECR process analysis

The software used for implementation of the Markov chain DSM is Python. General skills or background in some kind of data scientist or analyst is needed in order to implement and create the model. The underlying data is for free, but cleaning and compiling efforts are needed that usually take less than a week. The time to create the model from data takes less than a day. However, we put time into implementing it, adjusting the axis labels and representation can take long depending on the desired outcome. Then there are other aspects that can be considered as user friendliness so that anyone can upload data and run the model.

The benefits of using such a model for ECR process analysis are that state transitions can be compiled in holistic visual representation that reveal patterns about pathways of these transitions.

Limitation factors are that it is usually not enough just to create the model but to apply domain knowledge in the analysing process. Engineering with domain knowledge are needed for evaluation of patterns and to point to pathways that are interesting for their work. Another limitation of using Markov chains is that one assumes that the Markov property (memoryless) holds for the ECRs. The path up to a specific status can of course affect the probabilities of transitioning to new states. However, an initial analysis made indicated that the Markov property is in most cases a valid assumption to make.

5.5. Additional use cases articulated by engineers

When reviewing the model with engineers, interests were expressed to introduce the quantity of ECRs into the model together with the probability of the path. That would give better insights into the amount of ECRs in each pathway.

Additionally, it was pointed out that introducing time, as in lead-time, for each process state would be of interest. Introducing time would generate an indicator of speed in the process states and together with domain knowledge they could point out what ECR states take longer time and potentially suggest improvement that could affect the total lead-time of an ECR from start to end.

Another use case is to utilize the Markov chain models for predicting how the ECRs will evolve in the future, i.e., to give predictions on the severity of an ECR and the time until it will be closed.

6. Conclusion and future work

It is clearly shown in this study that modelling ECR data with Markov chain DSM can be done and proves useful in analysing transitions of ECRs. We displayed four cases that were identified as key results from the study. First, the model identified the percentage of ECRs which had been closed directly after creation. Secondly, the model helped to identify that there is a tendency to skip the transition to identify a solving responsible, that is the most common way of working and transition instead into "Identification of a possible solution". Third, there is another tendency similar to the second case, to skip transition of "Identification of a possible solution" and transition straight into "Assessment of solution". Last, there is a possibility of identifying frequently occurring iterations in the transition process, meaning that we see loops in the model were ECRs transition back to an earlier point.

The key results show that the Markov chain model proves beneficial in analysing the ECR process. It is possible to learn from the transitions to improve writing of ECR so they are not closed after creation, make certain transitions mandatory if they should follow the most common paths in certain transition categories, and to show where iterations in the process can be found.

We see three possibilities for future research, as observed together with engineers when the model was reviewed.

- First, include the quantity of ECRs together with the probability of a transition.
- Second, to introduce time into the model to evaluate lead-time between transitions.
- Third, to make a risk assessment of ECRs and based on that predict the solving time of an ECR from the time it was created.

Acknowledgements

References

Ahmadi, R. Roemer, T.A. and Wang, R.H. (2001), "Structuring product development processes", *European Journal of Operations Research*, Vol. 130 No. 3, pp. 539-558. https://doi.org/10.1016/S0377-2217(99)00412-9

Arnarsson, Í.Ö., Gustavsson, E., Malmqvist, J. and Jirstrand, M. (2017), "Design analytics is the answer, but what questions would product developers like to have answered?", Proceedings of ICED'17 / the 21st International Conference on Engineering Design, Vol. 7, Vancouver, Canada, August 21-25, 2017, The Design Society, Glasgow, pp. 71-80.

Arnarsson, I.Ö., Malmqvist, J., Gustavsson, E. and Jirstrand, M. (2016), "Towards big-data analysis of deviation and error reports in product development projects", *Proceedings of NordDesign 2016*, *Trondheim, Norway, August 10-12, 2016,* The Design Society, Bristol, pp. 83-92.

Browning, T.R. (2016), "Design structure matrix extensions and innovations: A survey and new opportunities", *IEEE Transactions on engineering management*, Vol. 63 No. 1, pp. 27-52. https://doi.org/10.1109/TEM.2015.2491283

Cho, S.-H. and Eppinger, S. (2001), "Product development process modelling using advanced simulation", *Proceedings of DETC'01 / 2001 ASME Design Engineering Technical Conferences, Pittsburgh, Pennsylvania, USA, September 9-12, 2001*, ASME.

Davenport, T.H., Harris, J.G. and Morison, R. (2010), *Analytics at Work: Smarter Decisions, Better Results*, Harvard Business Press, Cambridge, MA, USA.

Dong, Q. (2002), Predicting and managing system interactions at early phase of the product development process, PhD thesis, Massachusetts Institute of Technology, USA.

Eppinger, S.D. and Browning, T.R (2012), *Design structure matrix methods and applications*, MIT Press, Cambridge, MA, USA.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996), "From data mining to knowledge discovery in databases", *AI Magazine*, Vol. 17 No. 3, pp. 37-54.

Gilks, W.R., Richardson, S. and Spiegelhalter, D. (1995), *Markov chain Monte Carlo in practice*, CRC press, Boca Raton, Florida.

Hamraz, B., Caldwell, N.H.M, Wynn, D.C. and Clarkson, P.J. (2013), "Requirements-based development of an improved engineering change management method", *Journal of Engineering Design*, Vol. 24 No. 11, pp. 765-793. https://doi.org/10.1080/09544828.2013.834039

Jarratt, T.A.W., Eckert, C.M., Caldwell, N.H. and Clarkson, P.J. (2011), "Engineering change: an overview and perspective on the literature", *Research in Engineering Design*, Vol. 22 No. 2, pp. 103-124. https://doi.org/10.1007/s00163-010-0097-y

Norris, J.R. (1998), *Markov Chains*, Cambridge University Press, Cambridge, UK

Pikosz, P. and Malmqvist, J. (1998), "A comparative study of engineering change management in three Swedish engineering companies", *Proceedings of the DETC'98 / 1998 ASME Design Engineering Technical Conferences, Atlanta, Georgia, USA, September 13-16, 1998*, ASME, pp. 78-85.

Smith, R.P. and Eppinger, S.D. (1997), "Identifying controlling features of engineering design iteration", *Management Science*, Vol. 43 No. 3, pp. 276-293. https://doi.org/10.1287/mnsc.43.3.276

Steward, D.V. (1981), "The design structure system: A method for managing the design of complex systems", *IEEE Transactions on Engineering Management*, Vol. 28 No. 3, pp. 71-74. https://doi.org/10.1109/TEM.1981.6448589

Wu, X., Zhu, X., Wu, G.Q. and Ding, W. (2014), "Data mining with big data", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26 No. 1, pp. 97-107. https://doi.org/10.1109/TKDE.2013.109

Wynn, D.C. and Clarkson, P.J. (2018), "Process models in design and development", *Research in Engineering Design*, Vol. 29 No. 2, pp. 161-202. https://doi.org/10.1007/s00163-017-0262-7

Ívar Örn Arnarsson, PhD student
Chalmers University of Technology, Sweden, Product development
Fjällvindsgatan 4 LGH 1504, 41714 Göteborg, Sweden
Email: varo@chalmers.se