Model-Driven Software Evolution: The Multiple Views

Madhavi Karanam, Anandrao Akepogu

***Abstract*— Software systems need to evolve, and systems built using model driven approaches are no exception. The role of the models are increasing and becoming more and more important in the software development and evolution. So, the Model-Driven Software Evolution (MoDSE) is prominent. Several stakeholders are involved in the process of evolution. These user groups (stakeholders) have different concerns relevant to the models to be evolved in MoDSE. Multiple views provide a means to visualize complex information and are also a way to fulfill the concerns of different user groups. In this paper the concept of multiple views for the MoDSE is introduced and shows how these multiple views addresses the stakeholders concerns. The different viewpoints are identified to construct the multiple views. This paper also shows an analytical support how the proposed views are closer to the Lehman's laws of software evolution.**

Index Terms— **Laws of software evolution, Model-Driven Software Evolution, stakeholders, views, viewpoints.**

I. Introduction

Software development is becoming more and more model centric, such that modeling languages have gained a much broader use. The introduction of Model Driven Engineering (MDE) needs a new style of evolution i.e. Model-driven Software Evolution. The first fundamental premise [1] for Model-Driven Software Evolution is that evolution should be a continuous process. The second premise is that reengineering of legacy systems to the model-driven of the paradigm should be done incrementally. Model driven engineering (MDE) introduces a multitude of languages that are themselves artifacts of the development process. Due to these multitude languages in MoDSE, there is a need to have the model interaction, integration, mapping and transformation. Further there should be possible views to capture this information about models during the evolution.

One or more stakeholders[1] are involved in MoDSE. Each stakeholder typically has interests in, or concerns relevant to that system. The ability of models to evolve gracefully is becoming a concern for many stakeholders. Due to different and interrelated models used to design an entire system in MoDSE, the concerns of stakeholders are changes in models, model elements, model migration, model transformation, model interaction and integration etc., So, there is a need to have migration, model transformation, model interaction and integration etc., So, there is a need to have viewpoints and multiple views which captures the stakeholders concern.

This paper proposes the viewpoints and multiple views for model driven software evolution and need for the same. Section 2 discusses the related work about views and viewpoints in software evolution. Section 3 provides the concepts such as viewpoints, views and concerns .It also discusses the identified viewpoints to construct the proposed multiple views. Section 4 lists the laws of software evolution and shows how the proposed views are closer to the mentioned laws. Section 5 outlines the conclusions and future work.

II. Related Work.

This section reviews related work about the views and viewpoints in the area of software evolution.

iACMTool [2] is a prototype tool to tackle the impact analysis and change management of analysis/design documents in the context of UML based development. This taxonomy consist views such as static (class diagram) view, interaction (sequence diagram) view and the state chart diagram view. These views support the UML models.

Christain F.J.Lange, et.al. proposed a framework [3],[4] consisting of UML model elements, their properties, and software engineering tasks, that form a basis to develop new views of UML models and related information. Based on this framework they proposed eight views to support different tasks. These views are UML based views, which maintains model evolution and quality.

Multiview Software Evolution (MVSE) is a UML based framework for Object-Oriented software [11].In MVSE, evolution of complex systems is a process in which transformations are successively applied to multiple views of software (represented by models), until objective criteria are satisfied. A stakeholder view reflects the perspective of a stakeholder on a systems application and behavior.

Manuscript received September 19, 2008. Accepted October 8,2008.

Madhavi Karanam, Research Scholar in Computer Science and Engineering , Jawaharlal NehruTechnological University, Hyderabad, Andhra Pradesh, India.(Phone: 91-9811377031, email: bmadhaviranjan@yahoo.com)
Dr.Anandrao Akepogu , Professor & Vice-Principal, JNTUCE, Jawaharlal Nehru Technological University, Anantapur, Andhra Pradesh, India.(Phone: 91-9000493404, email: akepogu@yahoo.co.in)

[1] A stakeholder is an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system. (IEEE 1471 standard)

In MVSE stakeholders initiate changes to systems and describe these changes in the context of stakeholder views.

Rene Keller et.al. [10] introduces the concepts of multiple viewpoints and multiple views in engineering change management.

Change prediction Method (CPM) tool implements the change prediction. Stephen Cook et.al. [12] proposed an approach to understand software evolution. This approach looks at software evolution from two different points of view. One is dynamic view point, which investigates software evolution trends in models and the second is static view point which studies the characteristics of software artifacts to see what makes a software system more evolvable.

The above mentioned tools and frameworks describe the multiple views and viewpoints for traditional software evolution and change management, in which only the UML models are considered. Hence, there are no such views and viewpoints exist in the literature, to address the stakeholders concerns during evolution of the different, unrelated models in MoDSE. The following section discusses the viewpoints, and construction of proposed views to satisfy the stakeholders concerns.

III. MULTIPLE VIEWS FOR MODSE.

This section outlines the three underlying concepts of our proposed views for MoDSE. The concepts are viewpoints, views, and concerns. Reasons for multiple views also listed in the section 3.2.

A. *Viewpoints, views and concerns*

Views offer visual representations of a model. View is a vested interest to visualize how the system is used and evolved. A *view*[2] in this paper is defined as a visible projection of a model to fulfill a stakeholders concern, which is an evolution task. Concerns are those interests which pertain to the system's development and evolution.[13].In MoDSE, the concerns of stakeholders are considered as changes in models and model elements, model migration, model transformation, model interaction and integration, mapping etc.,

IEEE Standard also allows for the definition of an arbitrary number of views as well as viewpoints [14]. A *viewpoint*[3] defines a number of important aspects and *concerns* that are addressed by that *viewpoint. .A* viewpoint establish the conventions by which a view is created, depicted and analyzed. In this way, a view conforms to a viewpoint. So, the views are constructed from the identified viewpoints which address the stakeholders concerns. The three underlying concepts such as viewpoints, views, concerns and their relations are illustrated in Figure 1.

B. *Reasons for multiple views*

There are two reasons for using multiple views [11]:

- ➢ The amount of information in a complex product is too large to be displayed in one single graph. The information has to be broken down into smaller chunks that can be visualized and analyzed much easier. Different graphs (diagrams or models) can show different information, revealing structure that cannot be shown in one diagram. Some representations are good or some purposes and not for others.
- ➢ Different people are involved in the development process. This user group come from different objective world and has a different background and task focus and sees the productive in a different way. So these people have different viewpoints and demand different views on the product data.

C. *Proposed Views*

MoDSE consists of multitude of languages. There should be an interaction, integration, mapping and transformations etc. between these different models as well as code also. The stakeholders have various roles as well as concerns regarding the creation and evolution of the models in the system. To resolve this, the multiple views are proposed which addresses the stakeholders concerns and also conforms to view points.

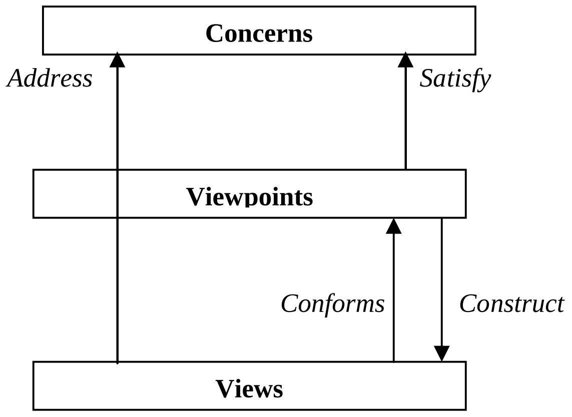

Figure 1. The three underlying concepts and their relations

(i) Context View

Models define what is variable in a system. The context of model element consists of model and all model elements relates to it. The elements are scattered over different models. It often occurs that only a limited number of model elements can be viewed at a time and understanding the entire model and its elements at a stretch is not an easy task. To understand a model completely, it might be necessary to know its context. Therefore, *Context View* is proposed. The context view is shown in Figure 2. The model element whose context is viewed in a context view is centered. All model elements that are directly related to the particular model elements are viewed around the model element, for example, class and its subclasses as shown in Figure 2.
Viewpoints: Expressivity, scope
Concerns: **C**omplete understanding of the model and its surrounding elements, and the impact analysis[4] of the model elements.

[2] View is a representation of a whole system from the perspective of a related set of concerns. (IEEE 1471 Std).
3 Viewpoint is a specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis. (IEEE 1471 Std).
4 Impact analysis is defined as the process of identifying the potential consequences of a change, and estimating what needs to be done.

(ii) Inter-Model View

The inter-model relations are visualized in this view. These dependencies provide the overview of the model and make it possible to show the relations between the different models. Further it is possible to have an interaction and integration between models. Therefore, *Inter model View* is proposed.
Viewpoints: Dependency, Integrity.

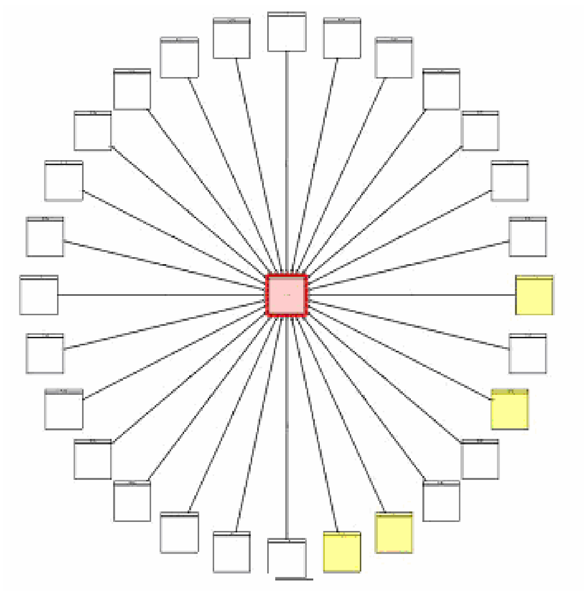

Figure 2. Context view

Concerns: The overview of the system, dependencies between models and elements, knowledge about impact analysis, interaction and integration of the models.

(iii)City View

The Inter-model view is an instance of a City view. This view is similar to a geographical map of a city. Especially in large systems, it can be very difficult to find a specific piece of information and is often spread over multiple models. The entire system is visualized in terms of models. The models are visualized in 3D dimensions in this view, from which traceability and extendibility of a model is visualized easily as shown in Figure 3.
Viewpoints: Extendibility, Traceability.
Concerns: The overview of the models in an entire system, search, trace, and highlight the change, extend the models by having additional elements, and the relationships between the models.

(iv) Metric View

In MoDSE, the models are the origin for evolution. So there is a need to have a set of metrics to have data values, which estimates the major issues in MoDSE such as Quality, complexity, and impact analysis. Metric set should be possible here to measure above mentioned major issues. Therefore, *Metric View* is proposed. The discussion of metric set and data set are out of the scope of this paper.
Viewpoints: Metric set, Data values
Concerns: Metrics for quality, complexity and impact analysis (change prediction and propagation) of the models, data values and data structures for maintaining theses values.

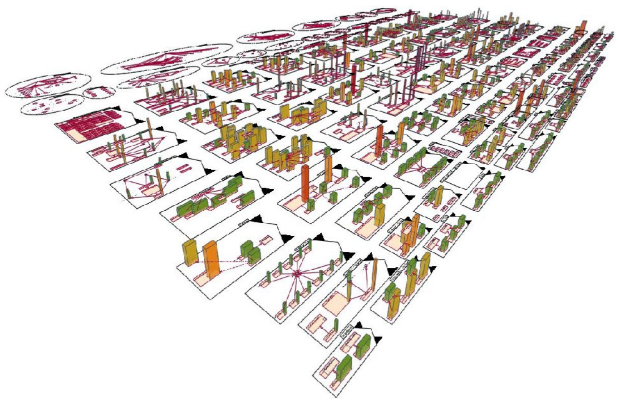

Figure 3. City view

(v) Transformation View

MoDSE requires many types of transformations due to different and interrelated models. Transformations include model to code, model to model (migration of models or mapping from higher level models to lower levels), and code to model. The key challenge is to transform platform independent models to platform specific models. Therefore, *Transformation View* is proposed.
Viewpoints: Consistency
Concerns: Transformation techniques, rules, tools, and languages, the model migration and mapping, characteristics of a model language Transformation techniques and languages are not discussed here.

(vi) Evolution View

MoDSE requires multiple dimensions of evolution. Stakeholder need to know the evolution type for example, platform evolution. To know what kind of evolution, *Evolution View* is proposed. This view is the combination of all the above proposed views because change in a single model or model element may be the cause of the evolution.
Viewpoints: Viewpoints that are identified in the above proposed views**.**
Concerns: Trends, causes, and dimensions for evolution, the side effects such as introduction of a new notation in a modeling language, introduction of a new modeling language, change in development platform, monitoring the quality and complexity of models at multiple dimensions.

(vii) Evaluation View

There is a need to validate and verify the evolution of models. The information captured in all the above proposed views should be verified for its accuracy. This view is also responsible to collect and check the feedback of the participants in the evolution of models in MoDSE. Based on the stakeholders' feedback, evolution process can be verified. Therefore, *Evaluation view* is proposed.
*Viewpoints***:** Validity
Concerns: The evaluation trends and techniques which validates the evolved models, improving the quality, controlling complexity of models, stakeholders' feedback.

The above proposed views are sufficient for the stakeholders to gain enough knowledge and visualization of the models during evolution process. Knowledge such as context, metrics, model interaction, integration, mapping and transformation etc., of models and model elements will be available for the user groups to produce more evolvable model based software systems. And all these views support the entire software development and evolution process. The advantages of proposed views are as follows:

- Provide visualization of evolution process.
- Stakeholders concerns will be addressed by these views and viewpoints.
- Stakeholders gain enough knowledge required for evolution of models in the MoDSE.
- Establish the communication between the stakeholders.

The proposed views also satisfy the Lehman's laws of software evolution and it is discussed in section 4.

IV. LAWS OF SOFTWARE EVOLUTION

The well known Lehman's laws of software evolution [8],[9] are listed as follows:

I. Continuing Change
II. Increasing Complexity
III. Self Regulation
IV. Conservation of organizational stability
V. Conservation of Familiarity
VI. Continuing Growth
VII. Declining Quality
VIII. Feedback System

A. Analytical support

This section analytically describes how the proposed views are closer to the Lehman's Laws of Software Evolution

(i) Continuing Change

The first and fundamental premise of the MoDSE says the evolution should be a continuous process. So, there is a continuous change. Generally the stakeholders initiate the change. The aim of the context view is to provide the different context of the models and model elements in the system and the continuous change can be viewed. Hence, the context view satisfies the first law. The inter model view satisfies the first law by providing the continuing change in terms of dependency between the models and model elements. City view also satisfies the first law which provides the information about change in models such as extendibility and traceability. Metric view provides the numerical values which can be used to manage the continuous change. So, the metric view satisfies the first law. Transformation view satisfies the first law where different types of transformations like model to model or platform independent to platform specific model are required when change occurs. Evolution view provides the causes for change. Hence, this view satisfies the first law. Evaluation view is responsible to check whether the changes have done properly or not. Therefore, this view also supports the first law.

(ii) Increasing Complexity

Complexity is one of the major issues in MoDSE. Any single complex model or model element can be visualized by knowing its context in the context view. So, this view satisfies the second law. A single change in a model or model element may affect the complexity of the other models which are directly related to the changed model. The relation or dependency of the models can be visualized from the inter model view. Hence, inter model view satisfies the second law. Change in models may extend the models exist in the system. Extendibility may affect the complexity and it can be visualized from the city view which satisfies the second law. To control or maintain the complexity, measures and metrics are required. The purpose of the metric view is to provide the metrics set and the data values which are used to estimate the complexity. So, the metric view satisfies the second law. Transformation view satisfies the second law where there will be an increase in complexity after the transformation of models. Evolution view is also has its own role to control and monitor the complexity. As a system evolves its complexity increases unless work is done to maintain or reduce it. So, evolution view satisfies the second law by providing the causes and trends for the evolution. Evaluation view satisfies the second law by verifying the increased complexity from the stakeholders satisfaction and feedback.

(iii) Self Regulation

The evolution of a model-based system involves a team of stakeholders within the organization. The interests, goals, purpose, tasks and objective of the team differs from each other. They together establish systematic parameters for more evolvable software. They also determine the growth and other development characteristics of the evolving product. The context view provides the different context of the evolvable system. The overview and complete understanding of the system can be derived from the context, inter model and city views. The extendibility and traceability can be determined from the City view. The Metric view provides the parameters for systematic evolution, and it requires data values which help to estimate issues like quality and complexity etc. The evolution and evaluation views provide the cause for evolution and user satisfaction in terms of feedback. Therefore, all the proposed views support the third law.

(iv) Conservation of Organizational Stability

The proposed views are aim to provide the knowledge during model driven software evolution which satisfies the concerns of the stakeholder. But these views are not intended to measure the organizational stability or invariant work rate. So, this law is not applicable for the proposed work. This is shown in the Table.1 as 'NA'.

(v) Conservation of Familiarity

Software undergoes continuous change during its life time. Due to this continuous change it is not possible for the stakeholders to retain the familiarity of these changes for a longer period. The proposed views capture the familiarity time to time, but not for a longer time. All these proposed views support fifth law.

(vi) Continuing Growth

This law, Continuing Growth, appears little different to the first law, Continuing Change. For example, accommodating a new requirement often necessitates substantial changes to system architecture and implementation, which leads to system uncertainties. The sixth law addresses change deriving from a different source. According to the second premise of the MoDSE, evolution should be done incrementally. So, evolution is a process of a continuing change and continuing growth also. Continuous growth can be visualized and traceable in the inter-model and City views. So, these two views satisfy the sixth law. The context of the model which is continuously growing can be visualized by using the Context view. So, context view satisfies the sixth law. The metric view provides the numerical values which represents the growth (change) of models, where by, metric view satisfies the sixth law. Transformation view provides the knowledge relevant to transformation of models, which indicates the growth of the models. By this transformation view satisfies the sixth law. Evolution and Evaluation are also possible to provide the sufficient information regarding the continuous growth of the model. Thus, all the proposed views support the sixth law.

(vii) Declining Quality

Quality is another major concern in MoDSE. Ultimately quality must relate to user satisfaction and also with the stakeholders feedback. A system that has performed satisfactorily for some period of time suddenly exhibit unexpected behavior, unexpected results. There are several causes to explain this, but here it is considered as due to change. So there is a chance of declining quality at least for a moment. Evolution view satisfies this law by providing the causes for declining quality. Among the above proposed views, the metric view has much responsibility to achieve the desired quality in the case of declining quality during evolution. Thus metric view satisfies the seventh law. Evaluation view satisfies the seventh law by identifying the declined quality with user satisfaction and feedback. Transformation of models from source to target done in the transformation view and here there is a chance of loosing the quality in models. Hence, this view satisfies the seventh law. Context view, inter model view, and city view are not intended to provide the knowledge about quality. So, these three views do not support the seventh law.

(viii) Feedback system

Systems are evolvable due to the feedback of the stakeholders. Feedback is collected from the different user groups. MoDSE is a feedback evolution process and it consists of multiple users, models, dimensions of evolution. Evolution and evaluation views capture the stakeholders' satisfaction and feedback also. Therefore these two views satisfy the eighth law. The remaining proposed views such as context view, inter model view, city view, metric view, and transformation view are not intended to collect the feedback from the user groups. Hence, these views do not satisfy the eighth law.

Table 1.Analytical Support

Laws / Views	I	II	III	IV	V	VI	VII	VIII
V1	√	√	√	NA	√	√	×	×
V2	√	√	√	NA	√	√	×	×
V3	√	√	√	NA	√	√	×	×
V4	√	√	√	NA	√	√	√	×
V5	√	√	√	NA	√	√	√	√
V6	√	√	√	NA	√	√	√	√
V7	√	√	√	NA	√	√	√	×

Analytical support of the proposed views with the laws of software evolution described above and same is shown in the Table.1. In the table, '√' mark represents the proposed view satisfy the respective law , the '×' mark represents the proposed view does not satisfy the respective law and 'NA' represents the respective law is not applicable for the proposed work. In the table, columns are numbered as I, II so on to represent the eight laws of software evolution and rows are named as V1, V2 so on to represent the seven proposed views. In the table it is observed that majority of the entries are '√' mark, which represents most of the laws are satisfied by most of the proposed views. From this observation it is proved that the proposed views are very closer to the laws of software evolution and they are sufficient to capture the information during MoDSE.

V. Conclusions and Future Work

The seven multiple views for Model Driven Software Evolution are proposed. Viewpoints are identified to construct each view. Different concerns of the stakeholders are satisfied by each view, providing sufficient knowledge regarding model driven software evolution. The analytical support of all these proposed views for the eight laws of software evolution was discussed. From the table it is observed that all the proposed views support majority of laws. So, the proposed views are sufficient for the stakeholder to capture the information and the laws are also important for Model Driven Software Evolution. Development of framework consisting of the proposed views for visualizing the MoDSE process and the evaluation of the framework with case tools and the stakeholders concerns is the subject of future research.

References

[1] Arie van Deursen, Eelco Visser, and Jos Warmer. "Model-Driven Software Evolution: A Research Agenda", In Dalila Tamzalit (Eds.). *Proceedings 1st International Workshop on Model-Driven Software Evolution*, University of Nantes, 2007. pp. 41-49.

[2] Briand L.C, Y.Labiche, O' Sulliavan. "Impact Analysis and Change Management of UML Models", *Proceedings of the International Conference on Software maintenance (ICSM'03),* IEEE, 2003.

[3] Christian F.J. Lange, Martijin A.M. Wijins, Michel R.V. Chaudron. "A Visualization Framework for Task-Oriented Modeling using UML", *Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07),* IEEE, 2007.

[4] Christian F.J. Lange, Martijin A.M. Wijins, Michel R.V. Chaudron. "Metric View Evolution: UML-based Views for Monitoring Model Evolution and Quality", *Proceedings of the 11th European Conference on Software Maintenance and reengineering (CSMR'07),* IEEE, 2007.

[5] Jaun F Ramil. "Laws of Software Evolution and their Empirical Support", *Proceedings of the International Conference on Software Maintenance (ICSM'02),* IEEE, 2002.

[6] Kazi Farooqui, Luigi Logrippo, Jan de Meer. "The ISO reference Model for Open Distributed Processing – An

Introduction", *Special issue on ISO reference model for open distributed processing,* 1985, ACM Portal, pp.1215-1229.

[7] M M Lehman, J F Ramil, D E Perry et.al. "Metrics and Laws of Software Evolution – The Nineties View", *4th International Software Metrics Symposium (METRICS'97),* 1997, IEEE Computer Society.org.

[8] M M Lehman. "Rules and Tools for Software Evolution Planning and Management", *Annals of Software Engineering,* Volume 11, Number 1/November, 2001, pp.15-44.

[9] M M Lehman. " Laws of Software Evolution Revisted ", *Springer Verlag,*Position Paper, EWSPT96, LNCS 1149, October, 1996, pp. 108-124.

[10] Rene Keller, Claudia M.Eckert, P.Jhon Clarkson. "Viewpoints and Views in Engineering Change Management", *Workshop on Complexity in Design and Engineering,* Glasgow, 2005.

[11] Robert France and James M. Bieman. "Multi-View Software Evolution: A UML-based Framework for Evolving Object-Oriented Software",*17th IEEE International Conference on Software Maintenance (ICSM'01)*,2001, pp..386.

[12] Stephen Cook, He Ji and Rachel Harison. "Dynamic and Static Views of Software Evolution", 17th IEEE International conference on Software Maintenece (ICSM'01), 2001, pp. 592.

[13] Tom Mens, Michel Wermelinger, Stephen Duacasse et.al. "Challenges in Software Evolution", *Proceedings of the 8th international Workshop on Principles of Software Evolution (IWPSE'05)*, IEEE, 2005.

[14] IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, IEEE Std 1471, 2000.