Cloud Migration Metamodel

A framework for legacy to cloud migration

Pooja Parnami
Comp. Sci. Dept.
IIS University
Jaipur, India
pooja.parnami@yahoo.com

Aman Jain
Comp. Sci. Dept.
MAISM
Jaipur, India
Amanjain.jpr@gmail.com

Navneet Sharma
Comp. Sci. Dept.
IIS University
Jaipur, India
Navneet.sharma@iisuniv.ac.in

***Abstract*—In this paper a legacy to cloud migration metamodel has been developed and presented. The purpose of the paper is to develop a consolidated framework for cloud migration support. The research adopts a Model Driven Software Engineering (MDSE) approach, which is used in Software Engineering and Information System field for developing domain-specific languages and knowledge reuse. The framework is developed with the help of review of existing cloud migration literature. The method was to find, harmonizing and iterative refinement of existing key concerns in cloud migration literature. The objective is to create a generic reference process for cloud migration.**

Keywords—legacy system; cloud; cloud migration; metamodel;

I. INTRODUCTION

In last ten years the information technology is developing rapidly in terms of agility, robustness and flexibility. But a great deal of changes also occurred in business environment, in terms of business model, merger and acquisition, laws and regulation. To cope up with these changes the organizations are changing their systems. But the modification, maintenance and evolution become difficult after a certain point of time. These applications could be terms as legacy application/ system. Migration of a large-scale legacy application to cloud is critical and challenging job. Legacy applications have been developed without considering the future need of cloud computing. The unique characteristics of cloud like multi-tenancy, interoperability, elasticity etc. are the major challenges to be addressed while migrating legacy applications to cloud.

We can find plenty of techniques developed for cloud migration but there a big gap could be visualized.

a) The existing research provides specialized migration approach. There is a need of holistic methodology for legacy to cloud migration [1].

b) The existing literature focuses on one or few aspects of cloud migration like cost estimation, decision making, migration tools etc. [2].

c) There are many approaches exist a need for guided methodology for design, develop and deploy solutions to take leverage of cloud computing. [3].

d) There is a need for a well-defined process for cloud migration [4].

e) Needs for a comprehensive migration framework [5].

f) Needs for a cloud enabled system with tailorable and self adaptive [2].

g) Automated support in migration process [2].

A migration project developed in trivial way could be designed and managed in ad-hoc manner but large scale, enterprise-based applications needs methodological approach. A step-by step, structured process of cloud migration could smoothly handle cloud migration complexities. Existing research recommends a need of methodology, technologies and framework for cloud computing framework.

II. METAMODEL

A metamodel is collection of building blocks and rules which can be used to create new models. Metamodel is a language that defines how to create models (OMG 2004). Metamodel helps developers to create domain specific models to conceptualize the generated results and the resultant domain specific model can be used for domain specific tasks and processes [6]. It aims to create reusable, interoperable, portable software activities and components. In this context, a metamodel is a fundamental building block to design the structure of models [7]. Metamodel could be defined as a modelling language with components like constructs, construct relationships, constraints, modelling rules and it helps to design domain specific model without essential knowledge of concrete syntax of the language [8], [9]. A model created using a metamodeling environment, which describes other modelling systems is called a metamodel. Large computer-based systems are integration of physical processes and information processing. These systems work under rapidly changing environment like hardware, personnel, support system etc. Due to complex nature of large systems, software modification involves a large amount of risk and to overcome this limit the system must be designed to evolve. The best way to design domain specific system is to create a model, and use interpreters to translate this model into real world execution environment. When changes in the overall system required, the model is updated to reflect the changes and the applications are regenerated automatically from the model. But when a model is used in a particular domain the changes done based on the particular environment where the model is implemented, should be reflected to the main model requires lots of effort and cost. One solution to this problem is to model the modelling

978-1-5386-5933-5/19/$31.00 ©2019 IEEE

environment. It is a way to model a particular domain-specific application, and called metamodel. Metamodel is a formalized Maintaining the Integrity of the Specifications description of objects, relationships and behaviour required in a particular domain specific modelling environment. [10]

III. CLOUD MIGRATION METAMODEL CREATION PROCESS

The creation of metamodel, through capturing common concepts of domain, without going into the specific details. This approach allows to get a generic and theoretical view of domain, but also provides the freedom of extending for a specific domain context. There are different models available in cloud migration field and each one focus on certain activities of a particular migration type. The suggested model in this research provides an abstract view of generic migration process, which hides the specific operational details of a migration scenario. With the help of vast body of cloud migration literature, this research proposed a cloud migration framework. This framework provides a step by step process of legacy to cloud migration with a clear division of core phases, activities, tasks and work-products.

Step 1 : Preparing Knowledge sources

This is the initial step for any process and used to identify the key constructs [11]. The main aim of this research is to create a metamodel and main knowledge source is literature available in cloud migration research. Essential feature of any research attempt is rigorous review of relevant literature. The papers published in the field of cloud migration provide best practices, solutions, methodologies etc. to migrate a legacy application to cloud. As recommended by [12], to identify key constructs for cloud migration a systematic literature review was performed.

Defining the Search String: The primary objective of the search was to find existing approaches, methodologies and techniques to migrate legacy on-premise software to the cloud. The search was designed using the keywords "Legacy", "Cloud", "Cloud Computing", "Service Computing", "Monolith application" , "On-premise application", "Legacy application" ,"Legacy program" , , "Legacy code" , "Legacy system" , "Legacy component" , "Legacy software", "Legacy information system", "Pre-existing assets" , "Legacy architecture" , "Legacy asset", "Methodology", "Reference Model", "Process Model", "Migration", "Cloud migration", "Cloud adoption", "Migration to Cloud", "Legacy to Cloud migration" , "Legacy migration to Cloud". Search string were defined with the help of these keyword and using OR /AND logical operator.

Selecting the Study Sources: The search string defined above was used to search in the following databases:

a) ACM Digital Library
b) Springer Link
c) Science Direct
d) IEEE Xplore
e) Google Scholar
f) Wiley InterScience
g) ISI Web of Knowledge

Reports and white papers published by national and international organizations and groups working in the field of *cloud migration(e.g. CSA, NIST, ENISA, IBM, Infosys, Cisco etc.)*

Above mentioned database covered all the major research work in software engineering field. Additionally a manual search was conducted into topic specific conference, journals, workshop proceeding and technical reports. Inclusion and exclusion criteria: There is a fair amount of studies available in the literature, therefore a good validation is required for selection of papers. The inclusion criteria were a set of studies:

a) Studies published between 2007 and 2017
b) Studies were was selected on the basis of well documentation, clarity and higher citation
c) With properly described research context and focused on the legacy application migration to the cloud environment
d) Published in software engineering journals/ conferences and technical/white papers published by leading companies working in the field of cloud e.g. Oracle, IBM, Amazon, Infosys etc.

The exclusion criteria were set of:

a) Opinion Papers
b) Short papers (less than 6 pages)
c) Prefaces
d) Abstracts
e) General discussions
f) Studies other than English

The result of first iteration was 201 primary papers. In second iteration after detail study of each paper's content, the papers were scrutinized on the basis of inclusion and exclusion criteria defined before and 85 papers were selected to proceed for next step.

Step 2: Identification of development and validation sets

The models collected in step 1 are divided into three sets. SET- 1 is used to initiate the metamodeling process, this will produce the initial metamodel. Remaining two sets were used for validation of proposed metamodel.

The different phases of cloud migration life cycle are generated with the help of general software development lifecycle, reengineering lifecycles and existing cloud migration literature. Generic software development life cycle suggested by Pressman[13], Sommerville[14] and existing reengineering lifecycles like Butterfly[15], Renaissance[16], Sneed's approach[17] and Architectural-Driven Modernization horseshoe[18] were also analyzed. Extensive study of existing cloud migration literature also helped in identifying different phases of cloud migration. Combining the suggested phases in these models, we have chosen three phases Pre-Migration, Design and Migration. In the Pre-migration phase tasks like understanding legacy application, technical and non-technical feasibility study are performed. The main concern of Design phase is to plan a high-level architecture of target cloud-based application and finally the actual migration tasks are carried out in the Migration phase, which includes making legacy applications cloud-enabled, deployment of application parts and enhancing performance.

For the creation of initial metamodel with the help of SET1 models, it requires full coverage across the concepts. The model which covers whole phases of cloud migration is considered as General Model. Whereas if a model describes specific cloud migration phase such as pre-migration phase is considered as Specific model with low coverage across the concepts. In selection of models for Set1, general models which cover all the phases in cloud migration (pre-migration, Design & Migration) have been selected. For designing initial metamodel we require all domain specific generic concepts.

Step 3: Extracting Constructs

The purpose of this step is to identify all constructs and their definitions for metamodel creation. It was performed by reading each paper, extracting the constructs and their definitions manually. While extracting the construct the main criteria is that a construct must be sufficiently generic i.e. a construct should not belong to a specific cloud migration scenario, technology or tool.

Step 4: Harmonizing Constructs' and Reconciliation of Definitions

Furthermore, definitions of constructs were also extracted from the source papers. Since the literature selected for the study come from people with different background, the terminologies, phrases and definitions of a given construct may differ. Therefore, many constructs have similar meaning but different definition. Resultantly it was essential to reconcile various definitions of assembled constructs. One alternate provided by[19],[20]has been followed and we try to provide a hybrid or harmonized definition for these type of constructs. If two or more constructs had same name or two constructs with different names referring the same thing, then a harmonization process has been undertaken. Under which a comprehensive definition has been designed on the basis of collective definitions. Harmonization does not apply on same name but different definitions. Further, if there is inconsistency between two or more definitions occur, the concept which has more rational usage will be chosen.

Step 5: Designation of Concepts into Cloud Migration

The reconciled and harmonized constructs designated into one of the cloud migration phases, identified in step 2 : Pre-Migration, Design and Migration.

a) Pre-Migration : The initial migration tasks like understanding organization's policies, feasibility study, analysing technical & non-technical requirements are performed in this phase.

b) Design : The tasks like define migration plan, choosing cloud provider, identify the changes and designing validation criteria are grouped under design phase.

c) Migration: The main tasks like designing and deploying cloud services, validation, monitoring and optimization are part of the migration phase.

Organizing the Constructs into Activities and Phases:

For the purpose of creating a coherent metamodel, 110 Identified constructs produced from Set 1, were organized into three hierarchical levels. With reference to SPEM (Software Process Engineering Model) formalism[10], common building blocks of software development processes are Phase, Activity, Task and Work-Product.

a) Task: A small unit of work to achieve one or more goal(s) could be define as a task

b) Work-Product: A Work-Product could defined as a concrete object required or utilized by a task. Constructs identified in previous section could act as a task or work-product.. It can be consumed or modified by one or more further constructs.

c) Activities: It could be defined as group of interrelated tasks.

d) Phase: Group of related activities forms a phase.

On the basis of descending order of granularity these three levels could be defined.

a) First level: Task and Work Products

b) Second level: Activities

c) Third level: Phases

Step 6: Designing Initial Metamodel:

In this step, the major task was identification of relationship among the constructs. It would help to understand how the constructs relate to each other. The UML notations Association (—), Specialisation(——▷) and Aggregation(——◇) were used to denote these relationships.

The relationships between the constructs were identified on the basis of studies and output of step 5. The established linkage between the constructs are as follows:

a) The first type of relation between the constructs is 'Uses Association', symbolised as (—). It shows that a construct uses data or artefacts provided by another construct. As an example, using 'Uses Association' between constructs "Legacy Architecture" and "Plan Migration" indicates that the architecture identification of legacy application is used for planning the migration.

b) The second type relation between the constructs is 'Follows Association' which defines the sequence of the migraiton process execution. For example, the relation between the constructs "Validate Target Cloud Architecture" and "Optimize" signifies that the optimization task could be conducted only after validation of target cloud architecture.

c) The third type of relation is used to define the relation 'Produces Association'. It means that a construct is outcome of the execution of another construct. For instance, the work-product “Requirement Specification” is the output of the task “Requirement Analysis”.

d) The relation fourth kind of relation is ‘isAGroupOf Association’, denoted by (——◇). It used to show an aggregation relationship among consturcts. i.e. constructs with a common theme could be grouped in a coarse grain activity. For instance, Cost, Size and Effort estimation belongs to the activity Feasibility Analysis with an aggregation relationship.

e) Finally, the last type of relation is Specialisation, denoted by (——▷), indicating a construct is a special type of another construct or a sub-construct (child) is a special kind of super construct (Parent). A good example of such relation is the various adaptation efforts which are required to be applied on legacy application parts to make them cloud-enabled. For example, the constructs “Technical Requirement” and “Business Requirement Analysis” are a kind of “Requirement Analysis”.

The Resultant Metamodel: The result of Step 1 to 6 is a generic process model for migration of a legacy application to cloud. It does not provide technical and platform specific details required for a specific cloud migration scenario, but it presents an abstract view of different transition phases during migration. For each individual instantiation of metamodel, such details could differ, and user can apply various techniques and tools in order to perform migration tasks.

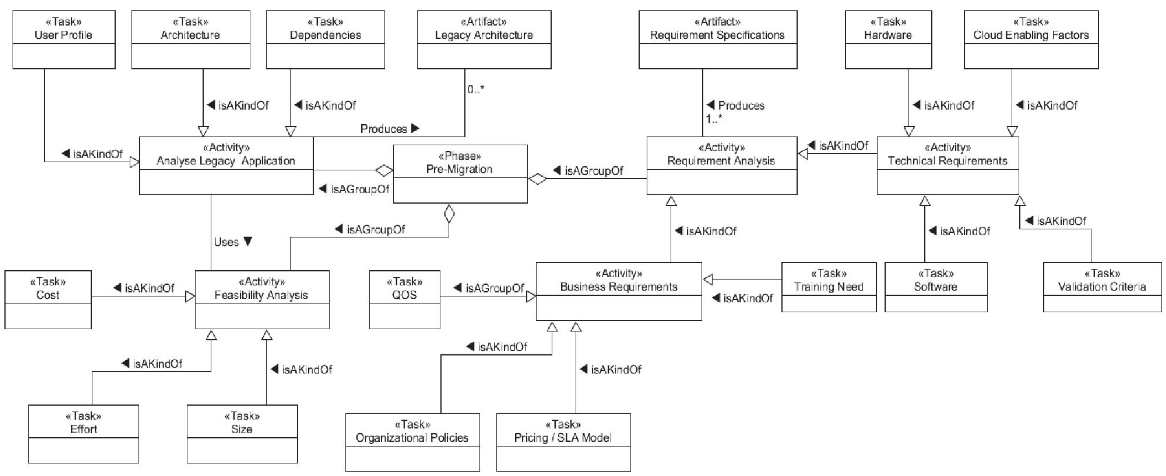

Fig. 1: Pre-Migration Phase

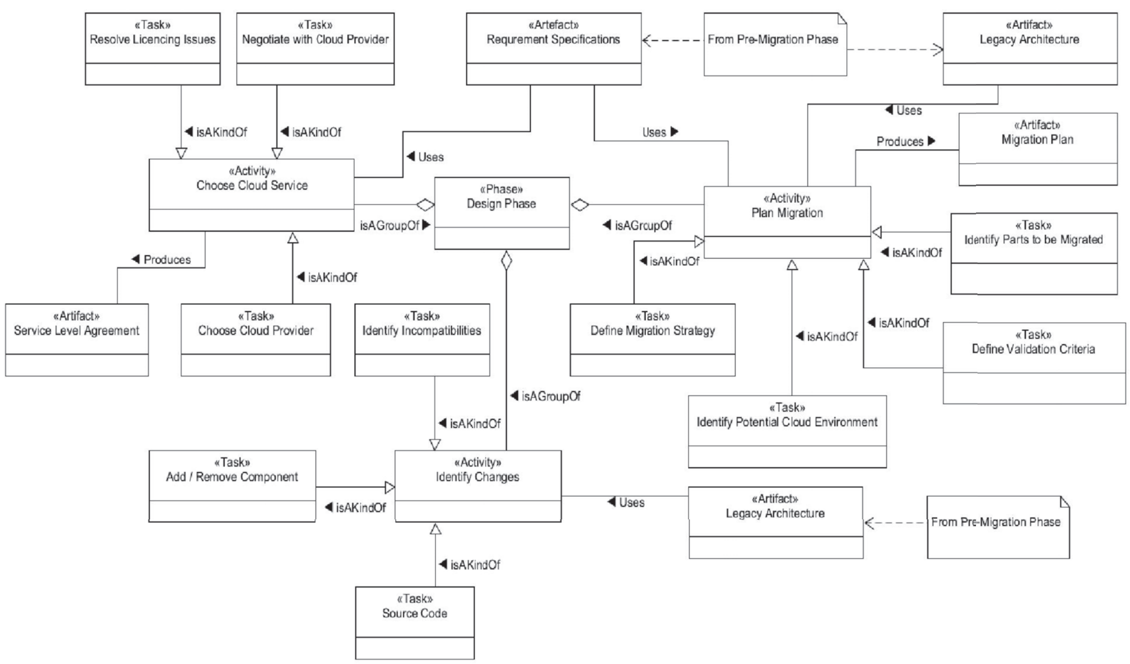

Fig. 2: Design Phase

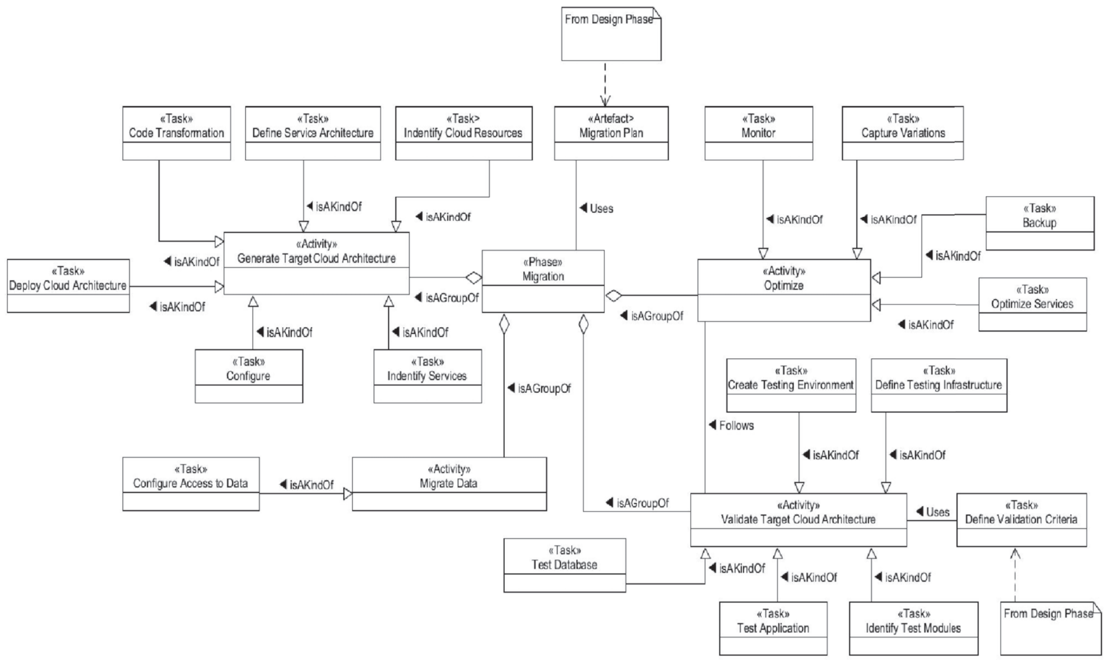

Fig. 3: Enable Phase

We have used simple UML notations[21] to represent metamodel in a simple and well structured manner. The UML notation class is used to represent metamodel constructs. Stereotype is a mechanism for extending UML notations, therefore we have stereotypes to extend the meaning of UML class. On the basis of stereotype <<Activity>>, <<Phase>>, <<Task>> and <<Artefact>> represents Activity, Phase, Task and Artefact respectively. Fig 1 to 3 shows the initial metamodel in the form of phases, activities, tasks and artefacts, along with their definitions and goals.

Step 7 Validation of Metamodel:

The validation is done to determine the completeness of generalization of proposed metamodel. It will help in determining the theories, assumptions and concepts used in the metamodel are correct [22]. In our research work we have applied the following two frequently used validation techniques.

a) Comparison against other models

b) Frequency-based selection

Validation 1 : Comparison against other models

After creation of first version of metamodel, the derived constructs of the metamodel need to be validated and compared from other exiting models of similar domain[19]. This process starts with selection of 8 models. These 8 models forms the first validation set called Set V1. Following process is applied on models of set V1.

i) *Extracting Constructs :* After detail study of each paper from set V1, constructs and their definitions were extracted.

a. Mapping the extracted constructs with the proposed model's constructs: The extracted constructs of set V1 are mapped to the proposed model. With the help of definition and meaning, each derived construct has been compared to existing metamodel designed construct. The result of this step is a list of constructs not addressed by existing models.

b. Two constructs "Stakeholder Analysis" and

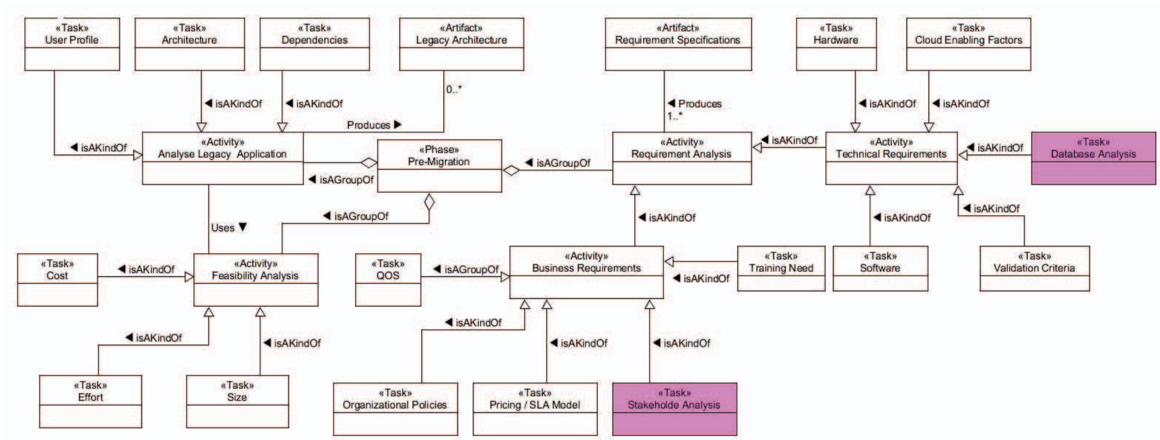

Fig. 4: Validated Version (1.1) Pre-Migration Phase

“DataBase Analysis” has been added to Pre-Migration Phase.

c. One construct “Identify Data Incompatibilities” has been added to Design Phase.

d. Two constructs “Extract Data” and “Adapt Data” has been added to Migration Phase.

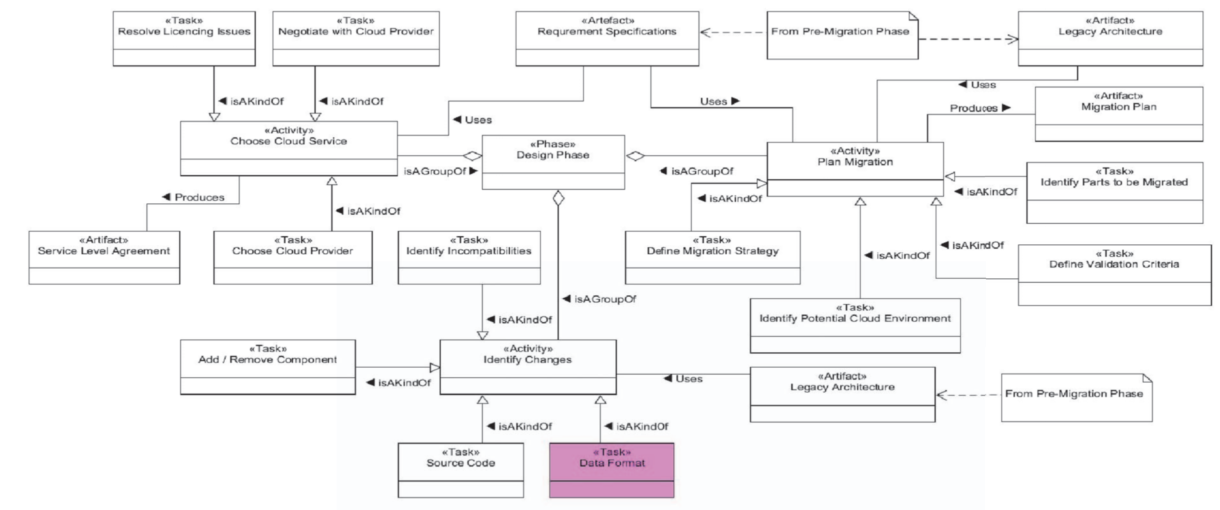

Fig. 5: Validated Version (1.1) Design Phase

Fig. 4 To 6 Shows the modified version of cloud migration metamodel. The added constructs are shown in a filled box.

Validation 2 -Frequency-based selection

The second type of validation is performed using frequency-based selection technique. For this validation 8 models have been selected from primary set of 85 cloud migration models. These selected models have wider cloud migration coverage. Specialized cloud migration models focus on specific concepts and will omit generic concepts. Therefore, wider coverage of individual cloud migration concept results in a better frequency count. This technique is frequently used in data mining[23] software analysis[22], and medical retrieval systems[24]. The purpose of frequency-based selection technique is to evaluates the importance of each generated concept of developed metamodel. It helps in removing the features that have no co-relation to the cloud migration metamodel.

First step is to gather all the concepts from Validation Set 2 and match it with derived concepts. The second step is to find the frequency of each concept. The concepts with low frequency are revisited and could be liable for deletion. In this technique frequency means how many times a concept is used in all investigating models of set V2. “Degree of Confidence DoC” [19] is to find the importance value of each concept in cloud migration metamodel. The DoC is derived by dividing each concepts frequency with total number of models in set V2. The formula for defining DoC is as follows:

$$Degree\,of\;Confidence = \frac{Frequency\,of\;Concept}{Total\,of\;\;Set\;V\,2\;Models} \times 100\%$$

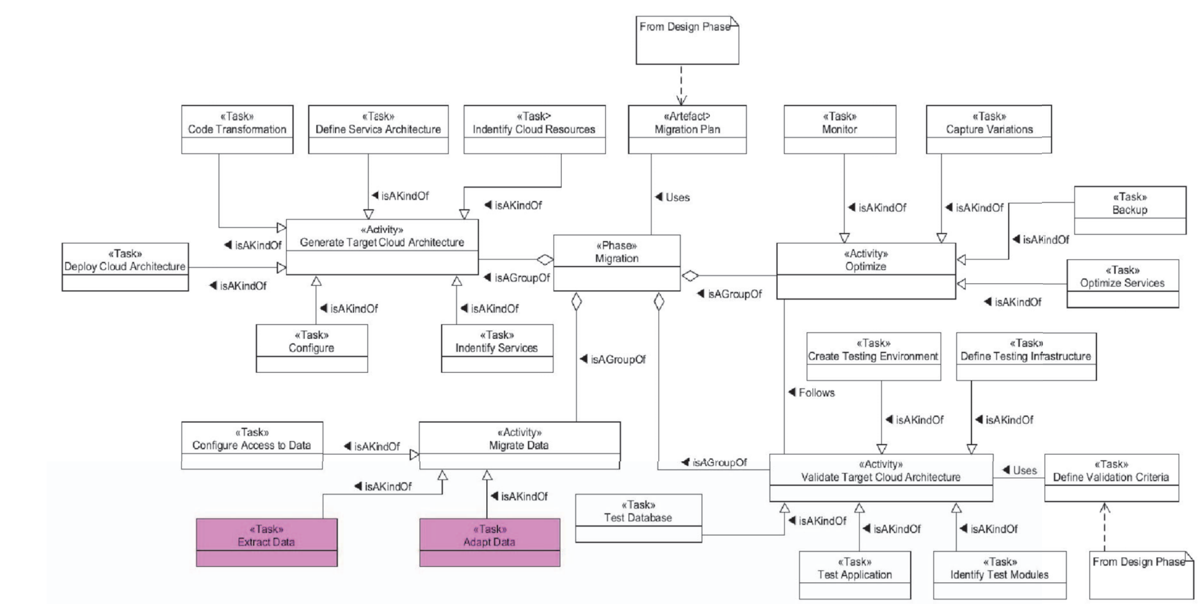

Fig. 6: Validated Version (1.1) Migration Phase

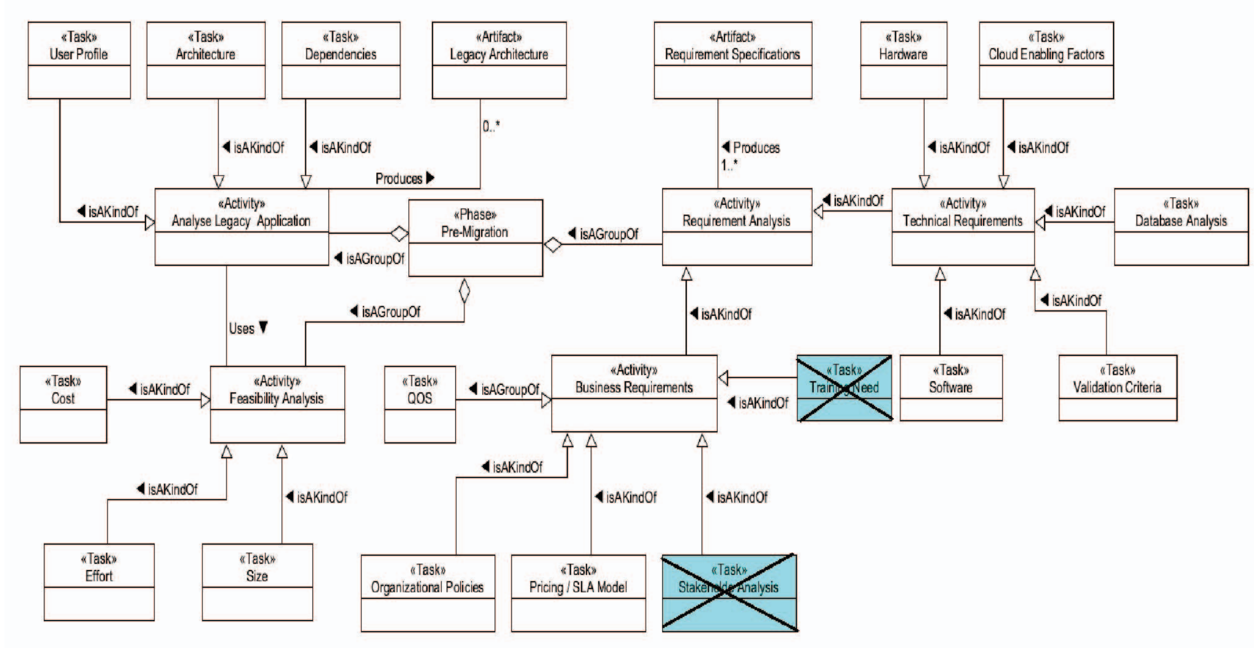

Fig. 7: Validated Version (1.2) Pre-Migration Phase

After finding the frequency of each concept, to find the importance of each concept, the Degree of Confidence (DoC) is estimated. DoC finds outs the probability of a cloud migration concept used in a randomly chosen model. Five categories of concepts could be defined on the basis of DoC values[25]:

a. Very Strong (100-75%)
b. Strong (74-55%)
c. Moderate (54-35%)
d. Mild (34-15%)
e. Very Mild (14-0%)

The categorization 46 concepts of cloud migration are as follows:

a. Very Strong -15
b. Strong -3
c. Moderate - 15
d. Mild - 7
e. Very Mild -6

Like domain specific modelling, we cannot achieve perfection in metamodel development[11]. So we agree on this view and if a concept has "Very Mild" DoC value, it could be considered again before tend to delete it. The following 6 cloud migration concepts are found with mild degree of confidence category:

1. *Size Estimation*
2. *Stake holder analysis*
3. *Defining training need*
4. *Defining validation criteria*
5. *Resolving licencing issues*
6. *Negotiate with cloud provider*

As a result of FBS, Pre-Migration has been changed, but Design phases and Migration phase remained un-change. Fig. 7 shows the new validated version of Pre-Migration phase. Design and Migration phases will remain unchanged.

CONCLUSION

There are different models available in cloud migration field and each one focus on certain activities of a particular migration type. The suggested model in this paper provides an abstract view of generic migration process, which hides the specific operational details of a migration scenario. With the help of vast body of cloud migration literature, this research proposed a cloud migration framework. This framework provides a step by step process of legacy to cloud migration with a clear division of core phases, activities, tasks and work-products. The proposed method is a unified representation of accumulated knowledge and will help the new comers, scholars and IS practitioners to get a consolidated view of transition process of legacy to cloud migration.

The proposed framework provides a platform to design and maintain situation-specific, systematic migration of legacy applications to cloud. The framework has extracted constructs from the literature and evaluated, designed and refined to design it. During cloud migration, besides other software development activities, a practitioner can select constructs from the framework and customize them to fit in a particular endeavor.

REFERENCES

[1] J. F. Zhao and J. T. Zhou, "Strategies and methods for cloud

migration," Int. J. Autom. Comput., vol. 11, no. 2, pp. 143–152, 2014.

[2] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud Migration Research : A Systematic Review," vol. 1, no. 2, pp. 142–157, 2013.

[3] M. A. Babar and M. A. Chauhan, "A tale of migration to cloud computing for sharing experiences and observations," Proceeding 2nd Int. Work. Softw. Eng. cloud Comput. - SECLOUD '11, no. January 2016, p. 50, 2011.

[4] S. Mehfuz and G. Sahoo, "A five-phased approach for the cloud migration," Int. J. Emerg. Technol. Adv. Eng., vol. 2, no. 4, pp. 286–291, 2012.

[5] R. Rai, G. Sahoo, and S. Mehfuz, "Exploring the factors influencing the cloud computing adoption: a systematic study on cloud migration," Springerplus, vol. 4, no. 1, pp. 1–12, 2015.

[6] N. Iscoe, G. B. Williams, and G. Arango, "Domain Modeling for Software Engineering," in Proceedings of the International Conference on Software Maintenance~1991, 1991.

[7] T. Stahl, M. Völter, J. Bettin, A. Haase, and S. Helsen, Model-Driven Software Development: Technology, Engineering, Management. 2006.

[8] G. Beydoun, G. Low, B. Henderson-Sellers, H. Mouratidis, J. J. Gomez-Sanz, J. Pavon, and C. Gonzalez-Perez, "FAML: A generic metamodel for MAS development," IEEE Trans. Softw. Eng., vol. 35, no. 6, pp. 841–863, 2009.

[9] G. Beydoun, C. Gonzalez-Perez, B. Henderson-Sellers, and G. Low, "Developing and Evaluating a Generic Metamodel for MAS Work Products," Softw. Eng. Multi-Agent Syst. IV, vol. 3914, pp. 126–142, 2006.

[10] D. M. Environments and G. G. Nordstrom, "No Title," 1999.

[11] S. Kelly and R. Pohjonen, "Worst practices for domain-specific modeling," IEEE Softw., vol. 26, no. 4, pp. 22–29, 2009.

[12] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," Inf. Softw. Technol., vol. 51, no. 1, pp. 7–15, 2009.

[13] R. S. Pressman, Software engineering : a practitioner's approach. McGraw-Hill, 2005.

[14] I. Sommerville, Software Engineering. 2004.

[15] Wu, D. Lawless, J. Bisbal, R. Richardson, J. Grimson, V. Wade, and D. O'Sullivan, "The Butterfly Methodology: a gateway-free approach for migrating legacy information systems," Proceedings. Third IEEE Int. Conf. Eng. Complex Comput. Syst., no. October 1997, pp. 200–205, 1997.

[16] I. Warren, The renaissance of legacy systems : method support for software-system evolution. Springer, 1999.

[17] H. M. Sneed, "Planning the Reengineering of Legacy Systems," IEEE Softw., vol. 12, no. 1, pp. 24–34, 1995.

[18] V. Khusidman and W. Ulrich, "Architecture-driven modernization: Transforming the enterprise," Semin. Softw. Anal. Trasformation, pp. 1–7, 2007.

[19] S. H. Othman, G. Beydoun, and V. Sugumaran, "Development and validation of a disaster management metamodel," Inf. Process. Manag., vol. 50, pp. 235–271, 2014.

[20] M. Fahmideh Gholami, P. Jamshidi, and F. Shams, "A procedure for extracting software development process patterns," Proc. - UKSim 4th Eur. Model. Symp. Comput. Model. Simulation, EMS2010, pp. 75–83, 2010.

[21] O. M. G. A. Specification and C. Bars, "OMG Unified Modeling Language (OMG UML)," Language (Baltim)., no. November, pp. 1–212, 2007.

[22] H. and S. Manning, C.D., P. Raghavan, Introduction to Information Retrieval. Cambridge University Press (Vol. 9)., 2008.

[23] B. Saleh and F. Masseglia, "Discovering frequent behaviors: Time is an essential element of the context," Knowl. Inf. Syst., vol. 28, no. 2, pp. 311–331, 2011.

[24] V. Jalali and M. R. Matash Borujerdi, "Information retrieval with concept-based pseudo-relevance feedback in MEDLINE," Knowl.Inf. Syst., vol. 29, no. 1, pp. 237–248, Oct. 2011. [25] "A test-driven approach for metamodel development," igi-global.com.