IPASJ International Journal of Information Technology (IIJIT)
Web Site: http://www.ipasj.org/IIJIT/IIJIT.htm
A Publisher for Research Motivation ..... Email:editoriijit@ipasj.org
Volume 5, Issue 2, February 2017 ISSN 2321-5976

SAREM: A SPEM ADDITION FOR SOFTWARE PLANNING EXTRACTION PROCESS

Mr. Vaidhav Shrikant

A1 Global Institute of Engineering and Technology, Prakasam

ABSTRACT

In order to keep up a system, it's crucial to know its design. but even if each system has associate design, not each system encompasses a reliable illustration of its design. To take care of this drawback several researchers have engaged in package design extraction wherever the system's design is recovered from its ASCII text file. whereas there's a overplus of approaches aiming at extracting package architectures, there's no mean or tool measure for these approaches; that makes the comparison between the various approaches a tough task. To tackle this lack, we have a tendency to developed a meta-model, supported SPEM meta-model, that specifies the package design extraction method. Such meta-model is a tool to match, analyze and valuate analysis field approaches. during this paper we have a tendency to detail our meta-model known as SArEM (Software design Extraction Meta-model) and clarify its ideas.

1. INTRODUCTION

A package design may be a bridge between the matter (i.e. the requirements) and also the resolution (i.e. the implementation). It covers all the many selections that result in the system construction. Formally, a package design is outlined because the system structure(s), that includes package components, the outwardly visible properties of these components, and also the relationships among them [1]. As each system may be shown to be composed of components and relations among them, each software package has associate design. In [2], Garlan reveals the importance of package architectures by sticking out its role on six totally different aspects: understanding, reuse, construction, evolution, analysis and management. However, for several systems, the illustration of their architectures isn't reliable, because of 2 reasons: the primary is that the lack of documentations. Indeed, this can be part because of the actual fact that current-day development environments still focus an excessive amount of on writing code and deficient on planning and documenting tasks. The second reason is that the subject erosion. In fact, throughout the various evolution phases the system deviates from its original illustration, inflicting a spot between the present system design and also the documented design. thus no reliable package design illustration is out there within the each cases. so as to avoid the increasing of non-synchronization between current system design and its illustration, many techniques and processes aiming at extracting package architectures are planned. every approach has its own needs, method, automation level, solution, limits and alternative properties. However, whereas there's associate more than approaches and technologies that support package design extraction, there's no reference or commonplace that enables their analysis. This lack makes the choice of associate extraction method a troublesome activity for the designer.

To solve this drawback, a comprehensive and effective comparison ought to be planned. Such comparison should be supported many criteria that mirror the necessary components of associate extraction method. Thus, we have a tendency to propose a metamodel, known as SArEM (Software design Extraction Meta-model), that represents the fundamental ideas of package design extraction method in terms of entities and relationships. the remainder of the paper is organized as follows: section II presents some connected works on package design extraction. Section III provides the ideas on that SArEM relies, a lot of specifically it describes the SPEM meta-model and its adaptation to the extraction processes. In section IV, our meta-model SArEM is careful. and at last section V concludes with an summary of the most important contributions of this paper.

2. CONNECTED WORKS

Several approaches are planned to extract system's design. Some propose a tool that supports the extraction et al propose an entire method that's human derived and tools supported. A. package design extraction tools the foremost

noted tools that support package design extraction ar the ASCII text file analyzers that dissect the ASCII text file to represent the package structure at totally different levels of abstraction. we have a tendency to mention: CPP2XMI [3] that extracts the knowledge from the C++ ASCII text file in UML diagrams like category, sequence and activity diagrams. Same for the reverse engineering framework Columbus [4] that analyzes massive C++ comes and generates a schema that prescribes the shape of the extracted information. Ptidej [5] builds category diagrams from Java ASCII text file. Rational Rose [6] supports reverse engineering of C++ and Java package systems. The output is that the category diagram with the attributes, functions and variables. augment this, there ar several tools that extract ASCII text file info as graphs just like the MDG graph. MDG may be a model dependency graph like the graph nodes represent the modules and also the edges represent the dependencies between them. we have a tendency to might mention Chava [7] for the analysis of Java programs and ACASIA [8] for C++ code analysis. there's several tools that support the package design extraction by providing the way to cluster supply entities, for example: The Bunch tool planned by Mancoridis et al. [9]–[14]. In fact, Mancoridis et al. introduce the extraction drawback as a module cluster drawback wherever search-based optimisation algorithms may be applied. this idea semiconductor diode to the event of a tool known as Bunch that supports a spread of search algorithms. The tool uses associate MDG graph as associate input, applies associate algorithmic program and finds the most effective partition of the MDG in step with a fitness perform. Maqbool et al. propose a replacement algorithmic program for cluster finding inter-cluster instance throughout the cluster method [15]. Among the varied means of package design extraction, matching tools ar one amongst the ways: Sartipi [16], [17] considers {the drawback|the matter} of extraction associate best matching problem between 2 graphs: the pattern graph that represents the abstract design planned by the designer and also the supply graph that represents the system ASCII text file. The reflexion model tool [18], [19], that evokes several alternative approaches, relies on the elaboration of a model that computes the variations between the developers high-level model and also the recovered model. Since style|the planning|the look} patterns mirror a design call that has been created in order that they ar some extent of interest for package design extraction. several tools support style pattern detection, one amongst them is finished by Kraemer and Prechelt [20]. The tool works on style pattern detection from C++ code files. The approach uses Prolog rules to observe the look pattern instance (the structural pattern) from a repository that contains the knowledge. Also, MARPLE [21] supports style pattern detection and package design reconstruction.

A. package design extraction processes

After finding out an oversized range of analysis works, we have a tendency to might classify the extraction processes beneath 2 classes in step with the techniques used. the primary class covers the processes that ar supported the cluster technique. In these processes the design is extracted in terms of parts like every part may be a cluster of connected system entities. Claudio Riva has contributed during this field by proposing associate approach [22] that extracts the package design of Nokia merchandise, it's supported the principle of abstraction so as to pass from the ASCII text file level to a better level. the method consists of six phases: (1) the definition of subject concepts; (2) the extraction of the ASCII text file model that contains entities that characterize the implementation; (3) the abstraction which's the grouping of the supply model entities by mistreatment cluster techniques; and also the remaining phases improve the design documents and establish subject flaws. several methodes ar supported Bunch tool which's a cluster tool: The Ig approach [23] integrates human interpretations within the extraction process. The genetic algorithmic program is dead iteratively and at every iteration new constraints given by the developers ar additional. Mahdavi et al. approach [24], which's conjointly supported by Bunch tool, shows that combining the results from multiple hill climbs will improve the results for straightforward hill rising and genetic algorithmic program.

The second class relies on the matching techniques. In such processes, the design is that the results of a mapping between a high-level design and also the ASCII text file. Among the processes that belong to the current class, we have a tendency to mention the reflexion model method [18], [19], [25] which's supported the subsequent steps: the definition of a high-level design, the extraction of a supply model from the ASCII text file, a definition of a map between the 2 models, a computation of a reflexion model mistreatment the reflexion model tool and, the last step is that the interpretation of the reflexion model by the reverse engineer. The output of the method may be a reflexion model that highlights the convergences, divergences, and absences between the 2 models. Koschke [26] extends the first reflexion model to stratified design models. His extension permits the engineer to decompose his abstract design at totally different level of details. Also, expert et al. approach [27] relies on the matching between a abstract design associated an extracted design. expert proposes the employment of (1) associate analysis tool, (2) a structuring tool to extract the relationships between components supported the implicit relationships between them and (3) a mental image tool. Kazman et .al propose a tool known as Salvidor Dali [28] that integrates many techniques just like the Rigi tool [29] and also the POSTGREDSQL question tool. These tools permit the extraction of the ASCII text file, the registration of the ends up in a on-line database, the execution of queries given by the reverse engineer that selects

IPASJ International Journal of Information Technology (IIJIT)
Web Site: http://www.ipasj.org/IIJIT/IIJIT.htm
A Publisher for Research Motivation ..... Email:editoriijit@ipasj.org
Volume 5, Issue 2, February 2017 ISSN 2321-5976

supply entities in step with the question and at last Rigi permits result mental image. so as to reconstruct the design in terms of patterns, Guo proposes a semi-automatic analysis technique known as ARM [30]. ARM relies on four phases that aim to (1) develop a recognition arrange, (2) extract a model that represents the ASCII text file mistreatment Salvidor Dali tool, (3) execute the arrange on the supply model conjointly mistreatment Salvidor Dali and (4) analyze the results mistreatment mental image tools. alternative processes may well be classified beneath the 2 classes. Focus approach [31] takes into consideration the subject ideas like subject style; it consists of six phases: (1) parts identification which's established by generating category diagrams, grouping connected categories associated packaging the categories mistreatment cluster tools; (2) idealised subject model proposition which's the creation of a abstract design that respects an subject style; (3) Mapping the parts onto the idealised design; (4) Use case identification; (5) parts interactions analysis and (6) refined architecture generation. The approach planned by Medvidovic [32] relies on Focus approach; it aims to mix techniques accustomed discover the design from purposeful needs with techniques that extract associate design model from the implementation.

3. SPEM

[33] (Software & Systems method Engineering Metamodel Specification) may be a meta-model that specifies the method engineering for system construction. It identifies the fundamental ideas of the package engineering processes in terms of components and relationships. 3 components ar outlined in SPEM: The activities, the work product and also the roles. associate activity may be a unit of labor that has got to be realised so as to attain the method goal. a piece product may be a concrete object that may be manipulated. and at last, a task represents a ability. Also, SPEM defines four relationships: The 'uses', 'produces', 'performs' and 'is accountable for' relationships. The 'uses'/'produces' relationships indicate that a product may be used/produced by associate activity. The 'performs' relationship indicates that associate activity is performed by a task. And, the 'is chargeable for' indicates that a task is accountable for associate whole.

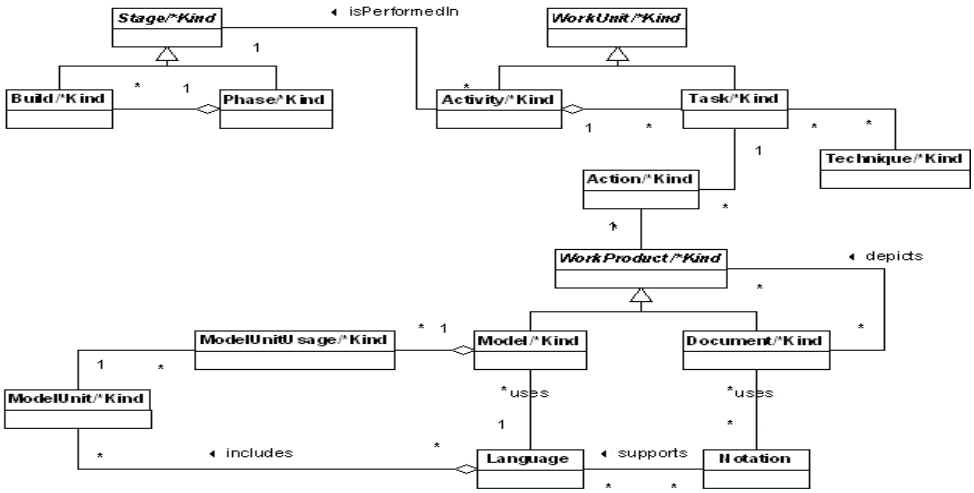

Figure 1: A meta-model based on SPEM.

While SPEM specifies package engineering processes, we have a tendency to believe that it may also specify package design extraction processes. Since associate extraction method may be seen as a collaboration between activities, artefacts and roles, SPEM may well be used as a meta-model for package design extraction modeling method. For this finish, we have a tendency to propose the primary meta-model given in Figure one that reflects that a package design extraction method may be a collaboration between extraction activities, extraction artefacts and extraction roles. we have a tendency to outline associate 'extraction activity' as a piece task performed to satisfy the extraction method, the 'extraction artefact' as a concrete object that may be manipulated by associate extraction activity throughout the extraction method, associated associate 'extraction role' as a ability able to execute an extraction activity. associate extraction role performs associate extraction activity and is chargeable for the manipulated artefacts through the extraction activity.

4. CONCLUSION

This paper has given a meta-model, known as SArEM, for package design extraction. SArEM is focused on the subsequent a pair of concepts: 1st, it's supported SPEM meta-model by respecting its 3 basic principles: work merchandise, activities and roles. Second, it extends every SPEM principle so as to spot a package extraction method

ideas. SArEM doesn't aim to produce its own package design extraction method, rather SArEM outline the flexibility for associate designer to decide on the extraction method that matches his desires. what is more, SArEM may well be thought-about as a start line to match, analyze and valuate existing approaches for package design extraction. once the definition of the necessary ideas that associate extraction method ought to be supported, we have a tendency to aim to outline a sensible tool that respects all SArEM ideas. Our future work is then to outline a tool that meets all the weather of SArEM in terms of activities, artifacts and roles.

REFERENCES

[1]. B. Len, C. Paul, and K. Rick, "Software design in observe," Beantown Mass. Addison, 2003.
[2]. D. Garlan, "Software architecture: a roadmap," in Proceedings of the Conference on the long run of package Engineering, 2000, pp. 91–101.
[3]. E. Korshunova, M. Petkovic, M. G. J. van den whole, and M. R. Mousavi, "CPP2XMI: Reverse Engineering of UML category, Sequence, and Activity Diagrams from C++ ASCII text file," in thirteenth operating Conference on Reverse Engineering, 2006. WCRE '06, 2006, pp. 297–298.
[4]. R. Ferenc, Á. Beszédes, M. Tarkiainen, and T. Gyimóthy, "Columbus-reverse engineering tool and schema for C++," in package Maintenance, 2002. Proceedings. International Conference on, 2002, pp. 172–181.
[5]. Y.-G. Guéhéneuc, "A reverse engineering tool for precise category diagrams," in Proceedings of the 2004 conference of the Centre for Advanced Studies on cooperative analysis, 2004, pp. 28–41.
[6]. T. Quatrani, Visual modeling with rational rose 2002 and UML. Addison-Wesley Longman business enterprise Co., Inc., 2002.
[7]. J. Korn, Y.-F. Chen, and E. Koutsofios, "Chava: Reverse engineering and pursuit of java applets," in Reverse Engineering, 1999. Proceedings. Sixth operating Conference on, 1999, pp. 314–325.
[8]. Y.-F. Chen, E. R. Gansner, and E. Koutsofios, "A C++ information model supporting reachability analysis and dead code detection," Softw. Eng. IEEE Trans. On, vol. 24, no. 9, pp. 682–694, 1998.
[9]. D. Doval, S. Mancoridis, and B. S. Mitchell, "Automatic cluster of package systems employing a genetic algorithmic program," in package Technology and Engineering observe, 1999. STEP'99. Proceedings, 1999, pp. 73–81.
[10]. S. Mancoridis, B. S. Mitchell, C. Rorres, Y.-F. Chen, and E. R. Gansner, "Using Automatic cluster to provide High-Level System Organizations of ASCII text file.," in IWPC, 1998, vol. 98, pp. 45–52.
[11]. S. Mancoridis, B. S. Mitchell, Y. Chen, and E. R. Gansner, "Bunch: A cluster tool for the recovery and maintenance of software package structures," in package Maintenance, 1999.(ICSM'99) Proceedings. IEEE International Conference on, 1999, pp. 50–59.
[12]. B. S. Mitchell and S. Mancoridis, "On the automated modularization of package systems mistreatment the bunch tool," Softw. Eng. IEEE Trans. On, vol. 32, no. 3, pp. 193–208, 2006.
[13]. B. S. Mitchell and S. Mancoridis, "Using Heuristic Search Techniques To Extract style Abstractions From ASCII text file.," in GECCO, 2002, vol. 2, pp. 1375–1382.
[14]. B. S. Mitchell and S. Mancoridis, "On the analysis of the Bunch search-based package modularization algorithmic program," Soft Comput., vol. 12, no. 1, pp. 77–93, 2008.
[15]. O. Maqbool and H. A. Babri, "The weighted combined algorithmic program: A linkage algorithm for package cluster," in package Maintenance and Reengineering, 2004. CSMR 2004. Proceedings. Eighth European Conference on, 2004, pp. 15–24.
[16]. K. Sartipi, "Software design recovery supported pattern matching," in package Maintenance, 2003. ICSM 2003. Proceedings. International Conference on, 2003, pp. 293–296.
[17]. K. Sartipi and K. Kontogiannis, "Pattern-based package design recovery," in Proceedings of the Second ASERC Workshop on package design, 2003.
[18]. G. C. Murphy, D. Notkin, and K. Sullivan, "Software reflexion models: Bridging the gap between supply and high-level models," ACM SIGSOFT Softw. Eng. Notes, vol. 20, no. 4, pp. 18–28, 1995.
[19]. G. C. Murphy, D. Notkin, and K. J. Sullivan, "Software reflexion models: Bridging the gap between style and implementation," Softw. Eng. IEEE Trans. On, vol. 27, no. 4, pp. 364–380, 2001.
[20]. R. K. Keller, R. Schauer, S. Robitaille, and P. Pagé, "Pattern-based Reverse-engineering of style parts," in Proceedings of the twenty first International Conference on package Engineering, New York, NY, USA, 1999, pp. 226–235.