TOWARDS INCORPORATING PROCESS EXECUTION MODELS IN INFORMATION SYSTEMS

[1]Satinder Kaur, [2]Simmi Bagga, [3]Hardeep Singh
[1,3]Assistant Professor, Department of Computer Science and Engineering, Guru Nanak Dev University, Regional Campus, Sathiala, [2]Assistant Professor, Sant Hira Dass Kanya MahaVidyalaya, Kala Sanghian, Distt Kpt. INDIA

Abstract: **During the last decade usability is the hottest issue among researchers. As the end product has been finally utilised by human, so effective guidance in terms of information systems is must. Today, a large manual which contains detailed instructions in form of chapters has very less value and users want somewhat interactive human guided information systems. To design an efficient interactive information system, some processing should be involved not in early stages of its development but also in execution phase of it. This paper addresses this issue and gives an idea how to use different process models for visualizing execution processes during the use of information systems to enhance its usability.**

Keywords: **Human guided systems, Work flow management systems, modelling language.**

1. INTRODUCTION

Today, there is a need to manage the execution process of an information system effectively and show its execution status positively to end user. A typical question for researchers today is "how to make an information system more efficient and effective to use?" Traditional systems use the process models only in the development phase of an information system. By introducing new means of information representation during the use of information systems, its usability can be enhanced. There are a lot of examples on usage of process models in complex engineering systems. To represent electricity supply networks, customers and suppliers are shown with monitoring screens whereas grid patterns show power lines between them. Process models are also used to retrieve information about certain parameters to examine certain processes. Processes can be started by just clicking certain entities like clicking on the particular power producing entity; electricity production capacity can be initiated. Similarly, on the same idea, one can start any activity by just clicking the related entity in the model representing the information system.

Information systems which are represented in the form of process models can be shown with the help of tree diagrams with increased detail at each level [1]. The main components of diagrams are graphical symbols representing repositories as data objects, entities as real objects, connections as control and data flows, time constraints and other artefacts as states, informal descriptions about activities processing, states and flows as comments. Now the need is to make it possible to execute an information process by just click on a particular component. The next section describes the general solution. Third section deals with previous work which has been done in this research area. Implementation and visualization issues are discussed in fourth section while fifth section describes the conclusions.

2. TRADITIONAL METHODS AND PROPOSED METHOD

Traditional systems use elements of interface controls like windows, menus, forms etc. to execute some user oriented tasks. Also, these systems presume that the user knows the exact sequence of activities that must be performed on menus, buttons and/or screens to accomplish specific tasks. The problem is that a layman has a lot of problems in operating these systems. So, usability which is the major aspect of information systems gets hindered.

So, there is a need for representation which is easy to operate. Humans feel easy to understand and operate a system which represents the live world as entities, objects, information flows and

processes. So, object oriented graphical diagrams can be made on which user can initiate process execution, choose certain paths, open certain documents, and finish certain processes by just clicking it rather than remembering various menus and sequences for certain operations. One can make the already executed activities visually depicted by colouring it to different colour from activities which are not executed yet. So, the proposed solution is to represent information systems with the help of process execution models.

The proposed solution is feasible only if:

- It is possible to represent precisely and definitely all elements necessary for execution of information system.
- User can easily execute the model without any system intervention in an automated way.

Main tasks for implementation of process execution models are:

- Design the process execution models and represent them with graphical diagrams.
- Represent all the elements needed for direct execution i.e. activities, data objects, control as well as data flows.
- Choose the modelling language according to domain [2] and tool for creating and editing of process descriptions and make it open for external use via API for developers.
- Link executable routines [5] to specific elements of the model to make it executable.

Now the users can execute certain actions by just clicking on the element. Two different types of experts involved in it. First are modellers which are responsible to design modelling language and tools which provide API to model such systems. Second are model developers which actually design and implement the process execution models for information systems.

3. RELATED WORK

Many information systems exist today which are in execution mode. One of the famous among them is Workflow Management Systems [3][4]. In these systems, processes are automated in such a way that documents, information or tasks are scheduled and passed to next active participant only if previous participant complete its action on them according to some defined procedural rule. The tools for these systems are JIRA [6] and KiSSFLOW [7] which is used in team oriented work systems. The important features of workflow management systems are graphical diagrams, data objects as screen forms, defined entities and their roles with accessibility criteria etc. [8]. So, process execution models take partially the idea from workflow systems for their implementation. Theoretical researches are hard to implement them as they defined axiomatic approaches [9]. In work flow systems, users of system are role-based whereas process executed systems are meant for end users. The main tool to represent workflow systems are Petri networks [10].

Another approach that gives idea for process execution models are from Model Driven Software Development (MDSD) approach [11] which is also based on graphical and CASE tools. The researchers of MDSD give the idea of abstraction of software description to enhance the software productivity as compared to other methods. So, linking of executable routines later on gives independence in software development. Although the use of tools of MDSD is limited, research and improvement is still alive in this area [12][13]. Also MDA (Model Driven Architecture) approach suggests modelling of information systems and then translates them into executable code.

4. FORMALIZATION AND EVALUATION OF DIRECT PROCESS EXECUTION APPROACH:

4.1. STRUCTURE OF MODEL

In hypothetical form it consists of two parts:

- Memory: It is a container that stores different type of instances of data objects and processes. It has further two parts to store data object instances and process instances.
  Data object part stores four types of objects i.e. new data objects, active data objects, processed data objects, archived data objects. New data objects are in wait state for processing, active data objects are in processing state, processed data objects which have finished processing and can be used again and archived data objects which have finished processing but cannot be used again.
  Process instances can be created by clicking on some particular process in graphical diagram. After that the process has to be linked to some particular data object for execution. So, memory stores two types of process instances i.e. active and passive process as shown in Fig 1.

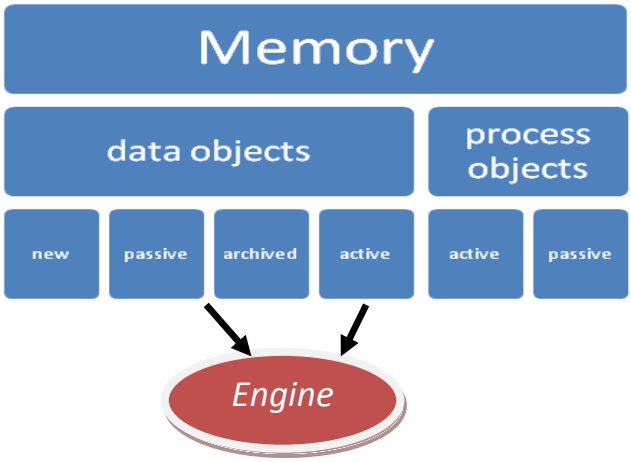

Fig 1- Structure of Model

- Engine : It is able to execute particular instances of active processes consecutively.

4.2. WORKING OF MODEL

As shown in Fig-2, working involves the following steps:

- User login into the information system.
- System checks authentication and authorization, if satisfy granted access to execute certain processes on certain objects.
- User choose a process for execution by clicking in process model diagram and link it to some particular data object, process can be new one or previously started but not finished yet.
- System creates a new instance of process and linked it to data object, and starts execution.
- If user clicks on a suspended process, system finds it in active process part of memory and start execution from where it was previously interrupted.
- Processes are executed step by step and user can see their progress in process diagram.

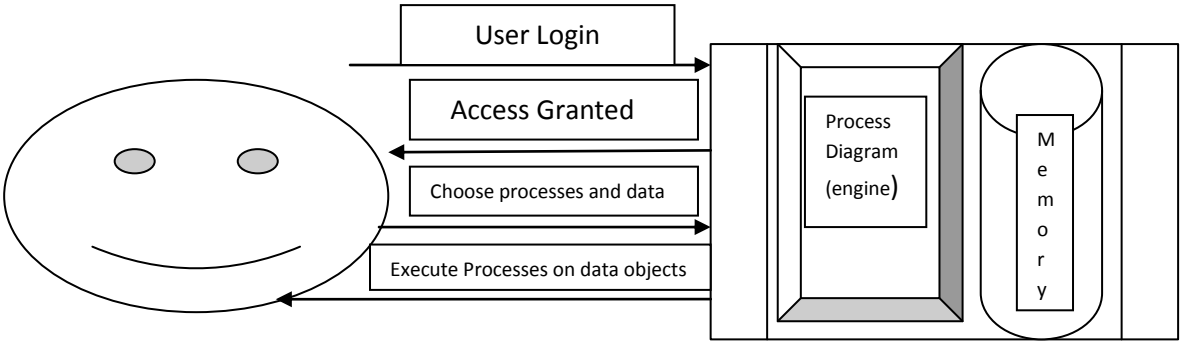

Fig-2 Working of Process Execution Model

Furthermore steps can be made to be executed manually i.e. by user clicking, semi-automatic also by user clicking (adding new data) but further activities need not much user involvement as in editing the old data or fully automatic i.e. without user interaction e.g. to save some data in repository.

4.3. REPRESENTATION OF DATA OBJECTS AND PROCESSES

To represent data objects screen forms are used, which contain many fields to be added/ edited/ deleted. It can also serve for selecting of data objects. The main instructions for making forms more effective are:

- show attribute values in screen forms so that users recognize the necessary objects
- allow filter data objects in screen forms to narrow the set of object instances
- colour fields in screen forms according to the data objects instance status (for instance,

the corrigible fields could be shown in a different colour)

- Furthermore, some data columns or fields could be disabled (not displayed at all) if the user does not have rights to access the data objects.

Traditional programming methods can be used for developing screen forms. To represent process execution, one can use graphical diagrams. One can use different colours to represent executed processes, active processes (shaded form) and processes not initiated yet. Also, the processes for which user has no right, must be shown in faded way.

4.4. PROCESS EXECUTION MODELLING ENVIRONMENT DIMOD

To create and use process diagrams, one needs a specific modelling tool and DIMOD is one of them. The tool DIMOD can be used in a domain specific language (DSL)[17]. It can publish the diagrams to the web; check the business process models' internal consistency and perform other modelling functions well. A specific feature is included in DIMOD which call APIs in predefined points to implement DSL specific semantics.

4.5. EVALUATION OF PROPOSED MODEL

The recommended usability metrics by ISO 9126 are:

- Effectiveness – A system is effective only if it meets user goals with full accuracy and completeness. It can be measured in terms of Percentage of completed tasks, Ratio of failures and successes etc.
- Efficiency – A system is efficient only if it is easy to operate, learn and recall certain actions in it. So, it must be user friendly.
- Satisfaction – Users are satisfied only if they can use the system comfortably.

As process execution models are completely transparent, user goals are fully accurate and complete, so effectiveness is achieved. Next, these systems are easy to operate and one should not put burden on brain to recall its activities, so they are efficient too. Finally, users can use them comfortably as there is no harm on existing diagram due to creation of new instances of processes, so satisfaction is achieved. In a survey, it has been shown that one graphical diagram can substitute 8 to 10 pages of textual instructions [18].

5. CONCLUSION

This paper presents a process execution model approach to represent information systems effectively. It can be represented to user with help of graphical diagram which consist of processes and data objects which are connected for execution of certain activities. A user can login the system to perform certain actions. He has to only click on certain process to activate it and data can be added or edited in forms. The user can also see the progress of execution. So the system is fully transparent during execution. A particular DSL can be used for modelling the system. Finally it concludes that model is fully usable.

REFERENCES

[1] Karnitis, G., Bicevska, Z., Cerina-Berzina, J., Bicevskis, J. *Practitioners approach to business processes modelling*. In: Selected papers from the 11th International Baltic

Conference DB&IS 2014, IOS Press, 2014, pp. 343-356.

[2] Barzdins J., Cerans K., Grasmanis M., Kalnins A., Rencis E., Lace L., Liepins R., Sprogis A., Zarins A., *Domain Specific languages for Business Process Managment: a Case Study*. In: Proceedings of DSM'09 Workshop of OOPSLA 2009. [WWW] http://www.dsmforum.org/events/DSM09/ (accessed 01.08.2014)

[3] *Workflow Management Coalition homepage*, retrieved: 16.11.2015, URL: http://www.wfmc.org

4. *Top Workflow Management Software Product*s, retrieved: 16.11.2015, URL: http://www.capterra.com/workflow-management-software/

5. Stahl T., Voelter M., Czarnecki K. *Model-Driven Software Development: Technology, Engineering, Management.* Chichester: John Wiley & Sons, 2006. 428 p.

6. JIRA homepage, retrieved: 16.11.2015, URL: https://www.atlassian.com/software/jira

7. KiSSFLOW, retrieved: 16.11.2015, URL: *https://kissflow.com/process_playbook/workflow-management-system-10-must-have-features/*

8. Draheim D., Atkinson C. *Business Process Technology: A Unified View on Business Processes, Workflows and Enterprise Applications*, Berlin: Springer, 2010. 300 p.

9. Cicekli, N. K., Yildirim, Y. *Formalizing Workflows Using the Event Calculus.* Springer, LNCS, Vol. 1873, 2001, pp. 222-231

10. Mateo, J., Srba, J., Sorensen, G. *Soundness of Timed-Arc Workflow* Nets Springer, LNCS, Vol. 8489, 2014, pp. 51-70.

11. Fowler M. *Patterns of Enterprise Application Architecture. Boston: Addison*-Wesley Professional, 2002. 533 p.

12. Stahl T., Voelter M., Czarnecki K. *Model-Driven Software Development: Technology, Engineering, Management.* Chichester: John Wiley & Sons, 2006. 428 p.

13. Buchmann T., Dotor A., Westfechtel B. *Model-driven software engineering: concepts and tools for modeling-in-the-large with package diagrams*. Computer Science - Research and Development, 2014, 29(1), 73-93.

14. Model driven architecture. [WWW] *http://www.omg.org/mda/ (accessed* 01.08.2014)

15. Usability Metrics, retrieved: 16.11.2015, URL: http://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/

16. Cerina-Berzina, J., Bicevskis, J., Karnitis, G. *Information systems development based on visual Domain Specific Language BiLingva*. In: Selected Papers from the 4th Conference CEE-SET 2009, Krakow, Poland, Springer, LNCS 7054, 2011, pp. 124-135.

17. Sprogis, A., Barzdins, J. Specification, *Configuration and Implementation of DSL Tool*, Frontiers in Artificial Intelligence and Applications, vol. 249, IOS Press, p. 330-343, 2013.

18. A.Sproģis, R.Liepiņš, J. Bārzdiņš, K. Čerāns, S. Kozlovičs, L. Lāce, E. Rencis, A. Zariņš, GRAF: a Graphical Tool Building Framework, Proc. of ECMFA 2010, Paris, France