
Reengineering a Software Process by using Unified Foundation Ontology

Meenakahi.T

Assistant Professor, Department of Computer Science,GFGC,Vijayanagar, Bangalore,India.

Abstract: Throughout project planning, learning about software processes is helpful in a few circumstances: software processes are characterized, activities are scheduled, and personnel's are allotted to these activities. In this setting, standard software processes are utilized as foundation for characterizing project processes, and tools are utilized to help planning, personnel allocation, etc. Conceivably, individuals and devices ought to impart a typical conceptualization in regards to this area for permitting interoperability, and right utilization of the tools. Domain ontology might be utilized to characterize an explicit representation of this imparted conceptualization. Additionally, for domain ontology to sufficiently serve as a kind of perspective model, it ought to be fabricated expressly considering foundational ideas. This paper discusses about the reengineering of a piece of a Software Process Ontology dependent upon the Unified Foundational Ontology (UFO). The part reengineered concerns standard processes, project processes, and activities, which are dissected at the light of UFO ideas.

Keywords: Reengineering, Domain Ontology, Unified Foundation Ontology.

Reference to this paper should be made as follows: Meenakahi.T (2014) 'Reengineering a Software Process by using Unified Foundation Ontology', *International Journal of Inventions in Computer Science and Engineering*, Volume 1 Issue 3 2014.

1 Introduction

Dealing with a project includes applying knowledge, aptitudes, tools and procedures to extend activities to reach its prerequisites. Such knowledge application requires the successful administration of suitable processes [1]. Around these processes, we can highlight the vitality of the project Management Process, which is capable, around others, for characterizing activities, scheduling them, and allotting human resources for each one planned action.

With respect to projects, project administration is a discriminating procedure. Throughout project planning, software processes ought to be characterized, their activities ought to be scheduled, and individuals ought to be dispensed to them. In this connection, standard software processes are utilized as premise for characterizing project processes. When the processes are characterized, devices are utilized to help planning, team assignment, etc. Throughout project execution, activities are performed and individuals fulfill their tasks, recording the time they used on them.

For the most part, distinctive tools are utilized to help such tasks. Acknowledging that these tasks are iterative and interrelated, in a perfect world these tools ought to interoperate. Then again, keeping in mind the end goal to rightly utilize the tools, developers and tools ought to impart a typical conceptualization in regards to the space of software processes. Indeed, for both purposes of interoperability between tools and consonant utilization of them by developers, we require an imparted conceptualization in regards to the software process space. In this setting, domain ontology might be utilized to characterize an express representation of this imparted conceptualization.

For having the capacity to sufficiently serve as a kind of perspective model, domain ontology ought to be developed utilizing an approach that expressly considers foundational ideas. The utilization of foundational ideas that consider positively ontological issues important is getting to be an ever increasing amount acknowledged by the ontological engineering group, particularly for speaking to complex domains[2] , for example, is the situation of software processes. Reference ontology is produced with the point at speaking to the subject domain with truthfulness, clarity and expressivity, paying little respect to computational necessities. The primary objective of this reference model is to help modelers externalize their knowledge about the domain, to make their ontological responsibilities unequivocal to help significance arrangement, and to enhance the assignments of domain correspondence, learning and critical thinking[2].

This paper examines the reengineering of some piece of the Software Process Ontology (SPO) initially proposed in [3][9], taking into account the Unified Foundational Ontology (UFO) [4][5]. This ontology was part of the way reengineered at the light of UFO in [5]. This first reengineering activity concentrated on recognizing process/action and process/movement event, around others. In this paper, regardless we take after the methodology embraced in[5] , i.e., to adjust the SPO ideas and relations to the ideas and relations of UFO. In any case, we concentrate on moving forward these qualifications dissecting ideas, for example, standard software processes, project processes and activities, at the light of UFO ideas of occasions, duties, arrangements and standardizing portrayals. Our focus is regulated towards ideas included in administration activities that are identified with process definition, planning and

asset assignment, since we expect to utilize the reengineered rendition of this ontology as foundation for incorporating instruments supporting these activities.

II. Unified Foundation Ontology

UFO is a foundational ontology that has been produced focused around various speculations from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive psychology. It is formed by three primary parts. UFO-A is an ontology of endurants [4][7]. UFO-B is an ontology of perdurants . UFO-C is an ontology of social substances(both endurants and perdurants) based on the highest point of UFO-A and UFO-B[5].

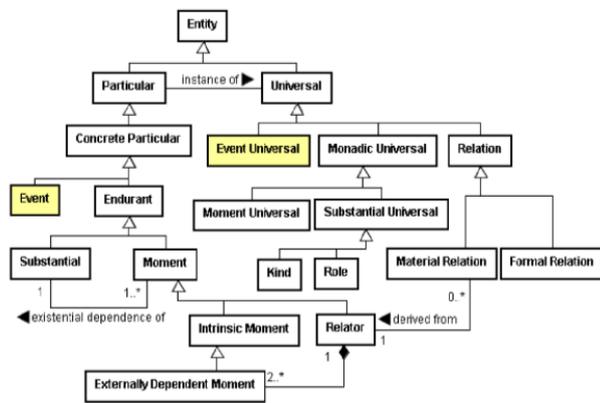


Fig 1 UFO A- Ontology of Endurants

As indicated in Fig 1, a central distinction in UFO-A is between particulars and universals. Particulars are elements that exist as a general rule having a special personality, while universals are examples of characteristics, which might be acknowledged in various distinctive particulars. Substantials are existentially autonomous particulars. Moments, interestingly, are particulars that can just exist in different particulars, and accordingly they are needy of them. In this sense, moments are existentially subject to different particulars. Existential reliance can likewise be utilized to separate inherent and social moments: intrinsic moments are subordinate of one single singular (e.g., a shade, a temperature), while relators rely on upon a majority of people (e.g., a business, a medicinal medicine).

Concerning universals, while persevering in time, substantial particulars can instantiate a few substantial universals. Some of these sorts, a substantial instantiates fundamentally (i.e., in each conceivable circumstance) and characterize what the substantial is. These are the sorts named kind. There are, in any case, sorts that a substantial instantiates in a few circumstances, however not in others, for example, is the situation of parts. A part is a sort instantiated in a given connection, for example, the setting of a given occasion interest or a given connection (e.g., learner). Both kind and part are sortal substantial universals, yet kind is an unbending sortal, while part is against inflexible sortal. Despite the fact that not spoke to in Fig 1,

Sortal Universal, Rigid Sortal and Anti-rigid Sortal are ideas of UFO-A.

Relations are substances that paste together different elements. Formal relations hold between two or more substances specifically, without any further mediating single person. Material relations, alternately, have material structure of their own. As such, for a material connection to exist, an alternate element (a relator) must exist, which intercedes the related elements. In this manner, relators are particulars with the force of uniting elements. The relations between relators and the joined elements are said intervention relations.

III. Software Process Ontology

Throughout project planning, we have to characterize the project procedure, plan its activities, and assign individuals to perform them. For purposes of project observing and control, we have to track the achievement of the activities, and the time used to perform them.

Throughout the definition of a software process for the project, the project supervisor ought to distinguish the activities that must be performed with a specific end goal to accomplish the project objectives. This is carried out by customizing authoritative standard processes, taking the project particularities and team characteristics into account. The project process is the foundation for the further project administration activities[10]. After characterizing the process, the project supervisor need to make a system of activities, characterize to what extent every movement will last, and allot individuals to perform it. For a great understanding of these assignments, we require an imparted conceptualization in regards to software processes.

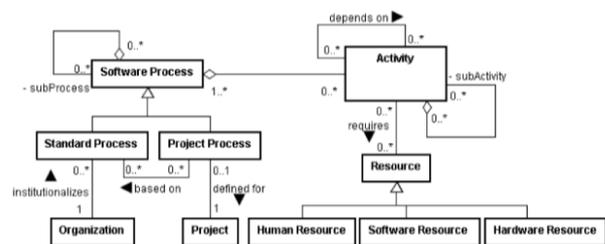


Fig 2 Software process ontology

The Software Process Ontology (SPO) initially created in [3] was developed pointing at creating a typical conceptualization for software associations to discuss software process. It was separated into four sub- ontologies, to be specific: action, resource, strategy also software process ontologies. Fig 2 shows a piece of the first form of this ontology that incorporates ideas from the action, resource, and software process sub-ontologies.

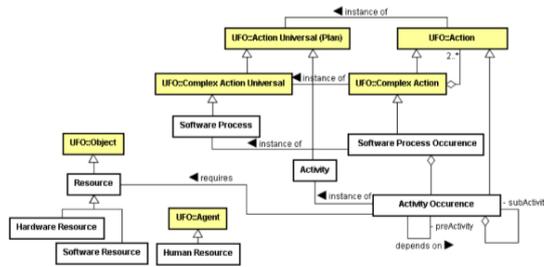


Fig 3 Reengineered SPO

As mentioned above, the first form of the SPO was object of a fractional reengineering in [7] by mappings some of its ideas to ideas of UFO. Fig 3 shows a part of this reengineered form of SPO, demonstrating ideas from UFO separated.

By translating the first form of the SPO regarding UFO, [5] pointed out that the previous breakdown the thoughts of action universals and actions. To tackle this issue, they presented the ideas of Activity Occurrence and Software Process Occurrence to indicate specific actions that occur in particular time interims. In addition, they said that a software process event is an example of a software process, which is, thus, a Complex Action Universal. Similarly, an action event is an occurrence of an action, which is an Action Universal (Plan).

The thought of resource in the first form of SPO was mapped to the idea of (Non- Agentive) Object in UFO, and the idea of human asset was no more recognized a resource. Contrariwise, human asset is viewed as a kind of Agent. This opened space for recognizing article investments from operator cooperations. Accordingly the connection requires in the unique variant of SPO subsumes diverse modes of partaking in an action event, specifically question investments and action commitments[6]. An action commitment of a human asset really means a social responsibility of that operator (with ensuing consents and commitments) of performing some piece of that movement event[9]. In this way, they obliges connection for the instance of human assets is a sort of reliance connection between operators that will prompt a designation connection at the point when the process is instantiated or planned.

IV. Conclusion

In this paper we introduced the new developments made in reengineering a section of the Software Process Ontology (SPO) initially characterized in [3], making further qualifications than the ones made in [5]. The freshest rendition was gotten from the applied arrangement of the ideas and relations characterized in the SPO with the ideas and relations of the Unified Foundational Ontology (UFO). The focus was to address the conceptualization identified with software processes at the light of software project arranging.

The utilization of UFO turned out to be helpful for recognizing issues and for driving the ontology

reengineering, particularly depicting the ontological duties that were understood. Such process had unequivocal the effect between standard process, project process, planned process and process event.

References

- [1] PMI, A guide to the project management body of knowledge, 4 edition, 2008.
- [2] Guizzardi, G., Baião, F., Lopes, M. and Falbo, R.A. The Role of Foundational Ontologies for Domain Ontology Engineering: An Industrial Case Study in the Domain of Oil and Gas Exploration and Production. *International Journal of Information System Modeling and Design (IJISMD)*, Vol. 1, Issue 2, pp. 1-22, 2010.
- [3] Falbo, R. and Bertollo, G. A software process ontology as a common vocabulary about software processes. *International Journal of Business Process Integration and Management (IJBPIM)*, v. 4, p. 239- 250, 2009.
- [4] Guizzardi, G., *Ontological Foundations for Structural Conceptual Models*, Telematica Intutuut Fundamental ResearchSeries no. 15, The Netherlands, 2005.
- [5] Guizzardi, G., Falbo, R.A. and Guizzardi, R.S.S. Grounding software domain ontologies in the Unified Foundational Ontology (UFO): the case of the ODE software process ontology. In *Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments*, 244-251, 2008.
- [6] Silva, E.O., Lisboa, F.J., Oliveira, A.P., Gonçalves, G.S.: Improving Analysis Patterns in the Geographic Domain Using Ontological Meta-properties. *Proceedings of Int. Conference on Enterprise Information Systems*, pp 256-261, 2008.
- [7] Liao, L., Qu, Y. and Leung, H.K.N. A Software Process Ontology and Its Application. In *Proceedings of the First International Workshop on Semantic Web Enabled Software Engineering*. November 2005.
- [8] Falbo, R. A. and Nardi, J. C. Evolving a Software Requirements Ontology. *XXXIV Conferencia Latinoamericana de Informática*. Santa Fe, Argentina, pp 300-309 (2008).
- [9] Barcelos, M.P., Falbo, R.A. and Dalmoro, R. A well-founded software measurement ontology. In *Proceedings of the 6th International Conference on Formal Ontology in Information Systems (FOIS 2010)*. Toronto - Canadá, 2010.
- [10] SEI, CMMI for Development, Version 1.3, Technical Report CMU/SEI-2010-TR-033 November, 2010.