Toward A Reusable Domain Analysis

Patricia Collins

Hewlett-Packard Company
1801 Page Mill Road, Bldg. D
Palo Alto, CA 94304
collins@hpcea.ce.hp.com

Abstract

We are developing a set of checklists and a process model for commercial business product reuse efforts. We describe our approach to scoping the activities of Domain Analysis (DA) in order to optimize reusability of the products (e.g., domain models) of that analysis. In particular, our model for the Domain Analysis activity is complemented by a distinct activity for analyzing the reusability constraints placed by the product development environment (people, tools, process, objectives, etc.). We refer to this as a "Reusability Analysis." We contend that one benefit of distinguishing between an analysis of the domain and an analysis of the environment in which workproducts will be reused is that the domain models and related workproducts of the Domain Analysis will be reusable, even if software components cannot be reused across the range of intended domain products.

Keywords: Domain Analysis, Reusability, Software Generation Process

1 Introduction

Hewlett-Packard Company has diverse lines of business in the commercial marketplace. HP's electronic instruments, medical and analytical equipment, and computer systems with vertical and horizontal application products all include large software (and firmware) components. Within a single line of business, there is extensive diversity in particulars of the software generation and maintenance processes. A single product family may span numerous hardware platforms, include multiple programming languages, interface with diverse database systems, address niche markets or diverse international standards, etc.

The time-to-market and return-on-investment criteria for products are driving these organizations to consider reuse practices. Some early adopters of reuse objectives have "backed into" performing domain analyses in order to ensure the development of reusable components. We are producing a set of checklists and a process model that incorporate best practices from these reuse practitioners. The practitioners serve as invaluable resources in the refinement of the reuse process model, so that prescriptive guidance is tempered by practical experience.

2 Approach

The first phase of our investigation has essentially constituted a domain analysis of Domain Analysis. Our efforts have included gathering data from the literature on prescriptive domain analyses, interviewing domain analysis experts, interviewing members of engineering teams that would be users of the domain analysis results, looking at existing practices (whether or not they are called "domain analysis"), and generating a model of the process that could be used across many application domains.

Initial steps of the process modelling involve defining the context for the domain analysis: what are the peer activities, and what is the encompassing activity. Our setting of the DA activity boundaries is based on the various published uses of the term "domain analysis," as well as on a consensus among the reviewers of the model as to the prescriptive goals of the process (i.e., what Domain Analysis *ought* to provide for commercial product development projects). [1]

We define three interdependent activities in reuse: Developing Reusable Workproducts (which includes Domain Analysis), Developing Software Products (with reusable workproducts), and Managing Workproducts. Within Developing Reusable Workproducts, we identify Analyzing The Domain (Domain Analysis), Generating Reusability Requirements (Reusability Analysis), and Engineering Reusable Components as the three primary activities.

From the Domain Analysis process model, we extract a hierarchical set of checklists that can be used in reviews. The checklists are designed to ascertain the quality and completeness of the domain model [2] and other DA workproducts, whether or not a defined process has been explicitly followed. Through review of projects that are intentionally designing reusable workproducts, we will refine our process model with best practices and provide timely feedback on weaknesses in projects' analyses.

We have chosen the checklist format as a near-term delivery vehicle for what we have learned. This enables us to give immediate support to projects with reuse objectives who are already in their analysis phase. For projects in progress, the checklist serves as a set of guidelines, reminding the team what questions they need to answer before they have finished their analysis. (Note that this does not imply a waterfall lifecycle. This is discussed further in Addressed Issues.)

To be of practical use in product teams, the process model will need to be modified with feedback from early adopters. Until the process model has stabilized, with careful assessment of its performance in application product development environments (firmware, application and systems software products), we do not want to oversell the model's benefits. Reaction in product teams to checklists is more forgiving. There is less expectation that checklists will provide complete coverage of a process, and the interactions among activities are not expected to be called out explicitly.

[1] We are employing the IDEF0 [IDEF90] practice of successive refinement of the model, with quick-turnaround reviews of top-down generated process activities, constraints, mechanisms, inputs, and outputs.

[2] Typically today, a project initially plans to produce a reusable architecture that embodies their domain model.

3 Addressed Issues

3.1 What Tasks Fall Within A Domain Analysis?

We contend that the proper focus of a domain analysis is the modelling of the domain in the "problem space," separate from the task of engineering a (reusable) solution based on the domain model. The domain analysis, together with a *reusability analysis* [3] constitute the inputs to Engineering Reusable Workproducts for this domain. Our motivation in disambiguating the role of Domain Analysis is to ensure that the products of the DA efforts are reusable, independent of the choice of system analysis and design paradigm. We find this to be consistent with the philosophy of design for reusability, in which bindings are delayed, and flexible application of workproducts is an explicit goal.

3.1.1 Systems Analysis and Design

The distinction between domain analysis and systems analysis and design tasks is often blurred. Our process model draws the distinction by advocating Domain Analysis outputs (e.g., feature models, E-R Diagrams, terminology dictionaries) that are independent of the software paradigm and methodology to be used in Engineering Reusable Components. We affirm the importance of revisiting Domain Analysis issues and decisions in light of early explorations of the reusable architecture and components design. Our process model distinguishes the analysis activities that *directly* address the needs of those developing software products (the "customers" for the reusable workproducts). We refer to this as "Reusability Analysis."

3.1.2 Do Boundaries Imply A Waterfall Lifecycle?

Our basic Domain Analysis model is from the perspective of those involved in the execution of the analysis (e.g., a domain analyst). The modelling notation system supports concurrent and iterative interpretations on the execution of the process.[4] The model includes feedback paths within the Domain Analysis activity, suggesting successive refinement of the DA workproducts. Additionally, there are important feedback paths between the DA activity and other activities within Developing Reusable Workproducts (DRW), and between DRW and Managing Workproducts (MWP) and Developing Software Products (DSP).

3.2 Role Of The Domain Analyst

With this scoping of the DA process, we clarify the role of the domain analyst as that of language interpreter, understanding the domain language of the users and customers and translating that into the features and relationships that will be used by the engineering team that produces the reusable components. The Domain Analyst says nothing about how to implement the reusable workproducts, imposing no particular software paradigm (e.g., object-oriented) in the domain model.

[3] See Section 4.1.2.

[4] We also model "planar views" [DURAN91] that enable us to show other perspectives, for example, emphasizing those feedback paths that would support a spiral lifecycle model, or highlighting the continuous process improvement analysis and feedback loops.

4 Identified Open Issues

Our research has resulted in identification of issues that require further investigation. As our process model matures, we intend to address many of these issues and to articulate solutions.

4.1 Distinction Between Domain Analysis and Market/Reusability Analyses

Analogous to our work in distinguishing Domain Analysis from Systems Analysis, we are investigating process models for Market Analysis to understand its role with respect to Domain Analysis. Because we have identified Reusability Analysis as a distinct task in Designing Reusable Workproducts, we are also investigating a process model for that activity and its interaction with Domain Analysis.

4.1.1 Market Analysis

Many enterprises have developed sophisticated market analysis processes that employ much the same philosophy as Domain Analysis[QFD86, MCCAIN85]. They conduct extensive interviews of users and customers of products, review the literature, analyze technology trends, and produce a characterization of the desired features of future products, with prioritization. More progressive groups employ creative techniques to ensure innovation, much like the DA practice of exploring novel combinations of features. We contend that some of the efforts in a full Market Analysis map to the early information-gathering phases of a traditionally-defined Domain Analysis. While Market Analysis provides significant input to the DA process, the two processes are distinct.

4.1.2 Reusability Analysis

We define the analysis of the constraints on reusability (i.e., what enables the product development teams to *use* the workproducts) as an important and separate activity in Developing Reusable Workproducts. While some existing process models outline the Analyzing Reusability activity [PRIETO91], our reuse engineers who are Engineering Reusable Components need explicit information on the content of the reusability requirements workproduct, and details on how to conduct the analysis. Arango [ARANGO89] models the reusability analysis as a meta-process, a learning system that iterates over the reuse process. We find that modelling Reusability Analysis as a peer activity to Domain Analysis provides practitioners with a clearer sense of the tasks that must be completed to Develop Reusable Workproducts. Our current model for Analyzing Reusability is analogous to analyzing customer requirements, where the "customers" are the product development team members and the "product" is the reusable workproducts.

Research into characteristics of reusable workproducts, and the impact of specific software development practices and environments on the design of reusable components is being conducted. Systematic analysis of claims (e.g., comparing reusability of an object-oriented approach with a traditional structured approach) would be a welcomed contribution (and is beyond our project's current scope).

4.2 Assessing The Cost/Benefits

Metrics to tease out the parameters of "good" Domain Analysis (and Reusability Analysis) have not been experimentally validated in controlled comparisons. We believe that data gathered in such assessments will contribute to the refinement of prescriptive domain analysis process models. As part of our program, we are partnering with early adopters of reuse approaches to gather before-and-after data. We also advocate explicitly modelling the Continous Process Improvement (CPI) process, showing acquiring data, assessing the costs and benefits of a particular approach to reuse, and supporting iteratively improving the DA process itself. Gathering informative data on the benefits of our Domain Analysis process implies an understanding of the overall reuse process, since benefits (and some expenses) are accrued in other phases of the reuse process. Since any reuse process is tailored to a class of environments, we plan to "plug" *our* reuse process into an existing HP CPI process model to find opportunities for process improvement. A companion project is researching appropriate metrics and analysis methods for accurately reflecting costs and benefits.

5 Status

An extensive review of the literature has been completed. We have identified activities, looking for commonality across the reported uses (e.g., [FODA90, SPC90]), and focusing on guidelines appropriate to diverse commercial (rather than defense) enterprises. Domain Analysis consultants have been interviewed, toward the goal of refining the model. The top three layers of the process have been modelled and have stabilized through peer reviews. The remainder of the model is in development and review.

We have identified project teams in distinct commercial markets that have agreed to participate in bidirectional reviews (of their projects' processes, and of our process model). We will employ the Domain Analysis checklists that are linked to the process model, rather than using the process models directly in these early assessments.

6 Related Work

References

[ARANGO89] Guillermo Arango, *DOMAIN ANALYSIS- From Art Form to Engineering Discipline*, Proceedings of the 5th International Workshop on Software Specifications and Design, 1989. pp. 152–159.

[DURAN91] Pat Duran & C.A. Irvine, *Planar Views: Multi-Layer IDEF0 Modeling*, Eclectic Solutions Corporation, 1991.

[FODA90] Kyo C. Kang, et al., *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, CMU/SEI-90-TR-21, November 1990.

[IDEF89] C.A. Irvine, *IDEF0 Model Dynamics, Activations and Simulation*, Eclectic Solutions Corporation, 1989.

[IDEF90] IDEF0 Course, Eclectic Solutions Corporation, 1991.

[MCCAIN85] Ron McCain, *Reusable Software Component Construction: A Product-Oriented Paradigm*, in Domain Analysis and Software System Modeling, Prieto-Diaz & Arango, ed., IEEE Computer Society Press Tutorial, pp. 70-80.

[PRIETO87] Ruben Prieto-Diaz, *Domain Analysis For Reusability*, Proceedings of COMPSAC'87, 1987, pp. 23–29.

[PRIETO91] Ruben Prieto-Diaz, *Making Software Reuse Work: An Implementation Model*, Position Paper for the First International Workshop on Software Reusability, Dortmund Germany, 1991.

[QFD86] L.P. Sullivan, *Quality Function Deployment*, Quality Progress, American Society for Quality Control, June 1986, pp. 39–50.

[QFD87] Robert King, *Listening to the Voice of the Customer: Using the Quality Function Deployment System*, National Productivity Review, Vol. 6,No 3, Summer 1987, pp. 277–281.

[SPC90] Software Productivity Consortium, *A Domain Analysis Process*, DOMAIN_ANALYSIS-90001-N, Version 01.00.02, January 1990.

[SIMOS90] Mark Simos, *Alternative Approaches To Domain Analysis: Steps Toward A Job Description*, course notes, 1990.

7 About the Author

Patricia Collins is Projects & Process Manager for Hewlett-Packard Company's Software Initiative Reuse Program. Prior to this appointment, Patricia served as Project Manager for research in systems software and software engineering at HP Laboratories. Her prior research was in speech recognition and speech synthesis. She holds an MSEE from Stanford University and a BA in Mathematics and Philosophy from Dickinson College.