

Session 4: Day 1 Summary and Issues

Dewayne E. Perry

Wilhelm S. Schaefer

Lawrence G. Votta

Software Production Research
Bell Laboratories
Murray Hill, NJ 07974

Computer Science Dept
University of Paderborn
D-33095 Paderborn, Germany

Software Production Research
Bell Laboratories
Naperville IL 60566

In summarizing the product line experience sessions Votta and Schaefer posed three questions:

- What defines product reuse? What are the dimensions of product reuse?
- How do we measure the costs and benefits of reuse?
- What are the necessary and sufficient conditions for product lines?

Important dimensions in the considerations of product reuse are the business model, the market conditions and aspirations, and the process necessary to produce the product.

The process itself is very important in considering the costs and benefits of product reuse, since those factors are embedded in the process itself. Cost, quality and time to market are driving factors. Reuse is one means of leveraging all three factors — it reduces costs, attains previously established quality levels and should ease the time to market pressures.

The discussion centered on the third question — that of the necessary and sufficient conditions for product lines. Two critical aspects are functionality and shared structure. This shared functionality and structure enables a company to achieve the general goal of ‘better, faster, cheaper’.

Schaefer pointed out that product line architecture is important. Because of this he did not consider Microsoft Office to be a product line. Balzer claimed that, while important, architecture should not be a defining function of a product line. Bertztiss noted that according to common English usage Microsoft Office is a product, not a product line.

Lehman stated that we were tackling this from the wrong end and that a product line needs to be defined from the point of the user. A product line is market niche or domain driven. Obviously there is shared structure, architecture, etc., but that is a consequence of the product line that arises as viewed by users.

Another important factor, according to Balzer, is that the essence of a product line is the production mechanism. Here the notions of versions and collections are important.

Wolf noted that the interdependencies that Staudenmayer pointed out are both between elements in the product and between the products and the processes. Derniame claimed that the artifacts and the level at which they are shared is important to the notion of interoperability (that is one dimension of these interdependencies).

Tully took a much more simplistic approach and stated that a product line is a product line if the company claims it is. Doublait observed that this makes it completely subjective and we will not have any agreement about what a product line is — it is too subjective a definition. [Reporter’s note: this represents the tension between what is the case — that is, that Tully is very likely right about how the term is used by various companies — and what ought to be the case — that is, that we want to have a clear and crisp definition that encapsulates the critical factors.] Schaefer rightly noted that if we take this company specific approach, then we really cannot discuss differences, as there is really no common ground.

Conradi claimed that this was all too foggy. We needed instances and examples.

In considering sharing, Sutton noted that it usually assumed that it will be modules or some such artifacts that will be shared. But it might be something quite different: processes or functionality, for example.

Balzer proposed a minimalist definition: a product line is something that is trying to optimize across some variability. This takes it out of the realm of a single project and is what is different about a product line. Though it should be noted that we invest a lot of tooling for successive products and this investment should not be excluded from our considerations.

In addition to sharing and common user view, one should not overlook the need for asset investment and

capitalization. This is where cost leverage kicks in. Bandinelli noted that this is precisely where we get the important aspect of commonality.

Perry noted that the current state is a bit like Alice in Wonderland: product line means anything anyone wants it to mean. What is important though, is the technology and processes to support our doing things more productively, whether it is producing successive versions or varying a product across some dimension. Votta agreed that the issue is not the definition but the processes and technology for doing it cheaper, better, faster. Reuse is seen as principal in this.

Asset building and asset usage are different processes according to Balzer. The processes answer two different questions: how do I build an asset and how do I use that asset? We need a range of assets, and hence need a range of processes.

Osterweil pointed out that we should write reuse very large — reuse of requirements, architecture, design, code, etc. We need processes for these different kinds of use and reuse.

Perry agreed with Balzer's emphasis: that reuse is used to narrow a focus. But there are other ways of doing it as well. For example a domain specific language satisfies Balzer's definition, but is completely different from the notion of an asset base and its use and reuse. Schaefer pointed out that a domain specific language is still an asset and needs maintenance. [There was both agreement and disagreement here.]

'What are we talking about?' was the question asked by Wolf. Doublait presented a process (good or bad) for reusing assets. It is hard to create design processes, or to create better processes. We use formalisms, languages, and technology that is useful for what we want to do.

Balzer noted that the key thing had not been said: the main motivation for product line is the filling of a market niche. If a company creates assets for such products, then we ought to be able to help businesses create processes to make use of them. A company gains leverage in some market niche. To do this they have to narrow their domain to get the needed leverage. It is up to them to have made a good choice. [Reporters note: It is up to us to have provided useful processes and supporting technology.]

In summary, some important factors, then, in product lines are:

- a narrowing of focus
- core competency leverage
- economics of scale
- shared structure
- interdependencies

- interoperability
- user view
- production mechanism

At the other end, some inhibiting factors are:

- the waterfall model
- top-down programming calculus
- lack of domain understanding