

Embracing Requirements Variety for e-Governments based on Multiple Product-Lines Frameworks

Mikio Aoyama
 Dep. of Inf. and Telecomm. Eng.
 Nanzan University
 27 Seirei, Seto, 489-0863, Japan
 mikio.aoyama@nifty.com

Kenichiro Watanabe, Yu Nishio, Yasuyuki Moriwaki
 Fujitsu Communication Systems Limited
 3-9-18 Shin-Yokohama, Kohoku-ku,
 Yokohama 222-0033, Japan
 {k.watanabe, y.nishio, y.moriwaki}@fcs.fujitsu.com

Abstract

This article reports our experience of product-line development for a wide variety of e-Government applications based on multi-layered frameworks. The key contributions of this report include the modeling a variety of requirements for e-Government applications, software architecture for multiple product-lines framework anticipating the variety, scenario-and-event-based patterns for composing the frameworks and components, and the statistical evaluation of the experience.

Keywords

Requirements Engineering, Product-Line, Software Architecture, and Component-Based Development.

1. Problems and our Approach

We have been working for the development of e-Government applications of the Ministry of Land, Infrastructure and Transport Government of Japan [2] for the decades. The major problems specific to the application domains include:

- 1) Wide variety of the detail requirements based on the similar architectures and functionalities,
- 2) Long-term evolution of applications, and
- 3) A huge imbalance of workload due to the rush of deliveries of more than 50% of products at the end of

every physical year.

To cope with the problems, we analyzed requirements across multiple product-lines, and re-designed the layered application frameworks based on the multiple product-lines [1].

2. Solutions

Fig. 1 illustrates the model of the frameworks for multiple product-lines. Each product-line, such as river information management and dam monitoring, is generated by composing frameworks at the deferent abstraction levels. An application is generated by adding and modifying components on the product-line framework.

On the other hand, the platform layer encapsulates OSes and hardware, so that the frameworks run on both UNIX/Linux and Windows 2000.

To compose the frameworks and component through a unified interface, we identified a set of scenario-based collaboration patterns through events. The patterns are stored in the database (DB). So, the application development from the framework is highly effective since the customization cost is less than 20% in terms of the LOC delivered.

3. Evaluations

We have delivered more than 100 systems for last a few years. And, we have saved development cost by 40% on average along with whole development process, and by 60% from functional design to integration test.

References

- [1] M. Aoyama, Continuous and Discontinuous Software Evolution: Aspects of Software Evolution across Multiple Product Lines, *Proc. IWPSE 2001*, Sep. 2001, pp. 87-90.
- [2] Ministry of Land, Infrastructure and Transport Government of Japan, <http://www.mlit.go.jp/english/index.html>.

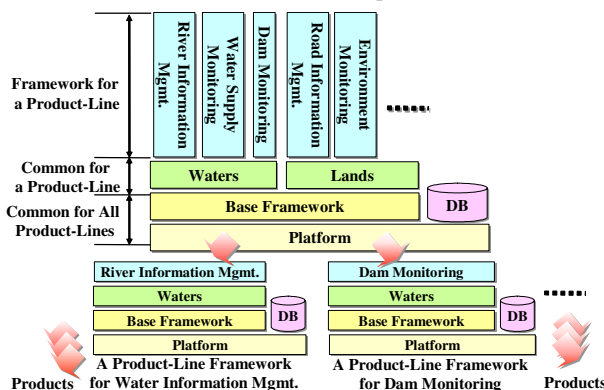


Fig. 1 Framework Architecture for Multiple Product-Lines