

2013. 6. 24. [제57호]

소프트웨어 제품 라인에서 형상관리의 현실적인 이슈와 발전 방향

임준학(Jun Hak Lim), 김은영(Eun Young Kim), 강동인(Dong In Kang)
2013 한국 소프트웨어공학 학술대회(KCSE 2013)

소프트웨어공학센터 경영지원TF팀

C o n t e n t s

I. 서론

II. 소프트웨어 제품 라인의 형상 관리 모델

III. 현실적 이슈

IV. 문제 해결을 위한 발전 방향

V. 결론

I. 서론

급 변하는 시장에서 시장의 요구를 적절한 시기에 반영하고 이를 빠르게 수용하여 다양한 제품을 생산하는 것이 기업의 중요한 이슈가 되고 있다. 이러한 문제를 해결하기 위해 가변성 관리를 통한 소프트웨어 제품 라인(SPLE - Software Product Line Engineering) 및 다른 기능을 가진 다양한 제품을 안정적으로 관리하기 위한 형상관리에 대한 관심이 높아지고 있다.

소프트웨어 제품 라인 공학과 연계하여 다양한 형상관리에 대한 꾸준한 연구가 진행되어 왔으나 이러한 결과를 적용하기에는 현실에서 존재하는 여러 가지 제약 및 예외적인 상황이 있다. 본 원고는 기존의 소프트웨어 제품라인의 형상관리 기법 및 이러한 기법의 적용이 현실적으로 어려운 이유와 앞으로 나아갈 방향에 대해서 제안한다.

II. 소프트웨어 제품 라인의 형상 관리 모델

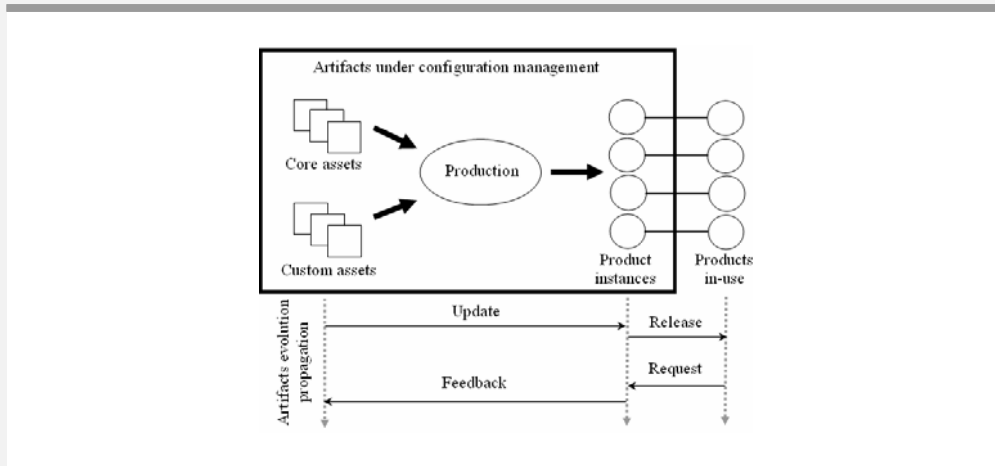
소 프트웨어 제품라인에서 형상관리에 관한 다양한 연구 결과 중 4가지 주요 연구 모델을 살펴봄으로서 소프트웨어 제품 라인에서 형상관리를 위해 필요한 주요 아이디어를 확인한다.

1. 소프트웨어 제품 라인의 일반적인 형상관리 모델

소프트웨어 제품 라인에서 전통적인 방식의 형상관리 모델은 그림 1과 같다. 이 모델은 형상관리 하에 모든 산출물이 관리가 되기 때문에 소프트웨어 제품 라인의 변화, 유지보수, 발전이 엄격하게 제어 되고, 새로운 제품이나 다양한 버전의 빠른 업데이트와 재구성이 가능하다는 특징이 있다.

그러나 Product instances와 Product in-use간의 복제 관계로 인해 제품 수의 증대에 따른 복잡도가 증가하고 Product in-use 상에서의 변경이 Core Asset으로 전파되지 않고, 다른 Product in-use 상으로도 전파되지 않는 한계점이 있다.

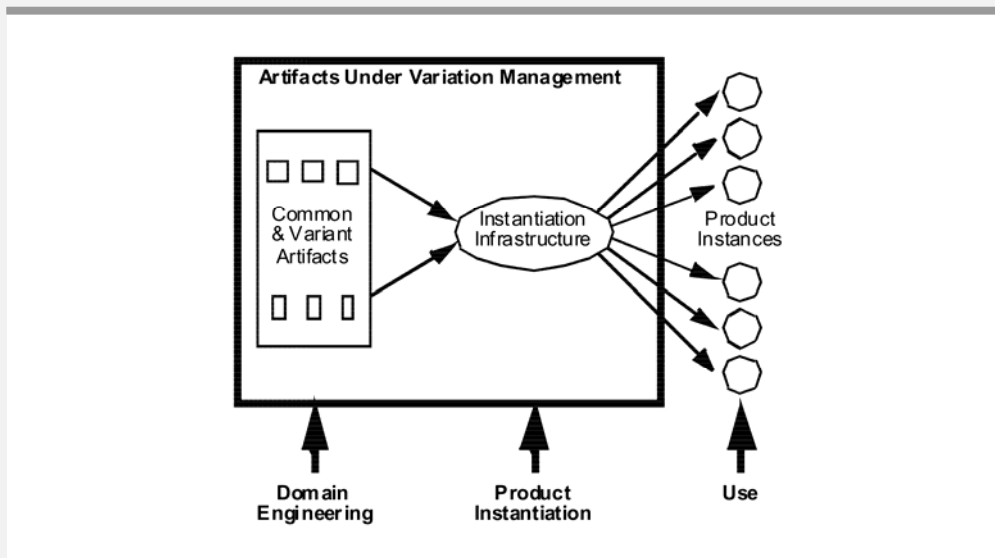
그림 1_소프트웨어 제품 라인의 일반적인 형상관리 모델



2. 크루거(Krueger) 모델

크루거(Charles W. Krueger)는 기존 방식에 중복을 제거하여 제품을 형상관리 하에 두지 않는 형태를 제안하였다. 이 방식은 중복 제거를 통한 관리의 용이함과 개발 중에 발생하는 Core Asset과 Custom Asset의 변화에 대한 빠른 전파에 초점을 맞추어 전통적인 방식의 단점을 개선하였다.

그림 2_크루거 모델

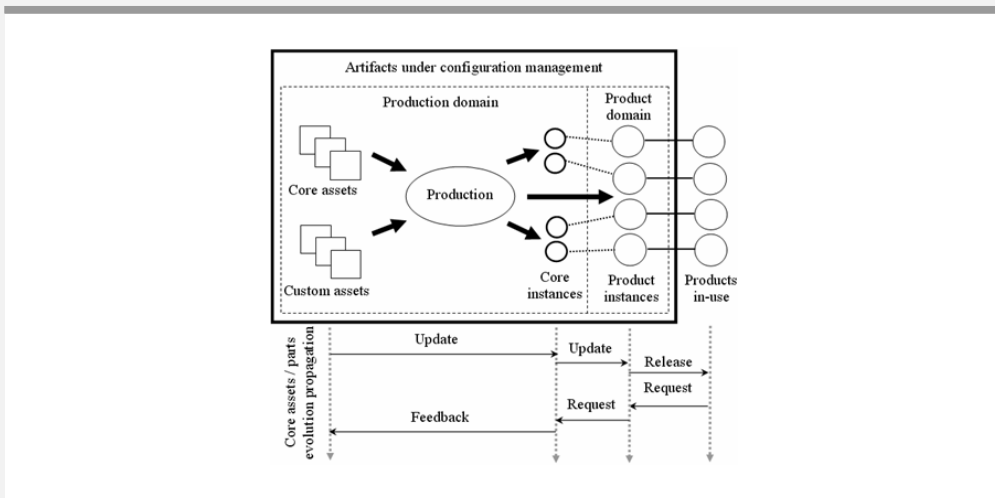


크루거(Charles W. Krueger)의 모델은 개발이후 유지 보수 측면에서 Core Asset과 Custom Asset의 변화 관리를 업데이트 하고 피드백을 반영하는 등의 활동과, 제품의 재구성 등이 어렵다는 단점이 있다.

3. Evolution-based 형상 관리 모델

Evolution-based 모델은 Core assets의 용이한 양방향 전파를 통해 부식을 방지할 수 있으며 그와 성격이 다른 Custom assets의 연결 관계를 별도로 구분하여 책임 분리를 통하여 다르게 관리를 할 수 있다는 장점이 있다.

그림 3_Evolution-based 모델



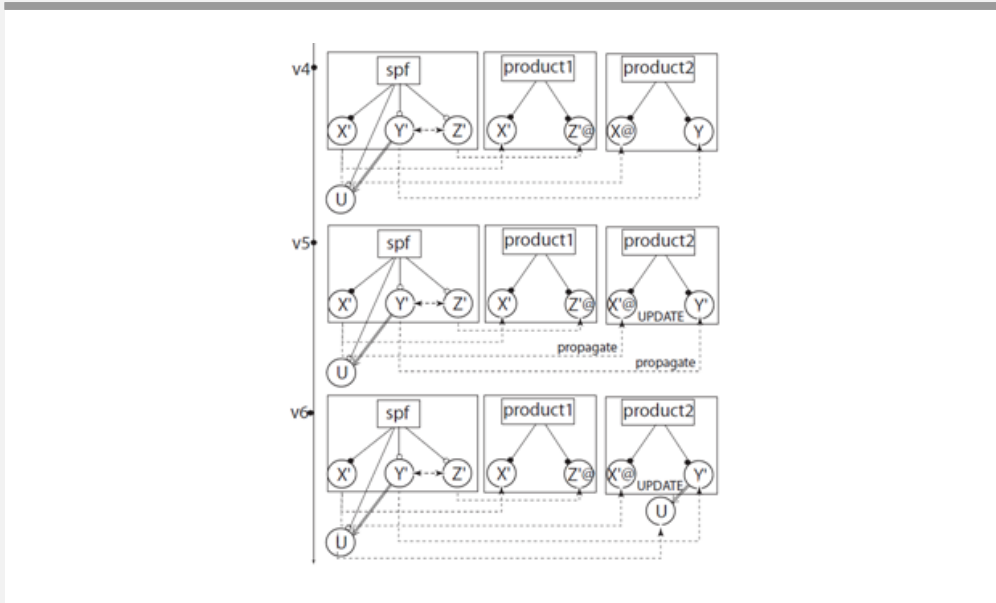
Core instances와 Product instances, Products in-use가 모두 별도로 관리가 되기 때문에 전통적인 모델에서 나타나고 있는 제품 수의 증대에 따른 복잡도 증가와 그로 인한 관리의 어려움 문제는 여전히 남아 있다. 또한 Core part와 Custom part가 완전히 구분되어 관리가 되기 때문에 제품에서 Core part와 Custom part가 서로 복합되어 존재하는 경우에는 이를 구분하여 관리하는 것이 어렵다는 문제가 있다.

4. MoSPL

MoSPL이라 이름 붙여진 환경에서는 컴포넌트들 사이의 관계를 6가지의 관계(Mandatory, Optional, Or, Alternative, Implies, Excludes)들로 미리 정의하여, 자동으로 변화 전파가 되도록 하여, 형상관리의 효율성을 증대 하였고, 이러한 자동화 작업이 실제 동작 가능한 소프트웨어 도구도 소개하고 있다.

이 모델의 특징은 엄선된 변화 전파를 자동화를 가능하게 하는 큰 장점은 있으나, 큰 규모의 소프트웨어에서는 6개 관계를 미리 정의하기가 쉽지 않는 한계가 있다는 점이라고 할 수 있다.

그림 4_MoSPL



III. 현실적 이슈

소 소프트웨어 제품 라인에서 형상 관리 모델들은 원칙적인 아이디어를 제공하고 있으나 실무에서 이를 적용하기에는 현실적인 어려움이 있다. 본 원고에서는 이를 환경적 변화 전파 이슈, 초고도 결합도에 대한 제어 이슈, 관리 시점적 이슈 3가지 측면에서 살펴본다.

1. 환경적 변화 전파 이슈

변화 전파는 소프트웨어 제품라인의 형상관리에서 중요한 부분이라고 할 수 있다. 변화 전파에 대한 주요 원칙은 Core Asset이나 Custom Asset에서 발생하는 문제 및 Product In-use 에서 발생한 문제에 대한 업데이트는 양방향 전파가 이루어 져야 하고 기능의 향상 및 추가는 선택적으로 이루어지되 decay방지를 위해 관리가 필요하다.

이를 위해 대부분의 실무에서는 개발 완료 이후에 기능의 추가가 아닌 중요 문제점에 대해서만 변화 전파를 통해 관리 하에 두려고 노력하지만 관리 외적인 문제와 내적인 문제로 인하여 현실적인 어려움이 따른다.

- 관리 외적인 문제 - 소프트웨어 제품 라인을 관리하는 주체 이외로부터의 정책이나 규제의 변경에 따라 발생하는 문제라고 볼 수 있다. 예를 들어 구글의 안드로이드 플랫폼과 같은 오픈 플랫폼의 경우 구글의 정책 변경으로 인하여 특정 버전부터는 구글의 애플리케이션이 적용되어야 하거나 특정 패치가 적용되어야 하는 등 이슈에 대한 전파뿐 아니라 특정 기능을 추가해야 하는 상황이 발생하게 된다. 또한 디지털 TV의 경우 캘리포니아 에너지 위원회(CEC, California Energy Commission)의 대기 전력 규제와 같은 경우도 동일한 형태의 문제라고 할 수 있다. 캘리포니아 에너지 위원회는 2011년부터 판매되는 TV제품에 대기 전력에 대한 규제를 적용하였다. 그로 인하여 규제를 발의하기 전에 생산된 TV의 경우 해당 규제를 맞추기 위하여 다양한 기능에 수정을 적용해야 하는 상황이 발생하게 된다.
- 관리 내적인 문제 - 소프트웨어 제품 라인 상에서 개발되는 제품들의 제한 요소에 따라 발생할 수 있는 문제로 다양한 하드웨어 사양의 제품이 동일한 소프트웨어 제품 라인에서 개발이 될 경우 Core Asset 혹은 Custom Asset의 변화가 제품으로 전파가 되면서 메모리의 크기 와 같은 하드웨어 제약사항으로 인하여 전파가 불가능한 경우가 발생하게 된다.

2. 초고도 결합도 제어 이슈

소프트웨어에서 모듈 간 결합도를 제거하고 변화 전파 시에 서로간의 의존도를 검사하고 관리하는 것이 소프트웨어 제품 라인에서 안정적인 형상 관리를 위해 중요하다.

- 초고도 결합도 제어 문제 - 소프트웨어의 규모가 커짐에 따라 모든 기능을 자체적으로 개발 한다는 것은 불가능한 상황이 되었다. 현재의 소프트웨어는 내부적으로 이루어지는 개발뿐 아니라 하드웨어 벤더, 오픈 소스, 아웃소싱 등 다양한 곳에서 개발된 모듈이 동시에 적용되고 있기 때문에 아무리 소프트웨어 결합도를 낮추려고 노력해도 이를 완전히 제거할 수 없다. 그로 인해 다양한 모듈, 프로그램, 라이브러리 등이 서로 공존하면서 스케일이 커질수록 한가지의 변화가 주는 영향력이 커지기 때문에 이를 전체적으로 관리하여 서로간의 결합도 관계를 완전하게 파악하는 것은 거의 불가능한 상황이 되고 있다.
- 잠재적인 결합도(Potential Coupling) 문제 - 일반적으로 소프트웨어 공학에서 연구되어온 여섯 가지의 결합도와는 별도로 직접적으로 정의되기 어려운, 프로그램과 그 환경 사이에서 발생하는 잠재적인 결합도도 전체적인 결합도를 관리하고 제어하는데 어려움의 원인이라고 볼 수 있다.

3. 관리 시점적 이슈

소프트웨어 형상관리에서 분리된 브랜치의 용이한 관리를 위해서는 전파에 소요되는 시간을 최소한으로 하여 최단 시간 내 전파를 할 수 있도록 하는 것과 변화의 전파 주기를 최대한 단축하여 즉시 전파가 가능해 지도록 하는 것이 중요하다.

그러나 현실적으로 자동화 툴의 불완전성과 메인 브랜치의 잦은 변경은 이러한 전파 원칙을 지키는데 아래와 같은 많은 어려움이 있다.

- 변화 전파 소요 시간 - 자동화 툴은 변화 전파 소요시간을 최소화 하는데 좋은 방법이다. 그러나 자동화 툴이 Core Asset과 Custom Asset을 자동으로 구분하지 못하면서 모든 수정사항이 적용되어 검증 시간이 증가하게 되는 문제가 발생한다. 이를 막기 위해 수동으로 전파 작업을 진행하게 될 경우 코드 사이즈 및 복잡성으로 인하여 전파 작업의 소요 시간이 상당히 길어지는 문제가 발생하게 된다.
- 변화 전파 시점 선정 - 소프트웨어 제품 라인 상에서 소프트웨어의 복잡도가 증가함에 따라 많은 수정과 기능 개발이 동시에 이루어진다. 그로 인한 메인 브랜치의 잦은 변경 즉시 전파 원칙을 따르는데 많은 어려움이 되고 관리 기준을 설정하는데 많은 어려움이 되고 있다.

IV. 문제 해결을 위한 발전 방향

다 양한 어려움을 해결하기 위해서는 전체 프로젝트의 결합도 관리와 소프트웨어 제품라인 상에서의 특성별, 기능별 타입 관리를 통한 변화 전파 자동화라고 할 수 있다. 관련된 사항은 아래와 같다.

1. 결합도 관리를 위한 저장소(Repository) 연동 관리

소프트웨어 규모가 커짐에 따라 한가지의 변화가 주는 영향력은 점점 커진다고 할 수 있다. 그로 인하여 소프트웨어 제품 라인 상에서 다양한 Core Asset에 대한 변화를 전파하는 데에 각 모듈 간 결합도를 관리 하는 것이 중요한 이슈임에도 불구하고 실무에서는 이를 개발자 개개인의 책임으로 맡기고 이에 대한 확인은 검증 단계를 통해서만 이

루어지고 있는 실정이다. 결합 관계로 인하여 문제가 발생 시에 이를 찾아내는 시점이 늦어지고 연관 모듈의 경우에는 아무런 수정 사항 적용이 없음에도 불구하고 갑자기 문제가 생기면서 이슈 해결에 많은 비용과 시간이 소모 되고 있다.

이러한 문제는 전파되는 변경사항에 대한 내용이 연관 모듈로 미리 전달만 된다면 쉽게 문제가 해결 될 수 있는 부분으로 개발 단계에서부터 파일 및 모듈과 연관된 데이터를 별도의 저장소를 통해 지속적으로 관리를 하고 이를 형상관리 도구와 연동하여 개발자 개개인이 능력이 아닌 프로세스 화를 통해 관리를 하는 과정이 필요 하다.

이 데이터를 형상관리 도구와 연동하여 개발자가 수정사항을 전파 시에 형상관리 틀에서 모듈 간 결합 저장부터 결합 관계 여부를 점검 하여 영향을 받지 않는 것으로 확인된 사항과 영향을 받을 경우 관련 모듈 개발자의 수정을 통해 확인이 된 내용만 적용을 할 수 있도록 하여 결합도 충돌에 관련한 문제를 해결 할 수 있다.

그림 5_결합도 관리를 위한 수정 사항 적용 순서도

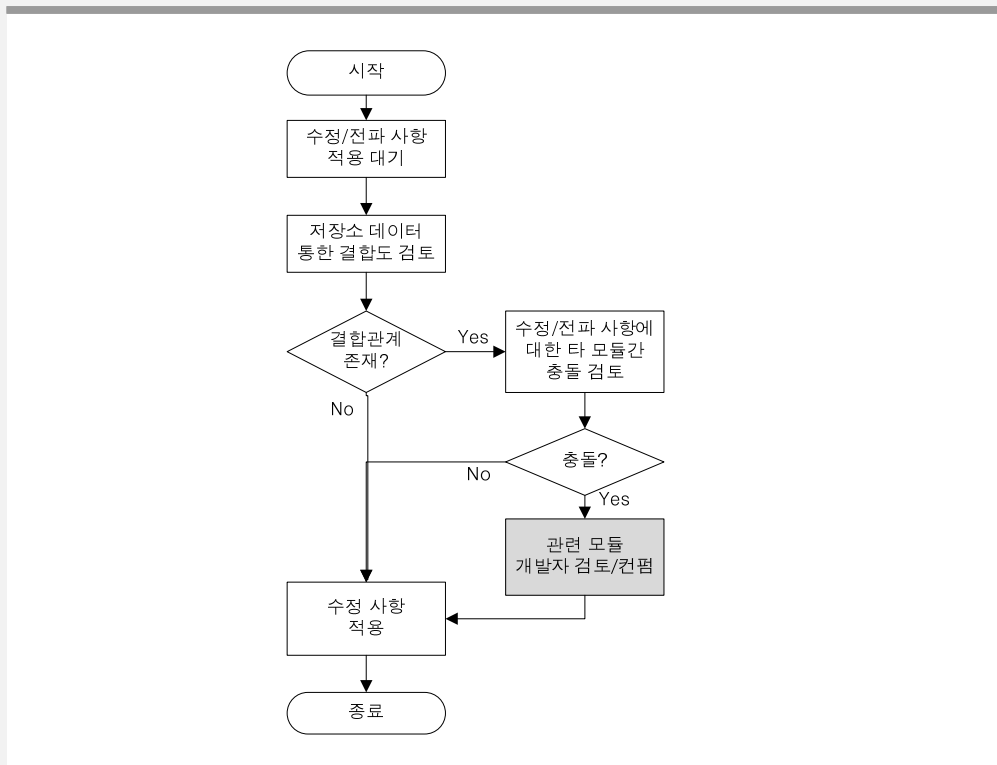


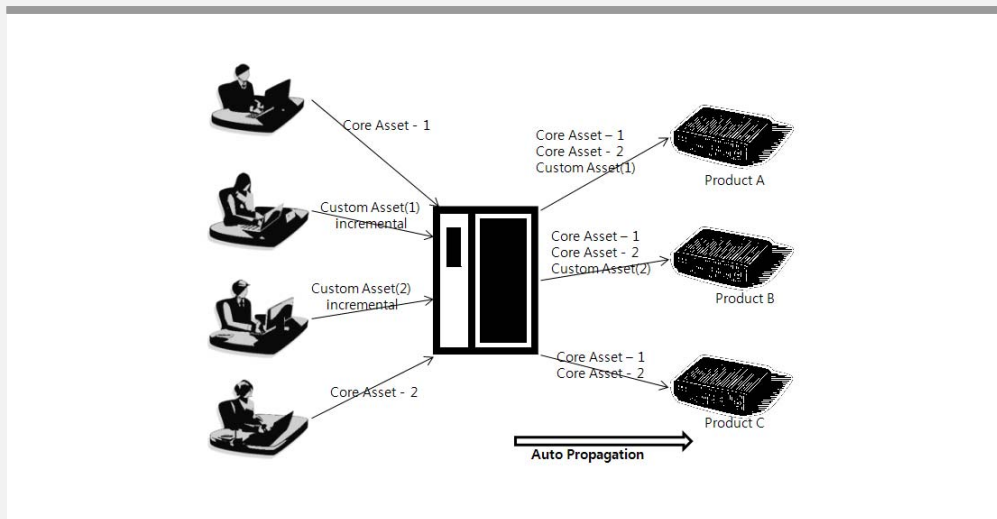
그림 5와 같은 절차로 수정사항에 대한 전파/적용 시 개발 단계부터 유지 관리 되어 온 결합 관계 저장소와의 연동을 형상관리 도구와 결합하여 기본적인 수정/전파 적용단계에 필수적인 점검 과정으로 넣어 이를 프로세스 화를 함에 따라 결합도를 고려하지 못하여 발생하는 문제를 해결할 수 있으며 그림 5의 순서도 상 최종적으로 문제 발생

시 직접 개발자가 검토하는 부분을 제외한 나머지는 자동으로 툴 상에서 수행이 되기 때문에 개발자의 추가적인 업무 로드는 최소화 시키면서 이에 대한 문제를 해결할 수 있을 것으로 판단된다. 또한 이러한 결합 관계 관리를 통해 수집된 데이터는 전체적인 구조 및 모듈의 독립성 여부 파악에도 좋은 자료로 활용될 수 있다.

2. 기능별 옵션 관리를 통한 변화 전파 자동화

형상관리는 소프트웨어 공학 분야에서 자동화가 가장 용이한 분야 중에 하나라고 할 수 있다. 그러나 현재의 자동화 툴은 소프트웨어 제품라인에서 Core Asset과 Custom Asset을 구분 할 수 없기 때문에 현재의 상용화된 자동화 툴을 사용한 변화 전파에는 한계가 있다. 이를 위하여 도메인에 수정 사항을 적용 할 때 형상관리 툴에서 수정 사항의 타입을 선택할 수 있는 옵션을 제공해 주고 이 옵션을 선택하도록 한다면 자동화 툴을 통한 변화 전파 시 적용 범위 선택을 통하여 변화 전파를 할 수 있다. 제공 옵션은 크게 Core Asset과 Custom Asset으로 구분을 하며 그 안에서도 세부적인 타입에 따라 다양한 옵션을 제공해 줄 경우 소프트웨어 제품라인 상에서 제품의 종류 별, 제품의 개발 단계별, 양산 후 등 다양성이 있는 많은 제품들과 다양한 수정사항들 사이에서도 변화 전파 시간을 최소화 하여 필요한 사항을 적용할 수 있다는 장점이 있다고 할 수 있다.

그림 6_옵션 관리를 통한 변화 전파 자동화



간단한 예로 그림 6에의 상황을 보면 4명의 개발자가 변경 사항을 적용할 경우 각각 Core Asset, Custom Asset(1), Custom Asset(2), Core Asset등으로 변경사항에 대한 타입을 지정하여 적용하게 될 경우 개발 중인 제품 A와 제품 B, 이미 양산된 제품 C에 대해서

수정 사항들이 개발자가 지정한 타입을 이용하여 자동 변화 전파 도구가 기존 연구 방향에서 제시했던 변화 전파 원칙에 맞게 에 맞게 자동 적용되도록 할 수 있다.

V. 결론

소프트웨어 제품 라인에서 형상관리의 중요성과 전통적 모델, 크루거 모델, Evolution-based 모델, MoSPL 등 과거 진행되어온 주요한 연구 모델 네 가지에 대해서 살펴보았다. 이러한 모델들이 소프트웨어 개발의 생산성 향상에 기여했다는 것은 자명한 사실이다.

그러나 지금까지 살펴 본 바와 같이 소프트웨어의 설계가 잘 이루어져 있다고 할지라도 환경적 변화 전파 이슈, 초고도 결합도 제어 이슈, 관리 시점적 이슈 등의 측면에서 볼 때 이러한 연구 결과를 실무에 적용하기에는 아직 현실적으로 많은 어려움이 따른다. 이러한 어려움을 해결하기 위해 기존에 설명한 바와 같이 결합도 관리를 위하여 저장소를 연동하여 관리 하거나 관리 시점적 이슈를 위해 자동화 툴의 개선을 통하여 이루어 질 수 있다.

* 본 원고는 해당 논문의 요약본임에 따라 자세한 내용은 논문원문을 참조하시기 바랍니다.

참고 자료

1. Clements, P. and Northrop, L. "Software Product Lines: Practices and Patterns", Addison-Wesley, 2001.
2. Pohl, K., Bockle, G., and Linden, F. J. "Software Product Line Engineering: Foundations, Principles and Techniques", SpringerVerlag New York, Inc, 2005.
3. L.Yu and S. Ramaswamy, "A configuration management model for software product line", Accepted to be published in INFOCOMP Journal of Computer Science, 2006.
4. Krueger, W. "Variation management for software production lines", Proceedings of the Second International Conference on Software Product Liens, San Diego, CA, pp. 37-48, 2002.
5. C. Thao, E. V. Munson, T. N. Nguyen. "Software Configuration Management for Product Derivation in Software Product Families", In ECBS'08, 2008.
6. S. Rotenstreich. "Toward measuring potential coupling", Software Engineering Journal, 4, March 1944.