

◆특집◆ 지능형 설계 기술

지능형 설계를 위한 설계 저장소 기술

강무진*, 김정기**, 안진철**, 엄광호**

Design Repository for Intelligent Design

Mujin Kang*, Jeong-Ki Kim**, Jin-Cheol Ahn** and Kwang-Ho Eum**

Key Words : Design Knowledge (설계 지식), Design Repository (설계 저장소), Function Representation (기능 표현), Knowledge Representation (지식 표현)

1. 서론

소비자 욕구의 빠른 변화와 시장의 극심한 경쟁은 기업으로 하여금 제품 개발 기간을 단축시키고 개발 비용을 축소하도록 압박하고 있다. 한편, 오늘날의 제품이나 시스템은 매우 복잡하여 더 이상 한 사람의 설계자나 단일 설계팀이 독자적으로 개발의 전 과정을 수행하거나 관리할 수 없게 되었다. 설계 활동은 물리적, 지역적, 시간적으로 떨어져서 그리고 각기 다른 분야에 핵심역량을 지닌 서로 다른 팀들이 협력하여 수행하는 프로젝트의 성격이 강해 졌다. 분산된 환경에서 작업하는 설계자들 간의 효과적인 협업을 위해서는 단순한 제품 형상 정보 뿐만 아니라 설계 지식과 설계 프로세스를 공유하는 것이 대단히 중요하다. 여기에는 설계 사양과 설계 규칙(Design Rules), 그리고 제한 조건(Constraints)이나 설계 의도(Rationale) 등이 포함된다. 즉, 설계는 점점 더 지식 집약적이고 협업적으로 수행되므로 지식의 효과적인 기록, 공유, 검색 및 재사용을 위한 지식 관리 도구 개발이 필

수적이다.

인터넷의 발달로 정보의 공유와 교환을 제한하던 지리적, 시간적 장벽은 제거되었지만, 효과적이고 효율적인 정보 검색을 위해서는 부적절한 다른 많은 정보를 걸러내고 필요한 정보만을 신속하게 찾도록 지원하는 시스템이 필요하다. 필요한 설계 지식 정보의 빠른 검색과 재사용이라는 산업계의 욕구를 충족시키기에는 CAD 모델이나 문서와 스키마를 제공하는 전통적인 설계 데이터베이스는 적절하지 못하다.

설계 저장소는 기업의 설계 지식을 정형화하여 표현하고 관리함으로써 공유하고 재사용하는 것을 가능케 하는 지능형 지식 기반 인공물(Artifact) 모델링 시스템이라 할 수 있다.¹ 설계 저장소는 설계의 명시적 지식뿐만 아니라 암묵적 지식을 명확하고 구조화된 형식으로 기록하여 지식의 공유, 검색, 재사용을 용이하도록 지원한다. 또한, 설계 저장소는 제품의 설계뿐 아니라 기업의 자원 관리, 지식 공학, 소프트웨어 개발과 유지 보수 등 다양한 분야에서 정보를 기록하고 관리하는 핵심이다. 아직 설계 저장소가 산업계에서 일반화될 만큼 발전하진 않았지만 많은 기업들이 전통적 설계 데이터베이스에서 설계 저장소로 바꾸기 시작하고 있다

본 논문의 목적은 설계 저장소의 개념과 기술 현황을 소개하고 선도적인 개발 사례를 조사하여 제시함으로써 설계 지능화를 위한 설계 저장소 기

* 성균관대학교 기계공학부
Tel. 031-290-7441, Fax. 031-290-5849
Email mjkang@skku.edu
생산 시스템의 설계 및 관리, 설계 지능화, STEP 표준, CIM, 생산정보시스템 분야에 관심을 두고 연구활동을 하고 있다.
** 성균관대학교 기계공학부 대학원

솔이 실용화될 수 있도록 기반을 제공하는 데에 있다.

2. 설계 저장소의 개념

전통적인 설계시스템은 형상 정보와 형상과 관련된 정보들 - 특징형상이나 제한조건(constraint), 파라미터 정보 등 - 만을 포함하고 있다. 산업체에서는 그 이외의 설계 지식을 필요로 하고 있어 지식 기반의 지능형 설계시스템들이 등장하고 있지만, 주로 데이터베이스 관련 이슈에 초점을 맞추고 있고 인공물(artifact)의 보다 포괄적인 표현 방식에 관한 정보 모델은 소화해 내지 못하고 있다. 이 시스템들이 설계나 제조 프로세스, BOM 등의 비형상 정보를 포함하고 있다고는 해도 인공물의 표현은 여전히 형상 정보에 국한되어 있는 실정이다. 설계 지능화를 위한 정보 모델 프레임워크는 기하학적 형상 정보 뿐만 아니라 기능(Function)과 거동(Behavior), 물리적인 전개(Physical Decomposition), 물리적 영역과 기능 영역 간의 매핑 등을 포괄하고 궁극적으로는 설계 의도와 같은 정보도 포함해야 한다.

설계 저장소(Design Repository)는 기업의 설계 정보를 표현하고 수집하고 공유하고 재사용할 수 있게 하는 기반이 되는 지능적인 지식 기반의 인공물 모델링 시스템이다. 이것은 그 근거가 되는 표현 방식과 인터페이스 및 메커니즘을 포함하는 모델링 시스템이거나, 어떤 특정한 (인공물의) 설계 모델과 그 내용물을 지칭할 수 있다. 기존의 설계 데이터베이스는 도면이나 CAD 모델 및 버전 정보 등과 같이 인공물을 제한적으로 표현하는 데이터 중심적이지만, 설계 저장소는 지식 중심으로 기능과 거동, 설계 규칙(Design Rule), 시뮬레이션 모델 등의 특징을 포함하는 보다 완벽한 설계 표현을 추구한다. 설계 데이터베이스는 대개 이미지나 CAD 모델, 비구조화된 문장이나 문서 등의 제한된 정보를 포함하는 반면, 설계 저장소는 정형화된 데이터나 정보 모델, 구조화된 텍스트 - 즉, 기능과 설계 규칙, 논리적 표현 등을 표현할 수 있는 표현법 - 과 수학적인 시뮬레이션 모델 등을 포함한다. 설계 저장소는 기존 데이터베이스 방식으로는 불가능한 보다 세련된 방법을 이용하여 정보의 저장 뿐만 아니라 설계 지식의 저장과 재사용을 가능하게 한다. 즉, 제품의 기능

적 및 물리적 전개와 그들간의 관계를 명시적으로 표현할 수 있고, 적절한 알고리즘을 사용하여 요구되는 기능을 만족하는 부품이나 조립품을 찾는 것과 같은 보다 세련된 검색을 가능하게 하고, 거동이나 성능의 시뮬레이션과 설계 추론 등을 지원할 수 있다.

그런 의미에서 설계 저장소는 전통적 데이터베이스로부터 진화한 것이다. 이것은 인공물에 대해 저장하고 작업하는 데에 지식 표현 기술을 적용함으로써 기존 한계를 극복하는 것을 목표로 한다.² 설계 저장소는 설계의 기하학적 정보뿐만 아니라, 기능, 행동, 근본적 이유 등 여러 이질적인 정보를 조사하고 구조적으로 조직하여 저장함으로써 전통적인 데이터베이스를 확장하고, 제조를 위해 기하학적 정보에만 초점을 맞추는 이전과는 달리 설계 수명 주기 동안의 전주기적 지원을 목표로 한다

Table 1 Comparison of database and repository

	Database	Repository
Object	Data centric	Knowledge centric
Design Representation	Limited - drawing - CAD model - version info. - related documents	More complete (characterization of) - function - behavior - design rules - simulation model
Types of Information	Few kinds - CAD model - unstructured documents	Extended & heterogeneous info. - CAD data - documents - simulation - animation - analysis
Search	By keywords	Sophisticated - reasoning - browsing
Classification	Manual	Hierarchical placing
Categorization	Non-existent	Automatic inclusion of hierarchy
Characteristics	Static	Evolving

3. 설계 저장소 개발 사례

3.1 NIST Design Repository Project

설계 지식의 표현, 포착, 공유, 재사용을 지원하기 위해 설계된 자료와 이질적인 지식의 저장소 개발에 대한 기술적인 기반을 제공하기 위한 NIST(National Institute of Standards and Technology)의 프로젝트로서,^{1, 3, 4, 5}

- 제품 개발에서 설계 정보 흐름 모델의 일반화
- 핵심 제품 모델의 개발
- 정보 검색 쉽고 효율적인 설계 저장소를 제작, 편집 검색 하기 위한 인터페이스 개발
- 웹 기반 소프트웨어 프레임워크 구현
- 표준화된 용어(Terminology)의 분류 체계 (Taxonomy) 개발

등의 연구 내용을 포함한다.

이 프로젝트에서 개발되는 인프라(Infrastructure)는 인공물(Artifact) 설계 지식에 대한 정형적 표현과 저장소 개발을 위한 웹 기반 인터페이스, 제품 기능과 같이 공학 설계와 관련이 있는 개념을 사용한 저장소 검색 기술로 구성되어 있다. 구체적인 결과물로 인공물의 정보 모델링에 형상, 기능, 거동의 세 가지 관점을 표현할 수 있는 객체 지향 인공물 표현 언어를 개발하였다. 이 모델링 언어는 객체(Object)와 관계(Relationship)의 집합으로 인공물을 나타낸다. 객체는 기능, 거동과 같은 비물리적 개념뿐만 아니라 조립품, 반조립품, 부품 등과 같은 물리적 엔터티(Entity)를 나타내고, 관계는 물리적 전개, 기능적 전개를 포함하는 객체 집합 사이의 관계와 객체 사이의 다양한 관계를 나타내는 데에 사용된다. 전체적인 인공물 표현은 물리적 엔터티를 나타내는 객체들의 집합과 객체의 다양한 특성과 값뿐만 아니라 다른 객체, 관계, 객체들 사이의 상호 연결 등으로 이루어진다.

Fig.1 은 데이터 구조를 나타내는 사각형과 하나의 데이터 구조에서 다른 데이터 구조로의 참조를 나타내는 화살표를 이용하여 모터에 대한 다양한 정보 유형을 나타내는 표현들간의 관계를 도해하고, 모터에 대한 기능/흐름/인공물 사이의 매핑을 보여준다

NIST 설계 저장소 프로젝트에서는 인간과 기계간에 해석될 수 있는 지식 표현을 제공하여, 설계자와 지식 기반 설계 시스템이 설계 저장소에 저장된 정보에 접근하고, 사용할 수 있게 하기 위

해 일반적인 스키마와 자료 구조를 사람과 기계가 해석할 수 있는 XML 포맷으로 매핑한다.

설계 인공물 모델링에 대한 대표적인 인프라 개발 외에도, NIST 설계 저장소 프로젝트에서는 설계 저장소를 만들기 위해 컴퓨터를 이용하는 프레임워크와 설계 저장소의 이점을 보여줄 수 있는 POC(Proof Of Concept) 프로토타입을 개발한다. 이 연구 결과는 설계 저장소의 분산 개발과 접근에 대한 툴 스위트 (Tool Suite) 구현으로 나타내고 있다. 구현된 시스템은 일반적인 웹 브라우저를 이용하는 다수의 분산 클라이언트들이 인터넷을 통해 설계 저장소에 접근하는 웹 기반 인터페이스와 아래와 같은 여러 요소들을 포함한다.

- ObjectStore : 객체 지향 DBMS
- 웹 기반 설계 저장소 편집기
- 웹 기반 설계 저장소 브라우저
- 설계 저장소 컴파일러
- 정보 추출기
- STEP/Works : STEP AP 203 뷰어

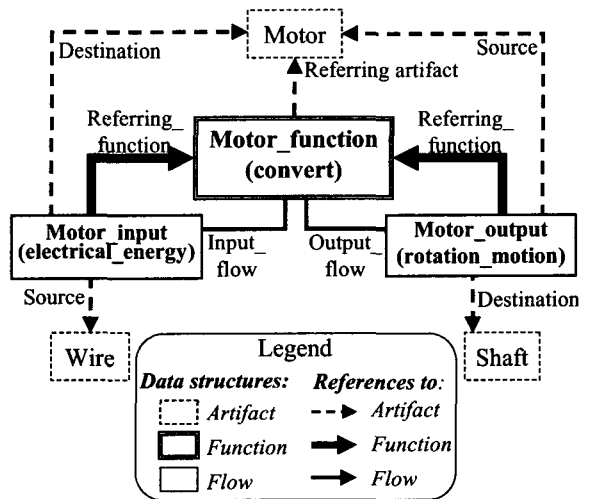


Fig. 1 Graphical illustration of motor function representation

웹 기반 설계 저장소 편집기는 일반적인 웹 브라우저를 통해 사용할 수 있고, 설계 저장소 개발 과정을 간단하게 한다. 이 인터페이스는 설계자를 설계 저장소에 원거리에서 접속하게 한다. 또한 기존 클래스들로부터 객체와 관계를 만들어

사용자가 설계 인공물 모델을 만들거나 편집할 수 있게 하고, 인공물의 새로운 클래스를 추가하게 하고, 물리적, 기능적, 거동적 분해에 대해 기술하게 한다. Figure 2는 인공물 모델을 포함하고 있는 설계 저장소에 대한 세션을 검색하는 것을 보여준다. 웹 기반 설계 저장소 브라우저는 설계자가 기존의 설계 저장소를 검색할 수 있는 포인트&클릭 인터페이스를 제공한다. 객체 지향 데이터 구조(객체, 관계, 클래스)는 표 형태로 나타내고, 표에서 하이퍼링크가 연결된 데이터 구조를 검색하게 한다. 객체에서 객체로의 검색 외에도 인터페이스의 하단 우측부는 가공물의 전 계층의 물리적 분해를 조립품, 하위 조립품, 각각의 요소들까지 제공한다. 사용자는 하이퍼링크 중 하나를 따라 재빨리 다른 인공물 객체로 건너 뛸 수 있다. 설계의 다른 부분을 방문할 때, 인터페이스의 상단 우측부에 기록이 추가되어 사용자는 세션을 검색하는 동안 이전에 보았던 가공 모델의 부분으로 빨리 돌아갈 수 있다

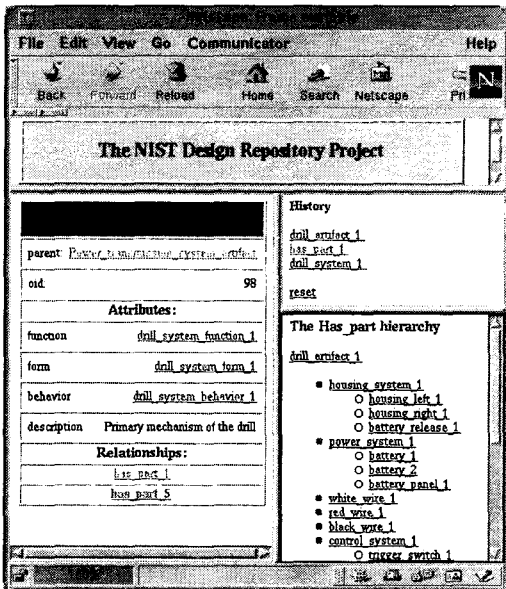


Fig. 2 Web-based artifact browser interface

3.2 SPOOL Design Repository

Bell Canada의 소프트웨어 품질 평가팀과 몬트리얼 대학 GELO 그룹 간의 산학공동연구로 수행

된 SPOOL Project (Spreading Desirable Properties into the Design of Object-Oriented, Large-Scale Software Systems)는 소프트웨어 설계 구성(Design Composition) 및 소프트웨어 설계 품질의 이해와 평가를 위한 방법론과 도구를 개발하기 위한 것으로, 소프트웨어 역공학(Reverse Engineering)과 시스템 이해 및 분석, 그리고 리엔지니어링을 지원하는 환경을 개발하는 것이다.

소프트웨어 시스템을 이해하고 평가하고 유지 보수하기 위하여는 그 시스템을 상위 레벨(분석 및 설계 레벨) 추상화(Abstraction)하여 표현하는 것이 매우 중요하다. 최종 사용자 틀들은 이 정보를 필요로 하며 따라서 대상 시스템의 저장을 위한 설계 저장소가 필요한 것이다.⁶

SPOOL 설계 저장소는 시스템의 소스코드로부터 추출한 설계 레벨의 정보를 상세한 저장하는 UML 메타모델로, 역공학 과정에서 소프트웨어 엔지니어들간의 협력적 작업을 지원하기 위해 틀(Tool) 간의 의사 소통과 그 기반을 제공한다. 그 내부 메커니즘과 관련 틀은 데이터를 질의하고 구성 요소 사이의 종속성(Dependency)을 관찰하고, 사용자가 역공학의 필수적인 작업인 시스템 이해, 시스템 해석, 리엔지니어링을 수행하는 데에 유용한 기능을 제공한다. SPOOL 설계 저장소는 저장소 스키마(Repository Schema)와 물리적 데이터 저장소(Physical Data Store)로 구성되며, 이 스키마는 객체들의 구조와 거동을 정의하는 객체 지향 클래스 계층 구조이다. 그 객체들은 역공학된 소스 코드 모델, 소스 코드로부터 식별되어진 추상적 설계 요소, 구현된 설계 요소, 그리고 복구되거나 재조직된(re-organized) 설계 모델들이다. SPOOL 환경은 사용자 틀, 역공학 모델의 스키마와 객체, 영속적인 데이터저장소 사이의 관심의 명확한 구분을 위해 3 계층 구조를 사용하였다 (Fig.1). 최하층은 역공학된 소스 코드와 설계정보의 물리적, 영속적 데이터저장을 위한 객체지향 데이터베이스 관리 시스템으로 구성된다. 중간층은 구조, 행동, 메커니즘을 포함하는 역공학된 모델의 객체지향 스키마인 저장소 스키마로 구성된다. 물리적 데이터 저장소와 저장소 스키마인 이 부분을 SPOOL 설계 저장소로 부른다. 최상층에는 저장소 스키마에 기반한 특정영역의 기능을 수행하는 사용자 틀, 즉 “소스 코드 획득(Capture)”과 “가시화 및 분석”들이 있다.

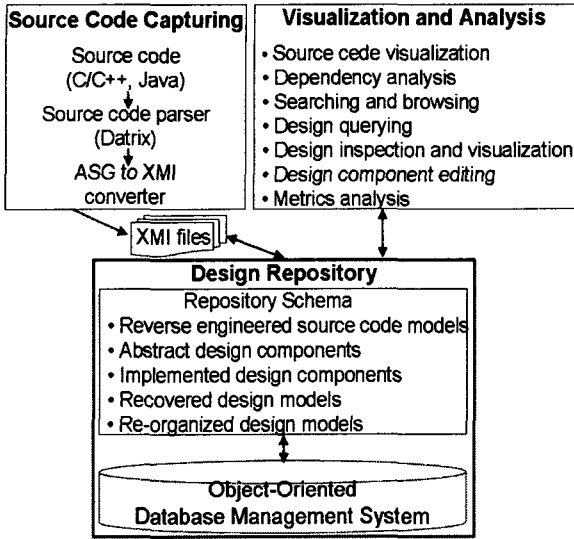


Fig. 3 Overview of the SPOOL environment

역공학의 첫번째 단계는 “소스코드 획득”으로, 이것은 소스 코드로부터 초기 모델 추출을 목적으로 한다. SPOOL 은 C++을 지원하고, C++소스코드 분석을 위해 Datrix 을 사용한다. Datrix 을 통해 ACS II 기반 표현의 형식으로 제공된 소스 코드 정보는 변환 유틸리티에 의해 Datrix/TA 소스 코드 표현 파일의 노드와 아크를 조립하고, XMI 파일들을 생성한다. 그 후 XMI 파일은 들여오기 유틸리티에 의해 읽혀지고, SPOOL 저장소에서 초기모델의 객체들로 구성된다. 또 다른 유틸리티는 XMI 파일의 내부 구조들을 해석하고 저장소의 내용을 보내기 하는데 사용한다.

설계 표현의 목적은 소스 코드 모델과 추상적 설계 요소 및 구현된 설계 요소들을 상호작용적으로 가시화하고 분석하기 위함이다. 사용자 툴 “가시화 및 분석”에서는

- 소스 코드 가시화,
- 상호작용적이고 점진적인 종속성 분석,
- SPOOL 브라우저를 이용한 검색과 브라우징을 통한 설계 검토,
- 주어진 문제를 해결하기 위하여 협력하는 클래스들에 대한 설계 질의,
- 역공학된 소스 코드 모델의 문맥 안에서의 설계 검사와 가시화,
- 설계 요소의 편집,

- 바람직하거나 바람직하지 못한 소스와 설계 특성에 대한 정량적인 분석을 하는 Metrics 분석 등의 툴들이 구현되고 통합되었다.

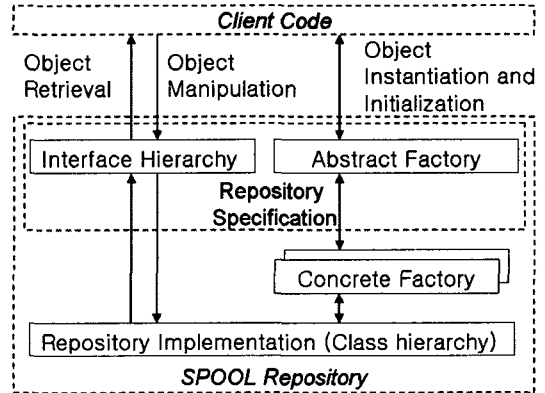


Fig. 4 SPOOL repository architecture

Fig.4 는 SPOOL 설계 저장소의 아키텍처를 보여 준다.⁷ SPOOL 저장소는 새로운 연구계획에 사용, 적용 가능하도록 변화, 적용, 확장에 탄력적인 스키마를 위해 해석과 시각화를 위한 고객의 코드로부터 설계 저장소의 실행을 완전히 차단했다. 설계저장소내의 객체의 검색과 조작은 공적인 Java 인터페이스를 통해 이루어 지고, 인스턴시에이션(Instantiation)과 초기화는 Abstract Factory 를 통해 구현된다. Interface Hierarchy 는 SPOOL 저장소의 검색과 조작 기능의 의미(Semantics)를 명시한다. 클래스 대신에 Client Code 를 인터페이스에 연결하는 것은 클라이언트 코드가 instantiate 하는 객체의 구체적인 유형과 구현 방법에 대해 몰라도 된다는 이점이 있다. 따라서 인터페이스 명세가 지켜지기만 하면 클라이언트 코드에 전혀 영향을 주지 않고 저장소 내용을 변경할 수 있게 된다. Abstract Factory 는 객체 인스턴시에이션과 초기화의 연결 방법을 제공한다. 클라이언트 코드는 구현된 저장소의 클래스를 직접 instantiate 하지 않고도 Abstract Factory 클래스로부터 저장소 클래스의 인스턴시에이션을 요청할 수 있는데, 이 경우 그 서브클래스인 Concrete Factory 가 해당 클래스의 실질적인 인스턴시에이션을 수행한다.

SPOOL 저장소의 스키마는 객체지향 클래스 계층구조로 설계의 복구와 해석에 적절한 프로그

래밍 언어 레벨까지 적용 가능한 확장된 UML 메타모델 구조를 사용하였다. 이것은 Top-Level 클래스, Core 클래스, Relationship 클래스, Behavior 클래스 Extension 클래스를 포함한다. SPOOL 설계 저장소 구현에는 객체 지향 데이터베이스 POET 6.0 이 사용되었다. UML 메타 모델에 기반한 스키마는 시스템 소스 코드의 상세한 정보를 저장할 수 있고, 데이터 교환 형식으로는 XMI 가 사용되었다.

4. 요약

소비자가 주도하는 현대의 시장에서 제품의 복잡성은 점점 더 커지고, 제품 개발 활동은 지리적으로 분산되어 수행되는 경향이 확산되고 있다. 이러한 환경에서 요구되는 설계 개발 툴로서의 지능형 설계 시스템은 종래의 CAD 시스템보다 다양한 기능을 가져야 하고, 도면이나 문서와 같은 전통적인 정보보다 다양하고 고도화된 정보를 취급할 수 있어야 한다. 이와 같이 제품이나 부품의 형상 외에도 설계 의도와 규칙, 지식 등을 포함하여 기능과 거동 및 구조를 표현할 수 있는 플랫폼을 설계 저장소라 한다. 조립 제품을 대상으로 하는 설계 저장소(NIST Design Repository)와 소프트웨어를 대상으로 하는 설계 저장소(SPOOL Design Repository)의 개발 사례를 들어, 설계 저장소가 설계의 지능화에 어떻게 기여할 수 있으며 얼마나 중요한 지를 소개하였다.

후 기

본 연구는 과학기술부 핵심연구개발사업 “미세형상 설계 지능화 기술 개발” 과제 수행의 일환으로 이루어진 것임을 밝히며, 지원 기관에 감사드립니다.

참고문헌

1. Szykman, S., Sriram, R., Bochenek, C., Racz, J. and Senfaute, J., "Design Repositories: Engineering Design's New Knowledge Base," IEEE Intelligent Systems, pp. 48 - 55, May/June 2000.
2. Kopena, J. and Regli, W., "Design Repositories on the Semantic Web with Description-Logic Enabled Services," First International Semantic Web and

- Databases Workshop, Berlin, Germany, September 2003.
3. Szykman, S., "Architecture and Implementation of a Design Repository System," Proceedings of the 2002 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, Paper No. DETC2002/CIE-34463, Montreal, Canada, September 2002.
4. Szykman, S., Racz, J. Bochenek, C. and Sriram, R., "A Web-based System for Design Artifact Modeling," Design Studies, Vol. 21, No. 2, pp. 145 - 165, 2000.
5. Szykman, S., Racz, J. and Sriram, R., "The Representation of Function in Computer-based Design," Proceedings of the 1999 ASME Design Engineering Technical Conferences (International Conference on Design Theory and Methodology), Paper No. DETC99/DTM-8742, ASME, Las Vegas, Nevada, September 1999.
6. Keller, R., Bedard, J. and aint-Denis, G., "Design and Implementation of a UML-Based Design Repository," CAiSE 2001, LNCS 2068, pp. 448 - 464, 2001.
7. Schauer, R., Keller, R., Lague, B., Knapen, G., Robitaille, S. and Saint-Denis, G., "The SPOOL Design Repository: Architecture, Schema, and Mechanisms," In Hakan Erdogmus and Oryal Tanir, editors, Advances in Software Engineering. Topics in Evolution, Comprehension, and Evaluation. Springer-Verlag, 2001.