

건설정보 디지털 데이터에 적용 가능한 동적 객체 데이터의 스키마 불가지론적 교환 방법론

유석준^{1*}, 김상욱²
¹(주)삼안, ²(주)피티씨

Schema-agnostic Dynamic Object Data Exchange Methodology Applicable to Digitalized Construction Engineering Information

Seok-Joon You^{1*}, Sang-Wook Kim²
¹Saman Corp.
²Precast & Pile Tech Corp

요약 인프라 시설물의 건설 프로젝트에 사용되는 다양한 소프트웨어 플랫폼 간에 필요한 데이터를 원활하게 교환하기 위해서는 표준화된 데이터 형식의 구현이 필수적이며, 이러한 표준화는 중요 정보의 손실이나 변질을 방지하는 기능을 한다. 이를 위해 예전부터 중립적 객체모델 기반의 데이터 형식과 교환 방법론이 제안되었다. 그러나, 기존 소프트웨어에 사용되는 데이터 모델은 호환성이 결여된 고정적인 데이터 모델을 기반으로 함으로써 다수의 출처에서 오랜 경험을 거쳐 생성되는 다양한 형식의 데이터를 매개하는 데에는 한계가 있다. 이러한 상황에서 고정된 중립 모델보다 유연한 데이터 교환 메커니즘을 통해 당사자 간 직접 정보를 교환하는 것이 더 현실적일 수 있다. 본 연구에서는 JSON(JavaScript Object Notation) 형식에 기반한 동적 객체를 이용해 고정된 객체모델 정의인 스키마 없이 다양한 데이터를 교환하는 방법을 제시하고자 한다. 본 방법론에서 인프라 시설물의 객체모델 정의는 데이터 구조에 대한 엄격한 규정을 적용하기 보다는 수신 데이터의 의미 파악을 위한 지침 역할을 하며, 주어진 데이터와 객체모델과의 일치 여부를 강제하지 않는다. 이를 바탕으로 2종의 JSON 샘플 데이터를 이용해 동적 객체의 구조와 형식을 유추할 수 있는지 실험한 결과, 두 샘플 모두 원래 클래스의 상위클래스까지는 75% 정도 유추할 수 있어 본 연구의 스키마 불가지론적 방법론의 실현 가능성을 보여주었다.

Abstract The implementation of a standardized data format is imperative to facilitate the seamless exchange of essential data across diverse software platforms in construction projects for infrastructure facilities. This standardization serves as a safeguard against the loss or corruption of critical information. However, the inherent rigidity of this unalterable data format within standard data models exposes vulnerabilities when mediating a wide range of data types from sources beyond CAD or BIM software. Given these complexities, a more pragmatic approach might involve the direct exchange of information between stakeholders through a flexible data interchange mechanism. In this paper, we introduce a technique for the exchange of diverse data by leveraging a dynamic object based on the JavaScript Object Notation (JSON) format. In this methodology, the object model definition for infrastructure facilities assumes the role of a reference guide that aids in the comprehension of incoming data rather than imposing strict regulations on data structures. Tests show that about 75% of the superclass of the original classes in two sample JSON datasets could be reconstructed, which demonstrates the feasibility of our schema-agnostic methodology.

Keywords : Construction Informatics, Engineering Software, Data Exchange, Standard Information Models, Dynamic Objects, Schema-agnosticism

이 연구는 국토교통부/국토교통과학기술진흥원이 시행하고 한국도로공사가 총괄하는 "스마트건설기술개발 국가R&D사업(과제번호 RS-2020-KA156050)"의 지원으로 수행하였습니다.

*Corresponding Author : Seok-Joon You(Saman Corp.)

email: sjyou@samaneng.com

Received September 7, 2023

Revised October 5, 2023

Accepted October 6, 2023

Published October 31, 2023

1. 서론

1.1 연구의 배경 및 목적

디지털 정보기술이 발달함에 따라 토목 인프라 시설의 설계, 시공, 유지관리 업무에서 생성되고 쓰이는 각종 문서 및 정보가 디지털화되고 있다. 이러한 데이터가 인프라 건설엔지니어링 분야에서 본격적으로 활용되려면, 이를 처리하는 다양한 엔지니어링 소프트웨어 사이에서 데이터의 상호운용성 및 호환성이 확보되어야 한다.

도로설계(Autodesk, Bentley 등), 구조 해석(Midas, ARoad 등) 등 사용되는 다양한 엔지니어링 소프트웨어와 하드웨어 시스템 간 데이터 교환은 CAD/CAE 등 다양한 분야의 정보를 디지털화하기 위한 중립적 데이터 형식으로 정의하는 방안이 1980~90년대에 개발되었다. 건설 분야에서는 시설물을 구성하는 각종 요소 및 이들 사이의 관계 정의를 핵심으로 하는 산업 기초 클래스(IFC: Industry Foundation Classes, 이하 IFC)[1]가 중립 데이터 형식의 표준으로 개발되어 쓰이고 있다.

IFC와 같은 중립 모델에 기반한 정보 교환은 IT, 정보 기술의 발달로 인해 건설단계에서 다양한 디지털 데이터가 생성되고, 실시간으로 유통되는 오늘날에는 아래와 같은 이유로 인해 현실에 부합되지 않게 되었다.

- 인프라 시설물의 설계, 시공단계 프로젝트의 단계별 참여 주체마다 수행하는 업무가 다양하며 동일한 대상에 대해서라도 주체별 필요 데이터의 종류가 서로 달라지기 때문에, 이를 하나의 단일 데이터 모델로 묶게 되면 그 다양성과 상세함을 표현할 방법이 없다.

- 설계단계에서 공사 준공까지 약 9년의 시간적 제약이 있고, 이 기간에 설계 변경, 공법 변경 등 엔지니어링 프로젝트에서 다뤄지는 정보의 종류와 내용은 계속 변화한다. 이에 맞춰 중립 모델도 바뀌어야 하는데, 이 과정에서 과거의 중립 모델 기반으로 생성된 데이터는 이후에 변경된 모델 기반 데이터와의 호환성을 상실한다.

이러한 문제를 해결하기 위해, 본 연구는 고정된 데이터 모델이 아닌, 다양한 데이터를 구조적으로 기술할 수 있도록 동적 객체를 이용한 중립 모델을 통해 데이터 교환의 단점을 보완할 수 있는 방법론을 제시한다.

1.2 연구의 범위와 방법

본 연구는 인프라 시설 중 도로 건설 프로젝트의 모든 참여 주체 및 업무에서 생성, 관리되는 정보에 대하여, 참여 주체 간 커뮤니케이션 과정에서 요구되는 다양한

정보의 데이터 교환을 만족시킬 수 있는 메커니즘을 제안하는 것을 목표로 한다.

본 연구는 객체모델 기반의 건설정보 데이터의 교환표준 초점을 맞추고 있지만, 여기에서 제안하는 데이터 교환 메커니즘은 데이터 모델에 기반한 정보 공유 방법론의 적용이 가능한 분야라면 건설뿐만 아니라 타 분야에서도 범용적으로 적용될 수 있도록 하였다.

본 연구는 아래 절차에 따라 수행하였다:

- 중립 모델 기반 정보 교환에 관한 문헌조사
- 동적 객체(반구조적 데이터) 기반 건설정보 정의 방법론 고찰
- 동적 객체에 기반하여 스키마에 구애받지 않는 디지털 건설정보의 교환 방법론 제시
- 스키마 불가지론적 데이터 교환 방법론의 타당성을 검증하기 위한 동적 객체 데이터의 원 스키마 유추 실험 및 결과의 분석

본 논문의 구성은 다음과 같다: 2장에서는 기존 연구 사례 및 관련 문헌을 통해 기존 디지털 건설정보 데이터의 정의 및 교환 방법의 현황과 문제점에 대해 고찰한다. 3장에서는 2장에서 고찰한 문제점의 대안으로 동적 객체 기반의 건설 데이터 및 데이터 사전의 정의 방법을 제시한다. 4장에서는 스키마 정보를 제거한 동적 객체 데이터에서 원 스키마 정보를 유추하는 실험을 통해 3장의 방법론의 타당성을 검증한다. 5장에서는 데이터로 정의된 건설정보를 교환하는 방안으로 스키마 불가지론적 교환 방법을 제시하고 6장의 결론으로 이어진다.

2. 기존 연구 고찰

2.1 중립 모델 기반 데이터 교환에 관한 연구

2.1.1 초기 중립 모델 연구사례

다수의 소프트웨어 또는 서비스 사이에서 데이터 교환을 위해 단일 데이터 포맷을 사용하자는 아이디어는 CAD/CAE의 초창기부터 있었다. 초기에 중립 포맷은 Autodesk DXF 형식[2]처럼 시장 점유율이 높은 특정 소프트웨어 중심의 내부 데이터 구조 또는 데이터 입력 형식을 변형, 가공한 포맷으로 사용되었다. IGES 포맷[3]이 개발된 이후부터, 특정 소프트웨어에 종속되지 않는 중립적 데이터 형식을 통한 3차원 설계 소프트웨어 사이에서 형상 데이터를 주고받을 수 있게 되었다.

IGES의 후속으로 개발된 STEP(Standards for the Exchange of Product model data, 이하 STEP)은 고정된 데이터 블록이 아니라 설계 대상을 추상적 정보 요소인 객체의 조합으로 기술함으로써 데이터를 정의한다 [4]. STEP은 이러한 객체모델을 바탕으로 CAD, CAE뿐만 아니라 설계 데이터를 후속 업무인 비용 산출, 공정 및 자원관리 등에도 활용할 수 있도록 중립 포맷의 역할을 확장하였다.

2.1.2 IFC 기반의 건설정보 교환 사례

IFC는 객체모델을 기반으로 하는 STEP의 방법론을 빌려 구조물을 구성하는 각종 요소의 형상과 속성을 비롯하여, 건설, 유지, 철거하는 작업 등 표현이 가능하도록 건설산업 분야의 포괄적 정보 모델 형태를 제시하였다. IFC1.0 버전은 도로, 교량, 철도 등 선형 기반의 시설물에 적용하기에는 부족함이 많았으나, IFC 4.3[5]부터 선형을 기반으로 한 시설물까지 표현이 가능해졌다.

IFC는 시설물의 형상 데이터를 비롯한 건설 엔지니어링 분야에서 사용하는 거의 모든 소프트웨어의 데이터를 포괄하고자 하였기 때문에 필연적으로 데이터 크기나 규모가 커지고 복잡해졌다. 이에 대응하기 위해 BLIS 프로젝트[6]에서는 IFC 모델이 사용되는 6개 시나리오별 유스케이스(Use-case)를 설정하고 개별 유스케이스별 서브셋(관점 정의, view definition)만 추출해 사용하도록 하고 있다[7]. 이러한 방법론은 이후 IFC의 공식 모델 뷰 정의(Model View Definitions, 이하 MVD)[8]와 사용자 주도의 유스케이스 작성 방법론인 Information Delivery Manual (이하 IDM)[9]로 발전하였다.

IDM 및 MVD는 IFC 모델을 다양한 유스케이스에 적용할 수 있는 체계화된 방법을 제공한다는 의의가 있다. 그러나, 필요 데이터가 IFC에 정의되어 있지 않으면, MVD로도 정의할 수 없다는 한계와 더불어 객체 사이의 연계 구조가 IFC와 맞지 않을 때 이를 변환하는 방법론이 미비한 점이 있다. 또한, 시간 경과에 따른 데이터양의 증가와 급변하는 건설정보에 데이터 교환을 원활하게 하기 위한 해결책으로는 충분하지 못한 한계가 있다.

2.2 반구조화 데이터 기반 건설정보 교환에 관한 기존 사례

건설 프로젝트의 데이터 교환에 사용되는 시설물의 객체모델은 고도로 구조화된 데이터 형식을 기반으로 하지만, 건설 프로젝트 수행 시 필요한 정보 대다수는 기존 수기 문서를 텍스트나 테이블 형식(엑셀)의 반구조화된

형태로 디지털화된다. 건설 프로젝트에서 기존 자료를 효과적으로 활용하려면, 이러한 비구조, 반구조적 정보를 객체모델과 연계되도록 해야 한다.

이를 위해 COBie[10]는 건물의 유지관리 정보를 스프레드시트 테이블 형식(XML로 기술)으로 정의하고, 대상 시설물인 IFC 객체에 ID를 부여하여 객체모델과 연계하는 방법을 사용하고 있다. 이와 달리 W3C Linked Building Data[11]에서는 시맨틱웹 기술을 이용하여 건축물의 공간구조에 대한 온톨로지를 정의하고, 이를 바탕으로 구성 부재의 정보를 링크드 데이터(Linked Data) 형식으로 연계토록 하고 있다.

다른 한편으로는, BIM 모델과 연계하여 다양한 비구조, 반구조적 정보를 교환하려면, 단순히 중립 모델을 확장하는 방식으로는 대응하기 어렵다. 이를 해결하는 방안으로 Sobhkhiz와 El-Diraby의 연구[12]에서는 프로젝트 문서의 자연어 분석을 통해 도출한 시맨틱 네트워크를 사용자의 커뮤니케이션 네트워크 분석과 결합하여 프로젝트별 상황에 맞는 고유 모델을 도출하는 방식을 제시하였다. 다만 특정 사례의 분석으로 도출한 상황별 데이터 모델을 프로젝트 참여자 모두가 이해하고 공유할 수 있게 하는 효과적인 방안을 찾는 것이 향후 과제로 남아 있다.

3. 건설 객체의 동적 정의 및 데이터 교환

3.1 동적 객체의 특성

동적 객체란 고정된 데이터 모델인 스키마 정의에서 파생되지는 않았지만, 항목명-내용 형식으로 계층적으로 구조화된 데이터를 통해 실제 상황을 추상적으로 나타낼 수 있도록 조합하는 것을 말한다. 동적 객체 데이터는 JSON[13], XML[14] 및 YAML[15] 등의 형식으로 기술될 수 있다.

동적 객체는 다음과 같은 특성을 가진다.

- **스키마리스(schemaless)**: 객체 인스턴스는 속성의 집합으로 구성되나, 해당 객체가 어떤 속성을 가져야 하는지, 각각의 속성이 특정 데이터형을 가져야 하는지에 대한 규율(스키마, 또는 데이터 모델)을 명시적으로 정의되지 않는다.
- **반 구조화된 데이터**: 동적 객체를 구성하는 요소의 명칭 및 데이터 형식은 동적 객체 형식이 요구하는 구성 규칙(예: JSON 형식)을 만족시키는 한 특정 스키마의 규정을 따르지 않아도 된다.

- **데이터 정의의 관용성과 유연함:** 고정된 스키마가 없어도 되기 때문에 스키마가 변경되었을 때도 파생된 데이터를 전부 수정하지 않아도 되며, 명칭은 같아도 상이한 데이터 형식의 멤버를 정의할 수 있다.
- **데이터 진화에 대응:** 시간의 경과로 데이터 구조가 변경되어야 할 때, 유연하게 대처할 수 있다. 다시 말해, 데이터 구조가 변경되었을 때 기존 객체 인스턴스에 신규 항목을 추가하는 작업이 자유롭다.

스키마가 없어서 발생할 수 있는 데이터 정합성 문제나, 데이터에 대한 설명이 불가능한 문제 등은 다음 절에서 기술할 방법론을 통해 보완할 수 있다.

3.2 동적 객체를 적용한 디지털 건설정보의 정의 방안

본 연구에서는 동적 객체의 이점인 유연성, 범용성을 가지면서도 표현 대상인 건설정보의 명확성 등 스키마가 존재할 때의 장점을 수용한 스키마 불가지론적(schema-agnostic) 동적 객체 데이터 정의 및 교환방식을 제시하고자 한다. 여기서, 불가지론이란 객체 데이터를 생성할 때 쓰인 스키마를 모르더라도 읽어 들이는 쪽에서 객체 데이터를 받아들여 처리할 수 있는 것을 말한다. 이것이 가능한 이유는, 데이터를 교환하고자 하는 양측이 동일한 업무영역을 공유하기 때문이다. 예를 들어, 도로 건설을 위한 설계업무와 공사비 산출업무가 비슷한 사례라 하겠다. 데이터를 받아들이는 측에서는 자신의 관점의 데이터 모델(스키마)을 가지게 되며, 이것은 데이터를 내보내는 측의 모델과 핵심적인 정의는 공유하지만, 개별 모델은 일치하지 않을 수 있다.

스키마 불가지론적 방법론에서 동적 객체는 다음과 같은 원칙을 따른다.

- JSON과 같은 동적 객체 데이터를 사용한다. 기본적으로 고정된 클래스를 명시하지 않는다.
- 동적 객체 개념에서 객체를 구성하는 속성은 고정된 데이터 형식을 갖지 않는 대신 데이터 값으로부터 속성 데이터의 형식을 유추할 수 있다. 단, 건설 엔지니어링 데이터 교환에 필요한 단위계, 좌표 원점 같은 정보는 명시적으로 공유되어야 한다.

3.2.1 고정적 객체모델을 대체하는 객체 사전의 정의

본 방법론에서 객체의 구성 정보는 사전 형태로 서술, 공개된다. 객체 사전이 수록하는 정보는 정적 객체모델에서 스키마에 의해 정의되는 객체의 클래스와 그 속성에 해당한다. Table 1은 객체 사건의 구성을 보여준다.

Table 1. List of the fields that make up a class dictionary item

Name	Data type	Description
clsId	alphanumeric string	Identifier of the given class. Different from classId in Table 3, which indicates a standard classification code
superId	alphanumeric string	Identifier of the immediate superclass of the given class
isAbstract	boolean (true/false)	Specifies whether the given class is an abstract class (i.e. it will not be instantiated)
members	list of value definitions (see Table 2)	Each value in this list represents the member definition of the given class

객체 사전에서 정의된 객체가 IFC 등의 기존 스키마 기반 객체와 구별되는 특징은 다음과 같다.

- 객체 사전에 정의되지 않은 데이터 항목도 객체의 멤버로 포함할 수 있다.
- 객체 사전상의 클래스 식별자를 반드시 갖지 않아도 되며, 사전에 정의되지 않은 클래스 식별자를 가져도 된다.
- 클래스 식별자 이외에, 부가적으로 객체의 형식번호, 분류명처럼 클래스를 지칭하는 다른 명칭을 별명으로 가질 수 있다.

객체 사건의 클래스는 Table 1의 members 항목의 집합으로 구성되며, 개별 멤버는 Table 2에 제시된 항목의 집합으로 정의된다. 각각의 속성은 그 명칭(name)과 속성의 형식(type)을 지정하는 값으로 Table 3의 Enumeration Item으로 정의된다.

Table 2. List of the fields that make up a class member item (in Table 1)

Name	Data type	Description
name	string (non-Latin characters such as Korean Hangeul are allowed)	Name of the member. May come from a standard object classification scheme if object properties are classified in it
type	enumeration (see Table 3)	Roughly specifies the member's data type, as the actual member values are type-agnostic

객체의 속성 멤버는 원칙적으로는 그 형식을 자유롭게 가져가지만, 객체 사전에서는 참고용으로 Table 2의 type으로 개별 속성의 형식을 지정할 수 있다. 여기에서는 JSON에서 구별하는 데이터 형식인 단일 값(Table 3

의 value), 내장 객체(embedded object, Table 3의 object) 및 배열(Table 3의 collection) 외에도 현 객체 바깥에 별도로 정의된 객체의 참조(object reference, 식별자를 속성값을 가짐으로써 표현됨)를 나타내는 열거 값인 objectRef가 추가된다.

Table 3. List of the enumerations that specifies the type of a class member

Enumeration item	Presumed data type	Description
className	string (non-Latin characters allowed)	name of the given class, or an alias of it, if it is classified by classId (see below)
classCode	alphanumeric string	classification code of the given class, if it is defined in a standard object classification scheme
objectId	alphanumeric string or long number (e.g. GUID)	identifier of the object instance created after the given class
value	non-object, non-collection types such as string or number	any value that represents the given member
objectRef	string or number	reference to the object specified by its objectId
object	JSON-style object instance,	actual content of the object instance as the value (instead of referencing an external object)
collection	list of arbitrary objects or values	the given member is made up of the list items

3.2.2 객체간 관계(relationship) 정의

본 방법론에서 객체 간 관계는 다음과 같이 나타낼 수 있다. 다만 관계 정의가 많아질수록 데이터가 객체모델 스키마에 종속되어 스키마 불가지성을 추구하기 어려워질 수 있음을 고려해야 한다.

- 부분-전체 조합 관계 (part-of/consists-of): JSON처럼 객체 내부에 부분을 나타내는 객체를 포함(embed)하는 방식을 사용한다.
- 일반-상세 관계 (is-a): 이를 나타내는 상/하위 클래스의 관계는 클래스 사전의 superId 항목을 통해 정의된다(Table 1 참조).
- 객체 사이의 참조 관계: 참조 대상 객체의 식별자를 속성으로 가짐으로써 객체 관계를 표현한다.

4. 객체 사전을 통한 데이터 교환 검증

4.1 객체 사전을 통한 동적 객체의 클래스 매칭

객체 데이터가 교환되는 환경에서 이를 생성하는 데 사용되는 객체모델 스키마를 알 수 없다면, 데이터를 받는 쪽에서는 해당 객체가 어떤 정보를 담고 있는지 바로 알아내기는 어렵다. 이 문제를 해결하기 위해 스키마 불가지론적 방법론에서는 데이터를 받는 쪽이 보유한 클래스 사전상의 멤버 목록과 객체를 비교하여 가장 유사한 클래스를 유추한다. 그 결과는 클래스의 전체 멤버 대비 객체 내 일치하는 멤버의 비율로 표현되며, 클래스의 멤버 정의를 100% 포함하지 않아도 된다.

이처럼 매칭을 수행하려면 객체의 멤버로 해당 속성을 포함하는 클래스 목록을 검색할 수 있는 매칭 사전이 필요하다. 이 사전은 멤버별 명칭을 key로 하고 이를 포함하는 클래스의 목록을 value로 갖는 key-value pair의 집합으로 구성된다. (Table 4 참조) 동일 멤버가 여러 클래스에 쓰일 수 있으므로 매칭 사전의 value는 복수의 매칭 클래스명 목록이 된다. 최종 매칭 클래스의 후보는 이들 속성별 클래스 목록의 교집합이 된다. 이러한 매칭 사전은 매칭을 수행할 때 클래스 사전을 바탕으로 동적으로 생성된다.

Table 4. Description of the attribute member dictionary, used for finding classes that contain the listed attribute

Name	Data type	Description
key	string (including non-Latin characters)	Name of the given attribute member
value	list of string (including non-Latin characters)	List of the class names that contain the member of the given attribute member

매칭 작업은 다음과 같은 절차를 통해 수행된다:

- (1). 주어진 객체의 모든 속성에 대해 이를 포함하는 클래스의 집합을 Table 4의 매칭 사전에서 찾아낸다.
- (2). 위에서 찾아낸 클래스 각각에 대해, Table 1의 클래스 사전으로부터 소속 멤버 목록을 찾아낸다.
- (3). (1)의 멤버 목록을 (2)의 후보 클래스별 목록과 비교하여 일치하는 멤버 수가 가장 많은 클래스를 일치 비율과 함께 반환한다.

4.2 사전을 이용한 동적 객체 데이터의 클래스 매칭 실험

동적 객체와 객체 사전을 이용한 데이터 정의 및 스키마 불가지론적 클래스 매칭 방법론의 타당성을 검증하기 위해 JSON 형식의 샘플 데이터 2종을 사용하여 아래와 같은 실험을 수행하였다. 실험방법 및 사용된 데이터의 개요는 다음과 같다:

- **ifcJSON 객체 데이터 샘플:** 시공작업 객체, 작업과 연계된 벽, 슬래브 등 71개의 객체로 구성됨
- **도로 포장공 객체 데이터 샘플:** 아스팔트 및 콘크리트 포장공 작업 객체, 작업과 연계된 노면, 포장층 등 270개의 객체로 구성됨
- 4.1에서 설명한 매칭 방법론을 구현한 프로그램을 코드를 작성하였고, Fig. 1은 해당 프로그램에 구현된 매칭 사전의 데이터 구조를 나타냄
- 기존 객체모델 스키마 2종(IFC4, 스마트건설 객체 모델)을 바탕으로 클래스 사전 생성
- 클래스 정보가 제거된 JSON 데이터 파일 2종(ifcjson 샘플, 스마트건설 객체모델 샘플)의 클래스 매칭을 수행
- 매칭된 결과를 데이터 원본(클래스 정보를 가진 JSON 파일)과 비교하여 적응률을 평가

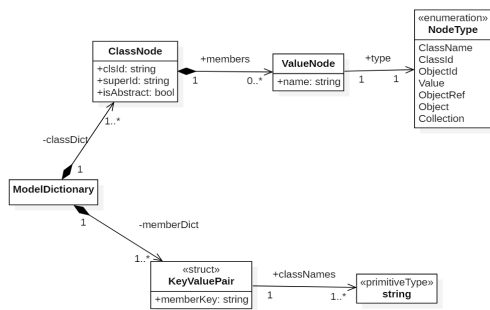


Fig. 1. UML class diagram showing internal data structures for the class matching test, as described in Section 4.2.

4.3 실험 결과

Fig. 2는 위의 실험 결과를 그래프로 나타낸 것이다. 진한 색 그래프는 ifcJSON 샘플 데이터, 연한 색 그래프는 스마트건설 객체모델 샘플 데이터를 나타낸다.

4.3.1 ifcJSON 데이터를 이용한 실험

샘플 데이터에 대한 매칭을 수행했을 때, 46.5%의 클래스를 정확히 맞추었다. 정확히 맞추지는 못했지만, 실

제 클래스의 슈퍼클래스로 매칭된 비율은 29.6%로. 이를 포함하면 76.1%의 적응률을 보였다.

4.3.2 스마트건설 객체모델 시안을 이용한 실험

샘플 데이터에 대한 매칭을 수행했을 때, 67.1%의 클래스를 정확히 맞추었다. 정확히 맞추지는 못했지만, 실제 클래스의 슈퍼클래스로 매칭된 비율은 8.1%로. 이를 포함하면 75.2%의 적응률을 보였다.

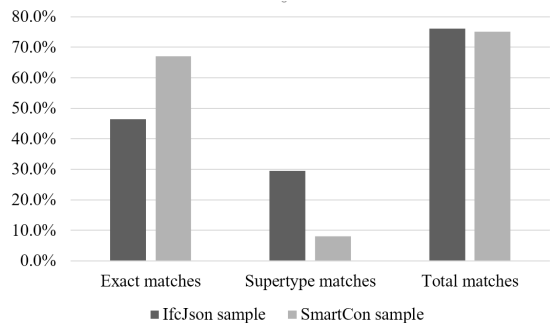


Fig. 2. Summary of the class matching tests against two sample data sets, using the class dictionary described in 3.2.1 and the attribute matching dictionary described in 4.1.

4.4 실험 결과 분석

실험 결과, 객체 인스턴스의 속성명을 이용해서 해당 객체의 클래스를 유추하는 것은 간단하지만, 데이터에 명시적인 클래스 정보가 없어도 데이터 생성에 쓰인 객체모델을 어느 정도 유추할 수 있음을 볼 수 있었다. 단, 75%대의 적응률을 90%, 95%로 향상시키는 방안이 강구되어야 할 것이다.

ifcJSON 대비 스마트건설 객체모델에서 슈퍼클래스 매칭 비율이 낮은 이유는 두 객체모델의 구성 차이 때문으로, 스마트건설 객체모델의 경우, 객체의 서브 클래스 단계가 더 깊었지만, 이들 서브 클래스 중 속성이 부여되지 않은 경우가 많았기 때문으로 보인다. 이 경우에는 멤버별 명칭뿐 아니라 객체 사이의 연관관계가 형성하는 그래프 구조를 매칭에 사용하는 방안을 생각해볼 수 있을 것으로 생각된다.

테스트 과정에서 객체모델과의 정합성은 참/거짓의 이분법 대신 비율로 표현할 수 있었다. 이 비율에 따른 품질 척도는 객체 데이터 품질을 확인할 때도 적용할 수 있을 것으로 보인다.

5. 스키마 불가지론적 교환 방법론 제시

지금까지 본 연구에서 설명한 스키마 불가지론에서 데이터의 교류는 기존의 단일 데이터 모델 기반이 아니라, 데이터 수요자 각자의 관점에서 정의된 데이터 사전을 바탕으로 이뤄진다. 이를 데이터 생성자와 수요자 사이의 데이터 교환에 적용해 본다면 데이터의 수요자는 필요한 데이터의 사양(교환 대상 데이터)을 사전 형식으로 선언할 수 있다. 이는 IDM의 교환요건(Exchange Requirements, 이하 ER)과도 일맥상통한다. IDM의 ER은 나중에 중립 모델인 IFC의 모델 뷰(MVD)로 매핑되는 반면, 스키마 불가지론은 중립 모델 없이 프로젝트 참여자 각자 관점의 객체모델을 정의하는 것으로, 프로젝트에 참여하는 임의 당사자 사이의 데이터 교환이 항상 가능해야 한다는 중립 모델의 전제가 현대에 와서 맞지 않게 된 점을 고려한 것이다.

데이터의 수요자는 전달받은 데이터를 수요자의 데이터 사전과 비교하여 요구사항과의 정합성을 평가하게 된다. 이때 필요한 최소 정합성은 데이터의 수요자가 결정하게 된다. 한편 데이터를 생성, 공급하는 측에서도 자신의 데이터 사전을 수요자에게 제공할 수 있다. 이는 수요자가 필요로 하는 데이터를 생성자가 제공할 수 있는지 판단하는 기준이 된다.

6. 결론

본 연구에서는 인프라 건설 프로젝트 수행과정에서 생성되는 다양한 데이터를 생성, 교류하기 위해 기존 고정적 데이터 모델 정의인 스키마 기반의 방법론 대신, 동적 객체와 객체 사전을 이용해 교환 대상 데이터의 의미를 유추하는 스키마 불가지론적 데이터 교환 방법론을 제시하였다.

스키마 불가지론적 방법론에서는 데이터를 교환하는 양자 사이에 유사성만 있다면, 데이터의 생성에 쓰인 스키마가 아닌 다른 객체모델에도 데이터를 근사적으로 매핑할 수 있다고 전제한다. 이러한 전제가 타당하며 구현 가능한지 검토하기 위해 클래스 정보가 제거된 2종의 JSON 샘플 데이터를 이용해 데이터 사전만으로 데이터 객체의 클래스를 유추할 수 있는지 테스트해 보았다. 그 결과, 두 샘플 모두 원래 클래스의 상위클래스까지는 75% 정도 유추할 수 있었다. 이는 본 연구에서 제시된 스키마 불가지론적 데이터 교환의 실현 가능성을 보여주

는 것으로 생각된다.

동적 객체는 다양한 출처에서 생성되는 균질하지 않은 구조의 데이터를 수용할 수 있어 기존 스키마 기반의 데이터 교환 방법론을 보완할 수 있지만, 스키마가 제공하는 데이터의 구조, 형식에 대한 정보가 누락되는 단점이 있다. 스키마 불가지론은 스키마 유무와 상관없이 동적 데이터의 구조를 유추함으로써 다양한 출처, 형식의 건설정보 데이터를 포괄할 수 있는 틀을 제시한다.

스키마 불가지론을 적용하더라도, 기존 디지털 및 비(非)구조화, 반(半)구조화 문서정보의 연계를 위한 건설정보 분류체계와 호환되도록 동적 객체에 분류체계 코드값을 갖도록 하면, 정형화된 객체모델의 부재에서 오는 부정합성의 문제를 상쇄할 수 있을 것으로 생각된다.

References

- [1] Industry Alliance for Interoperability, IFC Release 1.0 Specifications Volume II: IFC Object Model for AEC Projects, IAI, 1996.
- [2] Autodesk Inc., The AutoCAD User Reference, Autodesk Inc., 1982.
- [3] R. N. Nagel, W. W. Braithwaite, P. R. Kennicott, Initial Graphics Exchange Specification IGES Version 1.0, National Bureau of Standards, 1980.
- [4] M. J. Pratt, "Introduction to ISO 10303—the STEP standard for product data exchange", *Journal of Computing and Information Science in Engineering*, Vol. 1, No. 1, pp.102-103, 2001.
DOI: <https://doi.org/10.1115/1.1354995>
- [5] buildingSMART International, IFC 4.3.x Specification [Internet], buildingSMART International, c2022 [cited 2022], Available from: <https://ifc43-docs.standards.buildingsmart.org/> (accessed Oct. 10, 2023)
- [6] BLIS Project, Building Lifecycle Interoperable Software (BLIS) Project Brief [Internet]. VTT Building Technology, c2002 [cited 2002 Aug. 09], Available From: <http://www.blis-project.org/index2.html> (accessed Oct. 10, 2023)
- [7] J. Hietanen, The Concept Block Approach to View Definitions, Technical Report, VTT Building Technology, 2000.
- [8] J. Hietanen, IFC Model View Definition Format, Technical Report, International Alliance for Interoperability, 2006.
- [9] J. Wix, Information Delivery Manual (IDM): Enabling Information Exchange in AEC/FM Business Processes, Technical Report, Jeffrey Wix Consulting, Thatcham, UK, 2005.

- [10] E. W. East, Construction Operations Building Information Exchange (COBie), Technical Report, Engineer Research and Development Center, US Army Corps of Engineers, 2007.
- [11] M. H. Rasmussen, M. Lefrançois, G. F. Schneider, P. Pauwels, "BOT: the Building Topology Ontology of the W3C Linked Building Data Group", *Semantic Web-Interoperability, Usability, Applicability*, Vol. 12, No. 1, pp.143-161, Nov. 2021.
DOI: <https://doi.org/10.3233/SW-200385>
- [12] S. Sobhkhiz, T. El-Diraby, "Dynamic integration of unstructured data with BIM using a no-model approach based on machine learning and concept networks", *Automation in Construction*, Vol. 150, 104859, 2023.
DOI: <https://doi.org/10.1016/i.autcon.2023.104859>
- [13] Ecma International, The JSON Data Interchange Syntax, Standard, ECMA-404, 2nd. Edition, Dec. 2017.
- [14] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau, J. Cowan, Extensible Markup Language (XML) 1.1 (Second Edition) [Internet], World Wide Web Consortium, c2006 [cited 2006 Sep. 29], Available From: <https://www.w3.org/TR/xml11/> (accessed Oct. 10, 2023)
- [15] I. Net, T. Müller, P. Antoniou, E. Aro, T. Smith, YAML Ain't Markup Language (YAML™) version 1.2 Revision 1.2.2 [Internet], YAML Language Development Team, c2021 [cited 2021 Oct. 1], Available From: <https://yaml.org/spec/1.2.2/> (accessed Oct. 10, 2023)

김 상 욱(Sang-Wook Kim)

[정회원]



- 2000년 2월 : 상지대학교 토목공학과 (공학사)
- 2011년 8월 ~ 현재 : (주) 피터씨 기술개발센터 수석연구원

<관심분야>

토목구조, 프리캐스트 콘크리트, 스마트건설

유 석 준(Seok-Joon You)

[정회원]



- 1997년 8월 : 홍익대학교 대학원 건축학과 (공학석사)
- 2003년 12월 : Georgia Tech College of Architecture (박사 과정 수료)
- 2018년 1월 ~ 현재 : (주) 삼안 기술개발센터 수석연구원

<관심분야>

건설 IT, BIM, 스마트건설