What Information Can or Cannot Be Exchanged?

Ghang Lee, Ph.D.1

Abstract: What information can or cannot be exchanged between different systems? Since the dawn of the computer-aided design and engineering era, interoperability has been an issue. The exchangeable set of information between different systems has been loosely defined as the intersection of information. Yet, as information flow has directionality, a general definition of “intersection” is inadequate to define an exchangeable set of information. This paper proposes a new directive set operation, semantic intersection, and discusses which information can or cannot be exchanged between different systems using the concept of semantic intersection. This opens up the possibility of predetermining or calculating the exchangeable set of information between two or more systems or the possibility of automated generation of a standard product model between different systems. This paper focuses on data exchange between systems, but the proposed theory is also applicable to data exchange between human beings.

DOI: 10.1061/(ASCE)CP.1943-5487.0000062

CE Database subject headings: Information management; Information systems; Data processing.

Author keywords: Information exchange; Interoperability; Information standard; Semantic intersection; Semantic subset.

Introduction

The flow of information follows a natural direction just as heat flows from hotter objects to colder ones and water flows down from a height. However, unlike the flow of heat or water, the flow of information is not always unidirectional. Information generally flows from the more informed to the less informed; yet, depending on the contents of the information, it can sometimes flow in the opposite direction. In some cases, a set of information can be exchanged freely, with little constraint, between two agents. Conversely, in other cases, the exchange of information in one direction is practically impossible unless forced by making artificial assumptions. For example, although it is possible to derive two-dimensional representations from a three-dimensional representation, it is generally very difficult to reverse the process without making wild guesses to fill the missing information gaps.

Since the dawn of the computer-aided design (CAD) and engineering era, interoperability (data exchange capability) has been an issue (Goldstein et al. 1998). An early solution for interoperability issues was the single standard product model, which resulted in ISO 10303 STEP (ISO TC 184/SC 4 1994), CIS/2 (Crowley 2003), and industry foundation class (IFC) (buildingSMART 2009a). Many successful case studies were conducted on the deployment of such standard data models (AISC 2009; Smith 2002). However, the same number of interoperability problems has since been reported (Amor 2006; Gielingh 2008; Jeong et al. 2009; Kiviniemi et al. 2008; Lim et al. 2008; Pazlar and Turk 2008). Even though some of these problems can be attributed to the standard data model issues and system bugs, some information is not meant to be exchanged between two different systems.

The sources of exchange problems can be classified into the following four groups. (1) Incomplete coverage of a data model (or an exchange format): A data model (or an exchange format) such as ISO STEP and IFC does not fully support the intended data exchange scenarios. For example, if a data model does not include a definition of a reinforcing bar, reinforcing bar information cannot be exchanged using the data model. Gielingh (Gielingh 2008) identified this as one of the major causes of interoperability problems. (2) Translator problems: This is also a common cause of the interoperability issues today (Pazlar and Turk 2008). Translators are the modules of a system that import nonnative file formats or export the native file format to other formats. Translator problems occur because each translator developer develops translators without specific development guidelines. This type of problem can be relieved by the use of specific translator development guidelines; one such effort is the information delivery manual (buildingSMART 2009b). (3) System bugs: Data exchange problems can occur simply due to system bugs. For example, a data set that is read into a system correctly can be displayed different from the original data set simply because of a bug in the visualization module. (4) Software domain problems: A software application does not support the domain or the object. For example, a design software application usually does not support structural loads. In such cases, even if structural load information is read into a design system, the information cannot be interpreted and will be abandoned.

Given that some types of information are meant not to be exchanged between two systems that are different by nature, what information can or cannot be exchanged between different systems? Intersection has been used to loosely define an exchangeable set of information (Gielingh 2008). Obviously, the intersection of two information sets is exchangeable between the systems. For example, if one system includes “project code” information and the other system includes the same information, the two systems can exchange the information project code. In reality, each system defines a data field in its own way. For example, project code may mean the “identifier number of a project” in one system, but “short project name” in the other. Furthermore, if System A stores the “start date” and “end date” of a project and...
System B stores only the “project duration,” System A can derive the project duration by subtracting end date from start date, which can be passed to System B, but not vice versa. Here, we see that the traditional set-theoretic notion of intersection is not adequate to define the exchangeable set of information between different systems. Hence, we propose a generalized and formal definition of such exchangeable and nonexchangeable sets of information between two different systems by introducing the concept of “semantic intersection.”

**Exchangeable Set of Information—Theorems and Proofs**

Yang and Eastman (Yang and Eastman 2007) introduced the concept of a *semantically valid subset* in defining rules for checking the validity of subsets in a product model. In this paper, the term “semantic” refers to the “meaning of things” as it is used in linguistics. The semantically valid subset is a subset that maintains what Hammer and McLeod called “semantic integrity” (Hammer and McLeod 1975, 1976). For example, consider that, if a data model contains the entity building that represents a building in the real world, the building entity may be required to consist of at least one column and a roof to be in the minimum form of a building. If not, the building entity would be said to be semantically invalid. Semantic integrity is often represented as constraints in a data model. The semantic intersection that we are proposing is also a subset of two sets, but the definition of a semantic intersection in this paper differs from the semantically valid subset of Yang and Eastman or semantic integrity of Hammer and McLeod (Hammer and McLeod 1975, 1976). The semantically valid subset does not deal with the definition of an exchangeable set between two sets of information. The semantic intersection does not deal with an issue of the semantic validity of an information set, i.e., whether a set conforms to all the premises derived from the real world or not, although it is an important issue. Refer to Lee (2009b) for more discussions on the validity of exchanged information.

We define a semantic intersection ($\cap^s$) as a set of information items in different data sets that are functionally dependent. For example, let us assume that Set $A$ below is a set of information required by a “project management system,” and that Set $B$ below is a subset of information required by a “structural analysis system”

$$A = \{\text{project\_name}, \text{load}, \text{manager}\}$$

$$B = \{\text{structure\_name}, \text{load}, \text{frame}\}$$

The results of regular set operations of these two sets will be

$$A \cap B = \{\text{load}\}$$

$$A \cup B = \{\text{project\_name}, \text{load}, \text{manager}, \text{structure\_name}, \text{frame}\}$$

In this case, if the exchangeable set of information is defined as the intersection of two information sets, only “load” can be exchanged. However, as mentioned above, two different systems do not always use the same terms to mean the same thing or have the same internal data structure. Thus, let us assume that the term “project\_name” in Set $A$ is a synonym of the term “structure\_name” in Set $B$, while the term load in Set $A$ is a homonym of the term load in Set $B$, respectively, denoting “truck load” and “structural load.” Then, we know that project\_name and structure\_name can be exchanged, whereas the rest of the information will be lost in the process of information exchange.

In this case, the exchangeable set between Systems $A$ and $B$ is \{structure\_name\}. This can be represented as follows using the exchangeable set symbol $\cap^s$:

$$A \cap^s B = \{\{\text{project\_name};\text{structure\_name}\}\}$$

where $\{x_0, x_1, x_2, \ldots, x_n; y\}$ denotes that $y$ in Set $B$ is equivalent to or derivable from $x_0, x_1, x_2, \ldots, x_n$ in Set $A$.

Additionally, a semantic intersection is direction sensitive. That is, an exchangeable set between two systems is direction dependent. For example, let us assume that Set $A$ below is a set of information in a “CAD system” and Set $B$ below is a set of information in a “cost estimation system”

$$A = \{\text{room\_width}, \text{room\_depth}\}$$

$$B = \{\text{room\_area}\}$$

While a CAD system may contain room width and depth information, a cost estimation system may contain only the room area information, but not the room width and depth information. A CAD system can derive the room area information by multiplying the room’s width by its depth (assuming that the room has a rectangular shape) and pass it to System B. However, the initial room width and depth information cannot be retrieved from the room area information in System B. Thus, no information can be passed from System B to System A in this example. These facts can be represented using the semantic intersection symbol as follows:

$$A \cap^s B = \{\{\text{room\_width}, \text{room\_depth};\text{room\_area}\}\}$$

or

$$A \cap^s B = \{\text{room\_area}\}$$

$$B \cap^s A = \emptyset$$

We suggest that information exchangeable from $A$ to $B$ always be represented as a pair using the notation $(x:y)$ or only the information items in the target system $B$ to show which information can actually be passed between the systems. In the above example, we should not write $A \cap^s B = \{\text{room\_width}, \text{room\_depth}\}$ because the two items are not the information that can be passed to $B$. The semantic intersection and related operations and theorems can be formally defined as follows.

**Definition 1. Semantic equivalence function $f_x$:** Given two information sets, $X$ and $Y$, the semantic equivalence function $f_x$ is defined as a partial function from $X$ to $Y$ such that $f_x(x)$ is understood to be functionally dependent on $Y$ if $Y$ can be derived from $X$.

Note 1: If $Y$ is not functionally dependent on $X$, $f_x(x)$ is understood to be the empty set $\emptyset$. That is, $f_x$ as a partial function simply returns no value for that particular argument.

Note 2: If $Y = f_x(x)$, then $Y$ is said to be functionally dependent on $X$ and the relationship is denoted as $X \rightarrow Y$.

Note 3: The definition of functional dependency (FD) here is equivalent to that used in the definition of the third normal form in relational database theories (Bernstein 1976).

“Let $A$ and $B$ be attributes, let DOM($A$) be the domain of $A$ and DOM($B$) be the domain of $B$, and let $f$ be a time-varying function such that $f$: DOM($A$) $\rightarrow$ DOM($B$). $f$ is not a function in the precise mathematical sense because we allow the extension to vary over time in the same sense that we allow extensions of database relations to change over time. That is, if $f$ is thought of as a set of ordered pairs $\{(a, b) | a \in$ DOM($A$) and $b \in$ DOM($B$)$\}$, then, at every point in time for a given value of $a \in$ DOM($A$), there will be at most one value of $b \in$ DOM($B$). To distinguish $f$...”

2 / JOURNAL OF COMPUTING IN CIVIL ENGINEERING © ASCE / JANUARY/FEBRUARY 2011
from a mathematical function, we call } f } a FD. For notational convenience, we generally omit the ‘DOM’ and write } f : A \rightarrow B \text{. If there is a } \text{FD } } f : A \rightarrow B \text{, then } B \text{ is said to be functionally dependent on } A.”

The definition of semantic equivalence is not restricted to synonyms. The entities in the FD, the driving and driven entities, are also semantically equivalent; for example, } \{ \text{first name, last name} \} \rightarrow \{ \text{full name} \}. \text{ The major difference between synonyms and the entities in the FD is that entities in the driving and driven relations may not be exchangeable bidirectionally, whereas synonyms are always exchangeable bidirectionally. In natural language, many terms are ambiguous. But in data exchange, every term should be defined explicitly. For example, if “door” in Set } A \text{ is defined as a synonym of “gate” in Set } B \text{, they should signify exactly the same thing and, thus, are bidirectionally exchangeable. If gate is even slightly different from door, they are likely to be functionally dependent on each other. That is, they are equivalent conditionally and only when they are combined with additional information.}

**Definition 2. Semantic intersection (} \cap \text{):** The exchangeable set between two information sets is the semantic intersection between the two information sets. The semantic intersection (} \cap \text{) is the subset } Y \text{ of Set } B \text{, which is functionally dependent on a subset } X \text{ of Set } A \text{. Note that the semantic intersection is direction sensitive. Using the semantic equivalence function } f_\text{, the semantic intersection of Sets } A \text{ and } B \text{ can be defined as follows:}

\[ A \cap \cap B = \{ Y \subseteq B \mid X \subseteq A \land f_\text{}(X) = Y \} \]

A set of information exchangeable between two systems is a set of information such that the following theorems hold.

**Theorem 1.** A set of data exchangeable between two different systems is the semantic intersection of two application data schemas and is direction sensitive: Exchangeable information from Systems } A \text{ to } B = A \cap \cap B. \text{ Proof 1. If information cannot be comprehended by System } B \text{ in any form, the information cannot be imported by a system from other systems. When information is passed from System } A \text{ to System } B \text{ and } X \text{ denotes a set of information passed from System } A \text{ to System } B \text{, if } X \subseteq B \text{ or } f_\text{}(X) \subseteq B \text{, System } B \text{ will not be able to comprehend and store } X \text{ as native data. The information } X \text{ passed from System } A \text{ to System } B \text{ will be abandoned or imported as meaningless information to System } B \text{. On the other hand, if } f_\text{}(X) \subseteq B \text{, System } B \text{ will be able to comprehend and store } X \text{ as native data.}

**Theorem 2.** The semantic intersection and the intersection of two sets have the following superset-subset relation depending on inclusion of homonyms and synonyms:

1. (No homonym, no synonym, and no functionally dependent elements) If there is no homonym, no synonym, and no functionally dependent elements between two sets } A \text{ and } B \text{, then the semantic intersection and the intersection of two sets are the same:

\[ (A \cap \cap B) = (A \cap B) \]

2. (Synonyms only) If there are synonyms, but no homonyms or no functionally dependent elements between Sets } A \text{ and } B \text{, the semantic intersection of two sets is always a superset of the intersection:

\[ (A \cap \cap B) \supset (A \cap B) \]

3. (Homonyms only) If there are homonyms, but no synonyms or functionally dependent elements between Sets } A \text{ and } B \text{, the intersection of two sets is always a superset of the semantic intersection:

\[ (A \cap B) \supset (A \cap \cap B) \]

4. (Functionally dependent information) If Sets } A \text{ and } B \text{ include functionally dependent elements between each other but no homonyms and no synonyms, the semantic intersection of two sets is always a superset or the same set of the intersection:

\[ (A \cap \cap B) \supset (A \cap B) \]

5. (Coexistence of homonyms and synonyms) If there are synonyms and homonyms between Sets } A \text{ and } B \text{, there is no consistent superset-subset relation between the intersection and the semantic intersection of two sets.}

**Proof 2.** Only four cases are of importance: (1) two sets without any synonyms, homonyms, and functionally dependent elements; (2) two sets only with synonyms; (3) two sets only with homonyms; and (4) two sets only with functionally dependent elements.

1. If there are no synonyms, no homonyms, and no functionally dependent elements between Sets } A \text{ and } B \text{, there will not be any semantic differences between the semantic intersection and the intersection of Sets } A \text{ and } B \text{. Therefore, by definition of the semantic intersection, the semantic intersection and the intersection of the two sets will be the same.

2. If there are synonyms, but no homonyms and no functionally dependent elements between Sets } A \text{ and } B \text{, the synonyms will be included in the semantic intersection, but not in the intersection. The semantic intersection will always include more elements than the intersection by the number of synonyms in addition to other equivalent elements. Therefore, the semantic intersection will always be a superset of the intersection in this case.

3. If there are homonyms, but no synonyms and no functionally dependent elements between Sets } A \text{ and } B \text{, the homonyms will be included in the intersection, but not in the semantic intersection. The intersection will always have more elements than the semantic intersection by the number of homonyms in addition to other equivalent elements. Therefore, the intersection will always be a superset of the semantic intersection in this case.

4. If there are functionally dependent elements, but no synonyms and no homonyms between Sets } A \text{ and } B \text{, the functionally dependent elements may or may not be included in the semantic intersection depending on the direction of information flow, but they will always not be included in the intersection. The semantic intersection will always be the same as the intersection if all the elements in FD exist in both sets. Otherwise, the semantic intersection will have more elements than the intersection by the number of functionally dependent elements. Therefore, the semantic intersection will always be the same set as the intersection or a superset of the intersection in this case.

5. If homonyms and synonyms or functionally dependent elements coexist in Sets } A \text{ and } B \text{, both the semantic intersection and the intersection will have elements that the other does not have. Therefore, there will not be a consistent superset-subset relation between Sets } A \text{ and } B \text{.}

**Example 1.** Let } A = \{ \text{load} \}, B = \{ \text{load} \}, C = \{ \text{cargo} \}, \text{ and } D = \{ \text{lateral load, vertical load} \}.

1. (No homonym, no synonym, and no functionally dependent elements) If load in Sets } A \text{ and } B \text{ means the same thing.
(2) (Synonyms only) If load in Set A means truck load and is a synonym of “cargo”
\[
(A \cap \mathcal{C}) = \{\text{load}, \text{cargo}\}, \quad (A \cap \mathcal{C}) = \emptyset
\]
\[
\therefore (A \cap \mathcal{C}) \supseteq (A \cap \mathcal{C})
\]
(3) (Homonyms only) If load in Set A and load in Set B are homonyms (i.e., load in Sets A and B means different things such as truck load and structural load)
\[
(A \cap \mathcal{B}) = \emptyset \quad (A \cap \mathcal{B}) = \{\text{load}\}
\]
\[
\therefore (A \cap \mathcal{B}) \supseteq (A \cap \mathcal{B})
\]
(4) (Functionally dependent elements) If \( D : \{\text{lateral load, vertical load} \} \rightarrow B : \{\text{load}\} \) holds, the semantic intersection of Sets \( B \) and \( D \) will depend differently on the direction of information flow. However, the intersection of Sets \( B \) and \( D \) will be always empty
\[
(B \cap \mathcal{D}) = \emptyset
\]
\[
(D \cap \mathcal{B}) = \emptyset
\]
\[
\therefore (B \cap \mathcal{D}) \supset (D \cap \mathcal{B})
\]
(5) (Coexistence of homonyms and synonyms) Let \( A = \{\text{load}\} \) and \( B = \{\text{load, cargo}\} \), load in Set A means truck load and load in Set B means structural load
\[
(A \cap \mathcal{B}) = \{\text{load, cargo}\} = \{\text{cargo}\}
\]
\[
(A \cap \mathcal{B}) = \{\text{load}\}
\]
\[
\therefore \text{There is no superset-subset relation between Sets A and B.}
\]

Another factor that defines the directionality of information exchange is the subset relation between two targeted sets. If there are no homonyms and no synonyms in Sets A and B, and if Set B is a subset of Set A, by definition of a subset, any element in Set B will always be an element of Set A and, therefore, exchangeable with elements in Set A, but not vice versa. Here is a general definition of the subset relation between two sets
\[
A \supset B = \{b \in B | b \in (A \cap \mathcal{B})\}
\]

Note that homonyms, synonyms, and functionally dependent elements are not considered in general set operations. However, when homonyms or synonyms exist in two targeted sets, a traditional definition of the subset does not hold. We propose a concept called a semantic subset to deal with such cases.

**Definition 3.** Semantic subset (\( \supset \)): If any element in Set B is derivable from elements in Set A, Set B is a semantic subset (\( \supset \)) of Set A. Therefore, any element in Set B will always be exchangeable with elements in Set A, but not vice versa
\[
A \supset B = \{b \in B | b \in (A \cap \mathcal{B})\}
\]

This definition holds for cases with homonyms, synonyms, and functionally dependent elements. Different relations exist between the semantic subset and the general subset depending on the inclusion of homonyms, synonyms, and functionally dependent elements between two sets. Theorem 3 specifies the different relations.

**Theorem 3.** If there are homonyms, but no synonyms and no functionally dependent elements of the homonyms between Sets A and B, A and B are never in the semantic subset relation even if A and B are in the subset relation. If there are synonyms or functionally dependent elements, but no homonyms between Sets A and B, A and B are never in any subset relation even if A and B are in the semantic subset relation.

**Proof.** If there are homonyms, but no synonyms and no functionally dependent elements of the homonyms between Sets A and B, there always will be at least one element in each set, which will not be included in the semantic intersection of Sets A and B by definition of homonym. Therefore, by Definition 3, if there are homonyms between Sets A and B, A and B are always not in the semantic subset relation even if A and B are in the subset relation.

If there are synonyms or functionally dependent elements, but no homonyms between Sets A and B, there always will be at least one element in each set, which will not be included in the intersection of Sets A and B. Therefore, by definition of the subset, if there are synonyms between Sets A and B, A and B are always not in any superset-subset relation even if A and B are in the semantic subset relation.

**Example 2.** As in Example 1, let \( A = \{\text{load}\}, B = \{\text{load, cargo}\}, C = \{\text{cargo}\}, \) and \( D = \{\text{load, vertical load}\} \). If load in Set A and load in Set B are homonyms, \( A \supset B \) is FALSE although \( A \supset B \) is TRUE. If load in Set A is a synonym of cargo in Set C, \( A \supset C \) is FALSE although \( A \supset C \) is TRUE. If Set B is functionally dependent on Set D and \( (B \cap \mathcal{D}) = \{\text{load}\} \), then \( B \supset D \) is FALSE, and \( D \supset B \) is TRUE.

Standard data models are developed based on the concept of unification (Gielingh 2008). One such example is the development strategy of ISO 10303 STEP (ISO TC 184/SC 4 1994), which expands its coverage by developing individual information sets for specific domains, called application protocols (APs), first and then unifying them into a larger data model by adding cross-reference links to APs. However, note that the scope of a standard product model is not the union of data included in all target systems. It is meaningless to include all information because some information will not be read into the other target systems. For example, if System A includes “date of birth” and System B includes “age,” age can be derived from date of birth but not vice versa. In this case, it will be meaningless to include date of birth in a standard product model because System B will not be able to read date of birth into the system or produce date of birth for System A even if date of birth is included in the product model. However, it will be still meaningful to include age in a product model because age information can be passed at least in one direction, i.e., from A to B. Thus, the scope of a standard product model should not be the union of every set, but the union of exchangeable sets between target systems. We call the operation the semantic union (\( \cup^* \)). Applying the concept of semantic intersection, this proposition can be generalized for defining semantic union (\( \cup^* \)) as follows.

**Definition 4.** Semantic union (\( \cup^* \)): A set of information exchangeable between two systems regardless of the direction of the information exchange, denoted as \( \cup^* \), is the infinite union (or grand union, \( \cup \)) of all possible semantic intersections
\[
A \cup^* B = \cup (A \cap \mathcal{B}, B \cap \mathcal{A})
\]

This operation is direction insensitive
\[
A \cup^* B = B \cup^* A
\]

The following relation holds between the semantic union and the semantic intersection.

**Theorem 4.** The semantic union of Sets A and B is always a superset of the semantic intersection of Sets A and B
\[
(A \cup^* B) \supset (A \cap \mathcal{B}) \cup (A \cup^* B) \cup (B \cap \mathcal{A})
\]
Proof 4. If $Z = X \cup Y$, then $Z \supseteq X$ and $Z \supseteq Y$. Therefore, $(A \cup ^* B) \supseteq (A \cap ^* B)$ and $(A \cup ^* B) \supseteq (B \cap ^* A)$ hold by the definition of a semantic union. These theorems can be generalized to define the scope of a standard data model that can support multiple different applications.

Definition 5. The scope of a standard product model: The scope of a standard product model that can support $n$ number of different applications is the infinite union (or grand union, $\cup$) of all possible semantic intersections. Let $A_i$ be a set of information in an application and $n$ the total number of applications in the data exchange scenarios. The scope of a standard product model

$$E = \bigcup_{i=1}^{n} (A_i \cap ^* A)$$

This definition composed of the semantic intersection, which has directionality and allows data exchange of only the common semantic subset of two information sets, implies that a standard product model does not guarantee lossless data exchange or bidirectional data exchange between two systems. However, a standard product model is still useful in supporting data exchange between various systems because when we exchange data between two systems, we do not need to pass all the data or the entire semantic intersection of two information sets from one system to another, but only a part of it (i.e., information that is required to perform the next task). Also, data exchange is often unidirectional. Examples include data exchanges between a design system and a cost estimation system or with a structural detailing system.

Application Example

This section demonstrates how the definitions and theorems previously specified can be used to predetermine or calculate the exchangeable set of information between two or more systems and to define a product model to support the exchange of “beam” information between four different systems, namely, Revit building, ArchiCAD, Bentley Triforma, and Tekla structures. In this example, the semantic intersections were calculated in two separate steps.

First, models (information sets) created in four different applications were mapped to a common data format so that the information sets could be compared to one other. In this study, a function to export a model into the IFC format in each system was used for semantic mapping between the native data structure and the IFC, assuming that the translator developers went through a semantic mapping process between the native data structure and IFC during the translator development process. IFC is a standard data format for the architecture, engineering, and construction (AEC) industry (ISO TC 184/SC 4 2008). Instead of arbitrarily creating a beam model, beam models created by buildingSMART to test IFC files (buildingSMART 2007) were used. This semantic mapping process can be further improved by developing a mapping tool between different data structures, which will be similar to the tools used to integrate multiple databases.

Second, the intersections between two information sets were calculated using the exported IFC files. Table 1 shows the number of entities in the beam models and other basic information. The entities in information sets $A_1$, $A_2$, $A_3$, and $A_4$ were extracted using Compare P21, a software application developed by Building Informatics Group at Yonsei University, to analyze the similarities and differences between two IFC instance files (Lee 2009a). The extracted information sets are listed in Appendix I. The number of elements in each set ranged from 49 to 55. The number of elements in the union of the information sets, i.e., the discrete number of entities, was 66. The union of information sets was derived using a script developed in Excel.

Another script was developed to calculate the intersections between the information sets. Since semantic mapping was carried out in Step 1, these intersections were equivalent to the semantic intersections. The exchangeable set is the semantic intersection between two information sets (Definition 2). Table 2 shows the exchangeable and nonexchangeable entities between the two systems. The number of nonexchangeable entities is the total number of entities in the source information set ($A_i$) less the number of exchangeable entities. The number of nonexchangeable entities differs depending on the direction of data exchange because the source information set numbers differ.

The third script was developed to find the entities in a product model that can support data exchange among the four information sets and that is based on Definition 5: the scope of a standard product model. The list of entities in the derived product model is provided in Appendix II. The number of entities was 66. Coincidentally, the elements in the union of these four sets and the elements in the product model were the same. However, they may not always be the same if some entities are only supported by one system in the data exchange loop.

This example is only a small test case and it will require considerable effort and time to develop a full-scale product model that can support the entire building information data exchange. Nevertheless, considering that such efforts have been underway since 1994 and that there is still more to be done, a rigorous theoretical background that can shorten, strengthen, and possibly

Table 1. Number of Entities in “Beam” Models

<table>
<thead>
<tr>
<th>Information set</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$A_4$</th>
<th>Total number of entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling application</td>
<td>Revit building</td>
<td>ArchiCAD</td>
<td>Bentley system</td>
<td>Tekla structures</td>
<td>—</td>
</tr>
<tr>
<td>Total number of entities</td>
<td>55</td>
<td>53</td>
<td>49</td>
<td>54</td>
<td>66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Number of Exchangeable and Nonexchangeable Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1 \cap ^* A_2$</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>Number of exchangeable entities</td>
</tr>
<tr>
<td>Number of nonexchangeable entities</td>
</tr>
<tr>
<td>$A_j$ to $A_i$</td>
</tr>
</tbody>
</table>
automate the development and evaluation process is essential. This paper is one such effort.

Conclusion

We argue that three types of information sets can be exchanged between two different systems: (1) the intersection of two sets; (2) a set of information in the synonym relation; and (3) a set of information that is functionally dependent on another set. However, note that the semantic intersection does not guarantee the validity of exchanged information. The issue of a valid set has been discussed in length in another paper (Lee 2009b) focusing on information sets specified in EXPRESS language. In order to define an exchangeable set of information between different systems that satisfy all these criteria, we proposed a formal definition of an exchangeable set of information and its characteristics using a new set-theoretic set operation semantic intersection based on the theory of FD. We also defined the semantic subset and the “scope of a standard product model” based on the definition of the semantic intersection.

The defined definitions imply several apparent yet important facts about data exchange between two systems. First, there is always a set of information that cannot be exchanged between two systems regardless of the size of the coverage of an exchange data format or how well translators are developed. The information that is not the semantic intersection of two application data models cannot be exchanged. Therefore, there cannot be complete lossless data exchange between two different types of applications. Furthermore, a standard product model also cannot guarantee lossless data exchange. Second, we can also conclude that the more similar two application types are, the more information they can exchange. Third, the set of information that can be exchanged between two systems differs depending on the direction of data exchange. Fourth, with all these limitations a standard product between two systems differs depending on the direction of data exchange. Fourth, with all these limitations a standard product between two systems differs depending on the direction of data exchange. Fourth, with all these limitations a standard product between two systems differs depending on the direction of data exchange. Fourth, with all these limitations a standard product between two systems differs depending on the direction of data exchange. Fourth, with all these limitations a standard product between two systems differs depending on the direction of data exchange. Fourth, with all these limitations a standard product between two systems differs depending on the direction of data exchange. Fourth, with all these limitations a standard product between two systems differs depending on the direction of data exchange.

The specified definitions and theorems provide an explicit definition of an exchangeable set of data between different applications. This opens up the possibility of predetermining or calculating the exchangeable set of information between two or more systems and also the possibility of automating the generation of a standard product model between different systems based on Definition 5: the scope of a standard product model. As an application example, this paper demonstrated the development process of a product model that could support the exchange of beam information between four different systems.

There remain, however, many more things to be done. Data dictionaries and tools that can support mapping between homonyms, synonyms, and derived items are required to accelerate the semantic mapping process. Once such tools and dictionaries are developed, the theorems can be used to validate the mapping relations. Although the mapping process may still be tedious, once the mapping is completed, millions of data instances can be exchanged automatically. One effort to develop a data dictionary for the AEC industry is the international framework for dictionaries (IAI 2007). In the long term, we hope that the proposed definitions and theorems can contribute to the understanding of data exchange between both systems and human beings.

Acknowledgments

The writer thanks Dr. Kiyong Lee, Jongsung Won, and Jiyong Jeong for their constructive comments and discussions. This research was supported by grant “06-Unified and Advanced Construction Technology Program-E01” from the Korean Institute of Construction and Transportation Technology Evaluation and Planning (KICTEP).

Appendix I. Information Sets Exported from Each Application

<table>
<thead>
<tr>
<th>Revit</th>
<th>ArchiCAD</th>
<th>Triforma</th>
<th>Tekla Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFCAPPLICATION</td>
<td>IFCAPPLICATION</td>
<td>IFCAPPLICATION</td>
<td>IFCAPPLICATION</td>
</tr>
<tr>
<td>IFCARBITRARYCLOSEDPROFILEDEF</td>
<td>IFCARBITRARYCLOSEDPROFILEDEF</td>
<td>IFCARBITRARYCLOSEDPROFILEDEF</td>
<td>IFCARBITRARYCLOSEDPROFILEDEF</td>
</tr>
<tr>
<td>IFCARBITRARYPROFILEDEFWITHVOIDS</td>
<td>IFCAXIS2PLACEMENT2D</td>
<td>IFCAXIS2PLACEMENT2D</td>
<td>IFCAXIS2PLACEMENT2D</td>
</tr>
<tr>
<td>IFCAXIS2PLACEMENT2D</td>
<td>IFCAXIS2PLACEMENT2D</td>
<td>IFCAXIS2PLACEMENT2D</td>
<td>IFCAXIS2PLACEMENT2D</td>
</tr>
<tr>
<td>IFCAXIS2PLACEMENT3D</td>
<td>IFCBEAM</td>
<td>IFCBEAM</td>
<td>IFCBEAM</td>
</tr>
<tr>
<td>IFCBEAM</td>
<td>IFCBONDINGBOX</td>
<td>IFCBONDINGBOX</td>
<td>IFCBONDINGBOX</td>
</tr>
<tr>
<td>IFCBONDINGBOX</td>
<td>IFCBUILDING</td>
<td>IFCBUILDING</td>
<td>IFCBUILDING</td>
</tr>
<tr>
<td>IFCBUILDING</td>
<td>IFCBUILDINGSTOREY</td>
<td>IFCBUILDINGSTOREY</td>
<td>IFCBUILDINGSTOREY</td>
</tr>
<tr>
<td>IFCBUILDINGSTOREY</td>
<td>IFCCARTESIANPOINT</td>
<td>IFCCARTESIANPOINT</td>
<td>IFCCARTESIANPOINT</td>
</tr>
<tr>
<td>IFCCARTESIANPOINT</td>
<td>IFCIRCLE</td>
<td>IFCIRCLE</td>
<td>IFCIRCLE</td>
</tr>
<tr>
<td>IFCIRCLE</td>
<td>IFCCLOSED Shell</td>
<td>IFCCLOSED Shell</td>
<td>IFCCLOSED Shell</td>
</tr>
<tr>
<td>IFCCLOSED Shell</td>
<td>IFCCOLUMN</td>
<td>IFCCOLUMN</td>
<td>IFCCOLUMN</td>
</tr>
<tr>
<td>IFCCOLUMN</td>
<td>IFCOMPLEXPROPERTY</td>
<td>IFCOMPLEXPROPERTY</td>
<td>IFCOMPLEXPROPERTY</td>
</tr>
<tr>
<td>IFCOMPLEXPROPERTY</td>
<td>IFCOMPOSI TE CURVE</td>
<td>IFCOMPOSI TE CURVE</td>
<td>IFCOMPOSI TE CURVE</td>
</tr>
<tr>
<td>IFCOMPOSI TE CURVE</td>
<td>IFCOMPOSI TE CURVESEGMENT</td>
<td>IFCOMPOSI TE CURVESEGMENT</td>
<td>IFCOMPOSI TE CURVESEGMENT</td>
</tr>
<tr>
<td>IFCOMPOSI TE CURVESEGMENT</td>
<td>IFCONVERSIONBASEDUNIT</td>
<td>IFCONVERSIONBASEDUNIT</td>
<td>IFCONVERSIONBASEDUNIT</td>
</tr>
<tr>
<td>IFCONVERSIONBASEDUNIT</td>
<td>IFDIMENSIONALEXONENTS</td>
<td>IFDIMENSIONALEXONENTS</td>
<td>IFDIMENSIONALEXONENTS</td>
</tr>
<tr>
<td>IFDIMENSIONALEXONENTS</td>
<td>IFDIRECTION</td>
<td>IFDIRECTION</td>
<td>IFDIRECTION</td>
</tr>
<tr>
<td>IFDIRECTION</td>
<td>IFEXTRUDEDAREAASOLID</td>
<td>IFEXTRUDEDAREAASOLID</td>
<td>IFEXTRUDEDAREAASOLID</td>
</tr>
<tr>
<td>IFEXTRUDEDAREAASOLID</td>
<td>IFCFACE</td>
<td>IFCFACE</td>
<td>IFCFACE</td>
</tr>
<tr>
<td>IFCFACE</td>
<td>IFGEOMETRICREPRESENTATIONCONTEXT</td>
<td>IFGEOMETRICREPRESENTATIONCONTEXT</td>
<td>IFGEOMETRICREPRESENTATIONCONTEXT</td>
</tr>
<tr>
<td>IFGEOMETRICREPRESENTATIONCONTEXT</td>
<td>IFGEOMETRICREPRESENTATIONSUBCONTEXT</td>
<td>IFGEOMETRICREPRESENTATIONSUBCONTEXT</td>
<td>IFGEOMETRICREPRESENTATIONSUBCONTEXT</td>
</tr>
<tr>
<td>IFGEOMETRICREPRESENTATIONSUBCONTEXT</td>
<td>IFCHI S H APEPROFILEDEF</td>
<td>IFCHI S H APEPROFILEDEF</td>
<td>IFCHI S H APEPROFILEDEF</td>
</tr>
<tr>
<td>IFCHI S H APEPROFILEDEF</td>
<td>IFCFACETEDIREP</td>
<td>IFCFACETEDIREP</td>
<td>IFCFACETEDIREP</td>
</tr>
</tbody>
</table>
Appendix II. Examples of the Semantic Intersections and the Product Model

<table>
<thead>
<tr>
<th>$A_1 \cap A_2$</th>
<th>$A_3 \cap A_4$</th>
<th>Product Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFCAPPLICATION</td>
<td>IFCAPPLICATION</td>
<td>IFCAPPLICATION</td>
</tr>
<tr>
<td>IFCARBITRARYCLOSEDPROFILEDEF</td>
<td>IFCARBITRARYCLOSEDPROFILEDEF</td>
<td>IFCARBITRARYCLOSEDPROFILEDEF</td>
</tr>
<tr>
<td>IFCAXIS2PLACEMENT2D</td>
<td>IFCAXIS2PLACEMENT2D</td>
<td>IFCAXIS2PLACEMENT2D</td>
</tr>
<tr>
<td>IFCAXIS2PLACEMENT3D</td>
<td>IFCAXIS2PLACEMENT3D</td>
<td>IFCAXIS2PLACEMENT3D</td>
</tr>
<tr>
<td>IFCBEAM</td>
<td>IFCBEAM</td>
<td>IFCBEAM</td>
</tr>
<tr>
<td>IFCBOUNDINGBOX</td>
<td>IFCBOUNDINGBOX</td>
<td>IFCBOUNDINGBOX</td>
</tr>
<tr>
<td>IFCBUILDING</td>
<td>IFCBUILDING</td>
<td>IFCBUILDING</td>
</tr>
<tr>
<td>IFCBUILDINGSTOREY</td>
<td>IFCBUILDINGSTOREY</td>
<td>IFCBUILDINGSTOREY</td>
</tr>
<tr>
<td>IFCCARTESIANPOINT</td>
<td>IFCCARTESIANPOINT</td>
<td>IFCCARTESIANPOINT</td>
</tr>
<tr>
<td>IFCCIRCLE</td>
<td>IFCCIRCLE</td>
<td>IFCCIRCLE</td>
</tr>
<tr>
<td>IFCCOLUMN</td>
<td>IFCCOLUMN</td>
<td>IFCCOLUMN</td>
</tr>
<tr>
<td>IFCCOMPLEXPROPERTY</td>
<td>IFCCOMPLEXPROPERTY</td>
<td>IFCCOMPLEXPROPERTY</td>
</tr>
<tr>
<td>IFCCOMPOSITECURVE</td>
<td>IFCCOMPOSITECURVE</td>
<td>IFCCOMPOSITECURVE</td>
</tr>
<tr>
<td>IFCCONVERSIONBASEUNIT</td>
<td>IFCCONVERSIONBASEUNIT</td>
<td>IFCCONVERSIONBASEUNIT</td>
</tr>
<tr>
<td>IFCDIMENSIONALEXPONENTS</td>
<td>IFCDIMENSIONALEXPONENTS</td>
<td>IFCDIMENSIONALEXPONENTS</td>
</tr>
<tr>
<td>IFCDIRECTION</td>
<td>IFCDIRECTION</td>
<td>IFCDIRECTION</td>
</tr>
<tr>
<td>IFCEXTRUDEDAREASOLID</td>
<td>IFCEXTRUDEDAREASOLID</td>
<td>IFCEXTRUDEDAREASOLID</td>
</tr>
<tr>
<td>IFCFACEBOUND</td>
<td>IFCFACEBOUND</td>
<td>IFCFACEBOUND</td>
</tr>
<tr>
<td>IFCFACE</td>
<td>IFCFACE</td>
<td>IFCFACE</td>
</tr>
<tr>
<td>IFCFACEOUTERBOUND</td>
<td>IFCFACEOUTERBOUND</td>
<td>IFCFACEOUTERBOUND</td>
</tr>
</tbody>
</table>
References


