A Comparison of Field Mapping Approaches to Message Translation for Data Communication Protocols

JINGYU KIM†1 SUNGWON KANG†1

Abstract: Protocol translation for communication and data exchange over heterogeneous networks and systems has been widely studied. System and network engineers and related researchers proposed various message (or protocol) field mapping approaches for gateway systems, which bridges for data exchange between heterogeneous networks or systems. In our previous study, we proposed a new field mapping approach to message translation for data communication protocols. As a follow-up work, in this paper we develop evaluation perspectives for practicality in message field mapping and compare our approach with the existing message field mapping approaches based on them.

Keywords: Protocol translation, Interoperability, Field mapping, Heterogeneous system, Gateway system.

1. Introduction

In the computer communication world, protocols are "the standards for connection, control, communication, and data transfer by the rules governing the syntax, semantics, and synchronization of communication channels" [2]. For the reliability of data communication, it is important to properly manage data flow and data accuracy [3]. Beyond the confine of a single computer system, interoperability indicates ability to communicate, exchange data, and interpret the information exchanged meaningfully and accurately between heterogeneous systems or network environments [1]. In order to communicate valid and useful data as defined by different systems, correct protocol translation is critical for interoperability. By implementation translation methods, gateway systems provide bridges for data exchange between heterogeneous networks or systems. Various message field mapping approaches for gateway systems have been proposed by system and network engineers and related researchers in the past [2, 5, 7, 8].

In our previous study [1], we proposed a new message field mapping approach to translation methods for data communication protocols. Our previous study, the semantic translation method, consists of semantic translation preprocessing, field mapping and field data conversion, and semantic confirmation and adjustment phases [1]. The field mapping and field data conversion phase is our approach of this paper. As a follow-up work, in this paper we develop evaluation perspectives for practicality in message field mappings and compare our approach with the existing message field mapping approaches based on them.

2. Our approach to message field mapping

Figure 1 describes our procedure for message field mappings of our approach in [1]. It consists of two main steps: (1) field mapping description table step and (2) field data conversion algorithm step. The procedure generates a field data conversion function, which converts source field data to target field data.

†1 Korea Advanced Institute of Science and Technology (KAIST)

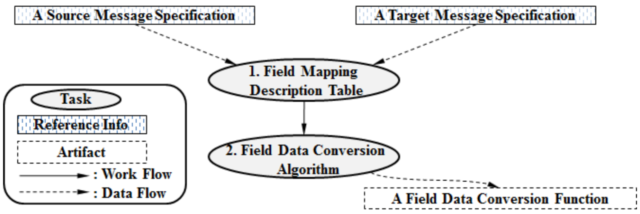

Figure 1. Our procedure for message field mapping [1]

2.1 Field mapping description table step

In this step, gateway engineers, who design gateway software, create a field mapping description table from mapped field information. A field mapping description table is a description template for mapping information between message fields. Gateway engineers analyze field information such as field type, purpose, description range, data type, information description unit, and default field data and fill out the field mapping description table. The table consists of the following seven parts: field mapping name, field mapping type, mapping cardinality, source type, target type, mapping condition, and mapping constraint [1]. To create the table, the following substeps are taken:

Substep 1) *Identify mapped fields*: Fields are identified. Gateway engineers create a field mapping description table with the composition of mapped source and target field names. The names would be discriminable for a field data conversion function in a gateway application.

Substep 2) *Select a field data mapping type*: One of three field data mapping types [1] and one of the seven mapping cardinalities [1] are identified.

Substep 3) *Record source protocol type, source message type and source field name*: The associated source protocol type, source message type, and source field name in the field mapping description table are recorded.

Substep 4) *Record target protocol type, source message type,* *target field name, and default field data*: An associated target protocol type, target message type, and target data field name are recorded in the table. Especially in the target type part, gateway engineers additionally record default field data. The default field data is preparation field data for mapping conditions and constraints in the substeps 5 and 6. The default field data should be neutral for the target data field [6], and gateway engineers choose one of field data from the target data field that does not indicate specific information. (Default field data for the homogeneous mapping type [1] is not recorded.)

Substep 5) *Define mapping conditions*: Mapping conditions are defined. The mapping condition is field data conversion or matching rule to describe source field data in the form of target data fields. (The data conversion or matching rules for the homogeneous mapping type [1] is not defined.)

Substep 6) *Define mapping constraints*: Mapping constraints are defined. The mapping constraints are conditions to describe source field data for target fields. The constraints decide whether field data converted by mapping conditions in substep 5 is describable in a target data field. (The mapping constraints for the homogeneous mapping type [1] are not defined).

Table 1 shows a field mapping description table for a heterogeneous and numerical computational type [1]. It describes information for mapping *Moving Distance* field and *Moving Time* field in a VMF Observation Report message to *Speed* field in a Link-16 Enemy Identification message [3, 4, 5]. In Table 1, *Speed* field data is calculated with field data from a *Moving Distance* field and *Moving Time* field, and a mapping constraint is defined in order to deal with the description range of *Speed* field.

Table 1. An example of a field mapping description table [1]

Field mapping name	*Moving Distance; Moving Time to Speed*	
Field data mapping type	Numerical computational	
Mapping cardinality	2:1 mapping	
Source type	Protocol / Message type	*VMF / Observation Report*
	Field_1	*Moving Distance*
	Field_2	*Moving Time*
Target type	Protocol/ Message type	*Link-16 / Enemy Identification*
	Field_1	*Speed*
	Default field data_1	/*Undefined *Speed* Field data*/ *1111111110*
Mapping condition_1	/*Converting 'kilometer' into 'mile' per hour */ *Speed* Field = (*Moving Distance* Field/*Moving Time* Field) / 1.6	
Mapping constraint_1	/* Range of *Speed* field description*/ IF (*Speed* Field > 2^{11}-1) *Speed* Field = Default field data_1	

2.2 Field data conversion algorithm step

In this step, gateway engineers generate a field data conversion algorithm from a field mapping description table. In this step, gateway engineers design pseudo code for each part of the table. The table is designed for gateway engineers to be read from top to bottom in order to make straight-line and statement-centered codes in field data conversion functions [6]. The field data conversion algorithm from this step can be directly implemented to a field data conversion function in gateway applications by system developers. To generate an algorithm from a field mapping description table, the following substeps are taken:

Substep 1) *Declare input fields*: The input source fields are defined.

Substep 2) *Declare output fields*: The output target fields are defined.

Substep 3) *Designate a default field value for output field data*: It is defined for the case when the converted field data is not valid value in the assigned output field.

Substep 4) *Define mapping conditions*: The mapping conditions of the assigned input and output fields are defined in a described order in the table.

Substep 5) *Define mapping constraints for output field data*: The mapping constraints of the converted source field data for the output field data are defined in a described order in the table.

Figure 2 presents the field data conversion algorithm generated from the field mapping information in Table 1.

Type: Moving Distance field; Moving Time field to Speed field

```
Input:    Moving Distance_VMF; Moving Time_VMF …(Step 1)

Output:   Speed_Link-16…… (Step 2)
/* undefined Speed field data*/
Speed_default _field_data = 1111111110…… (Step 3)

/* Mapping Condition 1: conversion from kilometer per hour to
   mile per hour */
Speed_ Link-16 = (Moving Distance_VMF/Moving Time_VMF) /
   1.6…… (Step 4)

/* Mapping Constraint 1: check description range of Speed Field */
if Speed_VMF > 2^11-1 then…… (Step 5)
        Speed _Link-16 = Speed _default_field_data
end if
```

Figure 2. Field data conversion algorithm that converts *Moving Distance* and *Moving Time* to *Speed* field [1]

3. Existing approaches to message field mapping

In this section, we give a brief description of the existing approaches to message field mapping. They are the rule-based approach [7], the table-based approach [5], and the ontology-based approach [8].

3.1 The rule-based message field mapping approach

This approach was first developed to mediate and integrate heterogeneous data sources on Web and databases as one of the schema matchers. It has been applied later to heterogeneous network messages or protocols. It defines corresponding message fields as textual rules and logic representations [7]. Figure 3 shows examples of mapping rules between the *Speed* field (in Link-16 messages) and the *Unit Speed* field (in VMF messages).

Match (*Speed*, *Unit Speed*):
- Type of field data
  (*Speed*.Numerical field; *Unit Speed*.Numerical field)
- Degree of unit speed
  (*Speed*. Mile per hour; *Unit Speed*. Kilometer per hour)
- Coverage of mapping
  (*Speed*.2^{11}; *Unit Speed*.2^{10})

Figure 3. Examples of mapping rules

A mapping rule basically defines a pair of corresponding fields with their properties. Two corresponding fields may have field data mismatches; for example, *Speed* field is described in miles per hour, and *Unit Speed* field in kilometers per hour as shown in Figure 3. A mapping rule stores a list of property mappings to provide mapping conditions of the corresponding source and target message fields for gateway developers.

3.2 The table-based message field mapping approach

This is the most practical and widely used technique for message field mappings. Table 2 describes a mapping table for VMF and Link-16 messages based on this approach.

Table 2. Mapping table for VMF and Link-16 messages

Source Message		Target Message			
Type	Field	Index	Type	Field	Mapping Condition
J2.0	Elevation 25FT	1.1	K05.1	Elevation Feet	NCR
N/A		1.2	K05.1	GPI	-
J2.0	Latitude	1.2.1	K05.1	Unit Latitude	CR
J2.0	Longitude	1.2.2	K05.1	Unit Longitude	CR
J2.0	Speed	1.3	K05.1	Unit Speed	CR

Basically, this approach is designed to store a list of target message fields and to assign source fields at each row of target field [5]. The advantage of this approach is that it simplifies field mapping information at target message structure in the same table so that no additional target message structure storage is necessary. As shown in Table 2, the mapping table stores index numbers of target message for gateway systems to refer to and interpret the message structure with mapped source or target fields at the same time. Each corresponding source and target field has a mapping condition such as CR (Conversion is Required) and NCR (No Conversion is Required) in order for a translation application to provide interpretation of the necessary fields [5].

3.3 The ontology-based message field mapping approach

This approach abstracts corresponding source fields and target fields as entities of an ontology model, and it associates the entities based on similarity of corresponding to their names, data coverage, and properties. For example, the Artemis project [8] proposed and developed the ontology-based message exchange framework for the interoperability among Health Level Seven (HL7) message based healthcare systems. Figure 4 describes *Speed* and *Unit Speed* field mappings as in Bicer et al. [8]. In Figure 4, a bidirectional mapping between *Speed* and *Unit Speed* fields is established because the data type and description attributes of the two fields are similar to each other's.

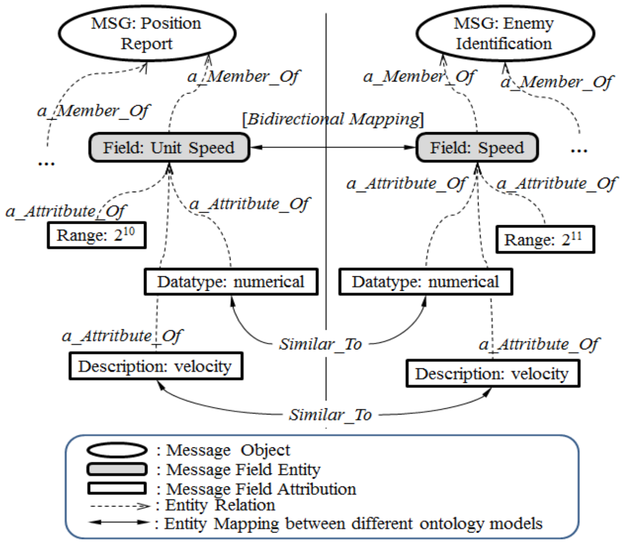

Figure 4. Message field mapping between two ontology models

The advantage of the ontology-based mapping approach is that it can be supported by ontology tools. Therefore, the entity mappings are easily automated [8]. Moreover, types of mapping cardinality and mapping conditions assigned entities are fully supported due to traditional characteristics of ontology models such as association representations among entities.

4. Comparison

This section compares our approach with the message field mapping approaches introduced in Section 3 from the viewpoint of the practicality. An approach is regarded as practical if it covers aspects related to message translation such as heterogeneous message structures and mapping cardinality establishment. For comparison, we adopt the following evaluation perspectives:

- Structure matching - Ability to handle mismatches of field structure or order.
- Property matching - Ability to associate properties of different message fields.
- Field data matching - Ability to associate field data between different message fields.
- Cardinality - Ability to support a variety of cardinalities [1] among message fields.
- Dynamic matching - Ability to support field data matching

in dynamic ways; for example, a source field provides field data to one of the mapped target fields in 1:*n* mapping cardinality according to the semantics of the source field data.

- Tool support - Ability to support automation of field mappings.

An analysis of each approach with the evaluation perspectives shows that no approach has problem in structure matching but each approach supports different perspectives as explained below.

The rule-based message field mapping approach: It is strong in property matching and cardinality because property associations and multiple associations such as many-to-many mapping relation can be described in textual rules or logic representations. However, there are difficulties in dynamic matching, field data matching, and tool support because this approach concerns only schema-level heterogeneity [1, 7]. No tool has been implemented for it.

The table-based message field mapping approach: This approach has an advantage in that it can be simply implemented by a database schema. However, our comparison indicates that this approach has many limitations in property matching, dynamic matching, and tool support because it only considers schema-level heterogeneity between message structures [1]. Furthermore, the mapping table only supports the data matching type [1] for field data matching and can support 1:*n* or *n*:1 cardinality (not *n*:*m* cardinality); for example, a same source field is stored at multiple rows of different target fields.

The ontology-based message field mapping approach: In view of property matching, field data matching, cardinality, and tool support, this approach is strong while the other approaches should depend on engineer's intuition. However, it does not support a dynamic matching because it handles schema-level field mappings but does not consider complicated field data matching of message level such as selective or dependent mapping types [1].

Our approach: It focuses on generating a field data conversion function and is strong in property, field data, dynamic matching, and cardinality because it can consider heterogeneity of property and field data for a field data conversion algorithm. For the dynamic matching, our approach can freely define mapping conditions and constraints for each target field. Moreover, the field data conversion function can freely define input and output parameters for a variety of cardinalities. On the other hand, with no tool support, modeling field associations takes a lot of efforts.

Table 3 shows the result of the comparison. From the comparison, the ontology-based message field mapping approach seems the most outstanding solution because it can be easily automated by ontology modeling and emerging tools. However, it is not suitable for message translation because mapping associations are fixed at schema level with consequence that complicated field data matching of message level [1] cannot be handled with it.

Except for automation of field mapping, our approach fully supports all the evaluation perspectives. Having no tool support poses difficulty to gateway engineers in identifying field mappings between different message (or protocol) specifications although it does not restrict message translation. We plan to overcome this limitation of our approach through further research.

Table 3. Comparison Result

Evaluation Perspective	Rule	Table	Ontology	Our approach
Structural matching	+	+	+	+
Property matching	+	-	+	+
Field data matching	-	P	+	+
Cardinality	+	P	+	+
Dynamic matching	-	-	-	+
Tool support	-	-	+	-

-: Not Supported, +: Fully Supported, P: Partially Supported

5. Conclusion

In this paper, we compared our field mapping approach to message translation methods introduced in [1] with the existing message field mapping approaches. For the comparison, we established evaluation perspectives of the practicality. Through the comparison, we identified that our approach is more practical in property, field data, and dynamic matching as well as a variety of cardinalities for message translation compared to the existing message field mapping approaches.

For future work, we plan to develop a tool to automate our field mapping establishment procedure.

References

1) Kim, J., Kang, S., Lee, J., and Choi, B. W.: A semantic translation method for data communication protocols, Journal of Systems and Software (2012), http//dx.doi.org/10.1016/j.jss.2012.06.018.
2) Singh, K. and Schulzrinne H.: Interworking between SIP/SDP and H.323, Proceedings of the 1st IP-Telephony Workshop. USA, pp.1-21 (2000).
3) Department of Defense: Interface Standard TADIL-J MIL STD 6016A, JEBCA Teston VA, (1999).
4) Department of Defense: Variable Message Format (VMF) MID-STD-6020 Appendix D (Test Edition), VMF TIDP-TE Reissue 5, USA, (2002).
5) Department of Defense: Interface Standard Data forwarding between Tactical Data Links (TDLs), MIL-STD- 6020, Appendix D-Link-16 and VMF, USA, (2005).
6) McConnell, Steve: Code Complete, Microsoft Press, (2004).
7) Saake, G., Sattler, K., Conrad. S.: Rule-based schema matching for ontology-based mediators, Journal of Applied logic, Vol.3, No.2, pp.253-270 (2004).
8) Bicer, V., Laleci, C., Dogac A., Kadak Y.: Artemis Message Exchange Framework: Semantic Interoperability of Exchanged Messages in the healthcare domain, ACM SIGMDO Record, Vol.34, No.3, pp.71-76 (2005).

Acknowledgment This research was supported by the KAIST High Risk High Return Project (HRHRP).