Appl. Math. Inf. Sci. **9**, No. 1L, 165-170 (2015)

Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/091L21

Data Scalability Design for Multi-Satellite Management based on E-R Model

Jungkeun Park[1], *Trung Kien Nguyen*[1], *Do Dang Thach*[1], *Ok-Chul Jung*[2] *and Chang-Joo Moon*[1,*]

[1] Department of Aerospace Information Engineering, Konkuk University, Seoul 143-701, Korea
[2] Korea Aerospace Research Institute, Daejeon 305-806, Korea

Received: 22 Nov. 2013, Revised: 23 Mar. 2014, Accepted: 24 Mar. 2014
Published online: 1 Feb. 2015

Abstract: We propose a Multi-Satellite Data Model (MSDM) for the efcient database management of multi-satellite ground control systems; the paper also explains a satellite control data management method using MSDM. The MSDM can manipulate various data attributes without modifying the entity structure by adding tuples instead of new attributes to tables when a new data attribute is required for a specic satellite. Data for retired satellites are only used for analysis purposes. We propose a backup method that selects data for retired satellites by using SQL and backs them up in an analyzable table. In addition, we propose an SQL statement for nding candidate attributes that should be added or deleted according to continuous satellite launches and retirements. For the metadata specication in the attribute table, XML schema is dened and the semantics and characteristic information of the extended data are described using XML tag information.

Keywords: Multi-satellite Management System, Data Extensibility Model, Metadata, XML

1 Introduction

In the past, a dedicated satellite ground control system was developed for each new satellite launch. Recently, multi-satellite ground control systems are developed that can control multiple satellites simultaneously from a single ground control system [1,2]. In a multi-satellite ground control system, common functions to all the satellites under management are re-used and only new functions that are required to manage a newly launched satellite are added. This can efficiently save time and cost because a new satellite ground control system can be equipped by modifying part of the currently working system. One example is the SCOS-2000 by the European Space Operations Centre (ESOC) of the European Space Agency (ESA) [3,4].

A database in a satellite ground control system stores a set of core data required for satellite operations. A multi-satellite ground control system handles different data for multiple satellites with different missions. Management data for each satellite contain common data for all satellites but they also have specific data required for a specific satellite. New data items have to be added or existing items have to be modified according to successive satellite launches. Parts of the data items are not used any more when a corresponding satellite reaches the end of its life. However, in mission-critical and stability-demanding satellite operations, there are many risk factors when changing the database schema of currently operating satellite ground control systems whose database designs have already been completed and verified.

Two types of data modeling are possible with respect to database management in multi-satellite ground control systems, as shown in Figure 1. In this paper, because access to detailed information of data entities in existing multi-satellite ground control systems is not possible, we present only conceptual specifications by assuming an arbitrary entity of a MissionPlan.

In Figure 1(a), a separate entity is defined to store a mission plan data for each satellite. In this case, m entities with similar structure are defined where m is the number of satellites. Each entity contains duplicate attributes from Attr_1 to Attr_n for mission plan; these attributes are commonly performed by all satellites. Attributes ExAttr of each entity store data for the mission plan that are performed by specific satellites. This structure increases the data system complexity and makes it difficult to

* Corresponding author e-mail: cjmoon@konkuk.ac.kr

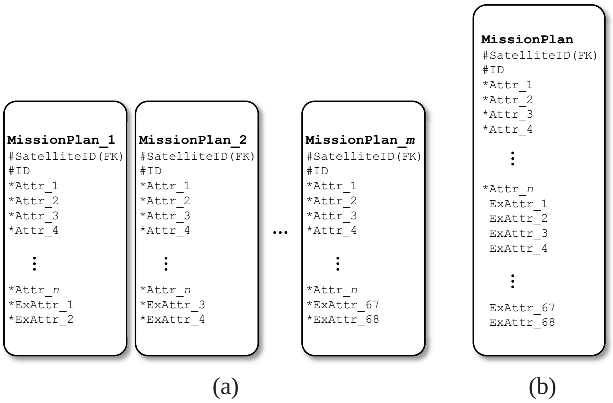


Fig. 1: Data model for attribute extension according to satellite launches: (a) a separate entity is defined to store a mission plan data for each satellite, and (b) multiple satellite mission plan data are managed by a single entity.

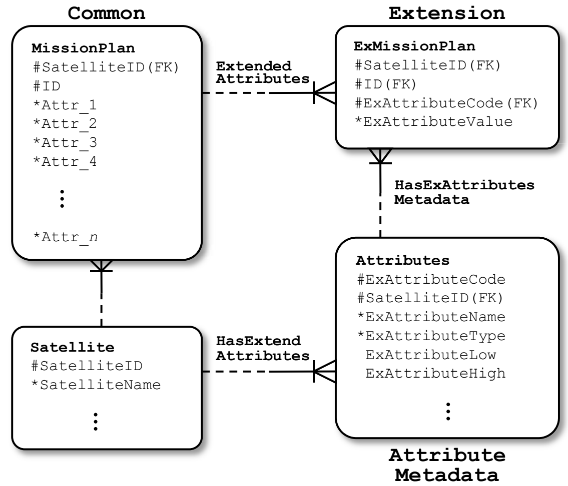


Fig. 2: Multi-satellite data model is modeled using E-R diagram. Attr's in the Common have to be managed by all satellites and the dedicated data for the new satellite are stored in the Extension entity.

efficiently query information for integrated management since data with the same characteristics are scattered.

In Figure 1(b), multiple satellite mission plan data are managed by a single entity. In this case, according to a new satellite launch, attribute ExAttr has to be added to the entity to store new mission data specific to the newly launched satellite. However, this addition causes an increase of the NULL value in the database because other satellites may have a NULL value in the newly added attribute ExAttr. Furthermore, this attribute is not deleted from the entity when a satellite retires. This also causes the generation of NULL values.

Consequently, general database modeling cannot be applied to multi-satellite ground control systems. In database modeling for multi-satellite systems, the following issues have to be considered.

1) Fixed database model independent of the type and number of satellites under management
2) Attribute addition/deletion according to satellite launch/retirement
3) Prevention of continuous NULL value generation in attribute
4) Data backup for retired satellites

In this paper, we propose a Multi-Satellite Data Model (MSDM) for the efficient database management of multi-satellite ground control systems considering the above four issues; we also explain the satellite control data management method using MSDM. MSDM can manipulate various data attributes without modifying the entity structure by adding tuples instead of new attributes to tables [5,6,7] when a new attribute is required for a specific satellite. In other words, the attributes required for a newly launched satellite can be freely added or deleted without modifying the database schema. Metadata for these attributes are stored in a separate attribute table. Thus, attribute values can be clearly understood even though various attributes for a large number of satellites exist. For the metadata specification in the attribute table, XML schema [8] is defined and semantics and characteristic information of the extended data are described using XML tag information. Data for retired satellites are only used for analysis purposes. In this paper, we propose a backup method that selects data for retired satellites by using SQL and backs them up to an analyzable table. In addition, we propose an SQL statement for finding candidate attributes that should be added or deleted according to continuous satellite launches and retirements.

The remainder of this paper is organized as follows: In Section 2, we explain the Multi-Satellite Data Model. In Section 3, we present the data management methods according to satellite lifecycle. Section 4 includes specification of attributes. Finally, in Section 5, we provide a conclusion and make suggestions for further work.

2 Multi-Satellite Data Model

Figure 2 represents MSDM as an E-R diagram. The tables in Figure 1 are modeled as an E-R diagram in Figure 2 to satisfy the four issues in the previous section. Notation # represents the primary key (PK) of an entity and FK denotes the foreign key. If notation # and PK are used together, it means that the FK column is used to consist PK. Notation * indicates the "not null" attribute. Relationships between entities are represented by the Barker method based on Crow's Foot Notation [9].

Common attributes for all satellites belong to the Common entity. In other words, Attr's in the Common entity are data items that have to be managed by all satellites. PK of the Common entity is the combination of SatelliteID and ID attribute. SatelliteID identifies which satellite the attributes belong to and ID uniquely identifies a data tuple. In a multi-satellite ground control system, depending on entities, ID attribute can be expressed in different forms such as mission_id or timestamp; however a simple ID value will be used in this paper in order to embrace entire entities. Structurally, the structure of Common entity is the same entity used in general database modeling. Domain specialists and database modelers determine attributes that belong to the Common entity. The problem of determining specific attributes of the Common entity is out of scope of this paper.

When a new satellite is added to a multi-satellite ground control system, common data are stored in the Common entity and dedicated data for the new satellite are stored in the Extension entity. For example, a combination of SatelliteID, ID, and ExAttributeCode becomes PK in the ExMissionPlan entity and the ExAttributeValue contains the attribute value corresponding to the ExAttributeCode attribute. So, the additional data of the ExAttributeCode attribute for a satellite with SatelliteID and a mission plan with ID are stored in ExAttributeValue. In this structure, a tuple is added instead of an attribute when a new data item is required in the Extension entity. Thus, various data attributes can be added with extensibility without modifying the ExMissionPlan entity structure.

Figure 3 show sample data in tables that are implemented from the entities in Figure 2. For a satellite KS3, common mission plan data are stored in columns A1∼An in the MissionPlan table while additional extended attributes EA1's and EA2's columns and values are stored in the ExMissionPlan table as a row. In the case of a mission with ID value KS3_145, extended attribute ExtendAt1 column has a value of 0.95 and another attribute ExtendAt2 has a value of 1.26. In the same way, extended attributes and their values are added to another mission KS3_146. This can also be applied to satellites KS5 and KS6. This structure is efficient when types and numbers of data items to be added in the future are unpredictable.

Because it is highly erroneous to input string typed names of satellites and additional attributes in the ExMissionPlan table, coded id values are used. The Attributes entity is required since various types of attributes are included in extended entities. This entity stores meta information for the attributes included in the extended entities such as satellite id, attribute name, type, range of value, unit of value, etc. If additional attributes are required according to a new satellite launch, attribute information is registered in the Attribute entity and attribute values are stored in the extended entity with attribute references. The Satellite entity stores the basic information of the satellite. SatelliteID information used by all entities references this entity.

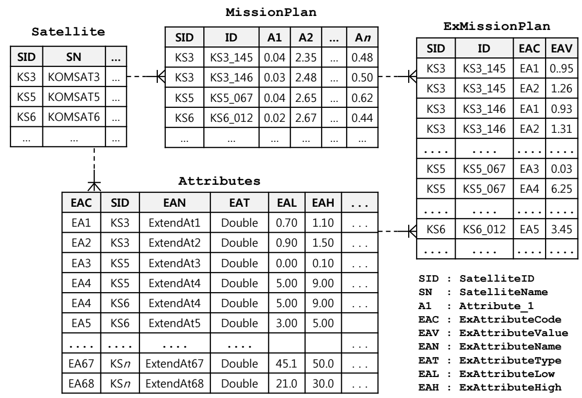


Fig. 3: Sample data of multi-satellite data model. Some attributes names are abbreviation because of table space problem. The full names of the attributes are located at lower right-hand corner.

(a)

```
select A1, A2, A3
from MissionPlan
where SID = 'KS3'
and ID = KS3_145

A1   A2   A3
---- ---- ----
-
0.95 0.04 2.35
```

(b)

```
select EAC, EAV
from ExMissionPlan
where SID = 'KS3'
and ID = KS3_145

EAC EAV
--- ----
EA1 0.48
EA2 1.26
```

Fig. 4: All attributes checking SQL for retired satellites: (a) sql query common data of KS3 satellite and (b) sql query extended data of KS3 satellite.

3 Data Management for Satellite Lifecycle

In this section, methods for backing up the MSDM data of retired satellites and modifying common entities according to satellite launches and retirements are presented. It is generally considered that the lifespan of a satellite is around five years. Therefore, satellite ground control data also will not be generated after this period. Since data for retired satellites will not be used anymore, they should not be stored in the same table with other satellites' control data. For example, the data related to the satellite KS3 in the MissionPlan and ExMissionPlan tables should be moved to other tables after KS3 is retired. The two SQL statements in Figure 4 are the queries for checking all the attributes of KS3.

The KS3 satellite has five attributes for its mission plan; A1, A2, A3, EA1, and EA2. If the condition of the

```
select EAC, COUNT(DISTINCT SID)
from ExMissionPlan
group by EAC

EAC COUNT(DISTINCT SID)
--- -------------------
EA1      2
EA2      3
EA3      2
EA4      5
EA5      1
```

Fig. 5: SQL for choosing candidate common attributes. The results for the query can tell whether an extended attribute can be candidates for common attributes.

```
select SID, COUNT(A1) CN1, COUNT(A2) CN2,
       COUNT(An) CNN
from MissionPlan
group by SID

SID    CN1    CN2    CNN
--- ------ ------ ------
KS3      2      3      2
KS5      0      3      2
KS6      0      2      3
```

Fig. 6: SQL for choosing candidate common attributes for deletion. Common attributes are not fixed once they are determined but change gradually as satellites retirement and launch.

'where' statement is the same, the two queries can obtain all the attribute values of KS3 because the MissionPlan and ExMissionPlan tables have the same SID and ID attributes. The data resulting from this query are removed from the MissionPlan and the ExMissionPlan tables and are stored in a separate analysis table. These queries are frequently used to control satellites. Information on retired satellites in the Satellite and Attributes entities are not removed immediately because the number of tuples is small and the extension attribute can be reused for a new satellite in the future.

Since most satellites are launched in series, recently launched satellites often include features similar to those of previous satellites. So, the extension attributes can be reused continuously for a series of satellites after they are first added by a newly designed satellite. In Figure 3, the attribute EA4 in the Attributes table is used not only in the satellite KS5 but also in KS6. If the attribute EA4 will be used continuously in future satellites and all the satellites launched before KS3 are retired, EA4 becomes a candidate that can be moved to the MissionPlan entity. The same decision process has to be performed for EA4 in order for it to become a common attribute of the MissionPlan entity.

The SQL statement in Figure 5 can check the extension attributes that are to be used by all the currently controlled satellites. The results for this query can tell whether an extension attribute can be candidates for common attributes. The count number of EA4 is five in the results for the query. This result means that EA4 attributes are used all satellite ground control so the EA4 attribute can be a candidate of the common attribute. If the attribute A1 is used until KS3 and after that is replaced by the extension attribute EA4, so that no currently operating satellite uses A1, then the deletion of the attribute A1 in the MissionPlan table should be considered at the time when the satellite KS3 retires.

The SQL statement in Figure 6 is a query for choosing candidates for deletion among common attributes in the MissionPlan table. Common attributes in the Common table are not fixed once they are determined but change gradually as the number of retired satellites increases and new satellites are launched. However, it is a very important task to change table structures. Thus, this task should be performed after a considerable number of common attributes and deletion candidates have accumulated. Considering the lifespan of satellites, this process may take a very long period. So, this process will be performed more infrequently and must be more stable than the process of changing table structures at every satellite launch time.

4 Specification of Attributes

The Attributes table in Figure 3 manages the metadata of the satellite control data. Data in this table are required in order to understand the satellite control data and to maintain the data integrity. These metadata are shared by applications in multi-satellite ground control systems and enable precise data processing. To improve data extensibility, it is important for applications to precisely understand and process data with different characteristics of multiple satellites in multi-satellite ground control systems. For data sharing, metadata are specified by using an XML schema. This schema is shared by applications to precisely understand tag and data information of XML documents that follow the schema. Figure 7 gives the definition of metadata for the Attribute table and a sample document that follows the schema.

5 Conclusion

In multi-satellite ground control systems, there are common attribute and extension attribute data; management data items are changed according to satellite launches and retirements. MSDM can effectively manage flexible data attributes and data of multi-satellite ground control systems without changing database schema. In

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ExAttributeInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ExAttributes"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExAttributes">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Description"/>
        <xs:element ref="ExAttributeName"/>
        <xs:element ref="ExAttributeCode"/>
        <xs:element ref="ExAttributeType"/>
        <xs:element ref="ValueRange"/>
        <xs:element ref="DefaultValue"/>
        <xs:element ref="Nullable"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Description"
            type="xs:string"/>
  <xs:element name="ExAttributeName"
            type="xs:string"/>
  <xs:element name="ExAttributeCode"
            type="xs:string"/>
  <xs:element name="ExAttributeType"
            type="xs:string"/>
  <xs:element name="ValueRange">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="LowValue"/>
        <xs:element ref="HighValue"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DefaultValue" type="xs:string"/>
  <xs:element name="Nullable" type="xs:boolean"/>
  <xs:element name="HighValue" type="xs:string"/>
  <xs:element name="HighValue" type="xs:string"/>
</xs:schema>
```

(a)

```
<ExAttributeInfo>
<?xml version="1.0" encoding="UTF-8"?>
<ExAttributeInfo xmlns:xsi="http://www.w3.org/2001/
  XMLSchema-instance" xsi:noNamespaceSchema
  Location="ExAttrschema.xsd">
  <ExAttributes>
    <Description>camera rotation angle</Description>
    <ExAttributeName>ro_angle</ExAttributeName>
    <ExAttributeCod>ra</ExAttributeCod>
    <ExAttributeType>double</ExAttributeType>
    <ValueRange>
      <LowValue>0</LowValue>
      <HighValue>70</HighValue>
    </ValueRange>
    <DefaultValue>0</DefaultValue>
    <Nullable>F</Nullable>
  </ExAttributes>
</ExAttributeInfo>
```

(b)

Fig. 7: Specification of Attributes with XML: (a) is an XML schema in order to share metadata of Attributes data. This schema is shared by applications of ground control system. (b) is an example of XML document that follow XML schema of (a).

MSDM, if new data attributes are required for a specific satellite, the attributes are not added to tables; rather, tuples are added. Thus, various data attributes can be managed without changing entity structures.

Metadata for attributes under management are stored in the Attribute table. So, attribute values can be clearly understood even though there are various attributes for many satellites. Based on MSDM, methods for effectively backing up data and modifying schema of common entities are described. The methods proposed in this paper enable us to effectively operate and manage many satellite data in multi-satellite ground control systems.

Acknowledgement

This paper was written as part of Konkuk University's research support program for its faculty on sabbatical leave in 2011.

References

[1] Ok-Chul Jung, Hae-Dong Kim, Su-Jin Choi, Dae-Won Chung, Analysis and Design of the Generic Mission Operations System, Aerospace Technique, **8**, 127-132 (2009).

[2] Hee-Sook Mo, Ho-Jin Lee, and Seong-Pal Lee, Development and Testing of Satellite Operation System for Korea Multipurpose Satellite-I, ETRI Journal, **22**, 1-11 (2000).

[3] Nestor Peccia, SCOS-2000 ESA's Spacecraft Control for the 21st Century, Ground System Architecture Workshop, Keynote Session, (2003).

[4] Fischer, M. Scholtz, A.L., Design of a Multi-mission Satellite Ground Station for Education and Research, Advances in Satellite and Space Communications (SPACOMM), 2010 Second International Conference on, 58-63 (2010).

[5] Mi-Young Choi, Chang-Joo Moon, Ki-Seok Park, Doo-Kwon Baik, An Enterprise Master Data Model based on the Data Taxonomy based on Their Origin, 2010 International Conference on Enterprise Information Systems and Web Technology(EISWT), 34-41 (2010).

[6] Mi-Young Choi, Chang-Joo Moon, Doo-Kwon Baik, Young-Jun Wie, Joong-Hee Park, Interoperability between a Relational Data Model and an RDF Data Model, Sixth International Conference on Networked Computing and Advanced Information Management (NCM), 335-340 (2010).

[7] Mi-Young Choi, Chang-Joo Moon, Doo-Kwon Baik, Young-Jun Wie, Joong-Hee Park, The RDFS Mapping for Recursive Relationship of Relational Data Model, IEEE International Conference on Service-Oriented Computing and Applications (SOCA), 1-6 (2010).

[8] Paul V. Biron and Ashok Malhotra, editors. XML, Schema Part 2: Datatypes. World Wide Web, Consortium, (2000).

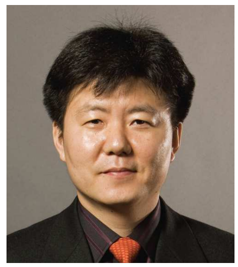
[9] http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#basicprinciples (2011).

Jungkeun Park is currently an Associate Professor of the Department of Aerospace Information Engineering at the Konkuk University. He received his B.S. and M.S. degrees in Electrical Engineering from the Seoul National University, Korea in 1997 and 1999, respectively. Dr. Park received his Ph.D. in Electrical Engineering and Computer Science from the Seoul National University in 2004. His current research interests include embedded real-time systems design, real-time operating systems, distributed embedded real-time systems and multimedia systems.

Trung Kien Nguyen received his B.S. degrees in Electrical Engineering from Hanoi University of Science and Technology, Viet Nam in 2009, and M.S. degree in Aerospace Information Engineering from the Konkuk University, Korea in 2013. His current research interests include embedded real-time systems, unmanned aerial vehicle navigation and control, and robotics.

Do Dang Thach received his B.S. degrees in Computer Engineering from Ho Chi Minh city University of Technology, Viet Nam in 2011, and M.S. drgree in Aerospace Information Engineering from the Konkuk University, Korea in 2013. He is currently a researcher of the Intelligent Vehicle and System Technology (INVEST) research center at the Konkuk University, Korea. His current research interests include embedded real-time systems, real-time operating systems and concurrent programming.

Ok-Chul Jung received the B.S. and M.S. degrees in aerospace engineering in 2003 and 2005 from Chonbuk National University, Jeonju, South Korea. In 2005, he joined the Satellite Ground Control Technology Team in Electronics and Telecommunications Research Institute. He is currently a research engineer in LEO Satellite Mission Operations Department in Korea Aerospace Research Institute. His research interests include space flight dynamics and control, and satellite operations engineering.



Chang-Joo Moon is currently a Professor of the Department of Aerospace Information Engineering at the Konkuk University. He received his B.S. and M.S. degrees in Computer Science from the Korea University, Korea in 1997 and 1999, respectively. Dr. Moon received his Ph.D. in Computer Science from the Korea University in 2004. His current research interests include embedded real-time systems, data engineering, database security, component-based software engineering, and distributed component system.