|  |
| --- |
|  |

Data Model Review

Approval

|  |  |
| --- | --- |
| Title Data Model Review | |
| Issue Number 0 | Revision Number 0 |
| Author Zlobin, Veniamin | Date 01/10/2018 |
| Approved By | Date of Approval |
|  |  |

Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| Reason for change | Issue Nr. | Revision Number | Date |
|  |  |  |  |

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Issue Number  0 | Revision Number  0 | | |
| Reason for change | Date | Pages | Paragraph(s) |
|  |  |  |  |

Distribution

|  |
| --- |
| Name/Organisational Unit |
|  |

Table of contents:

[1 Activity overview 4](#_Toc525924686)

[1.1 Applicable and reference documents 5](#_Toc525924687)

[2 mission data 5](#_Toc525924688)

[2.1 Summary 6](#_Toc525924689)

[3 Mapping to MP model 7](#_Toc525924690)

[3.1 Method 7](#_Toc525924691)

[3.2 Data mapping issues / examples 8](#_Toc525924692)

[3.2.1 Data validation - addressed 8](#_Toc525924693)

[3.2.2 Count attribute in lists – meta data 8](#_Toc525924694)

[3.2.3 Complex types in arguments - example 8](#_Toc525924695)

[3.2.4 ESA “profile” – complex argument 9](#_Toc525924696)

[3.2.5 Arguments in requestResponse - EnMAP 11](#_Toc525924697)

[3.2.6 Complex types in arguments – EnMAP 11](#_Toc525924698)

[3.2.7 RelativeValidityEvents – Expression example 12](#_Toc525924699)

[3.2.8 Window of RelativeValidityEvents - example 15](#_Toc525924700)

[3.2.9 Validity Period for a Planning Request 15](#_Toc525924701)

[3.2.10 insertOrDeleteFlag 15](#_Toc525924702)

[3.2.11 Complexity of the model - comment 15](#_Toc525924703)

[3.2.12 GAIA executionTime 15](#_Toc525924704)

[3.2.13 releaseTime 17](#_Toc525924705)

[3.2.14 executionTime/releaseTime in constraints 17](#_Toc525924706)

[3.2.15 Multiple errors in requestResponse 18](#_Toc525924707)

[3.2.16 XMM-Newton Revolution - example 18](#_Toc525924708)

[3.2.17 XMM-Newton Orbital Phase - example 19](#_Toc525924709)

[3.3 MAL issues 20](#_Toc525924710)

[3.3.1 MAL limitation – no polymorphism 20](#_Toc525924711)

[3.3.1.1 Technical reasons and a limited workaround 21](#_Toc525924712)

[3.3.1.2 Polymorphism in MP model 21](#_Toc525924713)

# Activity overview

1. Analyse planning interfaces (PLIDs) of space missions.
   1. Focus on PI – SOC and SOC – MOC.
   2. Draw UML data models.
   3. Record operations.
2. Map the UML data models to MP model. For each mission entity/field find an MP counterpart.
3. Map the operations to MP.
4. PoC development – MO adapter for MPSF.
   1. MPSF is ESA planning tool.
   2. Demonstrate how MP services control MPSF.

How PI, SOC, MOC relate – Figure 1.

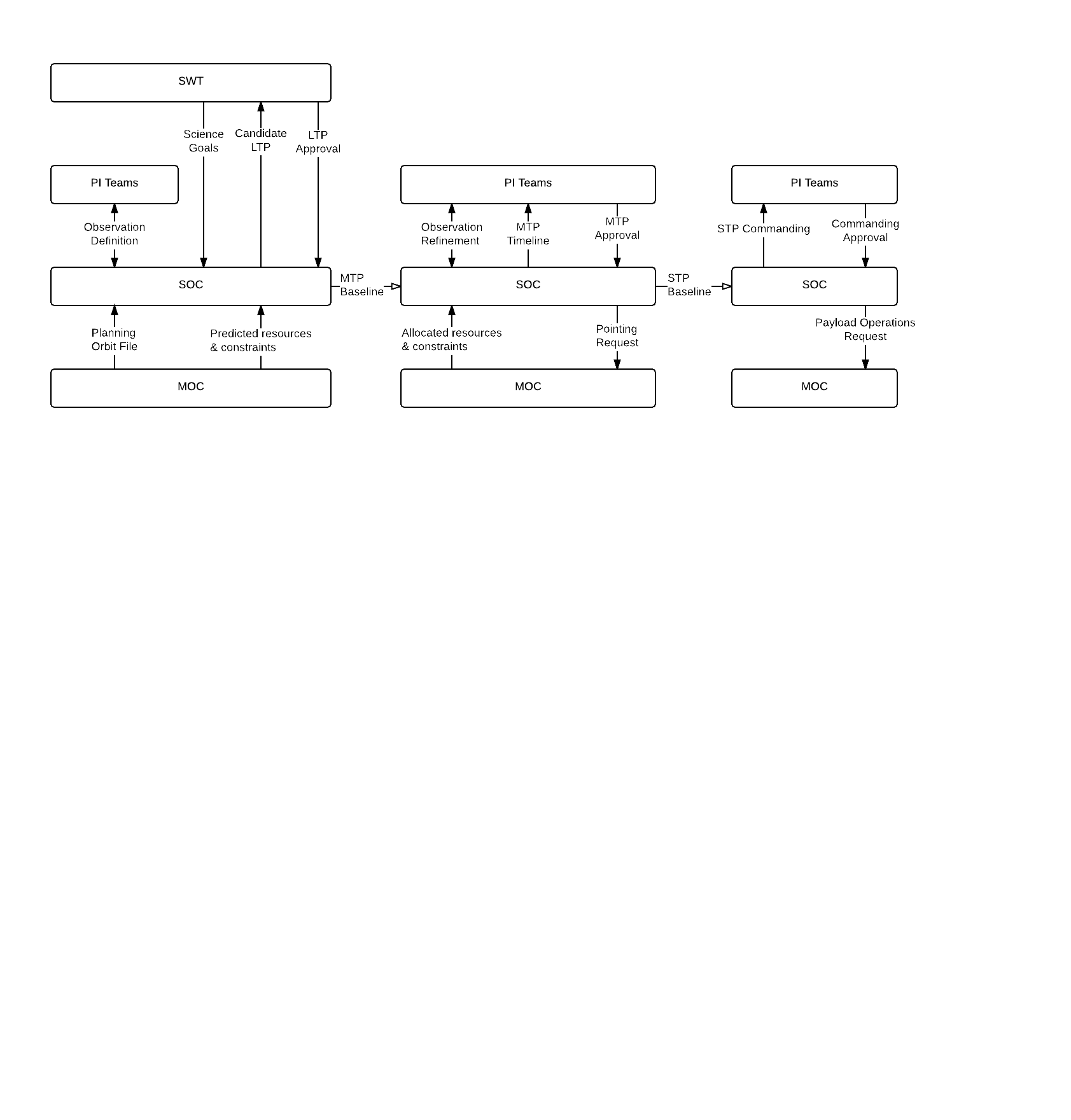


Figure 1 - ExoMars 2016 Planning Cycles

## Applicable and reference documents

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Document Title | Document Reference | Date |
| MP-COMM | CGI-comments-15062018 with RST and CGI responses 6 |  | 06/2017 |
| MO-ISSUES | CGI-MO-issues-11072018 with RST and CGI responses 4 |  | 07/2018 |
| MPSS-TN | Mission analysis for mission planning & Scheduling data model development concept – mapping to CCSDS-MP Model | ESA-MPSS-TN-0001-i1r0 | 08/2018 |
| MAL | Mission Operations Message Abstraction Layer | CCSDS 521.0-B-2 |  |

# mission data

Overview of the analysed missions, types of interfaces, and request formats.

|  |  |  |  |
| --- | --- | --- | --- |
| Mission | Mission type | PI to SOC | SOC to MOC |
| GAIA | Astrometric observatory | - | CRF |
| Bepi Colombo | Planetary / Science | CRF | CRF |
| Euclid | Astronomy | ICR | CRF |
| ExoMars | Mars reconnaissance | ODEF,OGEO | CRF |
| XMM-Newton | Astronomy | XPRS | - |
| EarthCARE | EO | PPF | PPF |
| EnMAP | EO | AcquisitionRequest | |

Table 1 - Mission data formats

CRF Command Request File

ICR Instrument Commanding Request

ODEF Observation Definition

OGEO Observation Geometry File

PPF Payload Planning File

XPRS XMM-Newton Remote Proposal Submission software

## Summary

* Most sample interfaces use the CRF format.
* PI to SOC interfaces are highly specific.
* UML diagrams were created for request formats. Source of diagrams: PLIDs and XML schemas.
* CRFs had a response – CRR (Command Request Response).

Comments about information models:

* Usually a request header + a list of activities.
* There were no examples of nested activities.
* The formats use only a subset of MP model.
* Data length restrictions.
* Data range restrictions.
* String value restrictions (enumerations).
* Special data types: nonNegativeInteger, positiveInteger.
* Special date fields: XSD::dateTime, CommonTime, DeltaTime, PositiveDeltaTime, ConstrainedDeltaTime, ConstrainedPositiveDeltaTime.
* Special attribute for the number of elements in a list (for detecting file errors).

# Mapping to MP model

## Method

Mapping has been described in a tabular format by showing the mission source element/attribute and a target MP element/attribute, as in the example figure.

|  |  |  |  |
| --- | --- | --- | --- |
| Mission | Attribute / Element | Description | MO Mapping |
| GAIA  BC  EXM | CRFHeader::  genTime | File Generation Time | RequestUpdate::Timestamp |



Figure 2 - Mission data to MP mapping method

Issues resulting from mapping mission data to MP model were described in TN of WP1, which was based on MPS Information Model draft C.

## Data mapping issues / examples

This section provides the discovered mapping problems and examples from [MPSS-TN]. Some of the problems given have been addressed or are in the plans, however, there are several that require the attention of the entire WG to come to an agreement how to move forward. Conclusions should be drawn for the open issues during the CCSDS 2018 Fall meeting.

### Data validation - addressed

The model allows constraints on Activities, but not on field values. The mission XML schemas express field value restrictions, such as limits on field length, range of values, allowed values (enumerations). This is data validation, is it part of the model?

**Conclusion**

* *The WG concluded that data validation on Argument values is needed*

### Count attribute in lists – meta data

Lists have a count attribute (number of elements in list) that is not expressed in MO model. Example: GAIA ParameterList in Occurrence. The count attribute may be used as a verification on expected number of elements in list.

**Conclusion**

* *The WG has discussed whether to support complex types, including lists. The decision was not support at that stage, however, it could be reconsidered. Support for list/arrays is probably simpler than custom structures.*

***Discussion Item***

* *Addition of Complex Types to the MP Model?*

### Complex types in arguments - example

Lists/complex types not supported in Arguments. Example: GAIA ParameterList in Occurrence contains elements of type “Parameter”:

* Name (string)
* Position (positiveInteger)
* Description (string)
* Unit (string)
* Value (RelativeToEventTypeValue or CRFParameterValue, both are structures).

Image2

Figure 3 - MP Arguments

**Conclusion**

* *The above example can be expressed using multiple discrete arguments.*

***Discussion items***

* *Description missing in ArgDef*
* *If the parameters are in an Argument list, how to distinguish them from other Arguments?*

### ESA “profile” – complex argument

A case of another complex argument (Figure 4).

***Discussion item***

* *How to express Profile in MP Model?*



Figure 4 - Bepi Colombo Command Request

### Arguments in requestResponse - EnMAP

There are no Arguments that support complex types in request response. An example of this is EnMap’s AcquisitionRequestStatus.



Figure 5 - EnMAP AcquisitionRequestStatus

***Discussion item***

* *How to express in MP model?*

### Complex types in arguments – EnMAP

AcquisitionRequest (Figure 6) has three sub types: TargetPoint, Calibration, AreaCoverage.



Figure 6 - DLR Acquisition Request

***Discussion Item***

* *AdministrativeParams, ControlParams – a case of complex arguments?*
* *Should all constraints be represented in MP model?*

### RelativeValidityEvents – Expression example?

GAIA, EXM, Euclid example: “RelativeValidityEvents” (Figure 7). List of fields:

* **baseEvent** (string) Mnemonic of the event that this time is based on.
* **baseCount** (nonNegativeInteger) Event count of the event the time is based on.
* **delta** (Time) Delta/Relative Time. [-][DDD.]hh:mm:ss[.mmm] format.
* **propagationFactor** (integer) Used to specify whether the occurrence's time is to be set with +/- a number of propagation delays. If no value is specified, 0 will be used as the default, i.e. no delays added.

***Discussion item***

* *Can be formulated as a temporal expression of the form Ta±Td\*/N where Ta is a reference to the Event, Td is the Delta and N is the propagation factor.*
* *How to represent the Nth occurrence of an event instance (baseCount)?*





Figure 7 - GAIA Command Request

### Window of RelativeValidityEvents - example

RelativeValidityEvents (Figure 7) are used to represent a duration.

***Conclusion***

* *Express RelativeValidityEvents as a time expression, then the TimeWindow structure would support this.*



### Validity Period for a Planning Request

There is a concept of validity period that applies to the Request as a whole (Figure 7).

***Discussion Item***

* *This could be added as additional attributes of the request.*

### insertOrDeleteFlag

An insertOrDeleteFlag (Figure 7) “Indicates whether the request is to insert or delete the occurrence.”

***Discussion Item***

* *Cancelling/deleting an activity will be handled by operations?*
* *Is RequestInstance suitable for removing activities, if it were part of a delete operation?*

### Complexity of the model - comment

The model draft C was work in progress at the time of mapping. Without full documentation it was difficult to understand the need for ActivityDetails (former ActivityConstructor). The complexity of the model should also be a consideration to avoid model abuse in implementations.

***Discussion Item***

* *An easy “shortcut” in implementations is substituting Arguments in place where an Expression, Constraint, etc. would be more suitable.*

### GAIA executionTime

Exercise: how to represent GAIA executionTime (Figure 8).

1. Based on time:
   1. **propagationFactor** (time) Used to specify whether the occurrence's time is to be set with +/- a number of propagation delays. If no value is specified, 0 will be used as the default, i.e. no delays added.
   2. **actionTime** (time) The time at which the occurrence is to be actioned.
2. Based on event:
   1. **propagationFactor** (time) Used to specify whether the occurrence's time is to be set with +/- a number of propagation delays. If no value is specified, 0 will be used as the default, i.e. no delays added.
   2. **eventID** This field permits the specification of an event mnemonic relative to which the operation is to be released
   3. **deltaTime** This field permits the specification of an offset time to the event specified in 'event ID' at which the occurrence should be scheduled. If not specified then the occurrence will be scheduled at exactly the time of the event
   4. **separation** (time) This field is used to specify the separation between repeated occurrences
   5. **repeat** (positive integer) minInclusive value="2"
   6. **eventCount** This field identifies a specific instance of the event specified in 'event ID'
   7. **eventCount2** This field is used in order to specify a range of event instances to be triggered



Figure 8 - GAIA Release Time and Execution Time

### releaseTime

Another example: Gaia release time (Figure 8).

* **propagationFactor** (time) Used to specify whether the occurrence's time is to be set with +/- a number of propagation delays. If no value is specified, 0 will be used as the default, i.e. no delays added.
* **earliestOffset** (delta time) The offset of the start of the window for the occurrence, which is only applied to the 'actionTime' or the 'event ID' if specified
* **latestOffset** (positive delta time) The offset of the end of the window for the occurrence, which is only applied to the 'actionTime' or the 'event ID' if specified

***Disussion Item***

* *How to express in the model?*

### executionTime/releaseTime in constraints

How to express both executionTime and releaseTime in constraints? (Figure 7) For example TimeConstraint specifies a TimeExpression for an ActivityRef, but it does not specify the semantics of the time constraint, which is assumed to be the execution time of the Activity.

***Discussion Item***

* *The model supports the concept of execution time. Extend the Activity to support the concept of release time?*
* *Alternative: consider the Release and Execution as two separate activities.*

### Multiple errors in requestResponse

Multiple errors not handled in Request Response. RequestUpdate allows specifying single error using ErrCode (Integer) and ErrInfo (String). GAIA allows multiple errors. First it says the number of errors, and for each error there is reference (line nr. in file) and string value.



Figure 9 - GAIA request response



Figure 10 - MP Request Update

***Discussion Item***

* *Multiple errors can be reported, if multiple responses are provided – we have not yet defined the interactions in the service. Do we want to extend to allow for multiple errors?*

### XMM-Newton Revolution - example

XMM-Newton web interface has the following Observation Time Constraint (“**XMM-Newton Revolution**”):

* EarliestRevolution
* LatestRevolution
* RepeatXTimes
* RepeatEveryXRevolutions
* RepeatToleranceXRevolutions

***Discussion Item***

* *How to represent in the model? (Figure 11)*



Figure 11 - MP Activity Details

### XMM-Newton Orbital Phase - example

XMM-Newton web interface has the following Observation Time Constraint (“**Orbital Phase**”).

* EarliestStartTime
* LatestStartTime
* ReferenceTimeForPhaseZero
* Period in days (int)
* Phase (float, <1.0)
* Tolerance in days (int)
* Repeat (int)
* Consecutive (boolean) – “XRPS (the web interface - CGI) will count all exposures belonging to one observation consecutively.”

***Discussion Item***

* *How to represent in the model? (Figure 11)*

## MAL issues

### MAL limitation – no polymorphism

[MO-ISSUES] There is a documented limitation using abstract types in MO messages. Complex types do not allow abstract members (diagram), with one exception: Attribute. It is as designed in MAL.



Figure 12 - Constraints: abstract members not allowed

#### Technical reasons and a limited workaround

Section 4.1.7 of [MAL] says that “*The MAL data type specification does not allow the containment of abstract elements*”. “The exception to this is the containment of the abstract Attribute type. It is permitted to create composites that have fields defined as Attribute.”

Section 4.1.6 of [MAL] explains the extension of complex types. Complex types extend MAL:Composite. Complex types may be abstract (has no Short Form Part attribute) or concrete (has Short Form Part). Abstract types can be used for specifying a family of related objects that share fields. Concrete type in MO means non-extendable: “*It is not permitted to extend a non-abstract composite*”. And: „*The MAL data type specification does not permit composites to contain an abstract type; all fields of a composite must be concrete types*“. The reasons, as explained in section 4.1.6 of [MAL], lie within the transport layer.

As a limited workaround, section 4.1.11 of [MAL] explains that „last type in a message body may be an abstract type“. Lists are also permitted, given similar condition: „It is permitted to specify the final part of the message to be a list of an abstract type. When the message body is encoded, any list of an abstract type shall be replaced with a list of a derived concrete type“. *This means that at run time the abstract list may contain objects of exactly one concrete type*.

#### Polymorphism in MP model

The model (draft D2) contains the following abstract member types:

|  | Diagram | Complex Type | Member | Abstract Type of Member |
| --- | --- | --- | --- | --- |
| 1 | Arguments | ArgSpec | ArgSpec | Expression |
| 2 | Constraints | Constraint Node | Constraints | Constraint[] |
| 3 | Planning Activities | Activity Definition | DurationSpec | Expression |
| 4 | Planning Activities | Activity Update | StartTime | Time |
| 5 | Planning Activities | Activity Update | EndTime | Time |
| 6 | Planning Events | Event Update | Timestamp | Time |
| 7 | Planning Requests | Request Update | Timestamp | Time |
| 8 | Planning Resources | Resource Update | Timestamp | Time |
| 9 | Plan | Plan Revision | RevisionStart | Time |
| 10 | Plan | Plan Revision | RevisionEnd | Time |
| 11 | Expressions | Expression | Result | Variant |
| 12 | Activity Details | Activity Node | Repetition | Repetition |

**Concl*usion***

* *The MP model design should be completed disregarding the MAL limitation. The MP model can be taken as requirements for an equivalent MAL compatible model.*