2MVG2-2RIM2

UML conceptual model

1 Introduction, scope & objectives

This document describes the UML Conceptual model of the guidelines developed by 2MVG2-2RIM project.

We refer to the general project results (see document “guidelines”) to correctly understand the scope and role this conceptual model plays in the overall project guidelines.

This document explains the details of the UML data model. See “UML conceptual model - diagram”.

We recall that the conceptual model was created to capture the needs of the MVG-II/RIM-II coding.

However, it has the intention to be generally applicable to the needs of the (nursing) care aspects of a patient record.

- The entities and relationships that are drawn with a black border are required for MVG-II/RIM-II. They are explained in detail.
- The entities and relationships that are drawn with a blue border are not required for MVG-II/RIM-II and only present to show the compliancy with the general needs of a patient record. They are described in brief only.

Note that when applying this conceptual model to build a database for a general patient record, a number of additional considerations should been taken into account. These additional considerations are not required for MVG-II/RIM-II.

- Concrete design implementation decisions should be taken. Examples are e.g. the concrete representations of the data elements. For MVG-II/RIM-II data values are typed but represented as text. For a general patient record, more appropriate representations may exist.
- When creating a patient record, other data and processes than those involved with patient care data and actions, play a role. Examples are e.g.
    - the implications on the care process of privacy, security and audit;
    - the use of a master patient index, when exchanging information between systems that use different identifiers for the same patient;
    - values lists may contain additional entries that are not required for MVG-II/RIM-II;
    - a model for the organizational structure of the hospital
    - etc…

2 Overview

The conceptual model describes the concepts that are used to represent:

- patient care
- administrative patient and organization information

Along the objectives of this research project, the concepts used for the representation of patient care are general and not limited to the MVG-RIM needs. They are also more general than the needs of a nursing record; they should be able to capture the concepts of a patient record.

The section "Acts & Facts" describes essentially the information that is known about a patient and the care activities that concern the patient. For MVG-RIM these data elements are the basis for measuring the executed activities.

The section "Paths & Plans" describes the mechanisms that that lead to the planning and execution of care activities. It can capture the concepts used in "Care paths", "Clinical pathways", "systematisch verpleegkundig handelen", "transmissions ciblées", etc.. For MVG-RIM these data elements form to a large extend the "validation conditions".

The conceptual model also describes administrative patient and organization information. With respect to the administrative patient and organization information, the conceptual model has not been generalized to concepts outside the MVG-RIM needs.

The section "Patient & Stays" describes the administrative patient information that is required to control the MVG-RIM process and to complete a full MVG-RIM reporting to the Portahealth server of the Belgian PFS.

The section "Caregivers" describes the administrative organization information with respect to the caregivers that are involved in the care process.

3 Terminology

3.1 Glossary

The following terms have a specific meaning in this text:

- The sending system sends clinical information to the receiving system.
- Unique identifier. A unique identifier is a string or number that is unique within a single system. It can thus be used to identify data and entities in this system. The data type for unique identifiers is "Eid".
    - Examples of unique identifiers are attributes ending on "Ref", "Code", etc…
- Universally unique identifier. A universally unique identifier is a string or number that is unique over all involved systems. It can thus be used to identify data and entities over different systems. The data type for unique identifiers is "Uid".
    - Examples of unique identifiers are attributes ending on "Uid".
    - We use universally unique identifiers to identify patients, caregivers, hospital stays, units, etc… The receiving system will thus not play the role of a "master patient index". If a "master patient index" is in use, the sending system should "normalize" its internal unique identifiers to the universally unique identifiers.

3.2 Translation of concepts

The following table gives the translation of the concepts used in Dutch and French.

English	**Dutch**	**French**
Act	Activiteit	Activité
Fact	Feit	Fait
Result	Observatie	Résultat
State	Toestand	Etat
Aspect	Aspect	Aspect
Caregiver	Zorgverlener	Prestataire
Value	Waarde	Valeur
Order	Instructie	Instruction
Plan	Zorgplan	Plan de soin
Prescription	Voorschrift	Prescription
Path	Zorgpad	Itinéraire clinique

Evaluation	Evaluatie	Evaluation
Objective	Doelstelling	Objectif
PlanTemplate	Zorgplansjabloon	Gabarit de plan de soin

4 General conventions

4.1 Representation of periods in time

Whenever an attribute is called “Until” it refers to the moment in time that a period ends. The moment in time that is mentioned, is excluded from the period. This means concretely:

- When Until is a DateTime, Until refers to the first moment in time after the period ends. E.g. the month of May 2006 is represented by [From=’ 01/05/2006 00:00:00’, Until=’01/06/2006 00:00:00’).
- When Until is a Date, Until refers to the first date after the period has ended. E.g. the year 2006 is represented by [From=’ 01/01/2006‘, Until=’01/01/2007’).

The reason to represent Until values in this way is that two consequent periods in time are easily represented: the From value of the next period is equal to the Until value of the previous period.

Implementation note: this model is chosen to be able to exactly represent periods in time. It is not the logical representation of the period to the user, who typically expects the Until value to be included in the period. It is the responsibility of the user interface of concrete systems to deal with this translation.

5 Acts & Fact

5.1 Overview

Acts & Facts describe the patient care information and care activities.

5.2 Entities

5.2.1 Act

5.2.1.1 Definition

An Act represents a care activity. Typically the recipient of care is a patient. However, in some case it may also be a significant person (see attribute RecipientOfCare).

This patient care activity is executed by or under the supervision of a Caregiver (see attribute ByCaregiverId). The patient care activity can be executed by the caregiver, by the patient himself or by a significant person (see attribute BySubject).

It is possible to represent planned, executed, cancelled activities (see attribute StatusId). We can represent the moment when the activity was planned, when it was planned to start, when it started, and when it ended.

5.2.1.2 Attributes

5.2.1.2.1 Sys

The Sys attribute represents the sending system in which the activity was recorded.

This attribute is needed in order to be able to receive or represent Acts that originate in different sending systems.

5.2.1.2.2 Ref

The Ref attribute represents the unique identification of the activity that took place in the sending system. This activity is an effectively executed or planned care action for a specific patient at a specific moment in time.

An example of an activity is “administering Dafalgan 500 mg to patient Dupont at May 31st, 2006 at 9h32”.

5.2.1.2.3 RefVersion

The RefVersion attribute represents the moment in time since when the specified attributes values of this activity are valid in the sending system.

Typically more than one version of the activity will exist: the activity is planned beforehand (version 1); then, it is executed (version 2); occasionally, the recorded information concerning the activity is corrected or complemented (version 3).

5.2.1.2.4 Code

The Code attribute represents the unique identification of the kind of activity that took place in the sending system. This kind of activity is one of the possible activities that can be ordered and is typically selected from a catalog of all known care activities that can be used in the sending system.

An example of a kind of activity is “administering a drug”.

5.2.1.2.5 Name

The Name attribute represents the literal representation of the kind of activity that took place in the sending system. Typically, this name can be looked up by the Code attribute in the catalog of all care activities of the sending system.

The Name attribute is optional. It makes data and messages more legible, but it not required for their semantics.

5.2.1.2.6 StatusId

The StatusId attributes represents the current status of the activity. It can be:

- Planned. The activity is planned for execution at PlannedExecutionStartDateTime.
- Executed. The activity has effectively been executed successfully.
- Cancelled. The activity has been planned before, but is cancelled.

Note: in some cases it may be useful to track whether an activity was executed with success, or whether if failed. This should be modeled through the attributes of the activity.

5.2.1.2.7 ByCaregiverUid

The ByCaregiverUid attribute represents the universally unique identifier that identifies the caregiver who executed the action or under whose responsibility the action took place.

5.2.1.2.8 BySubjectId

The BySubjectId attribute represents who effectively executed the action, or who is planned to execute the action. It can be:

- ByCaregiver. The caregiver effectively executes the action.
- ByPatient. The patient himself executes the action.
- ByFamily. The patient’s family or a significant person executes the action.

5.2.1.2.9 AtUnitUid

The AtUnitUid attribute represents at which unit the action was executed, or is planned to be executed.

5.2.1.2.10 PlanningTypeId

The PlanningTypeId attribute represents the type of planning that underlies the executed or planned activity. It can be:

- NotPlanned. The activity was executed without that is was planned.
- PlannedIfNecessary. The activity was explicitly planned "if necessary". A special case is an activity that is conditionally ordered in the context of a StandingOrder.
- PlannedInTime. The activity was planned to be executed in a determined moment in time. A special case is an activity that is "unconditionally" planned in the context of a Plan or StandingOrder.

5.2.1.2.11 PlanningDateTime

The PlanningDateTime attribute represents the moment in time when the planning of the activity took place.

Note the difference with PlannedExecutionStartDateTime.

5.2.1.2.12 PlannedExecutionStartDateTime

The PlannedExecutionStartDateTime attribute represents the moment in time for which the execution of the activity was planned.

5.2.1.2.13 EffectiveExecutionStartDateTime

The EffectiveExecutionStartDateTime attribute represents the moment in time at which the effective execution of the activity has started.

5.2.1.2.14 EffectiveExecutionDurationTimeSpan

The EffectiveExecutionDurationTimeSpan attribute represents the duration in time during which the effective execution of the activity has lasted.

5.2.1.2.15 RecipientOfCare

The RecipientOfCare attribute represents the recipient of care. It can be:

- "#". The recipient of care is the patient himself.
- Any text value. The recipient of care is a significant person, as described in text by the text value.

5.2.1.2.16 ProtocolCode

The ProtocolCode attribute indicates the protocol that describes how the activity has to be performed. The ProtocolCode attribute represents this protocol by its unique identifier by which the protocol is defined (or known) by the sending system.

It is possible that the protocol is defined outside the sending system, e.g. on paper or in a separate system. The ProtocolCode attribute then represents the code by which the protocol is known to the sending system, i.e. the code by which the sending system refers to the protocol. This means that, even if the protocol is defined outside the sending system, this sending system is able to trace the protocol that underlies the activity.

If two versions exist of the “same” protocol, they are considered separate protocols and correspondingly have different ProtocolCodes. There does not exist an attribute as e.g. ProtocolCodeVersion.

5.2.1.2.17 ProtocolName

The ProtocolName attribute indicates the protocol that describes how the activity has to be performed. The Protocol attribute represents the name of the protocol as known by the sending system.

The ProtocolName attribute is optional. It makes data and messages more legible, but it is not required for their semantics.

5.2.1.2.18 (CancellationReason)

The CancellationReason attribute represents the reason why an activity was cancelled.

This attribute is present for compliance with the AR/KB and not required for MVG-RIM.

5.2.1.3 Keys

5.2.1.3.1 Primary Key

The primary key for the Act entity is: Sys, Ref.

Note that subsequent versions of the Act (thus with different RefVersion values) are really considered versions of the *same* Act. It is intended to that only the most recent version of the Act is valid.

5.2.2 Fact, Result, State

5.2.2.1 Definition

A Fact represents information known. Typically the subject of information is a patient. However, in some case it may also be a significant person (see attribute SubjectOfInfo).

A Result represents a Fact that is valid at a specific moment in time. This is typically a result from an observation or a lab exam. The specific moment in time at which the information is valid is represented by ValidAtOrFrom.

A State represents a Fact that is valid over a longer period in time. This is typically a diagnosis motivated by the interpretation of results. The period in time during which the information is valid starts at ValidAtOrFrom and lasts until ValidUntil. If ValidAtOrFrom is <null>, it is considered that this state has last since the beginning of the HospitalStay of the Patient; if ValidUntil is <null>, it is considered that this state will last until the end of the HospitalStay of the Patient.

> Note that is would be possible to define other “special values” for ValidAtOrFrom or ValidUntil, such as “beginning of UnitStay”, etc… This generalization is not required for MVG-RIM and outside the scope of the project.

Facts cannot be anonymous. A Fact is always scoped by an explicit Act executed by an identified Caregiver. The implied semantics of this Act are “reporting the result”. The Caregiver under whose responsibility the Act was executed is considered the author of the Fact and takes responsibility for it.

It is possible to represent announced, partial, complete, validated and cancelled information (see attribute Status).

5.2.2.2 Attributes

5.2.2.2.1 Sys, Ref, (RefVersion), Code, Name

See Act. The semantics of these attributes are identical.

Note that RefVersion is derived information and by definition equal to: LastlyReportedDuring.RefVersion.

5.2.2.2.2 TypeId

The TypeId attribute represents the type of Fact. It can be:

- Result. This type is represented by:
  - Result. Typically, an observation
  - PlanEvaluation. A Result for the purpose of evaluating a care plan
- State. The following types of states are defined:
  - State
  - NursingDiagnosis
  - MedicalDiagnosis
  - SocialDiagnosis

5.2.2.2.3 StatusId

The StatusId attributes represents the current status of the Fact. It can be:

- Announced. The information is not yet present, but it is expected.
- Partial. The information is partially complete.
- Complete. The information is complete
- Validated. The information is complete and explicitly validated by a Caregiver.
- Cancelled. The information has been revoked.

5.2.2.2.4 SubjectOfInfo

The SubjectOfInfo attribute represents to whom this information applies. It can be:

- "#". The subject of info is the patient himself.
- Any text value. The subject of info is a significant person, as described in text by the text value.

5.2.2.2.5 ValidAtOrFrom

The ValidAtOrFrom attribute represents:

- In case of a Result: the moment in time to which the result information applies.
- In case of a State: the moment in time since when the state information applies.

If the ValidAtOrFrom attribute is NULL, it is unless otherwise specified equal to

- Fact.LastlyReportedDuringAct.ContextHospitalStay.From

Note that this attributes is the moment when a value is measured. In case of blood or tissue sample values, the moment when the sample was taken is equally important. This should then be modeled by an Aspect (for MVG2-RIM2: see the semantic model of N300, V500).

5.2.2.2.6 ValidUntil

The ValidUntil attribute represents:

- In case of a Result: this attribute is meaningless.
- In case of a State: the moment in time until when the state information is valid.

If the ValidUntil attribute is NULL, it is unless otherwise specified equal to

- Fact.LastlyReportedDuringAct.ContextHospitalStay.Until

5.2.2.2.7 SourceId

The SourceId attribute represents the origin of the information. It can be:

- ManuallyEntered. The information has been manually entered into the sending system.
- AutoMonitored. The information has been automatically entered into the sending system, typically through a connection with a monitoring device. This information may be validated manually afterwards, as indicated by the Status attribute.

5.2.2.3 Keys

5.2.2.3.1 Primary Key

The primary key for the Fact entity is: Sys, Ref.

Note that subsequent versions of the Fact (thus with different RefVersion values) are really considered versions of the *same* Fact. It is intended to that only the most recent version of the Fact is valid.

5.2.2.4 Relationships

5.2.2.4.1 LastlyReportedDuring

The LastlyReportedDuring relationship defines in which scope the information has come available. Through this relationship the Act during which the information has come available is known. This includes:

- When this (version of the) information has come available?
- Which Caregiver takes responsibility for the information?
- Etc…

Note the difference in semantics between:

- Fact.ValidAtOrFrom: the moment in time at which the result was measured or when the result is valid.
- Fact.ScopingAct.EffectiveExecutionsStartDateTime: the moment in time at which this result was reported.

By definition, the (logical) RefVersion attribute of a Fact, equals the RefVersion attribute of the LastlyReportedDuring.Act. This means that the Act that reports a Fact, determines the RefVersion of the Fact. This definition is defined in this way, in order to guarantee that messages containing Acts & Facts are idempotent (see XML Message Format).

5.2.2.5 Constraints

5.2.2.5.1 The fact type of a Fact must be consistent

For all elements with a given Fact.Code, its attributes Fact.TypeId must be identical.

This means that the same Fact.Code cannot not refer sometimes to a Result and sometimes to a State.

5.2.3 Value

5.2.3.1 Definition

A Value represents the value or element that holds the information, of a Fact or Aspect.

The data type of a Value determines what kind of values is valid or possible. The data type is known by DataTypeCode and DataTypeName. The underlying nature of the data type must be the same in the sending and receiving system. This underlying nature is called the Archetype. Examples:

- Both the sending and receiving system may represent the discharge date of a patient stay by a data type that they can both define and name it independently. However, the underlying nature of this data type is in both cases a “date” and is called their Archetype.
- Both the sending and receiving system may represent the degree of disability of a patient differently. The sending system may use values like ‘totally’, ‘severe’, ‘partial’, ‘absent’. The receiving system my use values like ‘yes’, ‘no’. The underlying nature of this data type however matches: in both cases the underlying Archetype is an Enumeration.

Although values belong to a specific data type that belongs to a specific Archetype, Values are always represented as strings, even if the underlying data type is stricter. This representation is sufficient for the MVG-RIM needs.

- The string representation is represented in the Value attribute.
- In the case the Archetype is an enumeration, i.e. when the values come from a list of allowed values, the Value is represented in a structured way (see attributes Value, ValueName).
- In the case the value is calculated based on a scale, this scale is identified (see attributes ScaleCode, ScaleName)

Currently the Archetype of the value in the sending and receiving system must be equal. This is sufficient for MVG-RIM needs. It would be possible to relax this constraint in a more general system and allow simple data type transformations like e.g.: datetime to date; enumeration to boolean; etc… This is outside the scope of the current project.

A Value is not an entity on its own. It always aggregates with either a Fact or an Aspect.

Implementation note: the sending and receiving system may use different data types for a value, as long as those data types have the same underlying ValueArchetype for representing a value. This should not be a practical limitation:

- The nature of the value is an inherent characteristic of the value itself. It is therefore logical that this nature can be represented by the same type in both the sending and the receiving system.
- In all cases, but Enumerations, the value is represented as a string. The sending system can thus easily declare the ValueArchetype identical to the ValueArchetype defined by the receiving system.
- In case of Enumerations, the systems foresees in a mapping (see below).

5.2.3.2 Attributes

5.2.3.2.1 DataTypeCode

The DataTypeCode represents the unique identifier of the data type of the Value as known by the sending system.

The DataTypeCode must be fixed and immutable for every Aspect or Fact.

- Remember that a Value aggregates to either an Aspect or a Fact, each identified in the sending system by a Code, either by Aspect.Code or Fact.Code. It can be derived from the UML diagram that every Aspect.Code or Fact.Code maps on one and only one DataTypeCode.
- It is important to mention that the association between the Aspect.Code and its corresponding DataTypeCode is fixed and immutable. Subsequent messages that use

the same Aspect.Code must use the same DataTypeCode. The same holds for the Fact.Code and associated DataTypeCode.
- If an Aspect or Fact is to be associated with a new DataTypeCode, a new Aspect or Fact should be defined.
- Note that is theoretically possible in the model to remove the DataTypeCode, DataTypeName attributes from the message format. These attributes are present such that the specified constraint can be checked.

5.2.3.2.2 DataTypeName

The DataTypeName represents the name of the data type of the Value as known by the sending system.

The DataTypeName attribute is optional. It makes data and messages more legible, but it is not required for their semantics.

5.2.3.2.3 ArchetypeId

The ArchetypeId attribute represents the underlying nature of the data type of the Value, as represented in the sending system.

It can be:
- Boolean. The value of a Boolean can be either:
  - True
  - False
- DiagnosticState. The value of a DiagnosticState can be either:
  - Absent
  - Present
  - Potential
- Enumeration. The value of an Enumeration comes from a limited list that is defined by the sending system.
- Date. The value represents a date.
- Time. The value represents a time.
- DateTime. The value represents a moment in time
- Numeric. The value represents a numeric value.
- Scale. The value represents a scale. A Scale is defined as the result of a measurement instrument.
- Text. The value represents text. The intended meaning is a short text fragment, as e.g. a follow-up note.
- Report. The value represents a complete report. The intended meaning is a long text fragment, as e.g. a RX-result, a referral letter or discharge note.
- URN. The value represents a URN. A URN is a string that references in a unique way a specific document or object in cyberspace.
- Any. The nature of the value is unknown or unspecified.
- Protocol. The nature of the value is a protocol.
- Caregiver. The nature of the value is a caregiver Uid.

Note: This archetype attribute is the archetype as known by the sending system. It is currently a requirement that this archetype equals the archetype of this value as known by the receiving system.

5.2.3.2.4 Value

The Value attribute represents the actual value. The actual value is represented as a string.

If the Archetype attribute equals Enumeration, the Value attribute represents the unique identifier by which the enumeration value is known in the sending system. This enumeration value is typically picked from the list of all allowed enumeration values for the data type identified by the DataTypeCode in the sending system.

If the Archetype attribute equals Protocol, the Value attribute represents the unique identifier by which the protocol is known in the sending system.

If the Archetype attribute equals Caregiver, the Value attribute represents the universally unique identifier by which the caregiver is known to all sending systems.

5.2.3.2.5 ValueName

The ValueName attribute is only meaningful if the Archetype is “Enumeration”. Otherwise it is ignored.

The ValueName attribute represents the literal representation that corresponds to the Value identifier in the sending system.

The DataTypeName attribute is optional. It makes data and messages more legible, but it is not required for their semantics.

5.2.3.2.6 ScaleCode

The ScaleCode attribute is only meaningful if the Value is calculated by a measurement instrument. Otherwise it is ignored.

The ScaleCode attribute represents the unique identifier by which the measurement instrument is known in the sending system.

It is possible that the measurement instrument is defined outside the sending system, e.g. on paper or in a separate system. The ScaleCode attribute then represents the code by which the measurement instrument is known to the sending system, i.e. the code by which the sending system refers to the measurement instrument. This means that, even if the measurement instrument is defined outside the sending system, this sending system is able to trace the measurement instrument by this code.

If two versions exist of the “same” measurement instrument, they are considered separate measurement instrument and correspondingly have different ScaleCodes. There does not exist an attribute as e.g. ScaleCodeVersion.

5.2.3.2.7 ScaleName

The ScaleName attribute is only meaningful if the Value is calculated by a measurement instrument. Otherwise it is ignored.

The ScaleName represents the name of the measurement instrument as known by the sending system.

The ScaleName attribute is optional. It makes data and messages more legible, but it is not required for their semantics.

5.2.3.3 Relationships

A Value aggregates with either a Fact or an Aspect.

5.2.3.4 Constraints

5.2.3.4.1 The data types of an Aspect must be consistent

For all elements with a given Aspect.Code, its Value attributes DataTypeCode and ArchetypeId must be identical.

This means that the data type of the values of an Aspect defined by Aspect.Code are always the same.

5.2.3.4.2 The data types of a Fact must be consistent

For all elements with a given Fact.Code, its Value attributes DataTypeCode and ArchetypeId must be identical.

This means that the data type of the values of a Fact defined by Fact.Code are always the same.

5.2.4 Aspect

5.2.4.1 Definition

An Aspect represents information that clarifies the conditions under which an Act took place or under which a Fact is valid.

5.2.4.2 Attributes

5.2.4.2.1 Code, Name

See Act. The semantics of these attributes are identical.

5.3 Relationships

5.3.1 ContextAct

The ContextAct relationship allows expressing that an Act took place as part of or as support for another Act.

With regard to the MVG-RIM needs, this is only used for item “Z200”.

5.3.2 PlannedActs

The PlannedActs relationship allows expressing that an Act took place as part of a Plan.

5.3.3 ContextState

The ContextState relationship allows expressing that an Act took place in response to a patient’s State.

This relationship is present for compliance with the AR/KB and not required for MVG-RIM.

5.3.4 ByAdditionalCaregivers

The ByAdditionalCaregivers relationship allows expressing that an Act took place with the participation of other caregivers than identified by the ByCaregiverUid attribute.

With regard to the MVG-RIM needs, this could be useful for items like “Z300”. As we expect that (not all) sending systems are able to provide more than one participating caregiver in an Act, we

have used a simpler model to cope with items like “Z300”. This simpler model is based on providing the additional caregivers as an Aspect to the Act.

5.3.5 OrderedActs

The OrderedActs relationship allows expressing that an Act took place ordered by a specific Order.

This relationship is present for compliance with the AR/KB and not required for MVG-RIM.

5.3.6 JustifyingResult

The JustifyingResult relationship allows expressing that the patient’s State is justified by the underlying Result(s).

This relationship is present for compliance with the AR/KB and not required for MVG-RIM.

6 Plans

6.1 Overview

The section “Paths & Plans” describes the mechanisms that that lead to the planning and execution of care activities.

6.2 Entities

6.2.1 Path

6.2.1.1 Definition

The Path entity represents the context or the sequence in which related care plans take place. It captures the notion of “clinical pathways”, “care paths”, etc…

It extends the notion of Plan (see below). A Path is modeled as a set of Plans. The general idea is that the trails connecting the evaluation points in a Path are each considered a Plan.

The Path entity is not required for MVG-RIM and further out-of-scope of this project.

6.2.2 Plan

6.2.2.1 Definition

A Plan entity represents a care plan.

A care plan typically organizes the care to cope with one or more Problems, represented by a State. Once the Problems are identified, the patient state with respect to these problems is evaluated, resulting in an initial Evaluation. Objectives are formulated and actions are ordered, defined as Orders. An Order defines how often en under what condition an Act should be executed.

Note that is model captures the underlying concepts of “systematisch verpleegkundig handelen” and “Transmissions ciblées”.

The type of plan is described along two axes:

- The PlanKindId (CarePlan or StandingOrder) indicates whether the Orders (and Acts) in the Plan are unconditional or depend upon further interpretation of the Caregiver.

- The PlanSpecifityId (Standard or Individualized) indicates whether the Orders (and Acts) in the Plan are adapted to a specific patient.

The PlanTemplate defines, in case of a Standard plan, the Orders that should be applied to the typical patient population. The PlanTemplate is the general description of a plan that is not applied to a specific patient.

6.2.2.2 Attritubes

6.2.2.2.1 Sys, Ref, RefVersion, Name

See Act. The semantics of these attributes are identical.

6.2.2.2.2 PlanKindId

The PlanKindId attribute defines the kind of this plan. It can be (see also above):

- CarePlan
- StandingOrder

6.2.2.2.3 PlanSpecifityId

The PlanSpecifityId attribute defines the specifity of this plan. It can be (see also above):

- Standard
- Individual

6.2.2.2.4 From

The From attribute represents the moment in time since when this plan applies to the patient. Only Acts that are executed since this moment in time are considered part of the plan.

6.2.2.2.5 Until

The Until attribute represents the moment in time until when this plan applies to the patient. Only Acts that are executed until this moment in time are considered part of the plan.

6.2.2.2.6 PrescribingCaregiverUid

The PrescribingCaregiverUid attribute represents the universally unique identifier of the Caregiver who prescribed or authorized the execution of this plan.

6.2.2.2.7 PlanTemplateRef

The PlanTemplateRef attribute represents the PlanTemplate on which this plan is based, or from which this plan is derived. The PlanTemplate attribute holds the unique reference by which the sending system identifies the PlanTemplate.

This is valid both when the PlanTemplate is part of the sending system, e.g. the sending system has an electronic PlanTemplate library, and when the PlanTemplate is not part of the sending system, but e.g. defined on paper.

If this plan is an Individual Plan, the value of this attribute is ignored.

6.2.2.2.8 PlanTemplateRefVersion

The PlanTemplateRefVersion attribute represents the version of the PlanTemplate on which this plan is based, or from which this plan is derived.

This attribute is valid only when the PlanTemplate is part of the sending system. If the PlanTemplate is on paper, the value of this attribute is ignored.

If this plan is an Individual Plan, the value of this attribute is ignored.

6.2.2.2.9 IndividualPlanOnPaperRef

The IndividualPlanOnPaperRef attribute represents the unique identifier by which the sending system refers to the plan when this plan is maintained outside the sending system on paper. In this case, typically only the Orders are maintained in the sending system.

6.2.2.3 Relationships

The relationships are described with the other entities that relate to “paths & plans”.

6.2.2.4 Keys

6.2.2.4.1 Primary Key

The primary key for the Plan entity is: Sys, Ref.

6.2.3 PlanTemplate

6.2.3.1.1 Definition

The PlanTemplate entity represents the plan template on which this plan is based.

This entity is present for compliance with the general need of the sector, but is not required for MVG-RIM needs.

6.2.4 Evaluation

6.2.4.1.1 Definition

The Evaluation entity represents an observation result where the patient’s capabilities are assessed. This evaluation serves as the basis for defining the Objectives and Orders that are required to cope with the patient’s Problem.

An Evaluation is a special kind of Result.

With respect to the MVG-RIM needs, two kinds of evaluations are to be considered in the context of a plan. The initial evaluation is the first evaluation when a plan is started. Once started the plan has to be reevaluated periodically.

6.2.5 Objective

6.2.5.1.1 Definition

The Objective entity represents the objectives that this plan targets at.

This entity is present for compliance with the general need of the sector, but is not required for MVG-RIM needs.

6.2.6 Order

6.2.6.1.1 Definition

The Order entity represents the orders that are planned.

This entity is present for compliance with the general need of the sector, but is not required for MVG-RIM needs.

6.2.7 Prescription

6.2.7.1.1 Definition

The Prescription entity represents the authorization to execute this plan.

This entity is present for compliance with the general need of the sector, but is not required for MVG-RIM needs.

6.3 Relationships

6.3.1 Problems

The Problem relationship allows representing the nursing problems or diagnoses that are at the origin of this plan.

6.3.2 Authors

The Authors relationship allows representing the caregivers that have specified or created this plan.

6.3.3 PlannedActs

The PlannedActs relationships allows representing which Acts are executed in the context of this plan.

An Act is always executed in the context of at most one plan.

This entity is present for compliance with the general need of the sector, but is not required for MVG-RIM needs.

6.3.4 OrderedActs

The OrderedActs relationships allows representing which Acts are executed in the context of an Order.

An Act is always executed in the context of at most one Order.

This entity is present for compliance with the general need of the sector, but is not required for MVG-RIM needs.

7 Patient & Stays

7.1 Overview

Patient & Stays describe the required patient demographic information and the information with regard to the stay of the patient in the hospital.

The model is sufficient for the MVG-RIM requirements.

It is designed to be compatible with the needs of a general electronic patient record; of course, electronic patient records may extend these concepts to deal with more complex situations.

Typical extensions to this model would include: a MPI (master patient index), mechanisms to deal with patient or stay merges.

As we do not include MPI, patient or stay merge aspects in this model, we expect the sending system to access the MPI and translate its internal unique patient identifier to the universally unique patient identifier for the site. The receiving system only deals with the “normalized” unique patient identifier. This simplifies the receiving system and recognizes the fact that every practical sending system has to deal with a MPI anyway. The same is true for patient and stay merges.

Implementation note: a sending system can deal in a pragmatic way with patient merges. This pragmatic approach is much simpler and less error prone than passing info about patient and stay merges, or linking to an MPI. Whenever a patient is “merged”, all data related to the old (universally) unique identifiers that was previously sent to the receiving system, is revoked by sending a “delete” message: delete all information with regard the patient (or stay) identified by the old identifier. Then resend this information associated with the proper, new identifier. The simplest way of doing so, is resending all information regarding that patient or stay.

THIS MODEL IS BASED ON INFORMATION REQUIRED BY THE REGISTRATION MANUAL OF NOVEMBER 2006.

7.2 Entities

7.2.1 Patient

7.2.1.1 Definition

A Patient entity represents the patient.

For the MVG-RIM needs only a limited set of patient demographic attributes are required. These attributes are described.

Note that the conceptual model is general. Any additional patient demographic attribute that would be useful in a more general case, can be added. These additional attributes are modeled as Aspects.

7.2.1.2 Attributes

7.2.1.2.1 Uid

The Uid attribute represents the universally unique identifier by which the patient is identified by all sending systems. This Uid typically corresponds to the master patient index entry for the patient in the institution.

7.2.1.2.2 Dob

The Dob attribute represents the date of birth of the patient.

7.2.1.2.3 SexId

The SexId attribute represents the gender of the patient. It can be:

- Unknown
- Male
- Female
- Changed

Note that these correspond to the values of the field A2_CODE_SEX of the registration guide (version november; page A 8).

7.2.1.2.4 AlternateUid

The AlternateUid attribute represents the identifier by which this patient will be known by the Portahealth system.

Note that this field corresponds to the PATHOSPI.PATNUM field in the registration guide.

7.2.1.2.5 ContextChildOfPatientUid

The ContextChildOfPatientUid attribute represents the universally unique identifier by which the mother of this patient is known.

7.2.1.3 Relationships

7.2.1.3.1 AdditionalAttribute

The AdditionalAttribute relationship defines additional Aspects that can be specified for this patient.

It is not required for MVG-RIM needs.

Implementation note: although the conceptual model models additional attributes in a general way through Aspects, this does not mean that a practical implementation has to actual store these additional attributes in a physical table called Aspect. Any practical implementation may choose the physical representation that best suits its needs.

7.2.1.4 Keys

7.2.1.4.1 Primary Key

The primary key for the Patient entity is: Uid.

7.2.2 HospitalStay

7.2.2.1 Definition

A HospitalStay entity represents a stay of the patient in the hospital.

For the MVG-RIM needs only a limited set of attributes is required. These attributes are described.

Note that the conceptual model is general. Any additional attribute that would be useful in a more general case, can be added. These additional attributes are modeled as Aspects.

7.2.2.2 Attributes

7.2.2.2.1 Uid

The Uid attribute represents the universally unique identifier by which the patient stay in the hospital is identified by all sending systems.

7.2.2.2.2 From

The From attribute represents the moment in time corresponding to the intake of the patient in the hospital.

7.2.2.2.3 Until

The Until attribute represents the moment in time corresponding to the discharge of the patient from the hospital.

7.2.2.2.4 AlternateUid

The AlternateUid attribute represents the identifier by which this hospital stay will be known by the Portahealth system (STAYHOSP.STAYNUM).

7.2.2.2.5 PTypeId

The PTypeId attribute represents the kind of HospitalStay as intended by the registration guide (field A2_HOPSTYPE_P_nP; page A 6).

7.2.2.2.6 FTypeId

The FTypeId attribute represents the kind of HospitalStay as intended by the registration guide (field A2_HOPSTYPE_FAC; page A 6).

7.2.2.2.7 PatientUid

The PatientUid attribute represents the universally unique identifier by which the patient who is admitted is identified by all sending systems.

7.2.2.3 Relationships

7.2.2.3.1 AdditionalAttribute

The AdditionalAttribute relationship defines additional Aspects that can be specified for this hospital stay.

It is not required for MVG-RIM needs.

Implementation note: although the conceptual model models additional attributes in a general way through Aspects, this does not mean that a practical implementation has to actual store these additional attributes in a physical table called Aspect. Any practical implementation may choose the physical representation that best suits its needs.

7.2.2.4 Keys

7.2.2.4.1 Primary Key

The primary key for this HospitalStay entity is: Uid.

Note that it is not allowed to reuse the Uid of a HospitalStay for different patients. If in a specific hospital's situation the identification number of a HospitalStay is reused for different patient, it is the responsibility of the sending system to make the Uid unique. This can easily be done by concatenating the PatientUid and the non-unique identification number.
We have deliberately chosen not to model the more general case, because this would lead to a more complex model and because this situation is rare and can easily be solved by the sending system.

Note that two admissions of the same patient *cannot* have the same Uid for the HospitalStay. If in a specific hospital's situation the identification number of a HospitalStay is reused for several admissions, it is the responsibility of the sending system to make the Uid for each HospitalStay unique. This can easily be done by concatenating the non-unique identification number with the date of the admission. There is a caveat to this approach, when the sending system changes the

From attribute, the concatenated Uid (that also contains part of the admission date) must not change.
We have deliberately chosen not to model the more general case, because this would lead to a more complex model and because this situation is rare and can easily be solved by the sending system.

7.2.3 UnitStay

7.2.3.1 Definition

A UnitStay entity represents a stay of the patient in ward or unit during his hospital stay.

For the MVG-RIM needs only a limited set of attributes is required. These attributes are described.

Note that the conceptual model is general. Any additional attribute that would be useful in a more general case, can be added. These additional attributes are modeled as Aspects.

7.2.3.2 Attributes

7.2.3.2.1 Uid

The Uid attribute represents the universally unique identifier known to all sending systems by which the stay in a ward or unit is identified *as part of* the HospitalStay in HospitalUid.

The Uid attribute must thus be unique for a given HospitalStayUid, but may be reused for other HospitalStayUids.

Implementation hint: a simple way to generate this Uid is to use the From value or the sequence number of the patient's transfer during this hospital stay. There is a caveat: if the original From value changes of if the original sequence number changes, the Uid must not change. If the sending system is not able to maintain the original Uid, it is always possible to delete the complete HospitalStay and resend the HospitalStay completely with new values.

7.2.3.2.2 From

The From attribute represents the moment in time corresponding to the intake of the patient on the unit.

7.2.3.2.3 Until

The Until attribute represents the moment in time corresponding to the discharge of the patient from the unit.

7.2.3.2.4 HospitalStayUid

The HospitalStayUid attribute represents the universally unique identifier by which the hospital stay, to which this UnitStay belongs, is identified by all sending systems.

Note: if for any moment in time between HospitalStayUid.From and HospitalStay.Until, there is no corresponding UnitStay where this moment in time lies between the UnitStay.From and UnitStayUntil, the patient is considered to be temporarily absent from the hospital (vacation, external examination, etc…).

Note: if there exist UnitStays whose time interval [From, Unit] lies (partially) outside the time interval of HospitalStayUid [From,Until], then (this part of) these UnitStays does not count. (This part of) the UnitStays is considered not to exist.

7.2.3.2.5 UnitUid

The UnitUid attribute represents the universally unique identifier by which the unit, to which this UnitStay belongs, is identified by all sending systems.

7.2.3.2.6 IndexUid

The IndexUid attribute represents the universally unique identifier by which the bedindex, to which this UnitStay belongs, is identified by all sending systems.

7.2.3.2.7 SpecialtyUid

The SpecialtyUid attribute represents the universally unique identifier by which the specialty, to which this UnitStay belongs, is identified by all sending systems.

7.2.3.3 Relationships

7.2.3.3.1 AdditionalAttribute

The AdditionalAttribute relationship defines additional Aspects that can be specified for this unit stay.

It is not required for MVG-RIM needs.

Implementation note: although the conceptual model models additional attributes in a general way through Aspects, this does not mean that a practical implementation has to actual store these additional attributes in a physical table called Aspect. Any practical implementation may choose the physical representation that best suits its needs.

7.2.3.4 Keys

7.2.3.4.1 Primary Key

The primary key for the UnitStay entity is: HospitalStayUid, Uid.

7.3 Relationships

All relationships were described with the related entities. See above.

8 Caregivers

8.1 Overview

Caregivers describe the required administrative information with regard to the caregivers involved in the Acts & Facts.

The model here is sufficient for the MVG-RIM requirements.

In general, electronic patient records may extend these concepts to deal with more complex situations. This is why we have modeled that addition of additional information with regard to caregivers by using Aspects, along the same lines as is done in Patient & Stays.

THIS MODEL IS BASED ON INFORMATION REQUIRED BY THE REGISTRATION MANUAL OF NOVEMBER 2006.

The model allows the calculation of tables EMPLOPER and EMPLODAY and allows the validation that care that was given by a caregiver can be taken into account for scoring the MVG-

RIM-items, based on whether this caregiver is assigned to the Unit where the care took place, at the right moment in time and with the required qualifications.

8.2 Entities

8.2.1 Caregiver

8.2.1.1 Definition

A Caregiver entity represents a caregiver. For each caregiver is known at a specific moment in time:

- CaregiverQualificationType: see field P1_CODE_QUAL_FUNCTION in the registration guide.
- CaregiverFunctionType: see field P1_CODE_QUAL_FUNCTION in the registration guide.
- EmploymentHoursPerWeek: the number of hours this employee works

These values are valid from From until Until.

8.2.1.2 Attributes

8.2.1.2.1 Uid

The Uid attribute represents the universally unique identifier by which the caregiver is identified by all sending systems.

Implementation note: the Uid typically corresponds to the “logon name” by which the user is known in the sending system. When more than one sending system is involved, either a user has to have the same “logon name” in all of those sending systems, or the sending system has to normalize the “logon name” to a unique logon name for that user in the institution.

8.2.1.3 Keys

8.2.1.3.1 Primary Key

The primary key for the Caregiver entity is: Uid.

8.2.2 CaregiverEmployment

8.2.2.1 Definition

The CaregiverEmployment entity represents which caregiver is employed in which unit, with which qualification and for which period. The idea is that the CaregiverUnitEmployment entity tracks the long term work relationships of a caregiver.

This entity is required specifically for MVG-RIM needs. Its objective is to allow the calculation of table EMPLOPER.

8.2.2.2 Attributes

8.2.2.2.1 CaregiverUid

The CaregiverUid attribute represents the universally unique identifier by which the caregiver is identified whose employment is listed.

8.2.2.2.2 From

The From attribute represents the moment in time since when the caregiver is employed according to the other information in this record.

8.2.2.2.3 Until

The Until attribute represents the moment in time until when the caregiver is employed according to the other information in this record.

8.2.2.2.4 UnitUid

The UnitUid attribute represents the universally unique identifier by which the unit is identified in which the caregiver is employed.

8.2.2.2.5 QualificationTypeId

The QualificationTypeId attribute represents the type of qualification that the caregiver has been assigned. It can be:

- See UML diagram

8.2.2.2.6 FunctionTypeId

The FunctionTypeId attribute represents the type of function that the caregiver has been assigned in this unit. It can be:

- See UML diagram

8.2.2.2.7 EmploymentHoursPerWeek

The EmploymentHoursPerWeek attribute represents the normal number of hours this employee is working every week.

8.2.2.3 Relationships

p.m.

8.2.2.4 Keys

8.2.2.4.1 Primary Key

The primary key of the CaregiverUnitEmployment entity is: CaregiverUid, From.

Note that it is not possible that a caregiver has more than one function at the same time in a unit. It is equally impossible that a caregiver is employed to more than one unit at a time.

8.2.3 CaregiverAssignment

8.2.3.1 Definition

The CaregiverAssignment entity represents which caregiver is assigned to which unit and for which duration. The idea is that the CaregiverUnitAssignment entity tracks the working hours of a caregiver on a unit on a daily basis.

This entity is required specifically for MVG-RIM needs. Its objective is to allow the calculation of table EMPLODAY.

8.2.3.2 Attributes

8.2.3.2.1 CaregiverUid

The CaregiverUid attribute represents the universally unique identifier by which the caregiver is identified whose assignment is listed.

8.2.3.2.2 UnitUid

The UnitUid attribute represents the universally unique identifier by which the unit is identified to which the caregiver is assigned.

8.2.3.2.3 At

The At attribute represents the date of the assignment to this unit.

8.2.3.2.4 FromTime

The FromTime attribute represents the start of the working at that day.

8.2.3.2.5 UntilTime

The UntilTime attribute represents the end of the working at that day.

8.2.3.3 Relationships

p.m.

8.2.3.4 Keys

8.2.3.4.1 Primary Key

The primary key of the CaregiverUnitAssignment entity is: CaregiverUid, UnitUid, At, FromTime

8.3 Relationships

All relationships were described with the related entities. See above.

9 Change log

Based on feedback by the pilots and the sector and the registration manual, the following sections have been refined:

- 5.2.2.5
- 5.2.3.4
- 7
- 8

The model for Plan has changed (see 6):

- The attribute HasObjectives is removed. The information is now captured by the Evaluation itself.
- The attribute HasOrders is removed. It is not required any more for MVG-RIM.
- The attribute ContinuationOfPlanRef is removed. Under the current model, when a Plan is evaluated, no new version of the Plan need to be created. It is simply OK to add a Result “evaluation”. This largly simplifies working with Plans.
- PlanTypeId has been replaced by the combination of PlanKindId and PlanSpecifityId.
- Correction in the description of IndividualPlanOnPaperRef: “when this plan is not maintained” has been replaces by “when this plan is maintained”