Title: **FBM Example for Mapping to 19763-12 – ESA1**

Source: Baba Piprani, Canada

Status: FBM Study Group Expert

Action: For distribution to WG2

Date: 11 Oct 2012

1. FBM Study Group Example – ESA1

The following is a real project document provided to ISO from European Space Agency files that has kindly agreed to distribute to SC32 WG2 for use in the mapping exercise from FBM into 19763-12.

Section 1 of the document contains the FBM graphical specification and the accompanying textual specification.

Section 2 of the document contains the Logical (relational mapping) model as *derived* from the FBM Model.

I am seeking help from WG2 members and invite WG2 to provide mappings into 19763-12 for the graphical specification and textual specification from Section 1. , i.e. fully populate all the FBM conceptual model textual requirements in Part 12. Note that the textual specification is a derivative of the graphical specification, and includes FBM constraints that accompany or are derived from FBM constructs.

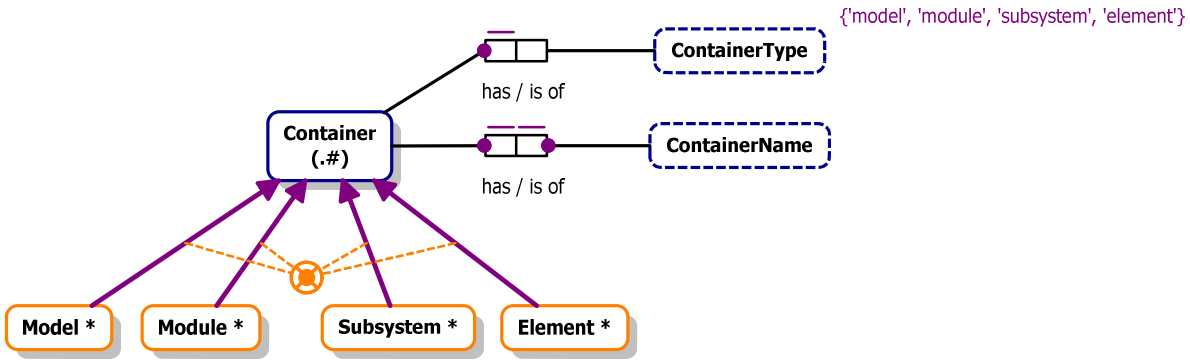
I was only able to map Section 2---the Logical model---into Part 12. Note that the Section 2 as shown is a pared down version that does not necessarily fully portray all the constraints declared in the FBM model.

1 FACT BASED Conceptual Modelling

1.1 Container

1.1.1 Introduction

1.1.2 Graphical specification



1.1.3 Textual specification

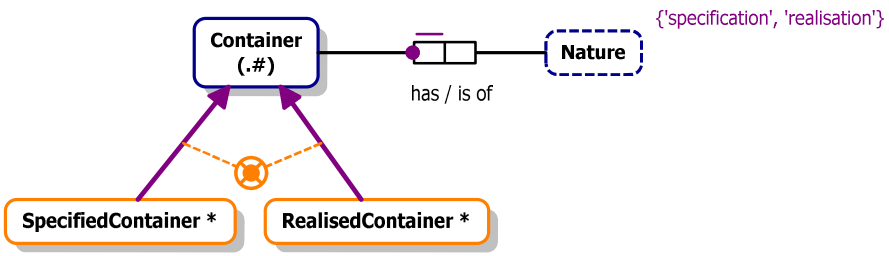
1) Container has container name.
   a) **Each** container has **exactly one** container name.
   b) **For each** container name, **exactly one** container has **that** container name.
2) Container has container type.
   a) **Each** container has **exactly one** container type.
   b) **It is possible that some** container type is of **more than one** container.
   c) **The possible values of** container type **are** 'model', 'module', 'sub-system', 'element'.
3) Container
   a) **For each** container, **exactly one of the following holds:**
      that container is **some** model;
      that container is **some** module;
      that container is **some** subsystem;
      that container is **some** element.
4) Model
   a) ***Each** model **is some** container **that** has **some** container type **where the possible value of that** container type **is** 'model'.
5) Module
   a) ***Each** module **is some** container **that** has **some** container type **where the possible value of that** container type **is** 'module'.
6) Subsystem
   a) ***Each** subsystem **is some** container **that** has **some** container type **where the possible value of that** container type **is** 'sub-system'.

7) Element
   a) ***Each** element **is some** container **that** has **some** container type **where the possible value of that** container type **is** 'element'.

1.2 Nature of container

1.2.1 Introduction

1.2.2 Graphical specification



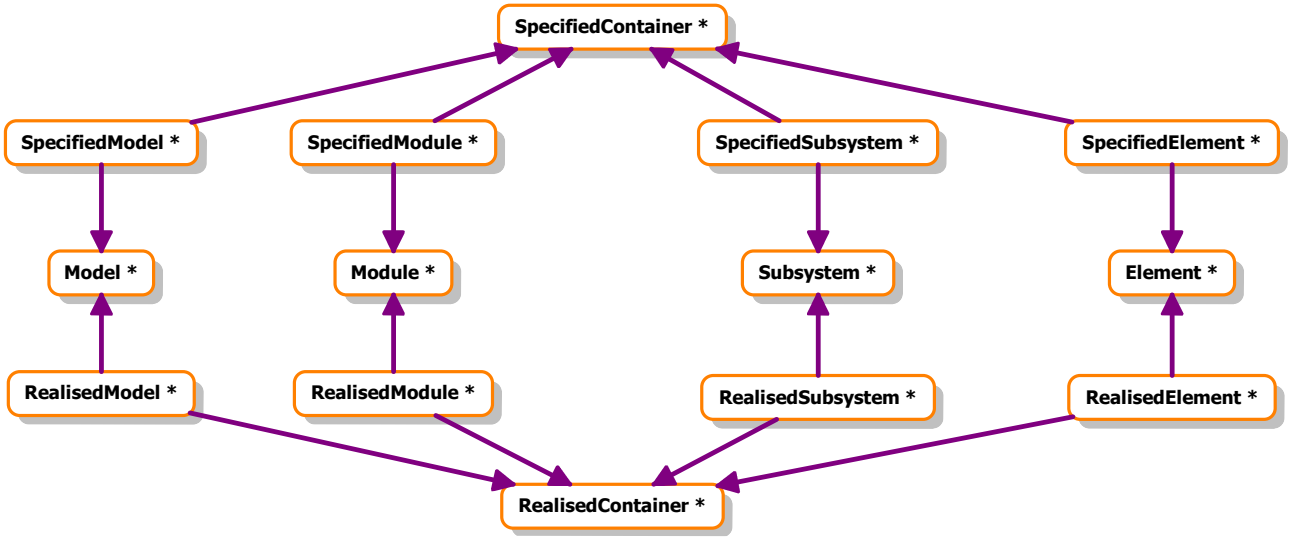
1.2.3 Textual specification

1) Container has nature.
   a) **Each** container has **exactly one** nature.
   b) **It is possible that some** nature is of **more than one** container.
2) Specified container
   a) ***Each** specified container **is some** container **that** has **some** nature **where the possible value of that** nature **is** 'specification'.
3) Realised container
   a) ***Each** realised container **is some** container **that** has **some** nature **where the possible value of that** nature **is** 'realisation'.

1.3 Specification and realisation

1.3.1 Introduction

1.3.2 Graphical specification



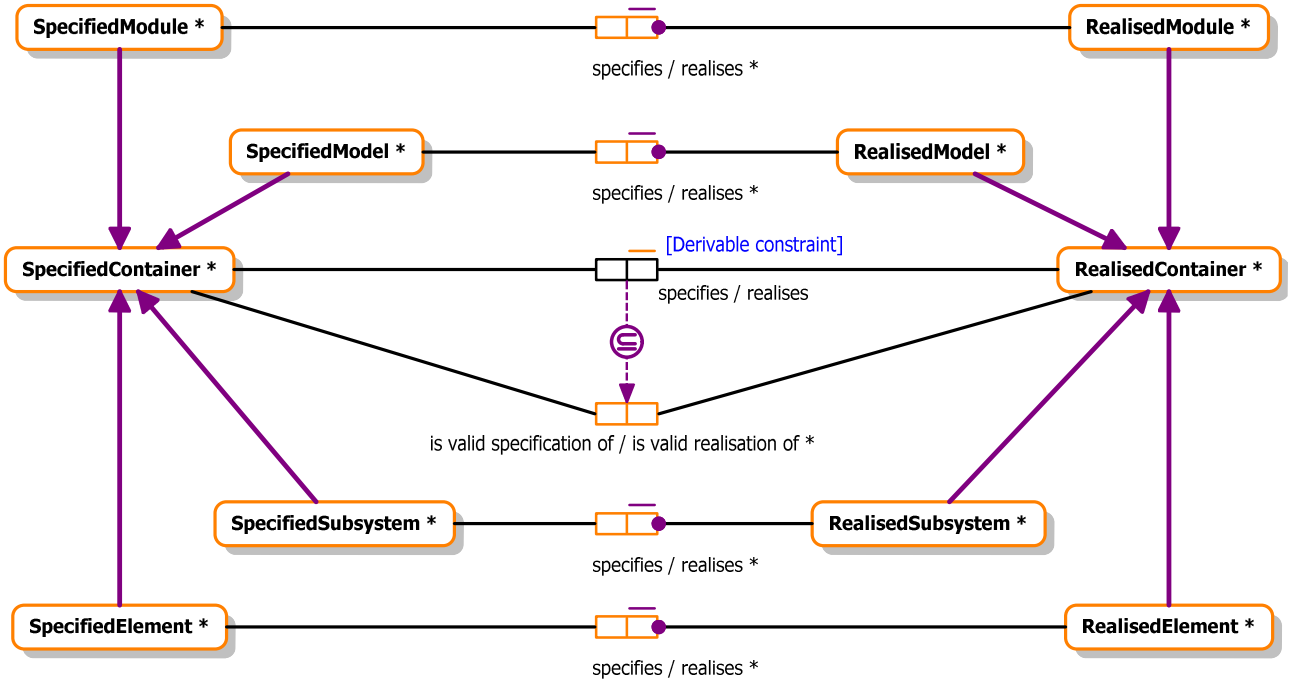
1.3.3 Textual specification

1) Specified model
   a) ***Each** specified model **is some** container **that** is **some** specified container **and** is **some** model.
2) Specified module
   a) ***Each** specified module **is some** container **that** is **some** specified container **and** is **some** module.
3) Specified subsystem
   a) ***Each** specified subsystem **is some** container **that** is **some** specified container **and** is **some** subsystem.
4) Specified element
   a) ***Each** specified element **is some** container **that** is **some** specified container **and** is **some** element.
5) Realised model
   a) ***Each** realised model **is some** container **that** is **some** realised container **and** is **some** model.
6) Realised module
   a) ***Each** realised module **is some** container **that** is **some** realised container **and** is **some** module.
7) Realised subsystem
   a) ***Each** realised subsystem **is some** container **that** is **some** realised container **and** is **some** subsystem.
8) Realised element
   a) ***Each** realised element **is some** container **that** is **some** realised container **and** is **some** element.

1.4 Physical instantiation

1.4.1 Introduction

1.4.2 Graphical specification



1.4.3 Textual specification

1) Specified container specifies realised container.
   a) **Each** realised container realises **at most one** specified container.
   b) **It is possible that some** specified container specifies **more than one** realised container.
2) Valid specification of realised container
   a) *Specified container is valid specification of realised container **if and only if** **that** specified container **is some** specified model **that** specifies **some** realised model **that is that** realised container
   or that specified container **is some** specified module **that** specifies **some** realised module **that is that** realised container
   or that specified container **is some** specified subsystem **that** specifies **some** realised subsystem **that is that** realised container
   or that specified container **is some** specified element **that** specifies **some** realised element **that is that** realised container.
   b) **If some** specified container specifies **some** realised container **then that** specified container is valid specification of **that** realised container.
3) Specification of realised model
   a) *Specified model specifies realised model **if and only if** **that** specified model **is some** specified container **that** specifies **some** realised container **that is that** realised model.

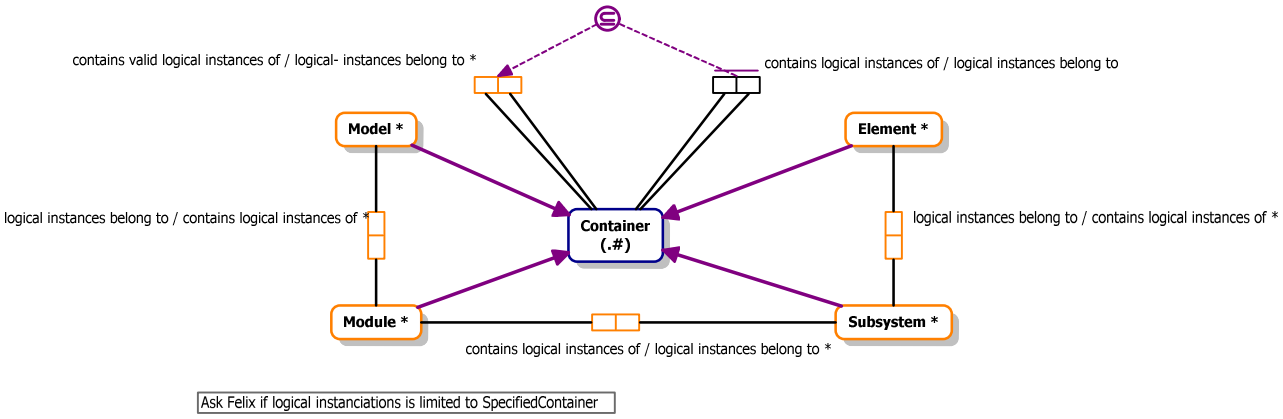
b) **Each** realised model realises **exactly one** specified model.
c) **It is possible that some** specified model specifies **more than one** realised model.

4) Specification of realised module
   a) ***Specified module specifies realised module **if and only if** **that** specified module **is some** specified container **that** specifies **some** realised container **that is that** realised module.
   b) **Each** realised module realises **exactly one** specified module.
   c) **It is possible that some** specified module specifies **more than one** realised module.
5) Specification of realised subsystem
   a) ***Specified subsystem specifies realised subsystem **if and only if** **that** specified subsystem **is some** specified container **that** specifies **some** realised container **that is that** realised subsystem.
   b) **Each** realised subsystem realises **exactly one** specified subsystem.
   c) **It is possible that some** specified subsystem specifies **more than one** realised subsystem.
6) Specification of realised element
   a) ***Specified element specifies realised element **if and only if** **that** specified element **is some** specified container **that** specifies **some** realised container **that is that** realised element.
   b) **Each** realised element realises **exactly one** specified element.
   c) **It is possible that some** specified element specifies **more than one** realised element.

1.5 Logical instantiation

1.5.1 Introduction

1.5.2 Graphical specification



1.5.3 Textual specification

1) Container contains logical instances of container.

a) **It is possible that some** container contains logical instances of **more than one** container
and that some container logical instances belong to **more than one** container.
b) **In each population of** container contains logical instances of container, **each** container, container **combination occurs at most once**.

2) Valid logical instantiations
a) *Container$_1$ contains valid logical instances of container$_2$ **if and only if**
that container$_1$ **is some** model **that** contains **some** logical instantiated- module$_1$ **that is that** container$_2$
or that container$_1$ **is some** module$_2$ **that** contains **some** logical instantiated-subsystem$_1$ **that is that** container$_2$
or that container$_1$ **is some** subsystem$_2$ **that** contains **some** logical-instantiated element **that is that** container$_2$.
b) **If some** container$_1$ contains logical instances of **some** container$_2$ **then that** container$_1$ contains valid logical instances of **that** container$_2$.

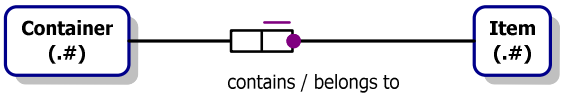
3) Module logical instances
a) *Model logical instances belong to module **if and only if**
that model **is some** container$_1$ **that** contains logical instances of **some** container$_2$ **that is that** module.

4) Subsystem logical instances
a) *Module contains logical instances of subsystem **if and only if**
that module **is some** container$_1$ **that** contains logical instances of **some** container$_2$ **that is that** subsystem.

5) Element logical instances
a) *Subsystem contains logical instances of element **if and only if**
that subsystem **is some** container$_1$ **that** contains logical instances of **some** container$_2$ **that is that** element.

1.6 Item

1.6.1 Graphical specification



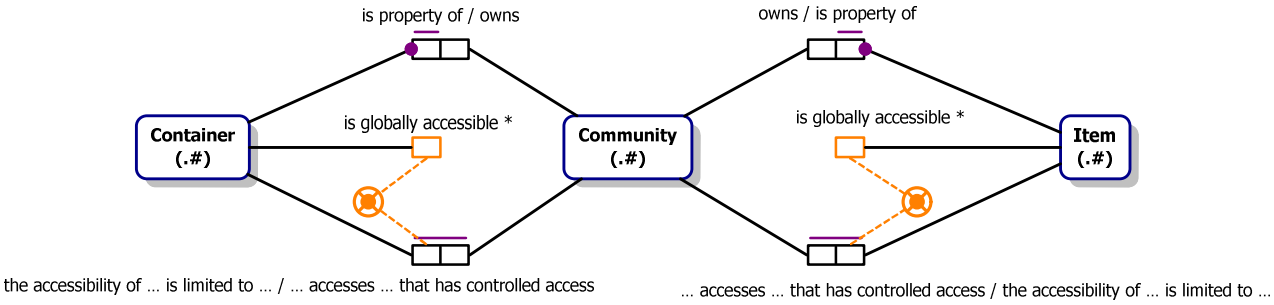
1.6.2 Textual specification

1) Container contains item
a) **Each** item belongs to **exactly one** container.
b) **It is possible that some** container contains **more than one** item.

1.7 Accessibility

1.7.1 Introduction

1.7.2 Graphical specification



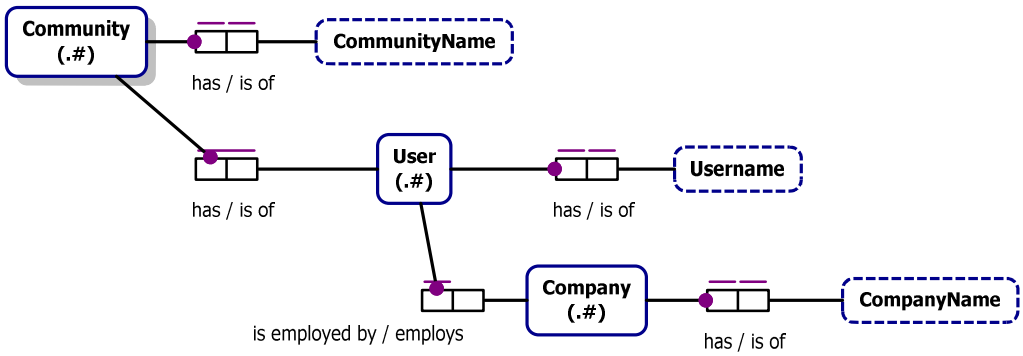
1.7.3 Textual specification

1) Container is property of community
   a) **Each** container is property of **exactly one** community.
   b) **It is possible that some** community owns **more than one** container.
2) Container controlled access
   a) **It is possible that** the accessibility of **some** container is limited to **more than one** community
      and that some community accesses **more than one** container that has controlled access.
   b) **In each population of** the accessibility of container is limited to community, **each** container, community **combination occurs at most once**.
3) Container global accessibility
   a) ***Container is globally accessible **if and only if**
      the accessibility of **that** container is limited to **no** community.
4) Item is property of community
   a) **Each** item is property of **exactly one** community.
   b) **It is possible that some** community owns **more than one** item.
5) Item controlled access
   a) **It is possible that** the accessibility of **some** item is limited to **more than one** community
      and that some community accesses **more than one** item that has controlled access.
   b) **In each population of** the accessibility of item is limited to community, **each** item, community **combination occurs at most once**.
6) Item global accessibility
   a) ***Item is globally accessible **if and only if**
      the accessibility of **that** item is limited to **no** community.

1.8 Community

1.8.1 Introduction

1.8.2 Graphical specification



1.8.3 Textual specification

1) Community has community name.
   a) **Each** community has **exactly one** community name.
   b) **Each** community name is of **at most one** community.
2) Membership
   a) **It is possible that some** community has **more than one** user **and that some** user is of **more than one** community.
   b) **In each population of** community has user, **each** community, user **combination occurs at most once**.
   c) **Each** community has **some** user.
3) User has username.
   a) **Each** user has **exactly one** username.
   b) **Each** username is of **at most one** user.
4) User is employed by company.
   a) **Each** user is employed by **exactly one** company.
   b) **It is possible that some** company employs **more than one** user.
5) Company has company name.
   a) **Each** company has **exactly one** company name.
   b) **Each** company name is of **at most one** company.

1.9 title

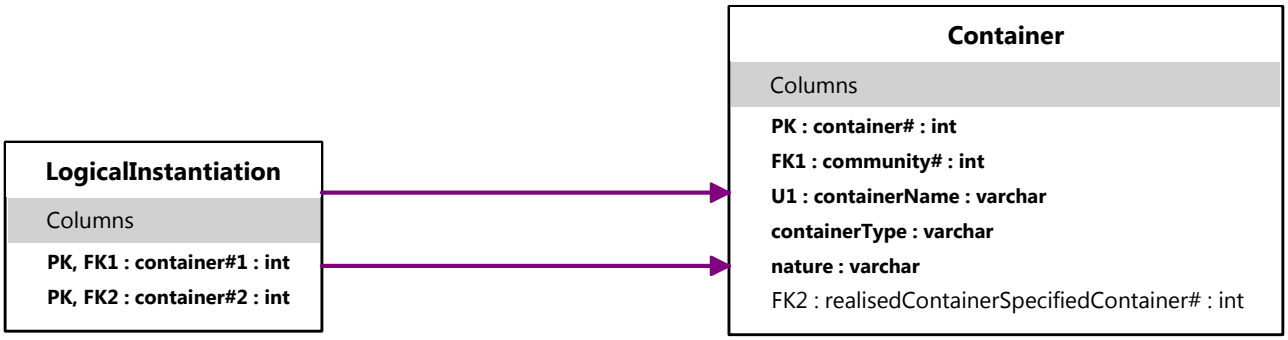
1.9.1 Introduction

1.9.2 Textual specification

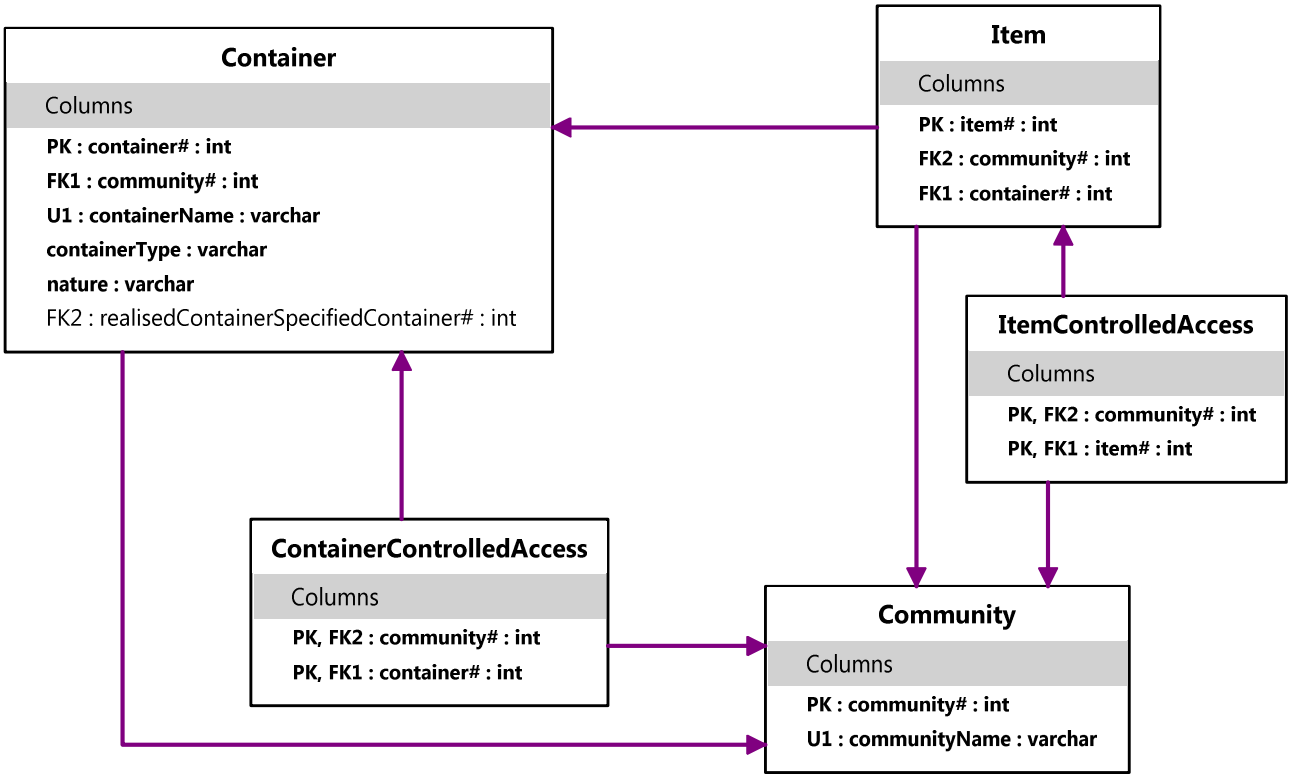
2 Logical Modelling

2.1 Relational model (automatically generated from the FBM conceptual model)

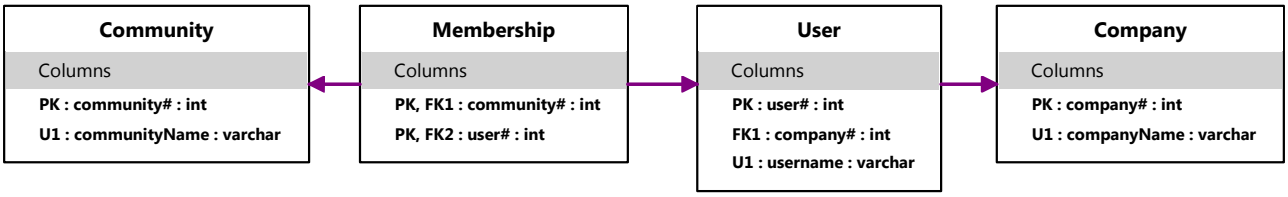
2.1.1



2.1.2



2.1.3


Community
Columns
PK : community# : int
U1 : communityName : varchar
Membership
Columns
PK, FK1 : community# : int
PK, FK2 : user# : int
User
Columns
PK : user# : int
FK1 : company# : int
U1 : username : varchar
Company
Columns
PK : company# : int
U1 : companyName : varchar