## Annex D. UML class diagrams and object descriptions

### Base Group

#### Base Artefacts Class Diagram

The *Identifiable Artefact* has three associations to *Contextual String* – one for each of name, description, and documentation. The value in the*Contextual String* is given a context by the *Context Key* which can be *Type* or *Language*.

There is no attempt in GSIM to model the administration of items in repositories such as the maintenance agency, versioning, repository functions. However, the *Identifiable Artefact* does have a link to *Administrative Details* where such details can be added using the GSIM extension methodology.



Figure 27. Base Artefacts Class Diagram

#### Organization Class Diagram

An *Organization Scheme* comprises *Organization Items*, each of which can be an *Individual* or *Organization Unit*can have a number of different *Contact Details.*

The *Individual*or *Organization Unit*can play zero or more recognized roles (*Organization Item Role*) in the maintenance (*Maintenance Agency*) data collection (*Data Provider*) and dissemination (*Data Consumer*) processes.



Figure 28. Organization Class Diagram

##### AdministrativeDetails

**Package:** Base

**Definition:** A placeholder for extensions to the GSIM model.

**Explanatory Text:** GSIM does not seek to replicate or embed constructs from the administration of objects held in metadata registries.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..\*** AdministrativeDetails.  IdentifiableArtefact. |  |

##### ContactDetails

**Package:** Base

**Definition:** A collection of modes and strings by which an *Organization Item* can be contacted.

**Explanatory Text:** Contact modes can include (but are not limited to) telephone, e-mail or fax. In these cases, the relevant strings would be the telephone number, e-mail address and fax number.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| contactType |  | 1..1 | String |
| name |  | 1..1 | String |
| mailAddress |  | 1..1 | String |
| emailAddress |  | 1..1 | String |
| telephone |  | 1..1 | String |
| cellPhone |  | 1..1 | String |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..\*** ContactDetails.  OrganizationItem. |  |

##### ContextKey

**Package:** Base

**Definition:** Gives semantic or structural meaning to the value of a *Contextual String.*

**Explanatory Text:** *Context Key* has two sub classes - *Type* and *Language*. For example: *Type* = Short Name, or *Language* = French

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Type.  ContextKey. |  |
|  | Language.  ContextKey. |  |
| has | **0..\*** ContextKey.context  ContextualString. |  |

##### ContextualString

**Package:** Base

**Definition:** A textual value, which is given context by one or more *Context Keys*.

**Explanatory Text:** A C*ontextual String* can be given context by one or more *Context Keys*. For example: *Type* = Short Name, or *Language* = French

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| value |  | 1..1 | string |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..\*** ContextualString.name  IdentifiableArtefact. | **Definition**  The Contextual String is the Name for the Identifiable Artefact. |
| has | **0..\*** ContextualString.documentation  IdentifiableArtefact. | **Definition**  The Contextual String is the Documentation for the Identifiable Artefact. |
| has | **0..\*** ContextualString.description  IdentifiableArtefact. | **Definition**  The Contextual String is the Description for the Identifiable Artefact |
| has | **0..\*** ContextKey.context  ContextualString. |  |

##### DataConsumer

**Package:** Base

**Definition:** An organization that uses data or metadata as input for further processing.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DataConsumer.  OrganizationItemRole. |  |

##### DataProvider

**Package:** Base

**Definition:** An organization, association, group or person who delivers information for a S*tatistical Activity*.

**Explanatory Text:** It is an organization, association, group or person that possesses statistical information (that it has collected, produced, bought or otherwise acquired) and is willing to supply those data and metadata to a statistical organization.

**Synonym:** data supplier

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DataProvider.  OrganizationUnit. |  |
|  | DataProvider.  OrganizationItemRole. |  |
| has | NonStructuredDataSet.  0..1 DataProvider. |  |
| providesDataFor | **0..\*** ProvisionAgreement.  **1** DataProvider. |  |

##### IdentifiableArtefact

**Package:** Base

**Definition:**  An abstract class that comprises the basic attributes and associations needed for identification, naming and other documentation.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| id |  | 1..1 | string |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| basedUpon | **0..\*** ChangeDefinition.  **0..\*** IdentifiableArtefact.input | Identifies the objects that require changing. Can be thought of as an input to the change. |
| has | **0..\*** AdministrativeDetails.  IdentifiableArtefact. |  |
| has | **0..\*** ContextualString.name  IdentifiableArtefact. | **Definition**  The Contextual String is the Name for the Identifiable Artefact. |
| has | **0..\*** ContextualString.documentation  IdentifiableArtefact. | **Definition**  The Contextual String is the Documentation for the Identifiable Artefact. |
| has | **0..\*** ContextualString.description  IdentifiableArtefact. | **Definition**  The Contextual String is the Description for the Identifiable Artefact |
| hasInput | 0..\* Assessment.  **0..\*** IdentifiableArtefact. |  |
| references | ProcessInput.  **0..1** IdentifiableArtefact.object |  |
| references | **0..1** IdentifiableArtefact.object  ProcessOutput. |  |
| represents | **0..\*** Representation.  **1..\*** IdentifiableArtefact. |  |
| resultsIn | ChangeDefinition.  **0..\*** IdentifiableArtefact.output | A ChangeNeed is likely to be defined by reference to other information objects. For example if the changeneed requires new data to be acquired or disseminated it will be necessary to define and reference variables and classifications etc. Can be thought of as an output of the change. |

##### Individual

**Package:** Base

**Definition:** A person who acts, or is designated to act towards a specific purpose.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Individual.  OrganizationItem. |  |

##### Language

**Package:** Base

**Definition:** The linguistic code used. This takes into account geographic variations, e.g. Canadian French or Australian English

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| value |  | 1..1 | languageCode |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Language.  ContextKey. |  |
|  | <anonymous>.  Language. |  |

##### MaintenanceAgency

**Package:** Base

**Definition:** The organization or expert body that maintains an artefact

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | MaintenanceAgency.  OrganizationItemRole. |  |

##### OrganizationItem

**Package:** Base

**Definition:** An abstract class which has two sub classes: *Organization Unit* and *Individual.*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | OrganizationUnit.  OrganizationItem. |  |
|  | Individual.  OrganizationItem. |  |
| has | **0..\*** ContactDetails.  OrganizationItem. |  |
| has | **0..\*** OrganizationItemRole.  **1** OrganizationItem. |  |
| has | **0..\*** OrganizationItem.  **1** OrganizationScheme. |  |

##### OrganizationItemRole

**Package:** Base

**Definition:** The function or activities of an *Organization Item*, in statistical processes such as collection, processing and dissemination.

**Synonym:** organization role

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | MaintenanceAgency.  OrganizationItemRole. |  |
|  | DataProvider.  OrganizationItemRole. |  |
|  | DataConsumer.  OrganizationItemRole. |  |
| has | **0..\*** OrganizationItemRole.  **1** OrganizationItem. |  |

##### OrganizationScheme

**Package:** Base

**Definition:**  A maintained collection of *Organization Items.*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..\*** OrganizationItem.  **1** OrganizationScheme. |  |

##### OrganizationUnit

**Package:** Base

**Definition:** A unique framework of authority within which a person or persons act, or are designated to act, towards some purpose.

**Synonym:** organization

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| unitType |  | 1..1 | char |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | OrganizationUnit.  OrganizationItem. |  |
|  | DataProvider.  OrganizationUnit. |  |
|  | StatisticalProgram.  **0..1** OrganizationUnit.reponsibleUnit |  |
| has | 0..\* BusinessCase.  **0..\*** OrganizationUnit.stakeholder |  |
| has | DataChannel.  **0..1** OrganizationUnit.operator |  |
| has | DataChannel.  **1..\*** OrganizationUnit.owner |  |
| hierarchy | **0..\*** OrganizationUnit.child  **1** OrganizationUnit.parent |  |

##### Type

**Package:** Base

**Definition:** Identifies a narrower meaning for the value in the *Contextual String*.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| value |  | 1..1 | type |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Type.  ContextKey. |  |
|  | <anonymous>.  Type. |  |

### Business Group

#### Information Request Class Diagram

An organization will react and change due to a variety of needs. In simple terms, these may be divided into at least two types of *Statistical Need*: an *Information Request* or an *Environment Change*.

When an organization receives an *Information Request,* it will often identify the information that a person or organization in the user community requires for a particular purpose. This request will commonly be defined in terms of a *Concept* or *Subject Field* that defines what the user wants to measure and the *Population* that identifies who the user wants data about).

This user community may include units within the organization as well as external to it. For example, a unit responsible for compiling National Accounts may need a new *Statistical Activity* to be initiated to produce new inputs to their compilation process.

Where an organization identifies an *Environment Change* this will identify the nature of the change. This may be specific to the organization in the form of reduced budget or new demands from stakeholders or may be a broader change such as the availability of new methodology or technology.

Once an organization has identified a need for change it will be further specified in the form of a *Change Definition*. This identifies the specific nature of the change in terms of its impacts on the agency or specific *Statistical Programs*. This *Change Definition* is used as an input into a *Business Case.* A successful outcome will either initiate a new *Statistical Program* or create a new *Statistical Program Design* which redefines the way the *Statistical Program* is carried out.



Figure 29. Information Request Class Diagram

#### Statistical Program Class Diagram

A *Statistical Program* is the overarching, ongoing activity that an organization undertakes to produce statistics (for example, Retail Trade survey). Each *Statistical Program* includes one or more *Statistical Program Cycles*. The *Statistical Program Cycle* is a repeating activity to produce statistics at a particular point in time (for example, Retail Trade survey March Quarter 2012).

A *Statistical Program* has an associated set of *Statistical Program Designs* which identify the methodology used for the *Statistical Program*. Only one *Statistical Program Design* is valid at any one time and is identified as being used by a particular *Statistical Program Cycle*. Changes to the methodology result in new *Statistical Program Designs* so over time each *Statistical Program* will have a series of Designs which provide a history of changes to the *Statistical Program*. The *Statistical Program Design* identifies the set of processes that are intended to be used to undertake the activity (*Process Step Design*, the resources required for the processes and a description of the methodology and context).

Each *Statistical Program Cycle* consists of one or more *Statistical Activities*. A *Statistical Activity* is the set of executed processes and the actual resources required as inputs and produced as outputs. It is analogous to the *Statistical Program Design* but represents the execution rather than design. For example in the design a *Data Set* of a particular type may be identified as an input whereas in the *Statistical Activity* the file name and location of the actual input *Data Set* would be identified.

The model identifies different types of *Statistical Activities* which represent the major steps in the statistical production process. Three types have been identified in the model but other types could be defined. The distinction between different types of activities and distinction of a*Statistical Activity* from a *Statistical Program Cycle* means that each iteration can be made up of multiple activities of the same or different types and these may or may not represent the process of collection through to dissemination. This model supports both the traditional approach of collecting data for a particular need, and also the emerging and future approach of collecting data and producing new outputs based on an existing *Data Resource* that is maintained and added to over time.



Figure 30. Statistical Program Class Diagram

#### Data-Channel Class Diagram

The *Instrument* object is the description of the tool that will be used to collect data. It could be a questionnaire, a set of requirements to develop software for gathering data, clinical procedures, etc. Although a questionnaire (or survey) is an *Instrument,* it is included in the model as a specialized object called *Survey Instrument. An Instrument* uses an *Instrument Control*.

Once the *Instrument* has been designed, it must be realized in the form of one or more*Instrument Implementation* objects. These objects can be printed forms, software programs, etc. The *Data Channel* uses the *Instrument Implementation* to request data and describes the technique used to do it by means of a *Mode.*

The *Mode* object represents the way the information collection process is going to be conducted and in this way, “how” the *Data Channel* is going to be used, the following table represents some examples of the relation among *Instrument, Instrument Implementation, Mode* and *Data Channel.*



Figure 31. Data-Channel Class Diagram

#### Instrument Control Class Diagram

The purpose of the *Instrument Control* is to record the flow of an *Instrument* (such as a questionnaire), its use of questions, and additional component parts. The *Instrument* is composed of a series of *Instrument Control* nested inside a top level *Instrument Control*. The*Question Block* is a sub class of *Instrument Control* thus enabling a block of *Instrument Control* to be specified and made reusable.

The *Control Transition*contains the logic to determine the next *Instrument Control.* The *Control Transition* uses *Rules* to determine either the next *Instrument Control,* or the next or sequencing of instance type of controls (*Instance Statement Item, Instance Question, Instance Interviewer Instruction. Instance Question Block*).

* the *Instance Statement*is associated to a (reusable) *Statement*,
* the *Instance Question* is associated to a (reusable) *Question*,
* the Instance Interviewer Instruction is associated to a (reusable) Interviewer Instruction and
* the Instance Question Blockis associated to a (reusable) Question Block.



Figure 32. Instrument Control Class Diagram

#### Question Group Class Diagram

Each *Question* can take its semantic and population context from one or more *Variables*. The *Question* must specify the valid content of a response to the *Question*. This response domain is linked to a *Value Domain* which can be enumerated (a list of valid of valid responses) or described (a type of valid response such as a date, a number, text).

*Questions* can be nested by means of the *Multiple Question Item*which is in itself a *Question* and therefore can link to *Variables* and a *Value Domain*. As the *Multiple Question Item* can link to another *Multiple Question Item* this enables question hierarchies to be built. At the bottom of each hierarchy there can only be single *Questions*.

Questions can be maintained in a Question Group.



Figure 33. Question Group Class Diagram

##### AcquisitionActivity

**Package:** Business

**Definition:** The set of executed processes and the actual resources required as inputs and produced as outputs to acquire data about a given *Population* for a particular reference period. It includes the process and resources required to acquire data in a *Statistical Program* consisting of gathering data via one or more *Data Channels* in order to create or feed one or more *Data Resources*.

**Explanatory Text:** This object holds *Statistical Activity* information that relates specifically to data collection or acquisition. It inherits the relationships and attributes from the *Statistical Activity* type.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | AcquisitionActivity.  StatisticalActivity. |  |
| defines | **1..\*** ChannelActivitySpecification.  AcquisitionActivity. |  |
| describedBy | **0..1** CollectionDescription.  **0..\*** AcquisitionActivity. |  |

##### AcquisitionDesign

**Package:** Business

**Definition:** The specification of the resources required and processes used and description of relevant methodological information for a set of activities to collect data about a given *Population*.

**Explanatory Text:** This object holds *Statistical Program Design* information that relates specifically to data collection or acquisition. It inherits the relationships and attributes from the *Statistical Program Design* type. Related to *Acquisition Design* is *Acquisition Activity*, which holds the detailed information about the conduct of *the Acquisition Activity* for a single reference period, The *Acquisition Design* describes the methodology and design elements that are intended to apply across all *Acquisition Activities* until such time as a decision is made to alter the design.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | AcquisitionDesign.  StatisticalProgramDesign. |  |
| defines | **1..\*** ChannelDesignSpecification.  AcquisitionDesign. |  |
| describedBy | **0..1** CollectionDescription.  **0..\*** AcquisitionDesign. |  |

##### Assessment

**Package:** Business

**Definition:** An activity to analyze quality or effectiveness and consider available options.

**Explanatory Text:** The *Assessment* is a generic class that regroups different types of more specific assessments. An example of *Assessment* is a SWOT assessment that identifies the Strengths, Weaknesses, Opportunities and Threats of a specified proposal. Another example is a *Gap Analysis* that formalizes the difference between the current situation and the state to reach due to certain requirements. An *Assessment* can use various objects as inputs, whether they are the main objects that the *Assessment* is about or auxiliary information objects that help the accomplishment of the assessment.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | EvaluationAssessment.  Assessment. |  |
|  | GapAnalysis.  Assessment. |  |
| hasInput | 0..\* Assessment.  **0..\*** IdentifiableArtefact. |  |
| isAssessed | 1..\* Assessment.  BusinessCase. | A BusinessCase will include an Assessment of the need for change |

##### BusinessCase

**Package:** Business

**Definition:** A proposal for a body of work that will deliver outputs designed to achieve outcomes. A *Business Case* will provide the reasoning for initiating a new *Statistical Program Design* for a *Statistical Program,* as well as the details of the change proposed.

**Explanatory Text:** A *Business Case* is produced as a result of a detailed consideration of a *Change Definition*. It sets out a plan for how the change described by the *Change Definition* can be achieved. A *Business Case* usually comprises various evaluations, for example a SWOT assessment, or *Gap Analyses* for the different solutions that are considered for satisfying the *Statistical Need*. The *Business Case* will also specify the stakeholders that are impacted by the *Statistical Need* or by the different solutions that are required to implement it.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| basedOn | **0..\*** ChangeDefinition.  0..\* BusinessCase. | A ChangeNeed may have been further defined as a ChangeDefinition. This will be used as a key input into the BusinessCase process. |
| has | 0..\* BusinessCase.  **0..\*** OrganizationUnit.stakeholder |  |
| initiates | 1..\* BusinessCase.  StatisticalNeed. |  |
| initiates | 1..\* BusinessCase.  **0..\*** StatisticalProgramDesign.designChange |  |
| initiates | BusinessCase.  **0..\*** StatisticalProgram. |  |
| isAssessed | 1..\* Assessment.  BusinessCase. | A BusinessCase will include an Assessment of the need for change |

##### ChangeDefinition

**Package:** Business

**Definition:** A structured, well-defined specification for a proposed change.

**Explanatory Text:** A related object - the *Statistical Need* - is a change expression as it has been received by an organization. A *Statistical Need* is a raw expression of a proposed change, and is not necessarily well-defined. A *Change Definition* is created when a *Statistical Need* is analyzed by an organization, and expresses the raw need in well-defined, structured terms.

A *Change Definition* does not assess the feasibility of the change or propose solutions to deliver the change - this role is satisfied by the *Business Case* object. The precise structure or organization of a *Change Definition* can be further specified by rules or standards local to a given organization.

Once a *Statistical Need* has been received, the first step is to do the conceptual work to establish what it is we are trying to measure. The final output of this conceptual work is the *Change Definition.*

The next step is to assess how we are going to make the measurements - to design a solution and put forward a proposal for a body of work that will deliver on the requirements of the original *Statistical Need*. The *Change Definition* is an input to this *Process Step* and the final *Business Case* is an output. Depending on the needs of individual agencies a *Change Definition* may be created before or after a *Business Case* has been created, or even created to a basic extent before the *Business Case* development and further developed after a *Business Case* has been approved and a decision made to proceed with the change.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| basedOn | **0..\*** ChangeDefinition.  0..\* BusinessCase. | A ChangeNeed may have been further defined as a ChangeDefinition. This will be used as a key input into the BusinessCase process. |
| basedUpon | **0..\*** ChangeDefinition.  **0..\*** IdentifiableArtefact.input | Identifies the objects that require changing. Can be thought of as an input to the change. |
| resultsIn | ChangeDefinition.  **0..\*** IdentifiableArtefact.output | A ChangeNeed is likely to be defined by reference to other information objects. For example if the changeneed requires new data to be acquired or disseminated it will be necessary to define and reference variables and classifications etc. Can be thought of as an output of the change. |
| specifies | **1..\*** ChangeDefinition.  StatisticalNeed. |  |

##### ChannelActivitySpecification

**Package:** Business

**Definition:** The description of the *Data Channel* made at run time.

**Explanatory Text:** This object is a specialization of a *Data Channel* and is used to describe the behaviour of a *Data Channel* at execution time.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ChannelActivitySpecification.  DataChannel. |  |
| defines | **1..\*** ChannelActivitySpecification.  AcquisitionActivity. |  |

##### ChannelDesignSpecification

**Package:** Business

**Definition:** The description of the *Data Channel* made at design time.

**Explanatory Text:** This object is a specialization of a *Data Channel*, and is used to make the design of the characteristics of a *Data Channel* before using it.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ChannelDesignSpecification.  DataChannel. |  |
| defines | **1..\*** ChannelDesignSpecification.  AcquisitionDesign. |  |

##### CollectionDescription

**Package:** Business

**Definition:** The set of information that provides a textual description of the processes and methods used to undertake an *Acquisition Activity*. It provides a set of contextual and reference metadata about the acquisition process.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| describedBy | **0..1** CollectionDescription.  **0..\*** AcquisitionActivity. |  |
| describedBy | **0..1** CollectionDescription.  **0..\*** AcquisitionDesign. |  |

##### ControlTransition

**Package:** Business

**Definition:** Governs how to determine the next *Instrument Control* based on factors such as the current location in the *Instrument*, the response to the previous questions etc.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **1** ControlTransition.  InstrumentControl. |  |
| invokes | ControlTransition.  **0..1** InstrumentControl. |  |
| uses | 0..\* Rule.  **0..\*** ControlTransition. |  |

##### DataChannel

**Package:** Business

**Definition:** A means of exchanging data.

**Explanatory Text:** A *Data Channel* is an abstract object that describes the means for communicating with *Data Resource(s)*. The *Data Channel* identifies the *Instrument Implementation*, *Mode*, and *Data Resource* that are to be used in a process. In some cases the *Data Channel* that is used by the *Data Provider* to send its responses could be different that the one used by the statistical office or organization to request information; the statistical office may put electronic formats that can be downloaded by the *Data Provider* and once answered returned by traditional mail. Two specialized objects are used to implement this abstract object: *Channel Design Specification* used at design time and *Channel Activity Specification* used at run time.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| dataSource |  | 1..1 | DataSource |
| status |  | 1..1 | channelStatus |
| activationDate | Date when the data channel was set up. | 1..1 | date |
| terminationDate | Date when the data channel was deactivated | 1..1 | date |
| temporalPattern |  | 1..1 | temporalPattern |
| direction | Direction of the exchanges on the data channel: one way into the statistical agency, one way out or both ways. | 1..1 | channelDirection |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ChannelActivitySpecification.  DataChannel. |  |
|  | ChannelDesignSpecification.  DataChannel. |  |
| has | DataChannel.  **0..1** OrganizationUnit.operator |  |
| has | DataChannel.  **1..\*** OrganizationUnit.owner |  |
| uses | **0..1** InstrumentImplementation.  DataChannel. |  |
| uses | 0..1 Mode.  DataChannel. |  |
| uses | 0..\* DataResource.  DataChannel. |  |

##### DesignContext

**Package:** Business

**Definitions:** Methodological metadata that provide the basis for the specification of the information objects required as input to and output from the *Process Step Design* including *Process Method* and *Rules*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..1** DesignContext.  StatisticalProgramDesign. |  |

##### DisseminationActivity

**Package:** Business

**Definition:** The set of executed processes and the actual resources required as inputs and produced as outputs in the dissemination of data for a given P*opulation* for a particular reference period, or of metadata. It describes the process and resources required in the dissemination of data and metadata in a *Statistical Program*.

**Explanatory Text:** This object holds *Statistical Activity* information that relates specifically to data and metadata dissemination. It inherits the relationships and attributes from the *Statistical Activity* type. A special type of *Dissemination Activity* is *Publication Activity*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | PublicationActivity.  DisseminationActivity. |  |
|  | DisseminationActivity.  StatisticalActivity. |  |
| performs | **0..\*** DisseminationActivity.  **0..\*** DisseminationService. |  |

##### DisseminationDesign

**Package:** Business

**Definition:** The specification of the resources required and processes used and description of relevant methodological information for a set of activities to disseminate data about a given *Population,* or metadata.

**Explanatory Text:** This object holds *Statistical Program Design* information that relates specifically to dissemination. It inherits the relationships and attributes from the *Statistical Program Design* type.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DisseminationDesign.  StatisticalProgramDesign. |  |

##### EnvironmentChange

**Package:** Business

**Definition:** A requirement for change (type of *Statistical Need*) that originates from a change in the operating environment of the statistical activity.

**Explanatory Text:** An *Environment Change* reflects variations in the context of execution of the S*tatistical Activity* that create a need for a modification in the way that this activity is conducted. *Environment Changes* can be of different origins and also take different forms. They can result from a precise event (budget cut, new legislation enforced) or from a progressive process (technical or methodological progress, application or tool obsolescence). Other examples of *Environment Changes* include the availability of a new *Data Resource*, the opportunity for new collaboration between agencies, etc.

*Environment Change* objects may be structured in very diverse ways, but an object will usually group text material describing the type of change that has occurred and created the need for change. This allows the statistical organization to document precisely the (possibly multiple) changes in environment that have led to the *Statistical Need*.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| changeOrigin | external or internal | 1..1 | Origin |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | EnvironmentChange.  StatisticalNeed. |  |

##### EvaluationAssessment

**Package:** Business

**Definition:** A type of *Assessment* that evaluates the process outputs of a statistical activity based on a formalized methodological framework.

**Explanatory Text:** The evaluation can be done in regard to various characteristics of the output, for example its quality, the efficiency of the production process, its conformance to a set of requirements, etc. The result of an *Evaluation Assessment* can lead to the creation of a *Statistical Need*: in this case, the *Statistical Need* will reference the *Evaluation Assessment* for traceability and documentary purposes.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | EvaluationAssessment.  Assessment. |  |
| evaluationOf | **0..\*** EvaluationAssessment.  **1..\*** StatisticalNeed. |  |

##### GapAnalysis

**Package:** Business

**Definition:** An expression of the difference (the 'gap') between the current state and a desired future state.

**Explanatory Text:** A *Gap Analysis* is a type of *Assessment* that compares the actual state of the activity with a potential state that would correspond to the implementation of a change. An organization will list the factors that define its current state and what is needed to reach its target state. This will for example document a *Business Case* and help to take the decision to implement the change or not.

**Synonyms:** need assessment

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| dateAssessed |  | 1..1 | date |
| currentState | Description of what currently exists | 1..1 | String |
| futureState | Description of what is required or desired | 1..1 | String |
| gap | Description of the difference between current and future state | 1..1 | String |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | GapAnalysis.  Assessment. |  |

##### InformationRequest

**Package:** Business

**Definition:** An outline of a need for new data or metadata required for a particular purpose.

**Explanatory Text:** An *Information Request* is a special case of *Statistical Need* that comes in a more organized way, for example by specifying on which S*ubject Field* the information is required, or what type of C*oncept* is to be measured, or even the type of U*nits* that are under consideration. The *Information Request* can for example be expressed internally, or by another statistical organization or authority.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | InformationRequest.  StatisticalNeed. |  |
| hasContext | InformationRequest.  **0..\*** SubjectField.informationAbout |  |
| hasContext | InformationRequest.  **0..\*** Population.informationOn |  |
| hasContext | InformationRequest.  **0..\*** Concept.informationAbout |  |

##### InstanceInterviewerInstruction

**Package:** Business

**Definition:** The use of an *Interviewer Instruction*in a particular*Instrument***.**

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..\*** InstanceInterviewerInstruction.  **1** InstrumentControl. |  |
| uses | **1** InterviewerInstruction.  InstanceInterviewerInstruction. |  |

##### InstanceQuestion

**Package:** Business

**Definition:** The use of a *Question* in a particular*Instrument***.**

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..\*** InstanceQuestion.  **1** InstrumentControl. |  |
| uses | 1 Question.  **0..\*** InstanceQuestion. |  |
| uses | **1** InstanceVariable.  **0..\*** InstanceQuestion. |  |

##### InstanceQuestionBlock

**Package:** Business

**Definition:** The use of a *Question Block* in a particular*Instrument***.**

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..\*** InstanceQuestionBlock.  **1** InstrumentControl. |  |
| uses | **1** QuestionBlock.  **0..\*** InstanceQuestionBlock. |  |

##### InstanceStatement

**Package:** Business

**Definition:** The use of a *Statement*in a particular*Instrument***.**

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..\*** InstanceStatement.  **1** InstrumentControl. |  |
| uses | 1 Statement.  **0..\*** InstanceStatement. |  |

##### Instrument

**Package:** Business

**Definition:** A tool conceived to record the information that will be obtained from the *Observation Units.*

**Explanatory Text:** The *Instrument* describes the tool used to collect data. It could be a traditional survey, a set of requirements for a software collection program, a clinical procedure, etc.

*Instrument* is described from the perspective of the statistical organization collecting the data. It includes the special type of Instrument used for the explicit purpose of gathering data through a questionnaire (Survey Instrument). The behavior and characteristics of a concrete *Instrument* is determined by an *Instrument Implementation*. Several implementations can be based in the same *Instrument* giving the possibility of using multiple channels and to apply different collection techniques (*Modes*) to gather data.

An example of this is when a printed format to collect information for a survey is substituted by a software program; in both cases the *Instrument* will collect the data from the *Unit* but the behavior of the *Instrument* will be different accordingly with its implementation.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| objective | A tool designed for recording information to be obtained from the observation units in a project to generate statistical information  A tool designed for recording information to be obtained from the observation units in a project to generate statistical information  Each one of the objectives that the instrument will cover | 1..1 | String |
| dateIssued | Date in which the instrument was made official/published | 1..1 | date |
| replacementDate | Planned date for the replacement of the instrument for other version. It can contain the date in which the new version of the Instrument was replaced | 1..1 | date |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | SurveyInstrument.  Instrument. |  |
| has | **0..\*** InstrumentImplementation.implementation  1 Instrument. |  |
| uses | 0..\* Instrument.  **0..1** InstrumentControl.topLevelControl |  |

##### InstrumentControl

**Package:** Business

**Definition:** A record of the flow of an *Instrument* and its use of *Questions*, *Interviewer* *Instructions* and *Statements*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **1** ControlTransition.  InstrumentControl. |  |
|  | QuestionBlock.  InstrumentControl. |  |
| has | **0..\*** InstanceInterviewerInstruction.  **1** InstrumentControl. |  |
| has | **0..\*** InstanceQuestion.  **1** InstrumentControl. |  |
| has | **0..\*** InstanceStatement.  **1** InstrumentControl. |  |
| has | **0..\*** InstanceQuestionBlock.  **1** InstrumentControl. |  |
| invokes | ControlTransition.  **0..1** InstrumentControl. |  |
| uses | **1** InstrumentControl.topLevelControl  SurveyInstrument. |  |
| uses | 0..\* Instrument.  **0..1** InstrumentControl.topLevelControl |  |

##### InstrumentImplementation

**Package:** Business

**Definition:** A concrete and usable tool for gathering information based on the rendering of the description made by an *Instrument*.

**Explanatory Text:** This represents an implementation of an *Instrument*. It describes the way in which an *Instrument* has been translated from a design to a concrete tool. It could represent a printed form, a software program made following a specific technological paradigm (web service, web scraping robot, etc.), the software used by a specialized device to collect data, etc. When it describes a *Survey Instrument*, it can contain descriptions of how each construct (e.g. *Questions*, *Value Domains*, validation *Rules* contained in the *Instrument*) is implemented.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| dateIssued | Date in which the instrument implementation was created. Not necessary if it’s the same the Instrument was published | 1..1 | date |
| replacementDate | Planned date for the replacement of the instrument implementation for other version. It can contain the date in which the new version of the Instrument implementation was replaced. | 1..1 | date |
| type | Describes the kind of instrument (survey, scraping tool, measurement device, etc.) | 1..1 | instrumentType |
| media | Description of the kind of media conceived for the use of the Instrument (printed, electronic, etc.) | 1..1 | mediaType |
| supportArtifacts | A list of devices, software programs, storage media, gadgets or other tools needed to support the use of the Instrument. | 1..1 | String |
| detailDocument | Reference to a document containing details of the implementation of the main elements of a survey instrument | 1..1 | URI |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **0..\*** InstrumentImplementation.implementation  1 Instrument. |  |
| uses | **0..1** InstrumentImplementation.  DataChannel. |  |

##### InterviewerInstruction

**Package:** Business

**Definition:** Directions given to an interviewer to aid the completion of the *Instrument.*

**Explanatory Text:** Example: "Show respondent prompt card before reading the question”

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| uses | **1** InterviewerInstruction.  InstanceInterviewerInstruction. |  |

##### Mode

**Package:** Business

**Definition:** A set of characteristics that describe the technique (the "how") used for the data acquisition through a given *Data Channel* based on a specific *Instrument Implementation.*

**Explanatory Text:** While the *Data Channel* describes the means used for data acquisition, the *Instrument* describes the "what" (i.e. the content, for example, in terms of questions in a questionnaire or a list of agreed time series codes in a data exchange template) and an *Instrument Implementation* describes the tool used to apply the *Instrument*; the *Mode* describes "how" the *Data Channel* is going to be used. The *Mode* is relevant for all types of *Data* *Channels*, *Instrument* *Implementations* and *Instruments* and can change over time. The list of *Modes* will potentially grow in the future and vary from organization to organization.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| completionType | How are the questions stated in the instrument completed/answered?  Examples: self completion, interviewer, data retrieval, data provision | 1..1 | String |
| format | Physical / technical format of the instrument  Examples: paper form, electronic form, SDMX DSD, OpenData, FTP | 1..1 | String |
| instrumentType | Type of content of the instrument  Examples: questionnaire, diary, query, navigation rules | 1..1 | String |
| inwardTransmission | Type of transmission of the completed Instrument (= the data) from the Responding Unit, Data Provider, or Data Resource to the statistical agency  Examples: mail, drop off, telephone, face-to-face, email, web service, software agent, file transfer | 1..1 | String |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| uses | 0..1 Mode.  DataChannel. |  |

##### MultipleQuestionItem

**Package:** Business

**Definition:** A construct that has all of the properties of a *Question* but additionally links to sub questions.

**Explanatory Text:** A *Multiple Question Item* is a specific type of *Question*.

**Synonyms:** question grid

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | MultipleQuestionItem.  Question. |  |
| hasSubQuestion | **0..\*** MultipleQuestionItem.  **1..\*** Question.subQuestion |  |

##### ProductionActivity

**Package:** Business

**Definition:** The set of executed processes and the actual resources required as inputs and produced as outputs in the production of data for a given *Population* for a particular reference period. It describes the process and resources required in the production of data in a *Statistical Program*.

**Explanatory Text:** These objects hold *Statistical* *Activity* information that relates specifically to data production. It inherits the relationships and attributes from the *Statistical* *Activity* type.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ProductionActivity.  StatisticalActivity. |  |

##### ProductionDesign

**Package:** Business

**Definition:** The specification of the resources required and processes used and description of relevant methodological information for a set of activities to process data about a given *Population*.

**Explanatory Text:** This object holds *Statistical Program Design* information that relates specifically to production - the act of taking data that have been collected and transforming them. It inherits the relationships and attributes from the *Statistical Program Design* type.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ProductionDesign.  StatisticalProgramDesign. |  |

##### PublicationActivity

**Package:** Business

**Definition:** The mechanism for creating structured, static content in response to an internal trigger.

**Explanatory Text:** A *Publication Activity* is a specific type of *Dissemination Activity*. A *Publication Activity* is triggered by an internal need to create a new *Product*. This is most commonly based on knowledge about a general need of potential consumers or the objective to actively provide information to consumers. Examples are the writing, editing and approval of a press release, web article or publication.

A *Publication Activity* may make use of *Dissemination Services* to get the necessary input. A *Publication Activity* may interpret or transform (e.g. visualize) statistical data, but cannot do any statistical processing.

A *Publication Activity* produces a *Product* and makes this available to *Dissemination Services* (possibly through an *Information Resource*) for the actual dissemination.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | PublicationActivity.  DisseminationActivity. |  |
| createdFrom | 0..\* Product.  **1** PublicationActivity. |  |

##### Question

**Package:** Business

**Definition:** Describes the text used to interrogate a respondent, the *Concept* that is measured and the allowed responses.

**Explanatory Text:** One specific type of *Question* is the *Multiple Question Item.*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | MultipleQuestionItem.  Question. |  |
| comprises | **1** QuestionGroup.  0..\* Question. |  |
| has | **1** ValueDomain.responseDomain  0..\* Question. | Specification of the valid response for the question in terms of either a set of pre-defined values or as a type of data such as a date, integer etc. |
| hasSemanticOf | 0..\* Question.  0..\* Variable. |  |
| hasSubQuestion | **0..\*** MultipleQuestionItem.  **1..\*** Question.subQuestion |  |
| uses | 1 Question.  **0..\*** InstanceQuestion. |  |

##### QuestionBlock

**Package:** Business

**Definition:** A set of Questions, Interviewer Instructions and Statements which are used together.

**Explanatory Text:** A statistical organization will often have a number of *Question Blocks* which they reuse in a number of *Instruments*. Examples of *Question Blocks* include:

* Household Question Block
* Income Question Block
* Employment Question Block

**Synonym:**  question module

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | QuestionBlock.  InstrumentControl. |  |
| uses | **1** QuestionBlock.  **0..\*** InstanceQuestionBlock. |  |

##### QuestionGroup

**Package:** Business

**Definitions:** A set of *Questions* which are gathered or stored together for the purpose of discovery.

**Explanatory Text:** *Questions* in *Question Groups* are similar in some way (for example, all the *Questions* relate to obesity).

*Questions* *Groups* are often found in databases that can be searched to find *Questions* which meet specific criteria

**Synonyms:** question pool, question bank

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| comprises | **1** QuestionGroup.  0..\* Question. |  |

##### Statement

**Package:** Business

**Definition:** A report of facts in an *Instrument*

**Explanatory Text:** *Statements* are often included to provide further explanation to respondents. Example:

“The following questions are about your health”

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| uses | 1 Statement.  **0..\*** InstanceStatement. |  |

##### StatisticalActivity

**Package:** Business

**Definition:** The set of executed processes and the actual resources required as inputs and produced as outputs to investigate the characteristics of a given *Population* for a particular reference period. It may describe process and resources required to acquire (*Acquisition Activity*), produce (*Production Activity*), and disseminate (*Dissemination Activity*) data in a *Statistical Program*.

**Explanatory Text:** A *Statistical Activity* includes the run-time information used to actually execute a set of processes. Activities occur in the context of each *Statistical Program Cycle* and execute a particular *Statistical Program Design*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ProductionActivity.  StatisticalActivity. |  |
|  | DisseminationActivity.  StatisticalActivity. |  |
|  | AcquisitionActivity.  StatisticalActivity. |  |
|  | **0..1** ProcessStep.topLevelProcess  **0..\*** StatisticalActivity. |  |
| comprises | **1..\*** StatisticalActivity.  **1..\*** StatisticalProgramCycle. |  |
| creates | **1..\*** StatisticalActivity.  0..\* DataResource. |  |
| specifies | **0..\*** ProcessInput.dynamkcInput  **0..\*** StatisticalActivity. |  |

##### StatisticalNeed

**Package:** Business

**Definition:** A requirement, request or other notification that will be considered by an organization. A *Statistical Need* does not have necessarily have structure or format - it is a 'raw' need as received by the organization. A *Statistical Need* may be of a variety of types including *Environmental Change* or *Information* *Request*.

**Explanatory Text:** The *Statistical Need* is a proposed or imposed change as it has been received by an organization. A *Statistical Need* is a raw expression of a proposed change, and is not necessarily well-defined. A related object - *Change Definition* - is created when a *Statistical Need* is analyzed by an organization. *Change Definition* expresses the raw need in well-defined, structured terms.

Once a *Statistical Need* has been received, the first step is to do the conceptual work to establish what it is we are trying to measure. The final output of this conceptual work is the *Change Definition*.

In some cases, the *Statistical Need* can result from the *Assessment* of the quality, efficiency, etc. of an existing process.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | EnvironmentChange.  StatisticalNeed. |  |
|  | InformationRequest.  StatisticalNeed. |  |
| evaluationOf | **0..\*** EvaluationAssessment.  **1..\*** StatisticalNeed. |  |
| initiates | 1..\* BusinessCase.  StatisticalNeed. |  |
| specifies | **1..\*** ChangeDefinition.  StatisticalNeed. |  |

##### StatisticalProgram

**Package:** Business

**Definition:** A set of activities to investigate characteristics of a given *Population*. It describes the purpose and context of a set of *Statistical Activities*.

**Explanatory Text:** The *Statistical Program* is one of a family of objects that provide the environmental context in which a set of statistical activities is conducted. *Statistical Program* is the top level object that describes the purpose and objectives of a set of activities. *Statistical Program* will usually correspond to an ongoing activity such as a survey or output series. Some examples of *Statistical Program* are:

- Labour Force Survey

- Multipurpose Household Survey

- National Accounts

- Demography

- Overseas Arrivals and Departures

Related to the *Statistical Program* object there are *Statistical Program Design* and *Statistical Program Cycle* objects that hold the detailed information about the design and conduct of the *Statistical Activity*.

In the case of the traditional approach, an organization has received a *Statistical Need* and produced a *Change Definition* and an approved *Business Case*. The *Business Case* will specify either a change to the design or methodology of an existing *Statistical Program*, which will result in a new *Statistical Program Design*; or a change to one or more existing *Statistical Programs* (for example, to add an additional objective to the *Statistical Program*);or result in a new *Statistical Program* being created.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| purpose |  | 1..1 | char |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | StatisticalProgram.  **0..1** OrganizationUnit.reponsibleUnit |  |
| has | **0..\*** StatisticalProgramCycle.  StatisticalProgram. |  |
| hierarchy | **0..\*** StatisticalProgram.child  **1** StatisticalProgram.parent |  |
| initiates | BusinessCase.  **0..\*** StatisticalProgram. |  |
| specifies | **0..\*** StatisticalProgramDesign.  **1** StatisticalProgram. |  |

##### StatisticalProgramCycle

**Package:** Business

**Definition:** A set of activities to investigate characteristics of a given *Population* for a particular reference period.

**Explanatory Text:** A *Statistical Program Cycle* documents the execution of an iteration of a *Statistical Program* according to the associated *Statistical Program Design* for a certain reference period. It identifies the activities that are undertaken as a part of the cycle and the specific resources required and processes used and description of relevant methodological information used in this cycle defined by the *Statistical Program Design.*

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| referencePeriod |  | 1..1 | Date |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | StatisticalProgramCycle.  0..\* Product. |  |
| comprises | **1..\*** StatisticalActivity.  **1..\*** StatisticalProgramCycle. |  |
| cycleFor | **0..\*** StatisticalProgramCycle.  **1** StatisticalProgramDesign. |  |
| has | **0..\*** StatisticalProgramCycle.  StatisticalProgram. |  |

##### StatisticalProgramDesign

**Package:** Business

**Definition:** The specification of the resources required and processes used and description of relevant methodological information about the set of activities investigating characteristics of a given *Population*. Includes the *Statistical Activities* that are required to acquire (*Acquisition Activity*), produce (*Production Activity*), and disseminate (*Dissemination Activity*) data in a *Statistical Program*.

**Explanatory Text:** The *Statistical Program Design* is one of a family of objects that provide the operational context in which a set of statistical activities is conducted.

A simple example is where a *Statistical Program* relates to a single survey, for example, the Labour Force Survey. The *Statistical Program* will have a series of *Statistical Program Design* objects that describe the methodology and design used throughout the life of the survey. When a methodological change is made to the survey, a new *Statistical Program Design* is created to record the details of the new design.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ProductionDesign.  StatisticalProgramDesign. |  |
|  | DisseminationDesign.  StatisticalProgramDesign. |  |
|  | AcquisitionDesign.  StatisticalProgramDesign. |  |
| cycleFor | **0..\*** StatisticalProgramCycle.  **1** StatisticalProgramDesign. |  |
| has | **0..1** DesignContext.  StatisticalProgramDesign. |  |
| initiates | 1..\* BusinessCase.  **0..\*** StatisticalProgramDesign.designChange |  |
| specifies | **0..\*** StatisticalProgramDesign.  **1** StatisticalProgram. |  |
| specifies | **0..\*** ProcessStepDesign.  **0..\*** StatisticalProgramDesign. |  |

##### SurveyInstrument

**Package:** Business

**Definition:** A specialized kind of *Instrument* used for the explicit purpose of gathering statistical data.

**Explanatory Text:** *Survey Instrument* is a tool used to gather information from a *Data Resource*. It can be applied in several ways using different formats and modes, for example, as paper forms in face-to-face interviews, as online self-administered interviews, as computer-assisted questionnaires in telephone interviews, as electronic templates downloaded from the web and returned via email. The *Survey Instrument* provides a generic description of the data collection form independent of the format and mode.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | SurveyInstrument.  Instrument. |  |
| uses | **1** InstrumentControl.topLevelControl  SurveyInstrument. |  |

### Concepts Group

#### Concept-Population Inheritance Class Diagram

A particular characteristic about a *Population* is described by a *Variable*. For example, the *Concept* person in the Population of adult in Netherlands can in a survey be a unique combination of age, sex, income, education of persons, etc.

*Populations* can be put into a hierarchy by creating a parent-child association between a *Population* and its sub-population. In addition, it is possible to distinguish between *Survey Population, Target Population, Frame Population* and *Analysis Population*, to describe the particular scenario in which the *Population* may be used, or may changeover the *Statistical Activity.*



Figure 34. Concept-Population Inheritance Class Diagram

#### Classification Class Diagram

A *Classification* is a categorization of real world objects so that they may be grouped, by like characteristics, for the purposes of measurement, for example ISIC. *Classifications* can be grouped into *Classification Family*, such as industrial activity.

A Classification such as ISIC is a set of related Classification Schemes. It relates Classification Schemes that differ as Classification Versions or Classification Variants of each other.

A *Classification Scheme* groups sets of *Classification Items* on one or more Levels. A *Level* is a set of *Categories* that are mutually exclusive and exhaustive, for example, section, division, group and class in ISIC rev 4.

A Correspondence Table between can be created by linking a Classification Item in a Classification Version with a corresponding Classification Item in another Classification Versionthrough the Category corresponding to both Classification Item . For example, in a table displaying the relationship between ISIC Rev.4 - NAICS 2007 (US), [0111](http://unstats.un.org/unsd/cr/registry/regcs.asp?Cl=27&Lg=1&Co=0111) in ISIC is related to [111110](http://unstats.un.org/unsd/cr/registry/regcs.asp?Cl=244&Lg=1&Co=111110) in NAICS.



Figure 35. Classification Class Diagram

#### Category-Code Class Diagram

A *Code* designates a*Category*, providing representation to the meaning from the *Category*. A *Code* can be a part of a *Code List*. A *Code List* is a type of *Node Set*, used for creating a group of *Codes* and their associated *Categories*. It consists of one or more *Code Items*.

A *Category* provides meaning to *Category Item*, for example “agriculture, forestry and fishing” or “female”. A *Category Item* further defines the *Category* in relation to other *Category Items*. A *Code Item* combines the meaning from a *Category* with a representation, for example “F – female” for gender.

A *Code* can be a part of a *Code List*. A *Code List* is a type of *Node Set,* used for creating a group of *Codes* and their associated *Categories*. It consists of one or more *Code Items*.



Figure 36. Category-Code Class Diagram

#### Variable Class Diagram

There is a distinction between conceptual and representation objects. Therefore, the *Variable* does not include any information on how the resulting value may be represented. This information is put into to the *Represented Variable.* This is to prevent duplication of *Variable* information where the essence of what is being measured remains the same but is represented in a different manner in order to promote reuse of the *Variable* definition.

*Conceptual Domains* are associated with a *Variable*, while *Value Domains* are associated with a *Represented Variable*. We distinguish between these two domains, because we want to be able to talk about the semantic aspect (*Conceptual Domain*) separately to the representational aspect (*Value Domain*).

We further distinguish between *Described Conceptual Domain* and *Enumerated Conceptual Domain*. An *Enumerated Conceptual Domain*, in combination with a *Category Set* contains information on the semantics of the *Categories* used by the *Variable*.

The *Represented Variable* adds information that describes how the resulting values may be represented through association with a *Value Domain.* We further distinguish between *Described Value Domains* and *Enumerated Value Domains.* The *Enumerated Value Domain*, in combination with a *Code Set* gives information on how the *Represented Variable* is represented, while the *Described Value Domain* provides a definition of how to form the values, rather than explicitly listing them.

An *Instance Variable i*s a particular *Represented Variable* associated with a collection of data (*Datum*).

A *Datum* is defined by the measure of a *Value Domain* combined with the link to a *Unit*. A *Datum* is also associated with a *Data Type* through the *Value Domain*.

*Data Types* contain information on the allowed computations one may perform on the *Datum*. We can distinguish between nominal-, ordinal-, interval-, and ratio-data. Gender Codes leads to nominal statistical data, whereas the age values lead to interval data.

A *Datum* uses a *Unit of Measure*. A *Unit of Measure* is the entity by which some quantity is measured. Examples are Tonnes, Count of\_, and Dollars.

A *Unit* is an object of interest in a *Statistical Activity.* We distinguish between two kinds of Unit: *Observation Unit* and *Analysis Unit*. A *Unit* is associated with a *Population*.



Figure 37. Variable Class Diagram

#### Node-Inheritance Class Diagram

A *Node Set* consists of one or more *Nodes*. *Node Sets* can be *Category Sets*, *Classification Schemes* (for example, ISIC rev 4) or *Code List* (Gender).

A *Node*  is a combination of a *Category* and any related attributes. It can be one of three different types, depending on what the *Node*  is intended for. It can be a *Category Item,* a *Classification Item***,** or a *Code Item*.

A *Node* can be associated to another *Node* in a parent/child relationship and a part/whole relationship.



Figure 38. Node-Inheritance Class Diagram

#### Node-Relationship Class Diagram

A *Node Set* consists of one or more *Nodes. Node Sets* can be *Category Sets, Classification Schemes* (for example, ISIC rev 4) or*Code Lists* (Gender).

A *Correspondence Table* can be created by linking a *Node* in a *Node Set* with a corresponding *Node* in another *Node Set*through the *Category* corresponding to both *Nodes*. For example, in a table displaying the relationship between ISIC Rev.4 - NAICS 2007 (US), [0111](http://unstats.un.org/unsd/cr/registry/regcs.asp?Cl=27&Lg=1&Co=0111) in ISIC is related to [111110](http://unstats.un.org/unsd/cr/registry/regcs.asp?Cl=244&Lg=1&Co=111110) in NAICS.

A *Code* gives a designation to a*Category*, providing representation to the meaning from the *Category*.



Figure 39. Node-Relationship Class Diagram

##### AnalysisPopulation

**Package:** Concepts

**Definition:** A *Population* used for the analysis, processing, or dissemination of statistical data.

**Explanatory Text:** *Population* determined by parameters of an analysis

**Synonyms:** object class, analytical population

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | AnalysisPopulation.  Population. |  |

##### AnalysisUnit

**Package:** Concepts

**Definition:** A *Unit* that is defined for the analysis, processing, or dissemination of statistical data.

**Explanatory Text:** Object corresponding to an *Analysis Population*

**Synonyms:** analytical unit, unit of analysis

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | AnalysisUnit.  Unit. |  |

##### Category

**Package:** Concepts

**Definition:** A *Concept* whose role is to extensionally define and measure a characteristic.

**Explanatory Text**: *Categories* for the *Concept* of sex include: Male, Female

Note: An extensional definition is a description of a *Concept* by enumerating all of its sub ordinate *Concepts* under one criterion or sub division.

For example - the Noble Gases (in the periodic table) is extensionally defined by the set of elements including Helium, Neon, Argon, Krypton, Xenon, Radon. (ISO 1087-1)

**Synonyms:** class

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Category.  Concept. |  |
| takesMeaningFrom | 0..\* Node.  1 Category. |  |
| takesMeaningFrom | CategoryItem.  1 Category. |  |

##### CategoryItem

**Package:** Concepts

**Definition:** An element of a *Category Set.*

**Explanatory Text**: A type of *Node*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | CategoryItem.  Node. |  |
| contains | **1..\*** CategoryItem.node  CategorySet. |  |
| takesMeaningFrom | CategoryItem.  1 Category. |  |

##### CategorySet

**Package:** Concepts

**Definition:** A list of *Categories*

**Explanatory Text**: A kind of *Node Set* for which the *Categories* have no assigned *Designations*. For example:

Male

Female

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | CategorySet.  NodeSet. |  |
| contains | **1..\*** CategoryItem.node  CategorySet. |  |
| is | EnumeratedConceptualDomain.  **1** CategorySet. |  |

##### Classification

**Package:** Concepts

**Definition:** A set of related *Classification Schemes*. The *Classification* relates *Classification Schemes* which differ as versions or variants of each other.

**Explanatory Text**: For example, NAICS (North American Industrial Classification System) is a *Classification*, but NAICS 2002 and NAICS 2007 are *Classification Schemes*, as they are different versions of NAICS.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| groups | **1..\*** ClassificationScheme.  Classification. |  |
| groups | **1..\*** Classification.  **0..\*** ClassificationFamily. |  |

##### ClassificationFamily

**Package:** Concepts

**Definition:** A set of *Classifications* that are related from a certain point of view.

**Explanatory Text**: The *Classification Family* includes *Classifications* devoted to describing the same subject matter, such as industries.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| groups | **1..\*** Classification.  **0..\*** ClassificationFamily. |  |

##### ClassificationItem

**Package:** Concepts

**Definition:** A *Category* at a certain *Level* within a *Classification Scheme*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ClassificationItem.  Node. |  |
| contains | **1..\*** ClassificationItem.node  ClassificationScheme. |  |
| groups | **1..\*** ClassificationItem.  Level. |  |
| maps | **1..\*** ClassificationItem.target  0..\* Map. |  |
| maps | **1..\*** ClassificationItem.source  0..\* Map. |  |

##### ClassificationScheme

**Package:** Concepts

**Definition:** A structured list of mutually exclusive *Categories*. Such a structured list may be linear or hierarchically structured.

**Explanatory Text**: *Classification Scheme* has two subtypes - *Classification Version* and *Classification Variant.* In a hierarchical *Classification Scheme*, *Categories* organized into *Levels* determined by the hierarchy. The *Categories* in each *Level* are mutually exclusive and exhaustive.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ClassificationScheme.  NodeSet. |  |
|  | ClassificationVariant.  ClassificationScheme. |  |
|  | ClassificationVersion.  ClassificationScheme. |  |
| contains | **1..\*** ClassificationItem.node  ClassificationScheme. |  |
| groups | **1..\*** ClassificationScheme.  Classification. |  |
| groups | **2..\*** ClassificationScheme.  CorrespondenceTable. |  |
| has | **0..\*** Level.level  ClassificationScheme. |  |

##### ClassificationVariant

**Package:** Concepts

**Definition:** A *Classification Variant* is based on a *Classification Version*. In a variant, the *Categories* of the *Classification Version* are split, aggregated or regrouped to provide additions or alternatives to the standard order and structure of the base version.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ClassificationVariant.  ClassificationScheme. |  |

##### ClassificationVersion

**Package:** Concepts

**Definition:** A *Classification Version* is a list of mutually exclusive *Categories* representing the version-specific values of the classification variable.

**Explanatory Text**: A *Classification Version* has a certain normative status and is valid for a given period of time.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ClassificationVersion.  ClassificationScheme. |  |
| groups | **2..\*** ClassificationVersion.  CorrespondenceTable. |  |

##### Code

**Package:** Concepts

**Definition:** A Designation for a Category.

**Explanatory Text**: *Codes* are unique within their Code List. Example: M (Male) F (Female)

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Code.  Designation. |  |
| contains | 0..\* Code.  CodeItem. |  |

##### CodeItem

**Package:** Concepts

**Definition:** An element of a *Code List*.

**Explanatory Text**: A type of *Node*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | CodeItem.  Node. |  |
| contains | **1..\*** CodeItem.node  CodeList. |  |
| contains | 0..\* Code.  CodeItem. |  |

##### CodeList

**Package:** Concepts

**Definition:** A list of *Categories* where each *Category* has a predefined *Code* assigned to it.

**Explanatory Text**: A kind of *Node Set* for which the *Category* contained in each *Node* has a *Code* assigned as a *Designation*.

For example:

1 - Male

2 - Female

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | CodeList.  NodeSet. |  |
| contains | **1..\*** CodeItem.node  CodeList. |  |
| is | EnumeratedValueDomain.  1 CodeList. |  |

##### CodeValue

**Package:** Concepts

**Definition:**  An alpha-numeric string used to represent a *Code*.

**Explanatory Text:** This is a kind of *Sign* used for*Codes*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | CodeValue.  Sign. |  |

##### Concept

**Package:** Concepts

**Definition:** Unit of thought differentiated by characteristics.

**Explanatory Text:** ISO 1087-1 defines Concept as a “unit of knowledge created by a unique combination of characteristics”. First, the term knowledge is poorly defined, and the word thought seems to capture the idea more cleanly. Second, different systems may try to capture the same thought but depend on different characteristics (i.e., attributes).For instance, typical demographic surveys care about age, sex, income, ethnicity, and education of persons. However, persons in a justice survey are either criminals or victims.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Population.  Concept. |  |
|  | Category.  Concept. |  |
| characteristic | RepresentedVariable.  Concept. |  |
| characteristic | Variable.  Concept. |  |
| characteristic | InstanceVariable.  Concept. |  |
| groups | 0..\* Concept.  0..\* ConceptSystem. |  |
| hasContext | InformationRequest.  **0..\*** Concept.informationAbout |  |
| organizedBy | Concept.  Level. |  |
| takesMeaning | 1 Concept.  0..\* Designation. |  |

##### ConceptSystem

**Package:** Concepts

**Definition:** Set of *Concepts* structured by the relations among them.

**Explanatory Text:** Here are 2 examples

1) Concept of Sex: Male, Female, Other

2) ISIC (the list is too long to write down)

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | NodeSet.  ConceptSystem. |  |
| groups | 0..\* Concept.  0..\* ConceptSystem. |  |
| groups | 1..\* SubjectField.  0..\* ConceptSystem. |  |

##### ConceptualDomain

**Package:** Concepts

**Definition:** Set of *Categories*, irrespective of any relations among them

**Explanatory Text:** Here are 3 examples –

1) Sex categories (enumerated CD)

* male
* female
* other

2) Non-negative whole number (described CD)

3) Endowment categories (enumerated CD)

* $0-$99,999
* $100,000-$999,999
* $1,000,000 and above

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | EnumeratedConceptualDomain.  ConceptualDomain. |  |
|  | ConceptualDomain.  Variable. |  |
|  | **1** ConceptualDomain.  0..\* ValueDomain. |  |
|  | DescribedConceptualDomain.  ConceptualDomain. |  |

##### CorrespondenceTable

**Package:** Concepts

**Definition:** A tool for the linking of *Classifications*. A *Correspondence Table* systematically explains where, and to what extent, the *Categories* in may be found in different *Classification Schemes* of the same *Classification* or in *Classification Schemes* of different *Classifications*.

**Explanatory Text:** Given 2 Category Sets

1) Marital Status A: Married, Single 2) Marital Status B: Married, Single, Widowed, Divorced

A *Correspondence Table* harmonizing the 2 Cate*gory Sets* will contain *Maps* that link *Categories* from each set:

Married (A) -> Married (B)

Single (A) <- Single (B), Widowed (B), Divorced (B)

where the arrow points to the *Category* which is more generic.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| contains | 1..\* Map.  CorrespondenceTable. |  |
| groups | 2..\* NodeSet.  CorrespondenceTable. |  |
| groups | **2..\*** ClassificationScheme.  CorrespondenceTable. |  |
| groups | **2..\*** ClassificationVersion.  CorrespondenceTable. |  |

##### DataType

**Package:** Concepts

**Definition:** The computational model for some data, characterized by axioms and operations, and containing a set of distinct values.

**Explanatory Text:** Here are 3 examples (with type families taken from ISO/IEC 11404):

1) State (nominal data): unordered, no arithmetic

2) Integer (interval data): Ordered, subtraction, bounded below

3) Enumerated (ordinal data): ordered, no arithmetic

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| describes | 1 DataType.  0..\* ValueDomain. |  |

##### Datum

**Package:** Concepts

**Definition:** Association of a *Unit* with an element of a *Value Domain*.

**Explanatory Text:** A *Datum* is the actual instance of data that was collected. It is the value with populates a cell in a table.

Here are 2 examples - 1. <M, male> (for *unit* Dan Gillman with respect to sex of US persons)

2. <3, $1,000,000 and above> (for *unit* John Hopkins with respect to endowments for US universities)

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| describes | 1..\* Datum.  **1** InstanceVariable. |  |
| has | **1..\*** Datum.identifier  DataPoint. |  |
| has | **1** Datum.observation  DataPoint. |  |
| has | **0..\*** Datum.attribute  0..\* DataPoint. |  |
| measures | 0..\* Datum.  1 Unit. |  |

##### DescribedConceptualDomain

**Package:** Concepts

**Definition:** A *Conceptual Domain*, with each *Concept* defined by a *Rule*.

**Explanatory Text**: For example: All real numbers between 0 and 1 (where 'number' is a *Concept*, and 0 and 1 are possible *Designations*.)

**Synonyms:** non-enumerated conceptual domain

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DescribedConceptualDomain.  ConceptualDomain. |  |
|  | **0..\*** DescribedValueDomain.  **1** DescribedConceptualDomain. |  |

##### DescribedValueDomain

**Package:** Concepts

**Definition:** A *Value Domain*, with each *Designation* defined by a *Rule*.

**Explanatory Text**: For example: All real decimal numbers between 0 and 1 (Where 'decimal number' is a *Designation*, such as the numeric string 0.5 for the number one half)

**Synonyms:** non-enumerated value domain

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DescribedValueDomain.  ValueDomain. |  |
|  | **0..\*** DescribedValueDomain.  **1** DescribedConceptualDomain. |  |

##### Designation

**Package:** Concepts

**Definition:** The name given to an object so it can be identified.

**Explanatory Text:** The association of a *Concept* with a *Sign* which denotes it.

**Synonyms:** term, code, appellation

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Code.  Designation. |  |
| contains | 0..\* Designation.  1 Node. |  |
| encodes | **1** Designation.  1 Sign. |  |
| takesMeaning | 1 Concept.  0..\* Designation. |  |

##### EnumeratedConceptualDomain

**Package:** Concepts

**Definition:** A *Conceptual Domain* expressed as a list of *Categories*.

**Explanatory Text**: Example: The Sex categories of 'Male' and 'Female'.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | EnumeratedConceptualDomain.  ConceptualDomain. |  |
|  | **0..\*** EnumeratedValueDomain.  **1** EnumeratedConceptualDomain. |  |
| is | EnumeratedConceptualDomain.  **1** CategorySet. |  |

##### EnumeratedValueDomain

**Package:** Concepts

**Definition:** A *Value Domain* expressed as a list of *Designations*.

**Explanatory Text**: Example – Sex Codes:

<m, male>

<f, female>

<o, other>

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **0..\*** EnumeratedValueDomain.  **1** EnumeratedConceptualDomain. |  |
|  | EnumeratedValueDomain.  ValueDomain. |  |
| is | EnumeratedValueDomain.  1 CodeList. |  |

##### FramePopulation

**Package:** Concepts

**Definition:** A *Population* represented by records in a frame, which is the observable part of a *Target Population* and provides a reasonable approximation to it.

**Explanatory Text:** Example: most recent population census frame)

**Synonyms:** object class

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | FramePopulation.  Population. |  |

##### InstanceVariable

**Package:** Concepts

**Definition:** The use of a *Represented Variable* within a *Data Set*. It may include information about the source of the data.

**Explanatory Text:** The *Instance Variable* is used to describe actual instances of data that have been collected.

Here are 3 examples:

1) Gender: Dan Gillman has gender <m, male>, Arofan Gregory has gender<m, male>, etc.

2) Number of employees: Microsoft has 90,000 employees; IBM has 433,000 employees, etc.

3) Endowment: Johns Hopkins has endowment of <3, $1,000,000 and above>,

Yale has endowment of <3, $1,000,000 and above>, etc.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | UnitDataPoint.  **1..\*** InstanceVariable.identifiers |  |
|  | UnitDataPoint.valueFor  InstanceVariable.measurement |  |
|  | UnitDataPoint.  **0..\*** InstanceVariable.attributes |  |
| characteristic | InstanceVariable.  Concept. |  |
| describes | 1..\* Datum.  **1** InstanceVariable. |  |
| measures | **1** InstanceVariable.  1 Population. |  |
| uses | **0..\*** InstanceVariable.instance  **1** RepresentedVariable. |  |
| uses | **1** InstanceVariable.  **0..\*** InstanceQuestion. |  |

##### Level

**Package:** Concepts

**Definition:** Set of ***Concepts*** which are mutually exclusive and exhaustive.

**Explanatory Text**: For example, section, division, group and class in ISIC Rev. 4. A *Level* often is associated with a *Concept*, which defines it.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| groups | 1..\* Node.  0..1 Level. |  |
| groups | **1..\*** ClassificationItem.  Level. |  |
| has | **0..\*** Level.level  1 NodeSet. |  |
| has | **0..\*** Level.level  ClassificationScheme. |  |
| organizedBy | Concept.  Level. |  |

##### Map

**Package:** Concepts

**Definition:** An expression of the relation between

a Category in a source Classification Scheme and a corresponding Category in the target Classification Scheme.

**Explanatory Text:** Given 2 *Category Sets*

1) Marital Status A

* Married
* Single

2) Marital Status B

* Married
* Single
* Widowed
* Divorced

The 2 Married *Categories* may be compared as follows

Married (A) -> Married (B)

where the arrow points to the *Category* which is more generic.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **1..\*** Node.source  Map. |  |
|  | **1..\*** Node.target  Map. |  |
| contains | 1..\* Map.  CorrespondenceTable. |  |
| maps | **1..\*** ClassificationItem.target  0..\* Map. |  |
| maps | **1..\*** ClassificationItem.source  0..\* Map. |  |

##### Node

**Package:** Concepts

**Definition:** A combination of a *Category* and related attributes.

**Explanatory Text**: A *Node* is created as a *Category*, *Code* or *Classification Item* for the purpose of defining the situation in which the *Category* is being used.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **0..\*** Node.child  **0..1** Node.parent |  |
|  | CategoryItem.  Node. |  |
|  | **1..\*** Node.source  Map. |  |
|  | **1..\*** Node.target  Map. |  |
|  | CodeItem.  Node. |  |
|  | ClassificationItem.  Node. |  |
|  | **0..\*** Node.part  **0..1** Node.whole |  |
| contains | 0..\* Designation.  1 Node. |  |
| contains | **1..\*** Node.node  1 NodeSet. |  |
| groups | 1..\* Node.  0..1 Level. |  |
| takesMeaningFrom | 0..\* Node.  1 Category. |  |

##### NodeSet

**Package:** Concepts

**Definition:** A set of *Nodes*

**Explanatory Text:** *Node Set* is a kind of *Concept System*. Here are 2 examples:

1) Sex Categories

* Male
* Female
* Other

2) Sex *Codes*

* <m, male>
* <f, female>
* <o, other>

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ClassificationScheme.  NodeSet. |  |
|  | CodeList.  NodeSet. |  |
|  | CategorySet.  NodeSet. |  |
|  | NodeSet.  ConceptSystem. |  |
| contains | **1..\*** Node.node  1 NodeSet. |  |
| groups | 2..\* NodeSet.  CorrespondenceTable. |  |
| has | **0..\*** Level.level  1 NodeSet. |  |

##### Observation Unit

**Package:** Concepts

**Definition:** A *Unit* for which information can actually be obtained during data collection.

**Explanatory Text:** The sub-set of the *Population* of interest for which information can actually be obtained. For example, if the *Population* is the persons living in Ontario, the *Observation Units* might be persons currently residing in Ontario neither in an institution nor in a remote northern location nor temporarily out of the province.

**Synonyms:** collection unit, Unit of observation, unit of collection

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Observation Unit.  Unit. |  |

##### Population

**Package:** Concepts

**Definition:** The total membership of a defined class of people, objects or events.

**Explanatory Text:** *Population* has a number of subtypes. Here are 3 examples –

1. US adult persons

2. US computer companies

3. Universities in the US

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | TargetPopulation.  Population. |  |
|  | Population.  Concept. |  |
|  | **0..\*** Population.child  **1** Population.parent |  |
|  | SurveyPopulation.  Population. |  |
|  | FramePopulation.  Population. |  |
|  | AnalysisPopulation.  Population. |  |
| hasContext | InformationRequest.  **0..\*** Population.informationOn |  |
| isAggregateOf | 1..\* Population.  1..\* Unit. |  |
| measures | 0..\* Population.  0..\* Variable. |  |
| measures | **0..\*** RepresentedVariable.  0..\* Population. |  |
| measures | **1** InstanceVariable.  1 Population. |  |

##### RepresentedVariable

**Package:** Concepts

**Definition:** The association of a *Variable* with a *Value Domain* which represents it. The *Represented Variable* is used as part of a *Statistical Activity*.

**Explanatory Text:** Here are 3 examples –

1. Sex variable which will be collected using

<m, male>,

<f, female>,

<o, other>

2. Number of Employees variable which will be collected using an Integer or Count of Individuals.

3. Endowment of Universities variable which will be collected using

<1, $0-$99,999>,

<2, $100,000-$999,999>,

<3, $1,000,000 and above>

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | 1 Variable.  **0..\*** RepresentedVariable.instance |  |
| characteristic | RepresentedVariable.  Concept. |  |
| definedBy | **0..\*** DataStructureComponent.  **1** RepresentedVariable. |  |
| measures | **0..\*** RepresentedVariable.  0..\* Population. |  |
| represents | ValueDomain.  **1** RepresentedVariable. |  |
| uses | **0..\*** InstanceVariable.instance  **1** RepresentedVariable. |  |

##### Sign

**Package:** Concepts

**Definition:** Something that suggests the presence or existence of a fact, condition, or quality.

**Explanatory Text:** It is a perceivable object. This object is used to denote a *Concept* as a *Designation*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | CodeValue.  Sign. |  |
| encodes | **1** Designation.  1 Sign. |  |

##### SubjectField

**Package:** Concepts

**Definition:** One or more *Concept Systems* used for the grouping of *Concepts* and *Categories* for the production of statistics.

**Explanatory Text:** A *Subject Field* is a field of special knowledge under which a set of *Concepts* and their *Designations* is used. For example, labour market, environmental expenditure, tourism, etc.

**Synonyms:** subject area, theme

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| groupedBy | 0..\* SubjectField.  0..\* DataFlow. |  |
| groups | 1..\* SubjectField.  0..\* ConceptSystem. |  |
| hasContext | InformationRequest.  **0..\*** SubjectField.informationAbout |  |

##### SurveyPopulation

**Package:** Concepts

**Definition:** A *Population* for which information can be obtained in a survey.

**Explanatory Text:** A *Population* which can realistically be studied (example: people currently residing in the province of Ontario not in an institution nor in a remote northern location nor temporarily out of the province). The *Survey Population* is therefore often a subset of the *Target Population.*

**Synonyms:** object class

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | SurveyPopulation.  Population. |  |

##### TargetPopulation

**Package:** Concepts

**Definition:** A *Population* for which a *Statistical Activity* is designed to make estimates.

**Explanatory Text**: *Population* for which estimates are desired in a *Statistical Activity*, though practical considerations may dictate that some *units* are excluded. If so, the resulting sub-set of *units* for which information can be obtained is the *Survey Population*.

**Synonyms:** object class

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | TargetPopulation.  Population. |  |

##### Unit

**Package:** Concepts

**Definition:** The object of interest in *Statistical Activities* and corresponds to at least one *Population*.

**Explanatory Text:** Here are 3 examples

1. Individual US person (i.e., Arofan Gregory, Dan Gillman, Barack Obama, etc)
2. Individual US computer companies (i.e., Microsoft, Apple, IBM, etc)
3. Individual US universities (i.e., Johns Hopkins, University of Maryland, Yale, etc)

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | Observation Unit.  Unit. |  |
|  | AnalysisUnit.  Unit. |  |
| isAggregateOf | 1..\* Population.  1..\* Unit. |  |
| measures | 0..\* Datum.  1 Unit. |  |
| observationFor | DataPoint.  1 Unit. |  |

##### UnitOfMeasure

**Package:** Concepts

**Definition:** Units by which some quantity is measured.

**Explanatory Text:** Here are 3 examples

1. Kilograms
2. Count
3. Dollars

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| refines | **0..1** UnitOfMeasure.  0..\* ValueDomain. |  |

##### ValueDomain

**Package:** Concepts

**Definition:** A set of allowed values (determinants). A *Value Domain* is a *Concept System* where all *Concepts* are designated, but in which there are no relations.

**Explanatory Text:** Here are 3 examples –

1) Sex codes (enumerated VD)

* m, male
* f, female
* o, other

2) Non-negative whole decimal number (described VD) , count of people

3) Endowment categories (enumerated VD) , dollars

* 1, $0-$99,999
* 2, $100,000-$999,999
* 3, $1,000,000 and above

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DescribedValueDomain.  ValueDomain. |  |
|  | **1** ConceptualDomain.  0..\* ValueDomain. |  |
|  | EnumeratedValueDomain.  ValueDomain. |  |
| describes | 1 DataType.  0..\* ValueDomain. |  |
| has | **1** ValueDomain.responseDomain  0..\* Question. | Specification of the valid response for the question in terms of either a set of pre-defined values or as a type of data such as a date, integer etc. |
| refines | **0..1** UnitOfMeasure.  0..\* ValueDomain. |  |
| represents | ValueDomain.  **1** RepresentedVariable. |  |

##### Variable

**Package:** Concepts

**Definition:** The use of a *Concept* as a characteristic of a *Population* that is intended to be measured as part of a *Statistical Activity*.

**Explanatory Text:** Here are 3 examples

1. Sex
2. Number of employees
3. Endowment

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | 1 Variable.  **0..\*** RepresentedVariable.instance |  |
|  | ConceptualDomain.  Variable. |  |
| characteristic | Variable.  Concept. |  |
| hasSemanticOf | 0..\* Question.  0..\* Variable. |  |
| measures | 0..\* Population.  0..\* Variable. |  |

### Production Group

#### Production -Overall Class Diagram

*Process Steps* can contain “sub-steps”, those “sub-steps” can contain “sub-steps” within them and so on indefinitely.

Each *Process Step* in a statistical business process has been included to serve some purpose. The purpose is identified by associating the *Process Step* with a *Business Function*.

The *Process Step Design* then identifies the *Process Method* that will be used to perform the *Business Function* associated with the *Process Step*.

A *Process Method* specifies the method to be used, and is associated with a set of *Rules* to be applied. At the time the *Process Step*is executed, however, someone or something needs to apply the designated method and rules. The *Process Step Design* designates the *Business Service* that will implement the *Process Method* at the time the*Process Step*is executed.. This implies the *Business Service* is capable of implementing the specified *Process Method. Business Services* are typically reusable to address functional needs associated with more than one statistical business process. A *Process Method* can be associated with a list of *Business Services* capable of implementing that method.

A *Process* is anominatedset of*Process Steps***,** including their associated process flow information (*Process Controls***),** which has been highlighted for possible reuse.

A *Statistical Program Design*is associated with a top level *Process Step* whose *Process Step Design*contains all the sub-steps and process flows required to put that statistical program into effect. It is therefore possible to discover the individual *Process Steps*, *Business Functions* and *Process Methods* associated with that *Statistical Program Design*.

A *Statistical Activity* initiates execution of top level *Process Step*which will result in all sub-steps being executed which are relevant to that instance of the *Statistical Activity*. (Some process flows are conditional, so not every sub-step will necessarily be subject to execution during a particular instance of a *Statistical Activity*.

The execution of the top level *Process Step*associated with a *Statistical Activity* will be recorded in a *Process Step Execution Record* which allows the actual flow of execution for that instance of the *Statistical Activity*to be traced**.**



Figure 40. Production -Overall Class Diagram

#### Process Overview Class Diagram



Figure 41. Process Overview Class Diagram

#### 

#### Process Design Class Diagram

A *Process Step Design* provides a *Process Input Specification* which identifies the types of the *Process Inputs* required at the time of *Process Step*execution. An example might be a *Process Input Specification* which requires a *Dimensional Data Set* be provided at the time of *Process Step*execution.

A *Process Step Design* may also identify *Process Inputs*. These refer to specific instances of inputs, rather than specifying a type of input. For example, a *Process Step Design* may specify that a particular *Code List* will always be used to provide a list of valid values.

Process Input Specifications and Process Inputs are often determined by the input requirements of the Business Service, Process Method and Rules associated with the Process Step Design.

*Process Output Specifications*play an analogous role to *Process Input Specifications* but describe the types of *Process Outputs* to be produced at the time of *Process Step*execution.

*Process Control* specifies what process flow should occur from one *Process Step Design* to the next at the time of execution. In some cases it may simply record the next *Process Step Design* to be executed on a fixed/constant basis. Alternatively, a *Process Control*may set out conditions to be evaluated at the time of execution to determine which *Process Step(s)* to execute next.

The specification and evaluation of conditional *Process Controls* refer to *Rules*. In the case of *Process Controls*, the *Rules* guide the process flow. (In the case of *Process Step Designs, Rules* guide the work done by the process step to produce *Process Outputs*).



Figure 42. Process Design Class Diagram

#### Process Execution Class Diagram

A *Process Input*may beprovided to a *Process Step* in order for the *Process Step* to “add value” to that input by producing an output which represents a “transformed” version of the input. Such a process input is classed as a *Transformable Input*.

*Parameter Inputs*are a form of *Process Input* used to specify which configuration should be used for a specific execution of a process step which has been designed to be configurable. *Parameter Inputs* may be provided where *Rules*and/or *Business Service* interfaces associated with the particular *Process Step Design* have been designed to be configurable based on inputs passed to them.

A *Process Support Input*influences the work performed by the *Process Step*, and therefore influences its outcome, but does not correspond to a *Parameter Input* or a *Transformable Input***.** Examples could include:

* A *Code List* which will be used to check whether the codes recorded in one dimension of a data set are valid
* An auxiliary data set which will influence imputation for, or editing of, a primary data set which has been submitted to the process step as the *Transformable Input*.

The same instance of an information object may perform different roles in regard to different *Process Steps*.

A *Process Output* isany instance of an informationobject which is produced by a *Process Step*as a result of its execution. *Process outputs* are subtyped as part of the *Process Output Specification*.

A *Transformed Output* is the result which provides the “reason for existence” of the *Process Step*. If that output were no longer required then there would be no need for the *Process Step* in its current form. Typically a *Transformed Output* is either a *Process Input* to a subsequent *Process Step* or it represents the final product from a statistical business process.

A *Process Metric* records information about the execution of a *Process Step,* eg how long it took to complete execution of the *Process Step*, or what percentage of records in the *Transformable Input* were updated by the *Process Step* to produce the *Transformed Output*

*Process Outputs* associated with the current *Process Step*execution may be evaluated as part of *Process Control* in determining which *Process Step* to execute next.



Figure 43. Process Execution Class Diagram

##### BusinessFunction

**Package:** Production

**Definition:** Something an enterprise does, or needs to do, in order to achieve its objectives.

**Explanatory Text:** A *Business* *Function* delivers added value from a business point of view. It is delivered by bringing together people, processes and technology (resources), for a specific business purpose.

*Business Functions* answer in a generic sense "What business purpose does this *Process Step Design* serve?" Through identifying the *Business Function* associated with each *Process Step Design* it becomes easier in for someone in future with an equivalent business need to identify *Process Step Designs* that they might reuse (in whole or in part).

A *Business Function* may be defined directly with descriptive text and/or through reference to an existing catalogue of *Business Functions*. The phases and sub processes defined within GSBPM can be used as an internationally agreed basis for cataloguing high level *Business Functions*. A catalogue might also include *Business Functions* defined at a lower level than "sub process". For example, "Identify and address outliers" might be catalogued as a lower level *Business Function* with the "Review, validate and edit" function (5.3) defined within GSBPM.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| canBePerformedUsing | **0..\*** BusinessFunction.  0..\* ProcessMethod. |  |
| performs | 0..\* ProcessStep.  **1** BusinessFunction. |  |

##### BusinessService

**Package:** Production

**Definition:** A defined interface for accessing business capabilities (an ability that an organization possesses, typically expressed in general and high level terms and requiring a combination of organization, people, processes and technology to achieve).

**Explanatory Text:** A *Business* *Service* may provide one means of accessing a particular *Business Function*. Requesting a particular service through the defined interface may result in a business process (workflow) being executed.

The explicitly defined interface of a *Business Service* can be seen as representing a "service contract". If particular inputs are provided then the service will deliver particular outputs in compliance within specific parameters (for example, within a particular period of time).

In the case of GSIM, a *Business Service* typically implements a particular *Process Method* to perform a particular *Business Function*.

Note: The interface of a *Business Service* is not necessarily IT based. For example, a typical postal service will have a number of service interfaces:

- Public letter box for posting letters

- Counter at post office for interacting with postal workers

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| Location |  | 1..1 | int |
| Person Role |  | 1..1 | int |
| Service Interface |  | 1..1 | int |
| Service Type |  | 1..1 | int |
| URN |  | 1..1 | int |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| canBePerformedUsing | ProcessMethod.  **1..\*** BusinessService. |  |
| executes | **0..\*** ProcessStepExecutionRecord.  **0..\*** BusinessService. |  |
| uses | **0..\*** ProcessStepDesign.implementationOf  **1..\*** BusinessService. |  |

##### ParameterInput

**Package:** Production

**Definition:** Inputs used to specify which configuration should be used for a specific *Process Step* which has been designed to be configurable.

**Explanatory Text:** Parameter Inputs may be provided where *Rules* and/or *Business Service* interfaces associated with a particular *Process Step* have been designed to be configurable based on inputs passed in to the *Process Step.*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ParameterInput.  ProcessInput. |  |

##### Process

**Package:** Production

**Definition:** A nominated set of *Process Step Designs*, and associated *Process Controls* (flow), which have been highlighted for possible reuse.

**Explanatory Text:** In a particular statistical business process, some *Process Steps* may be unique to that business process while others may be applicable to other business processes. A *Process* can be seen as a reusable template. It is a means to accelerate design processes and to achieve sharing and reuse of design patterns which have approved effective. Reuse of process patterns can also lead to reuse of relevant *Business Services* and business *Rules*.

By deciding to reuse a *Process*, a designer is actually reusing the "pattern" of *Process Step Designs* and *Process Controls* associated with that *Process*. They will receive a new instance of the Proce*ss Step Designs* and *Process Controls*. If they then tailor their "instance" of the *Process Step Designs* and *Process Controls* to better meet their needs they will not change the definition of the reusable *Process*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| consistsOf | 0..\* ProcessStep.  Process. |  |
| usedBy | 0..\* Process.  **0..\*** ProcessStepDesign. |  |

##### ProcessControl

**Package:** Production

**Definition:** A decision point which determines the flow between *Process Steps*.

**Explanatory Text:** The typical use of *Process Control* is to determine what happens next after a *Process Step Design* is executed. The possible paths, and the decision criteria, associated with a *Process Control* are specified as part of designing a production process. There is typically a very close relationship between the design of *Process Steps* and the design of *Process Controls*.

It is possible to define a *Process Control* where the next *Process Step* that will be executed is a fixed value rather than a "choice" between two or more possibilities. Where such a design would be appropriate, this feature allows, for example, initiation of a *Process Step* representing the GSBPM Process Phase (5) to always lead to initiation of GSBPM sub-process Integrate Data (5.1) as the next step.

This allows a process designer to divide a business process into logical steps (for example, where each step performs a specific *Business Function*) even if these *Process Steps* will always follow each other in the same order. In all cases, the *Process Control* defines and manages the flow between *Process Steps*, even where the flow is "trivial". *Process Step Design* is left to focus entirely on the design of the *Process Step* itself, not sequencing between steps.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| Decision Point |  | 1..1 | int |
| Sequence Number |  | 1..1 | int |
| Start Event |  | 1..1 | int |
| status |  | 1..1 | int |
| End Event |  | 1..1 | int |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | 0..\* Rule.  **0..\*** ProcessControl. |  |
| reviews | **0..\*** TransformedOutput.  **0..\*** ProcessControl. |  |
| specifies | **0..\*** ProcessStepDesign.nextStep  **0..\*** ProcessControl. |  |
| specifies | **1** ProcessControl.control  **1** ProcessStepDesign.completedStep |  |
| triggers | **1** ProcessControl.postControl  **0..\*** ProcessStep.preStep |  |
| triggers | **0..\*** ProcessControl.preControl  **0..\*** ProcessStep.postControl |  |
| usesAsParameters | **0..\*** ProcessMetric.  **0..\*** ProcessControl. |  |

##### ProcessInput

**Package:** Production

**Definition:** Any instance of an information object which is supplied to a process step at the time its execution is initiated.

**Explanatory Text:** *Process Input* has three subtypes: *Process Support Input*, *Parameter Input* and *Transformable Input*, to be able to identify the range of roles that the *Process Inputs* perform in the course of a *Process Step*. A *Process Input* may be provided to a *Process Step* to:

- "add value" to that input by producing an output which represents a "transformed" version of the input.

- control (for example, as a parameter) or influence the behavior of the *Process Step*.

- be used by the *Process Step* as either an input or a guide.

Note: The same instance of an information object may perform different roles in regard to different *Process Steps*.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| Data Provider |  | 1..1 | int |
| Location |  | 1..1 | int |
| Security Classification |  | 1..1 | int |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | TransformableInput.  ProcessInput. |  |
|  | ProcessSupportInput.  ProcessInput. |  |
|  | ParameterInput.  ProcessInput. |  |
| has | **1..\*** ProcessInput.input  **0..\*** ProcessStepExecutionRecord. |  |
| references | ProcessInput.  **0..1** IdentifiableArtefact.object |  |
| specifies | **0..\*** ProcessInput.staticInput  **0..\*** ProcessStepDesign. |  |
| specifies | **0..\*** ProcessInput.dynamkcInput  **0..\*** StatisticalActivity. |  |

##### ProcessInputSpecification

**Package:** Production

**Definition:** A record of the types of inputs required for a Process Step Design.

**Explanatory text:** The *Process Input Specification* enumerates the *Process Inputs* required at the time a *Process Step Design* is executed. For example, if five different *Process Inputs* are required at the time, the *Process Input Specification* will describe each of the five inputs. For each required *Process Input* the *Process Input Specification* will record:

1. the type of Process Input (Parameter Input, Process Support Input or Transformable Input); and

2. the type of information object (based on GSIM) which will be used as the *Process Input* (Example types might be a *Dimensional Data Set* or a *Classification*).

The *Process Input* to be provided at the time of *Process Step* execution will then be a specific instance of the type of information object specified by the *Process Input Specification*. For example, if a *Process Input Specification* requires a *Dimensional Data Set* then the corresponding *Process Input* provided at the time of *Process Step* execution will be a particular *Dimensional Data Set****.***

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| role | This specifies the role of the input. The value must be a class type which is a subclass of a *ProcessInput* (e.g. *ParameterInput* or *TransfromableInput*) | 1..1 |  |
| type | This denotes the type of object which can be used as an input. | 1..1 |  |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| specifies | **0..\*** ProcessInputSpecification.inputSpecification  **0..\*** ProcessStepDesign. |  |

##### ProcessMethod

**Package:** Production

**Definition:** A specification of the technique which will be used to perform the unit of work.

**Explanatory Text:** The technique specified by a *Process Method* is independent from any choice of technologies and/or other tools which will be used to apply that technique in a particular instance. The definition of the technique may, however, intrinsically require the application of specific *Rules* (for example, mathematical or logical formulas).

A *Process Method* describes a particular method for performing a *Business Function.* Similarly to the way in which *Business Function* documents the high level purpose of a process step ("what business purpose does this process step serve?"), *Process Method* documents the high level methodological "how" associated with the *Process Step*. Where a *Process Step Design* applies a method which is not specifically statistical in nature, however, this can still be recorded as the *Process Method*.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| Method Reference |  | 0..\* |  |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| applies | **0..\*** ProcessStepDesign.  1..\* ProcessMethod. |  |
| canBePerformedUsing | **0..\*** BusinessFunction.  0..\* ProcessMethod. |  |
| canBePerformedUsing | ProcessMethod.  **1..\*** BusinessService. |  |
| has | 0..\* Rule.  0..\* ProcessMethod. |  |

##### ProcessMetric

**Package:** Production

**Definition:** A *Process Output* whose purpose is to measure and report some aspect of how the *Process Step* performed during execution.

**Explanatory Text:** A *Process Metric* is a sub-type of *Process Output* which records information about the execution of a *Process Step*. For example, how long it took to complete execution of the *Process Step* and what percentage of records in the *Transformable Input* was updated by the *Process Step* to produce the *Transformed Output.*

One purpose for a *Process Metric* may be to provide a quality measure related to the *Transformed Output*. For example, a *Process Step* with the *Business Function* of imputing missing values is likely to result, as its *Transformed Output*, in a *Data Set* where values that were missing previously have been imputed. Statistical quality measures, captured as *Process Metrics* for that *Process Step* may include a measure of how many records were imputed, and a measure of how much difference, statistically, the imputed values make to the dataset overall.

Another purpose for a *Process Metric* may be to measure an aspect of the *Process Step* which is not directly related to the *Transformed Output* it produced. For example, a *Process Metric* may record the time taken to complete the *Process Step* or other forms of resource utilization (for example, human and/or IT).

Often these two kinds of *Process Metrics* will be used in combination when seeking to, for example, monitor and tune a statistical business process so its statistical outputs achieve the highest level of quality possible based on the time, staff and/or IT resources that are available.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ProcessMetric.  ProcessOutput. |  |
| usesAsParameters | **0..\*** ProcessMetric.  **0..\*** ProcessControl. |  |

##### ProcessOutput

**Package:** Production

**Definition:** Any instance of an information object which is produced by a *Process Step* as a result of its execution.

**Explanatory Text:** *Process Outputs* are subtyped.

- *Transformed Output* is the result which provides the "reason for existence" of the *Process Step*. If that output were no longer required then there would be no need for the *Process Step* in its current form. Typically a *Transformed Output* is either a *Process Input* to a subsequent *Process Step* or it represents the final product from a statistical business process.

- A *Process Metric* records information about the execution of a *Process Step*. For example, how long it took to complete execution of the *Process Step* and what percentage of records in the *Transformable Input* was updated by the *Process Step* to produce the *Transformed Output*.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| Location |  | 1..1 | int |
| Security Classification |  | 1..1 | int |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | TransformedOutput.  ProcessOutput. |  |
|  | ProcessMetric.  ProcessOutput. |  |
| creates | **1..\*** ProcessOutput.  **1** ProcessStepExecutionRecord. |  |
| references | **0..1** IdentifiableArtefact.object  ProcessOutput. |  |

##### ProcessOutputSpecification

**Package:** Production

**Definition:** Identifies the types of *Process Outputs* the associated *Process Step Design* will produce when it is executed.

**Explanatory text:** The *Process Output Specification* enumerates the *Process Outputs* that will be generated at the time the associated *Process Step Design* is executed. For example, if five different *Process Outputs* will be generated at the time of *Process Step* execution the *Process Output Specification* will describe each of the five outputs. For each *Process Output* the *Process Output Specification* will record:

1. the type of Process Output (Process Metric or Transformed Output)

2. the type of GSIM information object which will be generated as the *Process Output*.

An example type might be a *Dimensional Data Set*. The *Process Output* generated at the time of *Process Step* execution will then be a specific instance of the type of information object specified by the *Process Output Specification*. For example, if a *Process Output Specification* refers to generation of a *Dimensional Data Set* then the corresponding *Process Output* generated at the time of *Process Step* execution will be a particular *Dimensional Data Set*. For each *Process Step* execution a different *Dimensional Data Set* will be generated.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| role | This specifies the role of the input. The value must be a class type which is a subclass of a *ProcessOutput* (e.g. *ProcessMetric* or *TransfromableOutput*) | 1..1 |  |
| type | This denotes the type of object which can be used as an input. | 1..1 |  |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | **1..\*** ProcessOutputSpecification.outputSpecification  **0..\*** ProcessStepDesign. |  |

##### ProcessStep

**Package:** Production

**Definition:** One in a series of tasks which comprise a statistical business process

**Explanatory Text:** A Process Step implements the Process Step Design specified in order to produce the outputs for which the process step was designed.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **0..1** ProcessStep.topLevelProcess  **0..\*** StatisticalActivity. |  |
| consistsOf | 0..\* ProcessStep.  Process. |  |
| has | **0..\*** ProcessStepExecutionRecord.  ProcessStep. |  |
| has | 0..\* ProcessStep.  **1** ProcessStepDesign. |  |
| performs | 0..\* ProcessStep.  **1** BusinessFunction. |  |
| triggers | **1** ProcessControl.postControl  **0..\*** ProcessStep.preStep |  |
| triggers | **0..\*** ProcessControl.preControl  **0..\*** ProcessStep.postControl |  |

##### ProcessStepDesign

**Package:** Production

**Definition:** Defines how a Process Step will be performed. This includes specifying the *Process Inputs* to that work and the *Process Outputs* that will be produced.

**Explanatory Text:** A *Process Step* can be as big or small as the designer of a particular business process chooses. From a design perspective, one *Process Step* can contain "sub-steps", each of which is conceptualized as a (smaller) *Process Step* in its own right. Each of those "sub-steps" may contain "sub-steps" within them and so on indefinitely. It is a decision for the process designer to what extent to subdivide steps.

At some level it will be appropriate to consider a *Process Step* to be a discrete task (unit of work) without warranting further subdivision. At that level the *Process Step* is designed to process particular *Process Inputs,* using a particular *Business Service*, to produce particular *Process Output*s. The flow between a *Process Step* and any sub steps is managed via *Process Control*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| applies | **0..\*** ProcessStepDesign.  1..\* ProcessMethod. |  |
| has | 0..\* ProcessStep.  **1** ProcessStepDesign. |  |
| has | **1..\*** ProcessOutputSpecification.outputSpecification  **0..\*** ProcessStepDesign. |  |
| specifies | **0..\*** ProcessInputSpecification.inputSpecification  **0..\*** ProcessStepDesign. |  |
| specifies | **0..\*** ProcessInput.staticInput  **0..\*** ProcessStepDesign. |  |
| specifies | **0..\*** ProcessStepDesign.nextStep  **0..\*** ProcessControl. |  |
| specifies | **1** ProcessControl.control  **1** ProcessStepDesign.completedStep |  |
| specifies | **0..\*** ProcessStepDesign.  **0..\*** StatisticalProgramDesign. |  |
| usedBy | 0..\* Process.  **0..\*** ProcessStepDesign. |  |
| uses | **0..\*** ProcessStepDesign.implementationOf  **1..\*** BusinessService. |  |

##### ProcessStepExecutionRecord

**Package:** Production

**Definition:** A record of the execution of a *Process Step.* The record includes the actual *Process Inputs* to, and *Process Outputs* from, each *Process Step.* as well as the evaluation of each *Process Control* (which, in turn, determines the specific sequence of *Process Steps* performed during execution).

**Explanatory Text:** Each Process is an instance of executing a repeatable Process Step Design. At the time of Process Step Execution specific instances of input objects (for example, specific Data Sets, specific Variables) will be supplied.

Each instance of Process Step may produce unique results even though the Process Step Design remains constant. One reason is that specific instances of inputs are provided for each Process Step.

Even when the inputs remain the same, metrics such as the elapsed time to complete execution of process step may vary from execution to execution. For this reason, each Process Step Execution Record details of inputs and outputs for that instance of implementing the Process Step Design. It also records the outcome of Process Control evaluation at the end of the process step.

In this way it is possible to trace the flow of execution of a business process through all the process steps which were involved.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| Duration |  | 1..1 | int |
| End Time |  | 1..1 | int |
| Error Code |  | 1..1 | int |
| Error Message |  | 1..1 | int |
| Error Severity Level |  | 1..1 | int |
| Parent Process Step Execution |  | 1..1 | int |
| Start Time |  | 1..1 | int |
| Trigger Event |  | 1..1 | int |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| creates | **1..\*** ProcessOutput.  **1** ProcessStepExecutionRecord. |  |
| executes | **0..\*** ProcessStepExecutionRecord.  **0..\*** BusinessService. |  |
| has | **0..\*** ProcessStepExecutionRecord.  ProcessStep. |  |
| has | **1..\*** ProcessInput.input  **0..\*** ProcessStepExecutionRecord. |  |

##### ProcessSupportInput

**Package:** Production

**Definition:**. A form of *Process Input* that influences the work performed by the *Process Step*, and therefore influences its outcome.

**Explanatory Text:** *Process Support Input* is a sub-type of *Process Input*. Typical *Process Support Inputs* include metadata resources such as *Classifications* or structural information used in the processing of data.

Examples of *Process Support Inputs* could include:

- A *Code List* which will be used to check whether the *Codes* recorded in one dimension of a dataset are valid - An auxiliary *Data Set* which will influence imputation for, or editing of, a primary *Data Set* which has been submitted to the *Process Step* as the *Transformable Input*.

In these examples, which *Code List* to use, or which auxiliary *Data Set* to use, may be specified via a *Parameter Input*. The details of the *Code List* or the auxiliary *Data Set* are *Process Support Inputs*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | ProcessSupportInput.  ProcessInput. |  |

##### Rule

**Package:** Production

**Definition:** A specific mathematical or logical expression which can accept inputs and be evaluated based on those inputs.

**Explanatory Text:** There are many forms of *Rules* and their purpose, character and expression can vary greatly.

* Evaluation *Rules* consist of computing an output which will result in a particular course of action.
* The logicAl *Rules* implemented by a *Process Step* and their implementations in executable form.

A single *Rule* (at the conceptual level) may be expressed in different ways when using different notations and/or different software at the implementation level.

*Rules* can be "nested". In other words, a *Rule* can accept the outputs/evaluations from one or more other *Rules* as its inputs. This approach can be useful to achieve reuse of *Rules*.

A *Rule* can be used to generate new data (for example, determine values for a derived Variable) based on existing data. *Rules* can also be designed to apply "if then else" logic or "case" logic.

*Parameter Inputs* can be included in the definition of a *Rule* and values provided for those parameters at the time the *Rule* is evaluated.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| Algorhithm |  | 1..1 | int |
| Rule Type |  | 1..1 | int |
| Sequence Number |  | 1..1 | int |
| System Executable Indicator |  | 1..1 | int |
| Event Date |  | 1..1 | int |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | 0..\* Rule.  0..\* ProcessMethod. |  |
| has | 0..\* Rule.  **0..\*** ProcessControl. |  |
| uses | 0..\* Rule.  **0..\*** ControlTransition. |  |

##### TransformableInput

**Package:** Production

**Definition:** A type of *Process Input* whose content goes into a *Process Step* and is changed in some way by the execution of that *Process Step*. Some or all of the content will be represented in the *Transformed Output.*

**Explanatory Text:** *Transformable Input* is a sub-type of *Process Input*. Producers of official statistics often conceptualize data (and sometimes metadata) flowing through the statistical business process, having statistical value added by each *Process Step* and being transformed along the way.

The concept of *Transformable Input* allows this notional flow of information through the production process to be traced, without confusing these inputs with other inputs - such as *Parameter Inputs* and *Process Support Inputs* that are controlling or influencing a particular *Process Step* but do not "flow through the business process" in the same sense. Typical *Transformable Inputs* are *Data Sets* and structural metadata (if changed by a process and needed to describe another output or as an object in their own right).

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | TransformableInput.  ProcessInput. |  |

##### TransformedOutput

**Package:** Production

**Definition:** A *Process Output* (a result) which provides the "reason for existence" for the *Process Step*.

**Explanatory Text:** A *Transformed Output* is a sub-type of *Process Output*. Typically a *Transformed Output* is either a *Process Input* to a subsequent *Process Step* or it represents the final product from a statistical business process.

In many cases a *Transformed Output* may be readily identified as an updated ("value added") version of one or more *Transformable Inputs* supplied to the *Process Step* execution.

Note: If the output were no longer required then there would be no need for the *Process Step* in its current form.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | TransformedOutput.  ProcessOutput. |  |
| reviews | **0..\*** TransformedOutput.  **0..\*** ProcessControl. |  |

### Structures Group

#### DataSet Class Diagram

A Data Structure structures zero or more Data Sets and has one or more Identifier Components**,** one or more Measure Components. and zero or more Attribute Components**.**  The Identifier Component, Measure Component and Attribute Componentare specialisations of a Data Structure Component. The Data Structure Component is defined by one and only one Represented Variable, but a Represented Variable can be associated with zero or more Data Structure Components. A Represented Variable has zero or more instances i.e. Instance Variables.

A *Data Set* is structured by one and only one *Data Structure* and is a collection of one or more *Data Points*. A *Data Point* is an observation for one and only *Unit*. A *Data Point* has one observed *Datum* and one or more identifier *Datums*.

A *Datum* is a component of an *Instance Variable*. The *Instance Variable* is associated with one and only one *Represented Variable*. A *Datum* measures one and only one *Unit*, but a *Unit* may have zero or more *Datums.*



Figure 44. DataSet Class Diagram

#### UnitDataStructure Class Diagram

The *Unit Data Structure* is a specialization of the *Data Structure* for *Unit* data.

TheUnit Data Structure has zero or more component Logical Records and one or more Unit Identifier Components, one or more Unit Measure Components and zero or more Attribute Components. The Unit Identifier Component is a specialization of the Identifier Component and the Unit Measure Component is a specialization of the Measure Component.

The Logical Record has one or more identifiers i.e. the Unit Identifier Component and groups one or more Unit Measure Components.

Relationships between *Logical Records* are given by *Record Relationship*. The *Record Relationship* has one and only one target *Logical Record* and one and only one source *Logical Record.*



Figure 45. UnitDataStructure Class Diagram

#### UnitDataSet Class Diagram

The *Unit Data Set* is a specialization of a *Data Set*. It is a collection of one or more *Unit Data Points* and is structured by a *Unit Data Structure.*

The *Unit Data Point* is a specialization of the *Data Point.* The Unit Data Set records zero or more Unit Data Records. The *Unit Data Record* is structured by a *Logical Record* and groups one or more *Unit Data Points.*

The Logical Record is a component of the Unit Data Structure, but the Unit Data Structure may contain zero or more Logical Records.



Figure 46. UnitDataSet Class Diagram

#### DimensionalDataStructure Class Diagram

The *Dimensional Data Structure* is a specialization of the *Data Structure* for aggregated data.

The Dimensional Data Structure has one or more Dimensional Identifier Components**,** one or more Dimensional Measure Components and zero or more Dimensional Attribute Components.

The Dimensional Identifier Component is a specialization of the Identifier Component, the Dimensional Measure Component is a specialization of the Measure Component**,** and the Dimensional Attribute Component is a specialization of the Attribute Component.



Figure 47. DimensionalDataStructure Class Diagram

#### DimensionalDataSet Class Diagram

The *Dimensional Data Set* is a specialization of the *Data Set* for dimensional data.

The Dimensional Data Set has one or more Dimensional Data Points.The Dimensional Data Point is a specialization of the Data Point.



Figure 48. DimensionalDataSet Class Diagram

#### Data-Resource Class Diagram

A *Data Resource* is comprised of *Data Sets* that are made available as part of a *Acquisition Activity* (that is, made available by the data providers for data acquisition or resulting from the data acquisition activity) or as part of a *Dissemination Activity.*

The Data Resource, is discovered by means of the Data Flow/Provision Agreement/Data Location.

Each *Data Set* is made available at a specific *Data Location*. The *Data Location* specifies from where the data can be retrieved. This can be either a link to specific file containing the data or it can be a link to a service that will consume a query for the data and will return a *Data Set.* If the link is to a service then it is probable that the service is able to be queried for many types of data and so can provide many *Data Sets*. Each *Data Set* must be structured according to a known *Data Structure* (for example, a structure for Balance of Payments, Demography, Tourism, Education etc.). This link is achieved via the *Provision Agreement* and *Data Flow.*

The*Data Location* is associated to a specific *Provision Agreement* which identifies the *Data Provider* and the *Data Flow* that defines the both the type of data by the link to a category and the structure of the data by a link to the *Data Structure*. Data relating to a *Data Flow* can be structured by only one *Data Structure*.

It is mandatory that the*Data Set* is linked to a *Provision Agreement* to which it relates (that is, the union of the *Data Provider* and the *Data Flow*). A*Data Flow* can be linked to *Subject Fields* (for example, National Accounts, Balance of Payments, Demography) which supports data discovery as data can be classified in this way.



Figure 49. Data-Resource Class Diagram

#### DisseminationActivities Class Diagram

A Dissemination Activity is a specialization of a Statistical Activity. A Publication Activity is a specialization of a Dissemination Activity. A Dissemination Activity can be performed by aDissemination Service. A Product is created by a Publication Activity and includes one or moreRepresentations.



Figure 50. DisseminationActivities Class Diagram

#### Service Class Diagram

The Dissemination Service retrieves one or more Transformable Inputs, collects zero or more Process Inputs and one or more Output Specifications and returns zero or more Representations.

The Output Specification is a specialization of Parameter Input. The Representation represents one or more Identifiable Artefacts.



Figure 51 Service Class Diagram

##### AttributeComponent

**Package:** Structures

**Definition:** The role given to a *Represented Variable*in the context of a *Data Structure*. The role is to hold the pertinent information in addition to the identifiers and measures for a particular unit in a *Data Set.*

**Explanatory Text**: For example the publication status of an observation (e.g. provisional, final, revised), or information specific to the use of an Identifier in the context of a *Data Set.*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | AttributeComponent.  DataStructureComponent. |  |
|  | DimensionalAttributeComponent.  AttributeComponent. |  |
|  | UnitAttributeComponent.  AttributeComponent. |  |
| has | **0..\*** AttributeComponent.attribute  **1** DataStructure. |  |

##### DataFlow

**Package:** Structures

**Definition:** The *Data Flow* represents both the availability of data over time and the availability of sub sets of the possible data that could be made available according to a *Data Structure*.

**Explanatory Text:** There may be many data sets structured according to a *Data Structure*, perhaps made available at a pre-defined frequency (for example, monthly).

There can be many *Data Flows* that share the same *Data Structure*: for instance data for National Accounts may be compartmentalized into a number of *Data Flows* for organizational purposes or for data discovery purposes (there can be different *Data Flows* for different sub sets of National Accounts where each sub set is structured by the same *Data Structure*).

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| describesDataFor | **0..\*** ProvisionAgreement.  1 DataFlow. |  |
| groupedBy | 0..\* SubjectField.  0..\* DataFlow. |  |
| groups | 1..\* DataFlow.  0..\* DataResource. |  |
| structuredBy | 0..\* DataFlow.  **1** DataStructure. |  |

##### DataLocation

**Package:** Structures

**Definition:** Identifies where a *Data Set* can be retrieved from.

**Explanatory Text:** This could be a *Data Set* structured in a known format and retrievable via a URL, or the URL of a service that can be queried to return such a *Data Set*. It could also be the location of a publication.

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| location |  | 1..1 | char |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | 0..1 DataLocation.  0..\* DataSet. |  |
| providesDataAt | 0..\* DataLocation.  **1** ProvisionAgreement. |  |

##### DataPoint

**Package:** Structures

**Definition:** A placeholder in a *Data Set* for an item of factual information obtained by measurement or created by a production process.

**Explanatory Text**: Example for Unit Data: (1212123, 43) could be the age in years on the 1st of January 2012 of a person (*Unit*) with the social security number 1212123. The social security number is an identifying variable for the person whereas the age, in this example, is a variable measured on the 1st of January 2012.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DimensionalDataPoint.  DataPoint. |  |
|  | UnitDataPoint.  DataPoint. |  |
| has | **1..\*** DataPoint.dataPoint  1 DataSet. |  |
| has | **1..\*** Datum.identifier  DataPoint. |  |
| has | **1** Datum.observation  DataPoint. |  |
| has | **0..\*** Datum.attribute  0..\* DataPoint. |  |
| observationFor | DataPoint.  1 Unit. |  |

##### DataResource

**Package:** Structures

**Definition:** An organized collection of stored information made of one or more *Data Sets* which may be sourced from multiple *Acquisition* or *Statistical Activities*.

**Explanatory Text:** *Data Resources* are collections of structured or unstructured information that are used by a statistical activity to produce information. This information object is a specialization of an *Information Resource*.

**Synonyms:** data source

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DataResource.  InformationResource. |  |
| creates | **1..\*** StatisticalActivity.  0..\* DataResource. |  |
| groups | 1..\* DataFlow.  0..\* DataResource. |  |
| uses | 0..\* DataResource.  DataChannel. |  |

##### DataSet

**Package:** Structures

**Definition:** An organized collection of data.

**Explanatory Text**: Examples of *Data Sets* could be observation registers, time series, longitudinal data, survey data, rectangular data sets, event-history data, tables, data tables, cubes, registers, hypercubes, and matrixes.

A broader term for *Data Set* could be data.

A narrower term for *Data Set* could be data element, data record, cell, field

**Synonyms:** database, datafile, file, table

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | UnitDataSet.  DataSet. |  |
|  | DimensionalDataSet.  DataSet. |  |
| has | **1..\*** DataPoint.dataPoint  1 DataSet. |  |
| has | 0..1 DataLocation.  0..\* DataSet. |  |
| provisionedBy | DataSet.  **1** ProvisionAgreement. |  |
| structuredBy | 0..\* DataSet.  **1** DataStructure. |  |

##### DataStructure

**Package:** Structures

**Definition:** Defines the structure of an organized collection of data (*Data Set*).

**Explanatory Text**: The structure is described using *Data Structure Components* that can be either *Attribute Components, Identifier Components* or *Measure Components*. Examples for unit data include social security number, country of residence, age, citizenship, country of birth, where the social security number and the country of residence are both identifying components (*Unit Identifier Component*) and the others are measured variables obtained directly or indirectly from the person (*Unit*) and are *Unit Measure Components*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DimensionalDataStructure.  DataStructure. |  |
|  | UnitDataStructure.  DataStructure. |  |
| has | **1..\*** IdentifierComponent.identifier  **1** DataStructure. |  |
| has | **1..\*** MeasureComponent.measure  **1** DataStructure. | **Definition**  This association ..... |
| has | **0..\*** AttributeComponent.attribute  **1** DataStructure. |  |
| structuredBy | 0..\* DataSet.  **1** DataStructure. |  |
| structuredBy | 0..\* DataFlow.  **1** DataStructure. |  |

##### DataStructureComponent

**Package:** Structures

**Definition:** The identification of the *Represented Variable* used in the context of a *Data Structure*.

**Explanatory Text**: A Data Structure Component can be an Attribute Component, Measure Component or an Identifier Component.

Example of *Attribute Component*: The publication status of an observation such as provisional, revised.

Example of *Measure Component*: age and height of a person in a *Unit Data Set* or number of citizens and number of households in a country in a *Data Set* for multiple countries (*Dimensional Data Set*).

Example of *Identifier Component*: The personal identification number of a Swedish citizen for unit data or the name of a country in the European Union for dimensional data.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | MeasureComponent.  DataStructureComponent. |  |
|  | IdentifierComponent.  DataStructureComponent. |  |
|  | AttributeComponent.  DataStructureComponent. |  |
| definedBy | **0..\*** DataStructureComponent.  **1** RepresentedVariable. |  |

##### DimensionalAttributeComponent

**Package:** Structures

**Definition:**  A *Represented Variable*that is required to supply information in addition to the identification and measures of a *Dimensional Data Set*.

**Explanatory Text**: Example: The publication status of an observation such as provisional, revised.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DimensionalAttributeComponent.  AttributeComponent. |  |
| has | **0..\*** DimensionalAttributeComponent.attribute  **1** DimensionalDataStructure. |  |

##### DimensionalIdentifierComponent

**Package:** Structures

**Definition:** A *Represented Variable* that is required to identify or classify each observation value in a *Dimensional Data Set*.

**Explanatory Text**: Example: The name of a country in the European Union, the type of dwelling, the gender of a person, age-category of person

**Synonyms:** dimension

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DimensionalIdentifierComponent.  IdentifierComponent. |  |
| has | **1..\*** DimensionalIdentifierComponent.identifier  **1** DimensionalDataStructure. |  |

##### DimensionalMeasureComponent

**Package:** Structures

**Definition:** A *Represented Variable* that has been given a role in a collection of aggregated data to hold the summary values (means, mode, total, index, etc.) for a specific sub-population.

**Explanatory Text**: Examples: average age or total income in a sub-population

**Synonyms:** measure

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **0..\*** DimensionalDataPoint.valueFor  **1** DimensionalMeasureComponent.definedBy |  |
|  | DimensionalMeasureComponent.  MeasureComponent. |  |
| has | **1..\*** DimensionalMeasureComponent.measure  **1** DimensionalDataStructure. |  |

##### DimensionalDataPoint

**Package:** Structures

**Definition:** A placeholder or cell in a *Dimensional Data Set* determined by the crossing of (all) the values for the *Identifier* *Components* to contain the value (*Datum*) for an *Instance Variable* (defined by a *Measure Component*) with respect to a given *Unit*.

**Explanatory Text**: A *Dimensional* *Data Point* is uniquely identified by the combination of exactly one value for each of the dimensions (*Dimensional* *Identifier Component*) and one measure (*Dimensional* *Measure Component*).

There may be multiple values for the same *Dimensional* *Data Point* that is for the same combination of Dimension values and the same measure. The different values represent different versions of the data in the *Data Point*. Values are only distinguished on the basis of quality, date/time of measurement or calculation, status, etc. This is handled through the mechanisms provided by the *Datum* information object.

**Synonyms:** cell

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DimensionalDataPoint.  DataPoint. |  |
|  | **0..\*** DimensionalDataPoint.valueFor  **1** DimensionalMeasureComponent.definedBy |  |
| has | **1..\*** DimensionalDataPoint.dataPoint  **1** DimensionalDataSet. |  |

##### DimensionalDataSet

**Package:** Structures

**Definition:** A collection of dimensional data that conforms to a known structure.

**Synonyms:** hyper cube, macro data, n-cube, aggregated data, multi-dimensional data, dimensional data

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DimensionalDataSet.  DataSet. |  |
| has | **1..\*** DimensionalDataPoint.dataPoint  **1** DimensionalDataSet. |  |
| structuredBy | DimensionalDataSet.  DimensionalDataStructure. |  |

##### DimensionalDataStructure

**Package:** Structures

**Definition:** Defines the structure of a collection of aggregated data by Represented Variables (in their respective roles as Dimensional Measure Components, Dimensional Attribute Component or Dimensional Identifier Components) and their Value Domains.

**Explanatory Text**: This is similar to the SDMX Data Structure Definition: Set of structural metadata associated to a *Data Set*, which includes information about how *Concepts* are associated with the measures, dimensions, and attributes of a data cube, along with information about the representation of data and related descriptive metadata.

**Synonyms:** file description, data set description

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | DimensionalDataStructure.  DataStructure. |  |
| has | **1..\*** DimensionalMeasureComponent.measure  **1** DimensionalDataStructure. |  |
| has | **1..\*** DimensionalIdentifierComponent.identifier  **1** DimensionalDataStructure. |  |
| has | **0..\*** DimensionalAttributeComponent.attribute  **1** DimensionalDataStructure. |  |
| structuredBy | DimensionalDataSet.  DimensionalDataStructure. |  |

##### DisseminationService

**Package:** Structures

**Definition:** The mechanism for delivering, and possibly creating, structured content dynamically in response to a consumer request and in accordance with defined parameters as provided by that consumer.

**Explanatory Text**: A *Dissemination Service* will deliver a *Representation* created by a process that it invokes. The inputs into the *Dissemination Service* determine and feed the process that is to be invoked.

A *Dissemination Service* retrieves the information to be structured and delivered through an *Information Resource*. As part of the service execution, the consumer may be given a chance to browse or search through the collection of information available from the *Information Resource* exposed by the *Dissemination Service*. Based on the results, the consumer can than refine the *Output Specification* as (further) input to the *Dissemination Service* to complete the process of creating and delivering the information required in the form of a *Representation* to the consumer.

Example:

1. SDMX SOAP Data Web Services: The query XML message provides the Service with data selection and the specification of the preferred format (e.g. Generic format or Structured format, time series or cross-sectional). Based on this input the Service will retrieve a *Data Set* from the *Data Resource* and invoke a process that will format the data as an SDMX data message.

2. A manual service such as a response to a telephone request where the person answering the call based on the caller's request would mail a PDF (which might either be a *Product* or dynamically created from another source).

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| collects | DisseminationService.  **1..\*** OutputSpecification. |  |
| exposes | **1..\*** InformationResource.resource  **0..\*** DisseminationService. |  |
| performs | **0..\*** DisseminationActivity.  **0..\*** DisseminationService. |  |
| returns | **0..\*** DisseminationService.  **0..\*** Representation. |  |

##### IdentifierComponent

**Package:** Structures

**Definition:** The role given to a *Represented Variable* in the context of a *Data Structure*. The role is to identify the unit in an organized collection of data.

**Explanatory Text**: An *Identifier Component* is a sub-type of *Data Structure Component*. The personal identification number of a Swedish citizen for unit data or the name of a country in the European Union for dimensional data.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | IdentifierComponent.  DataStructureComponent. |  |
|  | DimensionalIdentifierComponent.  IdentifierComponent. |  |
|  | UnitIdentifierComponent.  IdentifierComponent. |  |
| has | **1..\*** IdentifierComponent.identifier  **1** DataStructure. |  |

##### InformationResource

**Package:** Structures

**Definition:** An abstract notion that is any organized collection of information.

**Explanatory Text:** The only concrete sub class is *Data Resource*. The *Information Resource* allows the model to be extended to other types of resource.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | <anonymous>.  InformationResource. |  |
|  | <anonymous>.  InformationResource. |  |
|  | DataResource.  InformationResource. |  |
| exposes | **1..\*** InformationResource.resource  **0..\*** DisseminationService. |  |

##### LogicalRecord

**Package:** Structures

**Definition:** Describes a type of *Unit Data Record* for one *Unit* within a *Unit Data Set*.

**Explanatory Text**: A *Logical Record* describes the record using variables of which one or more can uniquely identify the record (*Identifier Component*). It represents characteristics of a real or artificially constructed *Unit*, which could be represented by a *Concept*. The relationships between *Logical Records* are given by *Record Relationships*.

Examples: household, person or dwelling record

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **0..\*** LogicalRecord.  UnitDataStructure. |  |
| groups | **1..\*** UnitMeasureComponent.measure  0..\* LogicalRecord. |  |
| groups | **0..\*** UnitAttributeComponent.attribute  0..\* LogicalRecord. |  |
| identifiedBy | **1..\*** UnitIdentifierComponent.identifier  0..\* LogicalRecord. |  |
| relates | **0..\*** RecordRelationship.  **1** LogicalRecord.target |  |
| relates | **0..\*** RecordRelationship.  **1** LogicalRecord.source |  |
| structuredBy | **1** LogicalRecord.  **0..\*** UnitDataRecord. |  |

##### MeasureComponent

**Package:** Structures

**Definition:** The role given to a *Represented Variable* in the context of a *Data Structure*. The role is to hold the observed/derived values for a particular *Unit* in an organized collection of data.

**Explanatory Text**: A *Measure Component* is a sub-type of *Data Structure* Component. For example age and height of a person in a *Unit Data Set* or number of citizens and number of households in a country in a *Data Set* for multiple countries (*Dimensional Data Set*).

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | MeasureComponent.  DataStructureComponent. |  |
|  | DimensionalMeasureComponent.  MeasureComponent. |  |
|  | UnitMeasureComponent.  MeasureComponent. |  |
| has | **1..\*** MeasureComponent.measure  **1** DataStructure. | **Definition**  This association ..... |

##### NonStructuredDataSet

**Package:** Structures

**Definition:** A *Data Set* whose structure is not described in a *Data Structure.*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| has | NonStructuredDataSet.  0..1 DataProvider. |  |

##### OutputSpecification

**Package:** Structures

**Definition:** Contains the specifications for the dynamic creation and delivery of a *Representation* by a *Dissemination Service*.

**Explanatory Text**: An *Output Specification* is a specialization of *Parameter Input*. It is in fact a request for the dynamic creation and delivery of a *Representation*. It contains references to the information (e.g. a *Data Set*, a *Data Structure*, a *Code List,* a publication plan) desired with specifications concerning selections, (technical) form and/or method of delivery.

The references to the information come from the collection of information sources provided by the *Information Resource* that is exposed by the *Dissemination Service*. The consumer may select any (combination) of those information sources by including the references in the *Output Specification*.

Note that the *Output Specification* may be "soft" or "broad" in that it may identify groups of internal information objects rather than individual ones. For instance, all *Data Sets* within a certain (sub) category or theme. This may lead to multiple *Representations* being delivered.

As part of the *Output Specification*, the consumer may be given the option to select one of a number of possible formats for the *Representation* (e.g. SDMX, CSV, JSON or PDF) or to select one of a number of possible methods for delivery (web service response, email, FTP, mail delivery, etc.)

The *Dissemination Service* may be used to request future deliveries of *Representations* for information that is not yet available. This results in a subscription, where the specification of the *Representations* to be delivered in future is given in the *Output Specification.*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| collects | DisseminationService.  **1..\*** OutputSpecification. |  |
| defines | **0..\*** OutputSpecification.  **1** Representation. |  |

##### Product

**Package:** Structures

**Definition:** Static package of objects that can be disseminated as a whole.

**Explanatory Text**: A *Product* is a static presentation of artefacts created by fixed processes. The artefacts may be representations of data, visualizations, explanation, interpretation etc. Example: Publications, press releases, articles, list of classifications, etc.

**Synonyms:** publication

**Attributes**

| Name | Description | Cardinality | Value Type |
| --- | --- | --- | --- |
| expirationDate | Timestamp which expresses when the product should no longer be used. | 1..1 | dateTime |
| frequency | A duration which expresses the amount of time between the releases of the product. | 1..1 | duration |
| productionDate | Date and time that the product was produced. | 1..1 | dateTime |

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | StatisticalProgramCycle.  0..\* Product. |  |
| createdFrom | 0..\* Product.  **1** PublicationActivity. |  |
| includes | **1..\*** Representation.  0..\* Product. |  |

##### ProvisionAgreement

**Package:** Structures

**Definition:** A service-level agreement, a legal mandate, the terms of a mutual agreement, a memorandum of understanding, or any other terms/conditions which affect the provision of data.

**Explanatory Text:** The *Provision Agreement* does not need to have any formal consent of the *Data Provider*. For instance data collection via web scraping may identify the *Data Provider* but requires no formal agreement. A web service that provides data to anyone that queries it also may not need any formal agreement (save that perhaps of implicit agreement under the terms of the web service). Nevertheless, in both these cases the data may be structured according to a *Data Structure* which is associated to the *Data Flow*.

A *Provision Agreement* represents the union of a specific *Data Provider* and a specific *Data Flow* for which the *Data Provider* supplies data. The location of the *Data Sets* that are available for this *Provision Agreement* are associated in the *Data Location*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| describesDataFor | **0..\*** ProvisionAgreement.  1 DataFlow. |  |
| providesDataAt | 0..\* DataLocation.  **1** ProvisionAgreement. |  |
| providesDataFor | **0..\*** ProvisionAgreement.  **1** DataProvider. |  |
| provisionedBy | DataSet.  **1** ProvisionAgreement. |  |

##### RecordRelationship

**Package:** Structures

**Definition:** Describes relationships between *Logical Records* within a *Unit Data Structure*. It must have both a source *Logical Record* and a target *Logical Record* in order to define the relationship.

**Explanatory Text**: All relationships are defined in pairs. Hence multiple relationships may be needed to clarify all *Record Relationships* within a *Unit Data Set* e.g. household and person, household and dwelling etc.

Example: Relationship between person and household *Logical Records* within a *Unit Data Set*.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
| relates | **0..\*** RecordRelationship.  **1** LogicalRecord.target |  |
| relates | **0..\*** RecordRelationship.  **1** LogicalRecord.source |  |

##### Representation

**Package:** Structures

**Definition:** A "custom-built" artefact that has a consumable (human or machine) format. It is the output of a *Dissemination Service*. It is what is ultimately delivered to the consumer.

**Explanatory Text**: A *Representation* brings together various maintainable artefacts and their related artefacts. It is essentially the application of rules to an artefact (and possibly its related artefacts) which transform the object into a format fit for consumption. This consumption may be something that is understandable to a person or a machine.

*Representation* can be in different forms; e.g. tables, graphs, structured data files. Examples:

- A table of data. Based on a *Data* *Set*, the related *Data Structure* is used to label the column and row headings for the table. The *Data Set* is used to populate the cells in the table. Reference metadata is used to populate footnotes and cell notes on the table. Confidentiality rules are applied to the *Data Set* to suppress any disclosive cells.

- A data file based on a standard (e.g. SDMX). - A PDF document describing a *Classification*.

- Any structural metadata object expressed in a standard format (e.g. DDI 3.1 XML).

- A list of *Products* or services (e.g. a product catalogue or a web services description language (WSDL) file).

- A web page containing *Classifications*, descriptions of *Variables*, etc.

**Synonyms:** presentation, publication, delivery, product

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | <anonymous>.  Representation. |  |
| defines | **0..\*** OutputSpecification.  **1** Representation. |  |
| includes | **1..\*** Representation.  0..\* Product. |  |
| represents | **0..\*** Representation.  **1..\*** IdentifiableArtefact. |  |
| returns | **0..\*** DisseminationService.  **0..\*** Representation. |  |

##### UnitAttributeComponent

**Package:** Structures

**Definition:** A *Represented Variable*that is required to supply information in addition to the identification and measures in a *Unit Data Set*.

**Explanatory Text**: Example: The publication status of an observation such as provisional, revised.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | UnitAttributeComponent.  AttributeComponent. |  |
| groups | **0..\*** UnitAttributeComponent.attribute  0..\* LogicalRecord. |  |
| has | **0..\*** UnitAttributeComponent.attribute  **1** UnitDataStructure. |  |

##### UnitDataPoint

**Package:** Structures

**Definition:** A placeholder in a *Unit Data Record* to contain the value (*Datum*) for an *Instance Variable* with respect to a given *Unit*.

**Explanatory Text**: For example (1212123, 43) could be the age in years on the 1st of January 2012 of a person (Unit) with the social security number 1212123. The social security number is an identifying variable for the person whereas the age, in this example, is a variable measured on the 1st of January 2012. The value can be obtained directly from the *Unit* or indirectly via a process of some kind.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | UnitDataPoint.  DataPoint. |  |
|  | **1..\*** UnitDataPoint.dataPoint  UnitDataSet. |  |
|  | UnitDataPoint.  **1..\*** InstanceVariable.identifiers |  |
|  | UnitDataPoint.valueFor  InstanceVariable.measurement |  |
|  | UnitDataPoint.  **0..\*** InstanceVariable.attributes |  |
| groups | **1..\*** UnitDataPoint.  UnitDataRecord. |  |

##### UnitDataRecord

**Package:** Structures

**Definition:** Contains the specific values (as a collection of *Unit Data Points*) related to a given *Unit* as defined in a *Logical Record*.

**Explanatory Text**: For example (1212123, 48, American, United Kingdom) specifies the age (48) in years on the 1st of January 2012 in years, the current citizenship (American), and the country of birth (United Kingdom) for a person with social security number 1212123.

The *Unit Data Record* is a collection of *Unit Data Points* that provide either a complete or restricted view of the state of a *Unit* as observed over a specific period or at a specific point in time.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **0..\*** UnitDataRecord.record  **1** UnitDataSet. |  |
| groups | **1..\*** UnitDataPoint.  UnitDataRecord. |  |
| structuredBy | **1** LogicalRecord.  **0..\*** UnitDataRecord. |  |

##### UnitDataSet

**Package:** Structures

**Definition:** A collection of data that conforms to a known structure and describes aspects of one or more *Units*.

**Explanatory Text**: Example: A synthetic unit record file is a collection of artificially constructed *Unit Data Records*, combined in a file to create a *Unit Data Set*.

**Synonyms:** micro data, unit data, synthetic unit record file

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | **1..\*** UnitDataPoint.dataPoint  UnitDataSet. |  |
|  | **0..\*** UnitDataRecord.record  **1** UnitDataSet. |  |
|  | UnitDataSet.  DataSet. |  |
| structuredBy | 0..\* UnitDataSet.  **1** UnitDataStructure. |  |

##### UnitDataStructure

**Package:** Structures

**Definition:** Describes the structure of a *Unit Data Set*.

**Explanatory Text**: For example (social security number, country of residence, age, citizenship, country of birth) where the social security number and the country of residence are the identifying components (*Unit* *Identifier Component*) and the others are measured variables obtained directly or indirectly from the person (*Unit*) and are *Unit Measure Components* of the *Logical Record*.

**Synonyms:** file description, dataset description

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | UnitDataStructure.  DataStructure. |  |
|  | **0..\*** LogicalRecord.  UnitDataStructure. |  |
| has | **1..\*** UnitIdentifierComponent.identifier  UnitDataStructure. |  |
| has | **1..\*** UnitMeasureComponent.measure  UnitDataStructure. |  |
| has | **0..\*** UnitAttributeComponent.attribute  **1** UnitDataStructure. |  |
| structuredBy | 0..\* UnitDataSet.  **1** UnitDataStructure. |  |

##### UnitIdentifierComponent

**Package:** Structures

**Definition:** The role that has been given to a *Represented Variable*, in a *Unit Data Structure*, to identify the *Unit*.

**Explanatory Text**: For example the person identification number in Norway.

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | UnitIdentifierComponent.  IdentifierComponent. |  |
| has | **1..\*** UnitIdentifierComponent.identifier  UnitDataStructure. |  |
| identifiedBy | **1..\*** UnitIdentifierComponent.identifier  0..\* LogicalRecord. |  |

##### UnitMeasureComponent

**Package:** Structures

**Definition:** The role that has been given to a specific *Represented Variable* to hold the observed or derived values related to a Unit as identified by the *Unit Identifier Components*, in an organized collection of data.

**Explanatory Text**: For example age and height of a person in a *Unit Data Set*

Relationships

| Columns | Association | Notes |
| --- | --- | --- |
|  | UnitMeasureComponent.  MeasureComponent. |  |
| groups | **1..\*** UnitMeasureComponent.measure  0..\* LogicalRecord. |  |
| has | **1..\*** UnitMeasureComponent.measure  UnitDataStructure. |  |