Annex B Influence of existing standards

Note: To translate this paper into over 50 languages, please see the Automatic translation option at the top of the screen

Annex B Influence of existing standards

A. Introduction

132. GSIM must be implementable: In order to support the implementation of the GSIM reference framework, many known standards and tools have also been examined, to ensure that the reference framework is complete and useful in this respect. This section describes the influences of and relationships to a number of relevant standards.

133. Figure 24 illustrates how different relevant standards, models, and implementation syntaxes and tools relate to GSIM. Standards and models that have provided significant input to GSIM are presented on the left hand side of the figure. Implementation syntaxes and tools that are currently of relevance to an implementation of GSIM are presented on the right hand side of the figure. This list will become outdated as more and more implementation syntaxes and tools are developed. The particular software packages listed are widely used in statistical organizations, but are intended to be illustrative examples, and are not a complete list.

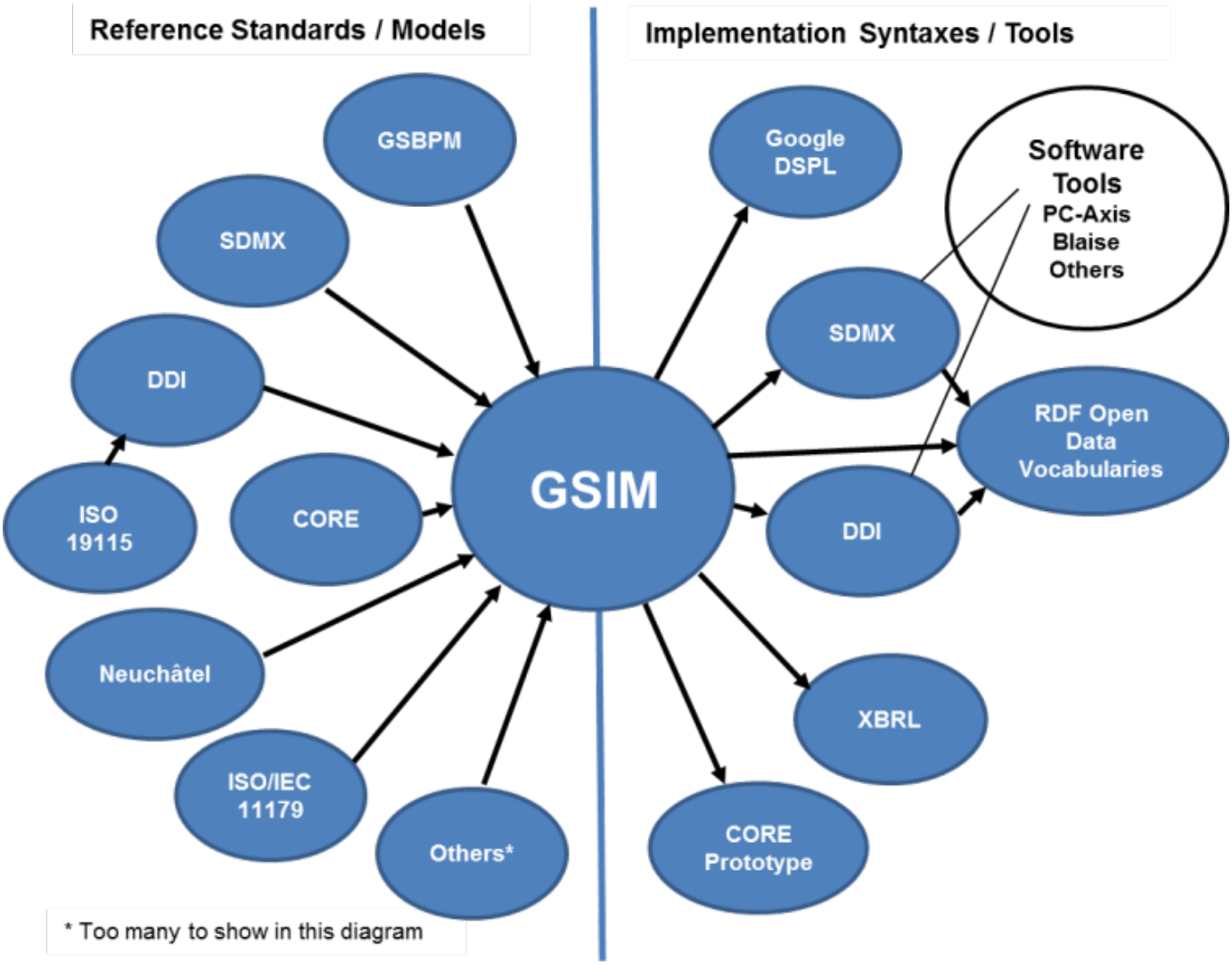

Figure 24: GSIM and its relationship to other relevant standards and models

B. Generic Statistical Business Process Model (GSBPM)

134. GSBPM provides descriptions of business processes that can occur throughout the statistical production process. It is a framework for categorizing processes. In order to describe a process in a level of actionable detail, more information is needed.

135. GSBPM explicitly excludes descriptions of flows within processes. This additional information is necessary if you wish to have reusable processes that talk about "flow" rather than just the specific functions which need to be performed during the flow (with no description of how they fit together).

136. Information needs to:

- Flow between GSBPM processes. For example, data are processed or transformed between the Collect and Disseminate phases.
- Govern the behaviour of GSBPM sub-processes. There are business rules and derivation formulas that are applied during processes (for example Impute, Derive New Variables). There are also rules or plans that determine which process should be performed next. An example of this is whether the quality of the data is sufficient to proceed to the next step or whether some form of remedial processing is required.
- Report on the outcome of GSBPM processes. For example, process related statistical quality metrics such as response rates or imputation variance.

137. The GSIM Production Group seeks to provide a standard way to capture this information about processes. It includes information objects such as *Process Step, Process Step Design, Process Step Execution Record, Rule, Process Input* and *Process Output.*

138. GSIM is designed to support current production processes and facilitate the modernization of statistical production. Implementation of GSIM, in combination with GSBPM, will lead to more advantages that are important. GSIM will:

- create an environment prepared for reuse and sharing of methods, components and processes;
- provide the opportunity to implement rule based process control, thus minimizing human intervention in the production process;
- generate economies of scale through development of common tools by the community of statistical organizations.

? Unknown Attachment

Figure 25. GSIM and GSBPM

C. Data Documentation Initiative (DDI)

139. The DDI Alliance supports the development of the GSIM information model and finds many parallels between the model and the DDI Lifecycle specification. The DDI Alliance is interested in working closely with the GSIM group to extend the modeling effort to encompass the definition of lower-level elements and attributes to provide actionable metadata that can be used to drive production and data collection processes.

Relationship with GSIM Business Group

140. The DDI standard is not designed to describe all aspects of a statistical program. However, there is a solid alignment with this portion of GSIM, especially as it relates to describing the data and metadata which are used by a particular activity. The primary link between GSIM and DDI, in this regard, is the DDI 'Study Unit' and the GSIM *Statistical Program Cycle* information object. All of the data and metadata associated with a particular cycle of a *Statistical Program* can be described using DDI XML, and the relationship of the different information objects can be described.

141. As DDI has a 'lifecycle' orientation, it is useful for describing many different aspects of the data, from collection through to dissemination. DDI provides a very rich description of a survey instrument, and this can be used to implement the GSIM *Survey Instrument* information object. It is easy to see that the DDI 'ControlConstruct' elements can be implementations of the GSIM *Instrument Control* information object, although these are more detailed in the DDI implementation model.

142. GSIM and DDI both model the existence of such information objects as *Questions* and *Interviewer Instructions*, as opposed to the use of these resources in a survey instrument, in the same way. This is important when it comes to re-use, as a question which is bound to a specific survey (for example) becomes non-reusable. Both DDI and GSIM see a similar set of when describing a *Survey Instrument: Questions, Statements, Question Blocks,* and *Interviewer Instructions.* These information objects are shared by both standards.

Relationship with GSIM Production Group

143. In the current versions of DDI, there is very little content related to the management of statistical production. However, there are plenty of metadata to describe some specific types of data processing. In addition, DDI provides a way of recording 'Lifecycle Events', which can record any kind of processing or production event and associate it with other identifiable metadata information objects. DDI is very useful when it comes to describing many of the data and metadata inputs and outputs for these processes. (It should be noted that *Process Metrics* are often themselves data sets, and can be described as such in DDI.)

144. Another strong feature of DDI relative to GSIM is the ability to describe data collection activities. There is in DDI elements for describing 'CollectionEvents', which can be associated with specific variables populated (although this feature is not required). While it would be intuitive to associate 'CollectionEvents' in DDI with data acquisition activities in GSIM, there is also a relationship with data processing activities.

145. In future versions of DDI, it is likely that the ability to associate processing and production information with metadata information objects will be enhanced. There were several extensions to this capability found in DDI version 3.2, and this feature will be revisited and enhanced in future versions of DDI.

146. 'Processing Events' in DDI can be used to describe some types of processes as well: 'Control Operations', 'Cleaning Operations', 'Codings', 'Data Appraisal Information', and 'Weighting'. While GSIM does not provide a breakdown of these types of processes, it is easy to see where these might fit into a process model such as GSBPM. What is captured by DDI, however, is the same type of information content as is found in GSIM.

147. The types of processing described by DDI include different types of 'Codings': 'Generation Instructions 'and 'General Instructions '. For each of these it is possible to provide a textual description of the process; to link to or insert the actual program code used to execute the process; and, in the case of generation instructions, it is possible to link to the variables manipulated by a derivation process. This model is in some ways similar to what is found in GSIM – it lacks tie-backs to the methodology used, and also to the explicit business function, but in some ways (inputs, code and controls applied, outputs) is fairly similar to GSIM.

148. 'Generation Instructions' describe processes used to create new variables from existing ones. Tabulation of data often requires the tabulation of new variables. This DDI structure is similar to a GSIM *Process Step Execution Record*, and includes some additional information (such as the processing code). The DDI structure is not perhaps optimal, because there will potentially be a lot of detail in a *Process Step Execution Record* placed into a text field in the DDI XML – the description field of the 'Generation Instruction'.

149. 'General Instructions' are used in DDI to describe other types of processing, in a manner similar to 'Generation Instructions'. This can cover the *Process Step Execution Record* portion of the GSIM model. 'Data Appraisal' includes information such as sampling error and response rate, which may be useful for some processes as *Process Metrics*. Other 'Processing Events' are simple descriptions.

150. 'Lifecycle Events' can be used to associate any process with the relevant inputs and outputs. Typically a process model (such as GSBPM) is used to distinguish types of 'Lifecycle Events', but there is no rule in DDI to prevent the process references being at a more detailed level.

151. DDI does not provide a mechanism for describing *Process Step Design*, but was designed to work with a separate description of this information, expressed in BPMN, or BPEL, for example.

Relationship with GSIM Concepts Group

152. DDI as a standard describes many of the foundational metadata objects which are modelled in the GSIM Concepts Group. *Concepts, Categories, Codes, Variables*, and *Populations* (in DDI, 'Universes') are all present in both DDI and GSIM. There is no dedicated way of representing a *Classification* in DDI – it is simply a pairing of a 'category scheme' and a 'code scheme' – but otherwise the two models are very similar. One major difference in this area is that DDI (and, indeed, all other models) lack the concept of what is described in GSIM as a *Node*. This is a key improvement to managing this type of metadata which is in GSIM, and is expected that it will be reflected in future versions of DDI.

153. One feature of GSIM which is more nuanced that DDI is in the set of *Variable* information objects. In GSIM there is a separation between *Variable, Represented Variable*, and *Instance Variable*. In DDI 3.1, there is only the instance variable included (called 'Variable' in DDI terminology). In DDI 3.2, the standard has added what it terms a 'Data Element', corresponding to the GSIM *Represented Variable* information object.

154. GSIM also has a richer set of concept links between various information objects than the current versions of DDI-Lifecycle. It is anticipated that the DDI model will be adjusted to include such linkages in future, as a response to GSIM. In DDI, there are links between concepts, questions, variables, and levels within classifications. In DDI 3.2, links to categories have been added. This is a more consistent model in GSIM, where concepts links are also applied to populations.

Relationship with GSIM Structures Group

155. The GSIM Structures Group maps well to DDI, especially in regards to the description of unit data. As GSIM is a conceptual model, it does not go into all of the implementation detail found in DDI for describing the storage of data. However, at the logical level, the two models are very compatible. Variables in DDI play a few specialized roles, including the identification of unit records, observations about those unit records, and additional supporting information such as weights. This maps very cleanly onto the GSIM model.

156. Further, DDI also has a concept of 'NCubes', multi-dimensional data sets. These also exist in GSIM in the form of *Dimensional Data Sets*. DDI here has variables playing roles of identifying (dimensions), measures (observations), and attributes (attributes) and as such is very much like the model found in GSIM. Both models tie the values here back to the variables in which they are stored as well.

157. What DDI is largely lacking are the constructs used to manage the data, such as *Data Flows, Data Channels, Provision Agreements*, etc. The mis-match here largely results from the fact that DDI is fundamentally organized around a lifecycle model, rather than a model of exchange like SDMX. It will seen how much of this type of metadata will be introduced into the DDI model in future – it is likely that GSIM itself may dictate that this type of information be better supported.

D. Statistical Data and Metadata eXchange (SDMX)

Relationship with GSIM Business Group

158. In general, SDMX does not cover explicitly the constructs in the Business group. However, the SDMX 'Metadata Structure Definition' and related 'Metadata Set' are used to describe and to provide quality, methodological and other reference metadata. These metadata not modelled explicitly in GSIM but are rather embedded in other GSIM constructs. These same SDMX constructs would also be used to map the metadata of the GSIM *Statistical Need, Assessment,* and *Business Case.*

159. GSIM has additional information about the *Data Channel, Instrument, Instrument Control, Question Scheme, Information Request,* and *Statistical Program* which are not found in SDMX.

Relationship with GSIM Production Group

160. The SDMX standard is primarily focused on the description of aggregate data sets and related metadata of various types. These various types of data and metadata are used as inputs and outputs by statistical processes. However, SDMX also contains some structures which are relevant to an implementation of the GSIM Production Group. Key among these is the ability SDMX provides to describe processes and process steps.

161. There is quite a good fit between the SDMX process model – which is made up of a set of nested, hierarchical sub-steps – and the GSIM approach, which is more detailed, but essentially similar. A *Process***,** its constituent *Process Steps* and its associated *Process Control* and *Rule* information objects describe essentially the same information as the SDMX 'Process', 'Process Step', 'Transition', and 'Computation' description: the flow of a process and the data and metadata inputs and outputs.

162. Whilst SDMX supports 'Process Artefact' for inputs and outputs, there is no link in SDMX to what provides the inputs. GSIM *Process Input* is provided by the *Statistical Program Design* (static design input) and the *Statistical Activity* (dynamic "run time" input).

163. Other parts of the GSIM production model could be implemented with SDMX as reference metadata, but the utility of this will depend very much on what the GSIM implementation is being built to do.

Relationship with GSIM Conceptual Group

164. The SDMX standard contains many of the foundational metadata objects used in this part of GSIM. The level of detail is somewhat different, because SDMX does not make a distinction between the meaning of a code (a *Category*) and the *Code* itself – both are bundled together into a 'Codelist' in SDMX. However, the same information about hierarchies (in SDMX, 'Hierarchical Codelists') can be expressed. There is no distinct classification information object in SDMX – classifications are described using a combination of SDMX 'Codelists' and 'Hierarchical Codelists' information objects. The 'Hierarchical Codelist' model in SDMX was developed with classification support as one use case.

165. Both the GSIM and the SDMX models have *Concepts* as an important construct, although the linkages to concepts are richer in GSIM than in SDMX.

166. As SDMX focuses on aggregate data, there is no information object representing a variable, which is different than in GSIM. When used to describe data, 'Concepts' in SDMX can be mapped to *Variables* as they appear in GSIM, however, such that the SDMX 'Concept' represents a collapsed GSIM *Concept* and *Variable.* SDMX has no explicit support for the GSIM *Population***.**

Relationship with GSIM Structures Group

167. GSIM has a number of constructs in the Information area which will be familiar to people using SDMX. The *Dimensional Data Set* corresponds to an SDMX 'Data Set', and an SDMX 'Data Structure Definition' corresponds to a GSIM *Dimensional Data Structure*.

168. The *Data Resource* model contains information objects from SDMX such as *Data Flows, Data Providers*, and *Provision Agreements*, in a very similar form. The way GSIM groups *Data Flow* by *Data Resource* and *Subject Field* would be supported in SDMX by 'Category' (this is not the same as a GSIM *Category)* and 'Categorisation'.

E. ISO/IEC 11179

169. ISO/IEC 11179 is a standard for describing and managing the meaning and representation of data. It specifies the kind and quality of metadata necessary to describe data. The GSIM Concepts Group contains a terminological description of data. This is similar in many respects to 11179.

170. However, 11179 also specifies the management and administration of metadata in a metadata registry, Registration is the process of managing the content and quality of descriptions, and this is supported explicitly in 11179. GSIM does not seek to replicate this work.

171. There are a number of constructs which are similar in 11179 and GSIM. Table 3 shows the pairs of constructs are equivalent in the two specifications:

Table 3. Similar Constructs in 11179 and GSIM

11179	*GSIM*
Object Class	*Population*
Property	*Variable*
Value Domain	*Value Domain*

Enumerated Value Domain	*Enumerated Value Domain*
Described Value Domain	*Described Value Domain*
Conceptual Domain	*Conceptual Domain*
Enumerated Conceptual Domain	*Enumerated Conceptual Domain*
Described Conceptual Domain	*Described Conceptual Domain*
Concept System	*Concept System*
Unit of Measure	*Unit of Measure*
Datatype	*Data Type*

172. Dimensionality is specified as well in 11179. It identifies those units of measure that are equivalent. For example, miles per hour, meters per second, and furlongs per fortnight all measure speed; and they are equivalent measures. Data measured in any one of those units can be converted without loss of information to any of the others. This is only lightly supported in GSIM.

173. The notion of classifications is more explicitly defined in GSIM than in 11179. The following objects related to classifications are defined in GSIM and not in 11179: *Category Set, Code List, Datum, Nodes* and *Node Sets.*

F. ISO 704

174. Both GSIM and 11179 base their description of data on the principles laid out in ISO 704. However, GSIM does a more careful job of making sure these principles are followed precisely. In GSIM, *Populations*, *Variables*, and *Categories* (called a property in 704) are all laid out as roles for *Concepts*, and these have parallels to the principles defined in 704.

175. In contrast to 704, GSIM explains more clearly the relationships between: (a) concepts (*Populations* in GSIM) and characteristics (*Variable***s** in GSIM) and (b) objects (not explicit in GSIM but the individual units from which measurements are taken) and properties (*Categories* in GSIM)

G. Neuchâtel Terminology for Classifications

176. A statistical classification is often described as a tool that is used to handle and structure objects systematically into categories in the production of statistics http://unstats.un.org/unsd/class/intercop/expertgroup/2011/AC234-6.PDF. Neuchâtel Terminology for Classifications http://www1.unece.org/stat/platform/download/attachments/14319930/Part+I+Neuchatel_version+2_1.pdf?version=1 is one of the most used standards for classification management.

177. The Neuchâtel terminology definition of classification:

"A classification version is a list of mutually exclusive categories representing the version-specific values of the classification variable. If the version is hierarchical, each level in the hierarchy is a set of mutually exclusive categories. A classification version has a certain normative status and is valid for a given period of time. A new version of a classification differs in essential ways from the previous version. Essential changes are changes that alter the borders between categories, that is, a statistical object/unit may belong to different categories in the new and the older version. Border changes may be caused by creating or deleting categories, or moving a part of a category to another. The addition of case law, changes in explanatory notes or in the titles do not lead to a new version."

178. One important difference between GSIM and the Neuchâtel terminology for classifications is that GSIM separates meaning and representation. Table 4 below show how GSIM maps to the Neuchâtel Terminology for Classifications:

Table 4. Mapping between Neuchâtel Terminology for Classifications and GSIM

Neuchâtel terminology	GSIM	Example	Comment
Classification family	*Classification Family*	Activity (Industry) classifications, Educational classifications	Group of Classifications
Classification	*Classification*	NACE, ISIC, ISCO, ANZIC06, NAICS	Group of Classifications Schemes
Classification version	*Classification Version*	NACE rev 2, ISIC rev 4, ISCO 08, ANZIC06, NAICS	

Classification variant	*Classification Variant*	High-level SNA/ISIC aggregation A*10/11 grouping	
Classification level	*Level*	Section, division, group and class in ISIC rev 4	
Classification item	*Classification Item*	0111 - Growing of cereals (except rice), leguminous crops and oil seeds	
Correspondence table	*Correspondence Table*	ISIC rev 4 - NAICS	
Classification index			List of aliases
Classification index entry			Aliases
Item change			
Case law			
Classification index entry	Alias on *Node*		
Correspondence item	*Map*	0111 in ISIC - 111110 NAICS	
Classification item - code	Attribute on *Classification Item*	0111 (in ISIC)	Not an information object in itself in GSIM
Classification item - title	Attribute on *Classification Item*	Growing of cereals (except rice), leguminous crops and oil seeds	Not an information object in itself in GSIM

Classification item - explanatory notes	Attribute on *Classification Item*	"This class includes: • growing of temporary and permanent crops • cereal grains: rice, hard and soft wheat, rye, barley, oats, maize, corn (except sweetcorn) etc. • growing of potatoes, yams, sweet potatoes or cassava • growing of sugar beet, sugar cane or grain sorghum • growing of tobacco, including its preliminary processing: harvesting and drying of tobacco leaves • growing of oilseeds or oleaginous fruit and nuts: peanuts, soya, colza etc. • production of sugar beet seeds and forage plant seeds (including grasses) • growing of hop cones, roots and tubers with a high starch or inulin content • growing of cotton or other vegetal textile materials • retting of plants bearing vegetable fibres (jute, flax, coir) • growing of rubber trees, harvesting of latex • growing of leguminous vegetables such as field peas and beans growing of plants used chiefly in pharmacy or for insecticidal, fungicidal or similar purposes • growing of crops n.e.c. *This class excludes:* • *growing of melons, see 0112* • *growing of sweet corn, see 0112* • *growing of other vegetables, see 0112* • *growing of flowers, see 0112* • *production of flower and vegetable seeds, see 0112* • *growing of horticultural specialties, see 0112* • *growing of olives, see 0113* • *growing of beverage crops, see 0113* • *growing of spice crops, see 0113* • *growing of edible nuts, see 0113* • *gathering of forest products and other wild growing material (cork, resins, balsam etc.), see 0200*"	

H. Business Process Model and Notation (BPMN)

179. BPMN provides a standard means to document business processes, including representing them graphically. GSIM does not try to duplicate the richness of modelling in BPMN. It simply aims to establish a high level connection.

180. There are two main objects in GSIM that have a direct relationship with BPMN. These are *Process Step Designs* and *Process Control* (sh own in Table 5).

Table 5. Similar constructs in BPMN and GSIM

BPMN	*GSIM*
Process	A high level *Process Step*
Sub-process	An intermediate level *Process Step*
Task	A low level (atomic) *Process Step*
Sequence Flow	A *Process Control* (in cases where the flow between process steps is invariable)
Gateway	A *Process Control* (in cases where the flow between process steps is evaluated at the time of execution)

181. The BPMN V2.0 specification explicitly notes that BPMN is not a 'data flow language'. BPMN can represent 'data objects' but does not explicitly model them in detail. GSIM does model these objects explicitly (*Process Input Specifications, Process Inputs, Process Output Specifications and Process Outputs*)**.**

182. The BPMN V2.0 specification also explicitly excludes

- modelling of functional breakdowns (GSIM *Business Functions*) business rule models (GSIM *Process Methods*)

I. COmmon Reference Environment (CORE)

183. The CORE model is a communication protocol for the exchange of information between a CORE service (a service designed with the help of CORE information objects) and its environment (an implementation of CORE on any specific platform). The CORE model knows of the existence of statistical information objects, but knows nothing else about them.

184. In CORE, a 'channel' is a communication line between a service and its environment. A 'channel' is specialized in the transportation of specific objects by referring to their 'kind definition' (for example, Data set kind – constraining Data set definitions; Column kind – constraining Column definitions; Rule kind – constraining rules; etc.). There is a channel kind labeled 'GSIM Object Description', which will accept a GSIM object without understanding its contents, structure or meaning.

185. Figure 26 shows the constructs which are similar in CORE and GSIM.

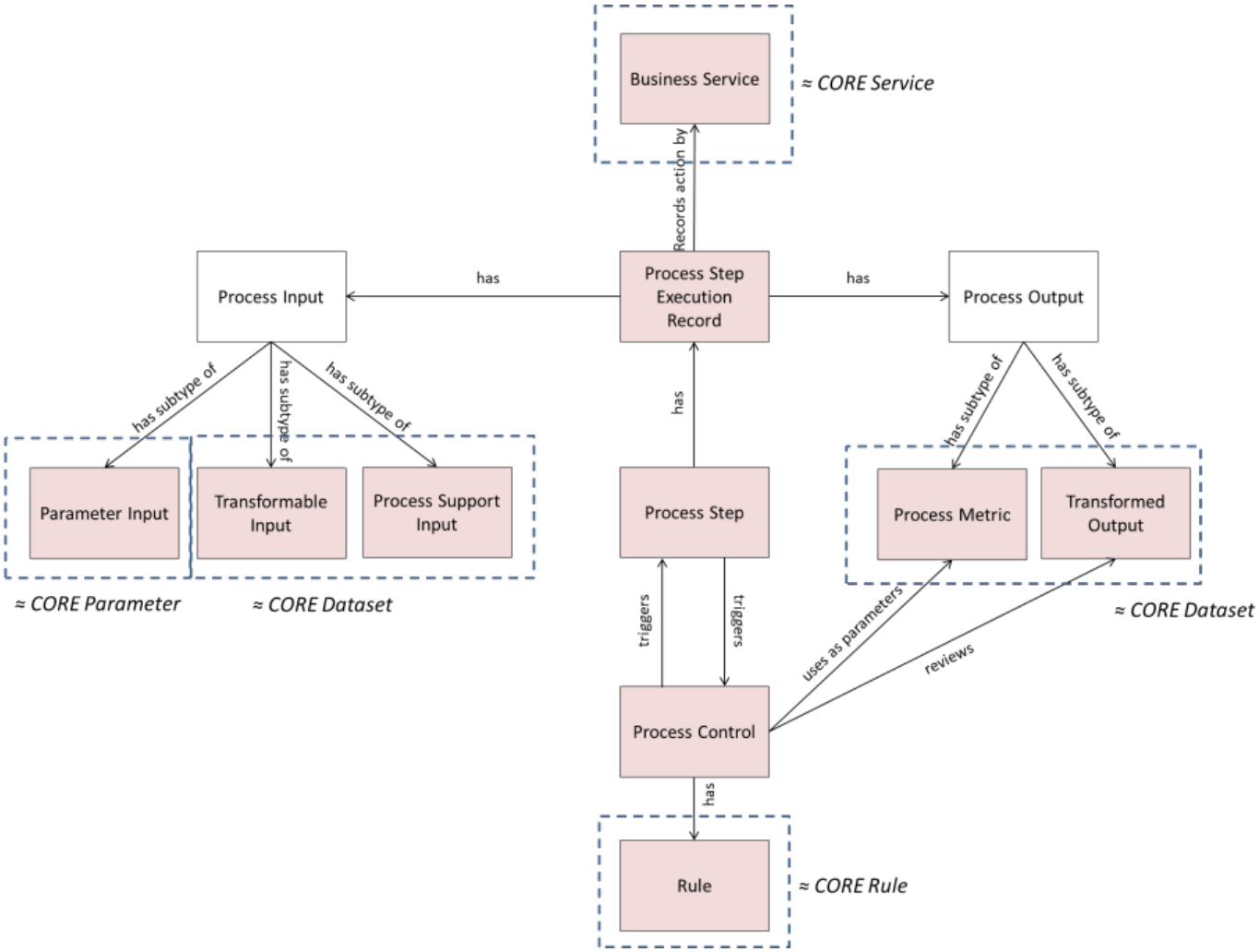

Figure 26. CORE and GSIM

J. The Open Group Architectural Framework (TOGAF)

186. TOGAF is widely recognized and used within statistical organizations as well as many other organizations around the world. Most other architectural frameworks are basically consistent with TOGAF although the terms and precise concepts used within other architectural frameworks may vary.

187. Most information objects defined within GSIM (for example, *Data Sets* and *Classifications*) would be considered 'business objects' within TOGAF. Within TOGAF, such 'business objects' would typically be modeled as Data Entities within the Data Architecture.

188. Three information objects within the Production Group, however, are included in the metamodel for Business Architecture within TOGAF, Business (Process), Business *Function* and *Business Service*. Within TOGAF, *Business Functions* and *Business Services* interact with (for example, produce and consume) 'business objects'.

189. *Process Method* is not directly referred to by TOGAF. Statistical organizations place particular emphasis on design and selection (and evaluation) of statistical methods (in the context of statistical methodology more generally) when producing official statistics. For many other industries, the method to be selected and used to perform a particular *Business Function* might not need to be separately identified (for example, it will not be subject to specific evaluation or reuse). In these cases, the concept of "method" could be subsumed in the definition of the Business Process in the TOGAF metamodel.

190. GSIM does not model in as much detail as TOGAF the way that *Organizational Units* interact with *Business Functions* and *Business Services*. A lot of the detail about how *Organizational Units* interact will be specific to a particular organization. Nevertheless, *Process Steps* and *Business Services* need to have owners designated in GSIM.

191. The TOGAF metamodel sets out a very flexible (rather than strictly hierarchical) relationship between *Business Functions*, business processes and *Business Services*. For example, the business process used to fulfill a particular *Business Function* (for example, GSBPM 6.2 Validate Outputs) might require another *Business Function* (for example, GSBPM 5.3 Review, validate and edit) to be performed. GSIM inherits this flexibility.

192. This allows an individual to apply GSIM to describe the relationship between statistical information and statistical business processes for those aspects of the statistical production processes that are of interest to that person. They don't need to model the workflows required to deliver services they consume, they merely need to document (via a single *Process Step*) the inputs and outputs associated with their use of the service.

? Unknown Attachment

Word version of this section of document only