Developing a Model-Driven DDI Specification

By Participants in 2012 Dagstuhl Seminar on DDI Moving Forward[1]

Abstract

The DDI Alliance has initiated new work to build a model-based specification from which technical bindings can be generated. This approach will carry many benefits in terms of communicating with other standards efforts and maintaining consistency. To launch this project, a group of experts met in Wadern, Germany, in October 2012, where different approaches to the modeling effort were discussed and design goals established. Also discussed were efforts to align DDI with the Generic Statistical Information Model (GSIM), a reference model describing the process of producing official statistics. There are clear synergies between the DDI and GSIM, and the Alliance has offered to produce the implementation layer for GSIM. A roadmap outlining steps in realizing the DDI model is under development and will lay out concrete plans to realize the vision.

[1] See Appendix B for a full list of authors participating in the seminar at Schloss Dagstuhl - Leibniz Center for Informatics in Wadern, Germany (Dagstuhl Event 12432), on October 21-21, 2012.

Background

The DDI (Data Documentation Initiative) began in the mid-1990s as a project to create a structured metadata standard for the social sciences. The concept of a structured standard found traction right away, especially among the data archives, which were dealing with heterogeneous unstructured documentation and needed a way to standardize content. It was clear that this approach had the potential to drive process and to enhance data discovery.

The DDI specification itself began as an XML archival codebook format (DDI Codebook, released in 2000), which had a somewhat document-centric focus, and then branched off to cover the research data life cycle (DDI Lifecycle, released in 2008), a more ambitious undertaking. Over time, the specifications have continued to add new coverage and functionality to respond to new user requirements. DDI is now growing beyond the social, behavioral, and economic sciences into the official statistics and medical research communities.

As the DDI specification has been evolving, the ecosystem around research data has been changing rapidly. Data are taking on increasing importance in research and policy-making, and as new data types are explored and combined, expectations for metadata rise as well. The DDI has an opportunity to respond to community expectations by creating a new version of the specification that can transcend traditional disciplinary barriers to document data about humans and their behavior more broadly. As an example, while data collection instruments in the social sciences have traditionally been surveys, we can also view blood pressure gauges and magnetic resonance imaging (MRI) scans as new types of instruments that capture and export data. There is also a growing emphasis on data from administrative registers and various Internet sources. This leads to the need for a new way of looking at metadata that does not require added complexity but rather a smart and economical approach to metadata scoping and modeling.

It has been clear for some time that to meet the diversity of disciplinary needs, the DDI specification must incorporate a level of abstraction, which is best addressed by having an information model -- a representation of concepts and associated relationships, constraints, rules, and operations[2]. During its June 2012 meeting in Washington, DC, the DDI Alliance, which is the membership organization stewarding the DDI specifications, agreed to move forward to develop a model-driven, next-generation DDI specification that could be expressed in a variety of technical formats. To facilitate this work, a week-long seminar[3] was held at Schloss Dagstuhl, Leibniz Center for Informatics, in Wadern, Germany. Organized by Arofan Gregory (Open Data Foundation - Tucson, US), Wendy Thomas (Population Center, University of Minnesota, US), Mary Vardigan (University of Michigan - ICPSR, US), and Joachim Wackerow (GESIS - Mannheim, DE), the seminar drew on expertise from within the DDI Alliance and outside of it, with 21 invited participants contributing to the work.

[2] http://en.wikipedia.org/wiki/Information_model

[3] http://www.dagstuhl.de/en/program/calendar/evhp/?semnr=12432

This paper reports on the Dagstuhl work, interweaving information from other sources in order to present a thorough description of the vision and process for the next version of the DDI specification.

History and Lessons Learned

As we look ahead to the development of a DDI information model, it is worth looking back to understand the drivers that brought us to where we are now.

Published in 2000, DDI Codebook, with its clear delineation of study description, files description, and variables description, was easily understood by non-technical people: librarians and archivists could implement it without a steep learning curve. The documentation for DDI Codebook was provided originally as a Tag Library with examples, which was quite straightforward. The Nesstar Publisher and Server tools were and continue to be incredibly helpful in propagating DDI Codebook in the community. The World Bank created a Microdata Toolkit based on the Nesstar Publisher to foster data access in developing countries. As a result, DDI is being used in over 70 countries of the world.

The move from DDI Codebook to Lifecycle was an important one for the DDI Alliance, extending the scope of the standard and making it relevant for new audiences. DDI Lifecycle added support for data collection, thus facilitating metadata-driven processes. This was a major paradigm shift for the community as DDI Lifecycle, which moved from an XML DTD to XML Schema, was designed to be used by machines. An active set of developers became involved with DDI Lifecycle, building new tools. The Danish Data Archive was one of the first to support DDI Lifecycle through creation of a DDI editor. The Colectica tool based on DDI provided an interface with Computer Assisted Interviewing, consuming information exported by CAI systems. StatTransfer developed an export to the DDI Lifecycle format, and several implementations of DDI in relational databases were also produced.

Meanwhile, the DDI began to experience growing pains, both as an organization and as a standard. After several years of functioning as a self-sustaining membership organization (the Alliance was established in 2003), the DDI group needed to review its progress and chart its future in light of new needs and requirements. An external review was important because new communities – the National Statistical Institutes, in particular – were expressing interest in joining the Alliance and using the specification to document their data and process. This development brought new expectations and questions about how the organization worked.

The review of the DDI project recommended that to continue its path to a more professional and formal organization, the Alliance needed to revisit its organizational structure and its foundational documents. This work was done in 2012 with the result that a new structure reflecting a more stable, forward-looking organization that can accommodate new groups and more quickly adapt to changes in the environment is about to be put in place.

Another result of the growing pains was that the standard itself began to take on increasing complexity with the weight of new requirements, and it was difficult to document such a large standard. Further, the XML format of the standard itself proved limiting when attempting to communicate across disciplinary boundaries and align with other standards. It was clear that the lack of an information model was making it difficult interact with other groups, other standards, systems, and tools.

The lessons learned across the years -- both organizational and technical -- also underscored the need for more community involvement. It became evident that the DDI needed to transition to a community-driven model in which development and maintenance could be spread across the community to make the organization more sustainable. Too few people had been engaged in creation and documentation of the standards and this would need to change for the organization to accomplish new goals. We would need to be able to develop faster, with more domain engagement and responsiveness to our increasingly diverse user community, in a way that ramps up involvement of more human resources.

All of these lessons brought the DDI Alliance to a place where it makes sense to begin development of a model-driven standard with a new vision for the future.

Content Goals for Next-Generation DDI

Scope Statement

What is the proper scope for the next-generation DDI? We know from lessons learned that DDI can’t be all things to all people. The Dagstuhl group wrote this statement summarizing what is in the purview of DDI and what the DDI aspires to become:

The Data Documentation Initiative (DDI) is an international standard for describing data related to the observation and measurement of human activity. With origins in the quantitative social sciences, DDI is increasingly being used by researchers in other disciplines. The DDI specification is also being used to document other data types, such as social media, biomarkers, administrative data, and transaction data. DDI is a model-based metadata specification that can be implemented in a variety of technologies. The specification itself is modular and can document and manage different stages of data lifecycles, such as conceptualization, collection, processing, analysis, distribution, discovery, repurposing, and archiving.

This statement is useful in strategic, big-picture planning for the future.

Generic Statistical Information Model (GSIM)

In thinking about the scope of the new specification, we also need to look at other factors and drivers in the ecosystem. One such driver is the Generic Statistical Information Model, or GSIM[4]. GSIM was developed in 2012 by the High-Level Group for the Modernisation of Statistical Products and Services (HLG) and the National Statistical Offices around the world to complement the Generic Statistical Business Process Model[5].

[4] http://www1.unece.org/stat/platform/pages/viewpage.action?pageId=59703371

[5] http://www1.unece.org/stat/platform/display/metis/The+Generic+Statistical+Business+Process+Model

GSIM is a reference framework of internationally agreed definitions, attributes and relationships that describe the pieces of information that are used in the production of official statistics (information objects). Examples of information objects in scope for GSIM include data (held in datasets) and metadata (e.g., classifications, variables, questions, populations).

Where GSBPM provides common terminology related to statistical business processes, GSIM provides both common terminology and a common reference model for the statistical information which is input to, or output from, statistical business processes. GSIM spans all the "business objects" needed, and used, by producers of official statistics to undertake their core business (i.e., to produce official statistics).

A core aspiration of HLG when they commissioned development of GSIM was that the "business objects" could be implemented in practice using existing standards, especially DDI and Statistical Data and Metadata eXchange (SDMX)[6]. Relating (model-based) objects defined in GSIM to representations in DDI would help producers of official statistics use DDI in a consistent, business-driven manner (rather than driven by technical specialists).

GSIM as a conceptual model already draws heavily in places on sound modeling undertaken previously by the DDI Alliance when developing DDI-L. Further defining and improving the "fit" between GSIM and DDI maximizes the probability that:

- The Common Statistical Production Architecture (CSPA) now being defined by HLG will promote actively applying DDI when representing and exchanging statistical data and metadata between components, and
- Many of the shared and interoperable processes, methods, repositories, and IT components to be designed based on the CSPA will prove useable by, and of value to, other implementers of DDI besides agencies producing official statistics.

DDI Lifecycle has influenced the design of GSIM, and thus aligning in a closer way makes sense. It may be possible for DDI to function as an actionable implementation structure for GSIM. At the time of this writing, the Alliance was in discussions with GSIM to collaborate on the lower-level elements to support the GSIM model. A diagram of the GSIM model is provided in Appendix A.

User Stories

A focus on the needs of actors in the research enterprise was viewed by the Dagstuhl group as essential to an effective model that will be used by a wide audience. To that end, the group developed several user stories, starting with the activity (e.g., processing for data harmonization) and adding the elements needed as well as the who, why, and how of the activity. These user stories will be used to develop the substantive content of the model.

[6] http://sdmx.org/

Process and Workflow

The life cycle model promulgated by DDI has been lacking a process layer, which is essential for the specification going forward. For example, an archive may want to document how a variable was transformed for reasons of confidentiality as part of an “audit trail” for the data. Describing process and workflow in general, with inputs and outputs, is a goal. Ideally, there would be support for workflow automation and replication as well. The group discussed the fact that workflow does not necessarily map in a one-to-one way with the data lifecycle stages. It was also noted that process documentation may be pulled in from another standard so that DDI does not need to build it. A process layer should be available at all levels of the model.

Other New Content

There are other substantive areas that the DDI specification needs to cover in response to community needs that have been expressed:

- New content on sampling, survey implementation, weighting, and paradata
- New content pertaining to qualitative data/mixed methods data
- Framework for data and metadata quality
- Framework for access to data and metadata
- Integration with existing standards like SDMX, CDISC, Triple-S
- Disclosure review and remediation
- Data management planning

Design Goals for Next-Generation DDI

In designing any model, there are tradeoffs between the ideal and the practical implementation. A core requirement for the DDI, however, is that the new version of the specification have a UML data model as its foundational version. Discussions at the Dagstuhl seminar suggest that the canonical model should be expressed in English, with a UML model expressing as much of that canonical model as possible. This model can then be rendered in XML Schema, RDF/OWL Ontology, relational database schema, and other languages, ideally via some degree of automation. The Alliance believes that such an abstract data model will make it easier to interact with other disciplines and other standards, to understand the specification, to develop and maintain it in a consistent and structured way, and to enable software development that is less dependent on specific DDI versions.

The following high-level design goals, written by the Dagstuhl group, were developed to guide development and maintenance of the DDI Information Model [“model”]:

1. Interoperability and Standards – The model is optimized to facilitate interoperability with other relevant standards.
2. Simplicity – The model is as simple as possible and easily understandable by different stakeholders.

3. User Driven – User perspectives inform the model to ensure that it meets the needs of the international DDI user community.
4. Terminology – The model uses clear terminology and when possible, uses existing terms and definitions.
5. Iterative Development – The model is developed iteratively, bringing in a range of views from the user community.
6. Documentation – The model includes and is supplemented by robust and accessible documentation.
7. Lifecycle Orientation – The model supports the full research data lifecycle and the statistical production process, facilitating replication and the scientific method.
8. Reuse and Exchange – The model supports the reuse, exchange, and sharing of data and metadata within and among institutions.
9. Modularity – The model is modular and these modules can be used independently.
10. Stability – The model is stable and new versions are developed in a controlled manner.
11. Extensibility – The model has a common core and is extensible.
12. Tool Independence – The model is not dependent on any specific IT setting or tool.
13. Innovation – The model supports both current and new ways of documenting, producing, and using data and leverages modern technologies.
14. Actionable Metadata – The model provides actionable metadata that can be used to drive production and data collection processes.

The Model and Underlying Principles

The Dagstuhl group drew on its collective technical expertise to outline various aspects of the model and the design principles supporting it. We start here with some general principles and move to more technical detail.

Simple Terminology

As expressed in the design goals, the group agreed that simple and understandable terminology for all aspects of the model is critical to acceptance. It is important to package the model and its components in a way that makes sense to potential users.

Literate Programming[7] Paradigm

This programming approach can inform the model creation and maintenance going forward. This involves describing and defining the specification in human language included in source code with a strong relationship between reasoning and purpose and solution. Good documentation of the specification and transparency for different audiences is critical, and maintenance over time should not be dependent on individual people.

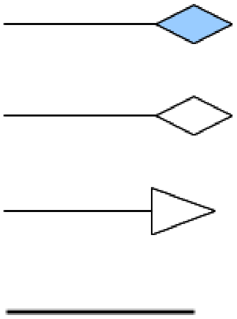
[7] http://en.wikipedia.org/wiki/Literate_programming

English as Normative Form

Related to the above principle, the model should be expressed in English as its normative form. As much as possible of the normative form should be expressed in a simple version of UML. The UML then becomes the foundation for expression in a variety of other forms – e.g. XML, RDF, and a relational database schema.

Modeling DDI in Simple UML

The UML model will be generated with as simple a set of notations as possible, for example:



(compose) – the existence of the parts is dependent on the aggregate

(aggregate) – parts belong to the aggregation but exist independently

(generalize) - i.e., inheritance, the specific inherits from the general

(reference)

Interoperability

An important principle for the model-driven DDI specification is that it should not duplicate what other standards do but instead align with them. We face an increasingly complex world in terms of other standards and will need to ensure compatibility with GSIM, ISO 11179, Dublin Core, geographic standards, library standards, and possibly other domain metadata standards like CDISC for clinical trials.

Coordinated Releases with Feedback Loop

The group recommended that there be no “Big Bang” releases of the DDI specification anymore but rather coordinated releases of smaller pieces of the model. Further, the group endorsed a feedback process from implementation back into development so that implementers can contribute to and enhance the specification during development.

Core and Base Plus Modules

In terms of design, it was agreed that the DDI model should have a substantive Core – possibly based on a subset of DDI Codebook -- along with a set of modules that extend the Core and are needed for specific tasks, such as documenting and managing a longitudinal survey. In general, these modules would correspond to families of user stories, providing descriptors to cover functional areas. The Core of the model should contain a carefully selected set of objects used by many other parts of the model (“When in doubt leave it out” was the principle expressed).

To support the model a technical Base will be required in addition to the substantive Core. Figure 1 shows an example of what the contents of a Base and Core module might be. Note that all of the objects in the DDI Base are abstract. The objects in the domain Core are all objects which are used in many places.

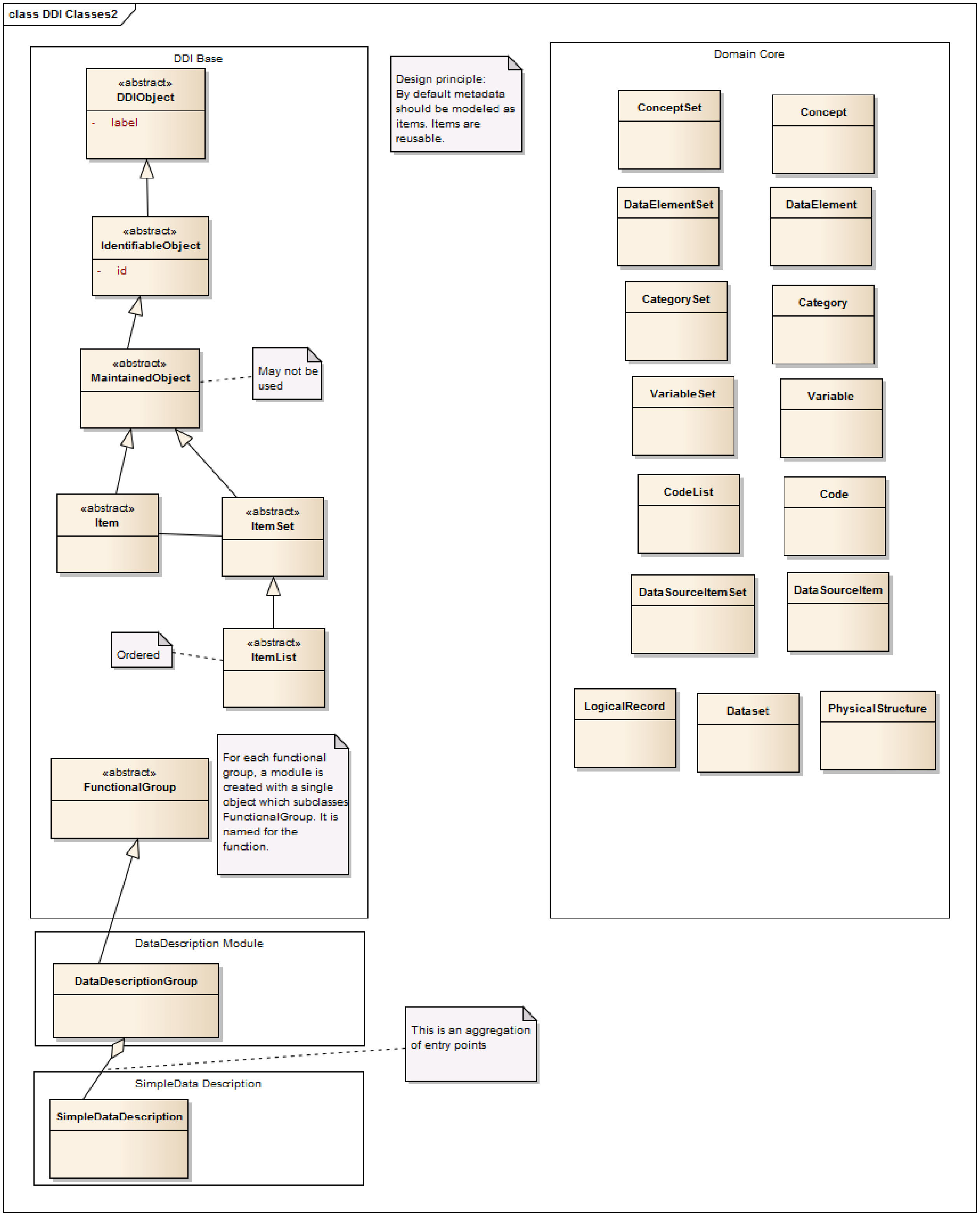


Figure 1: DDI core and base structure

(Figure 1 uses current DDI terminology, but this will ultimately be replaced with clear and concise labeling understandable by a wider audience.)

Modularity

The Dagstuhl group envisioned a series of modules radiating out from the Core, with advanced modules inheriting from simpler modules and the simplest module inheriting from the Core as in the figure below. The design principle here is to enable bindings to the simplest possible representations where, for example, the XML schema for a simple survey does not need all of the hundreds of elements that DDI 3.1 requires. The group believed that this method should make DDI much more accessible for developers. Simple document types should allow for consistent use of XPath expressions, a big benefit. An additional agreed upon principle for the overall DDI model was that objects would represent something real and would be defined in only one place in the model. Where objects need to appear in multiple places they would appear by reference to their one definition. Where definitions of objects need to be repeated outside of their one definition location, the metadata should be included inline but specifically marked as non-canonical. An example might be an instance intended for long-term preservation, needing to be interpretable independent of any external infrastructure. In this case definitions of objects which would normally be used by reference to an external source might need to be included for completeness. These definitions would be marked as copies of the authoritative definition.

Figure 2 shows a possible arrangement of part of the DDI model, separating elements used to describe a survey from those used to describe a dataset. In each group a set of objects could be used for the simple case, for example, metadata describing a simple dataset for a statistical package, or a basic survey. Another set might contain extensions for more advanced cases such as a matrix of questions in a survey. The outer circle shows the objects that might be expressed for preservation purposes (everything).

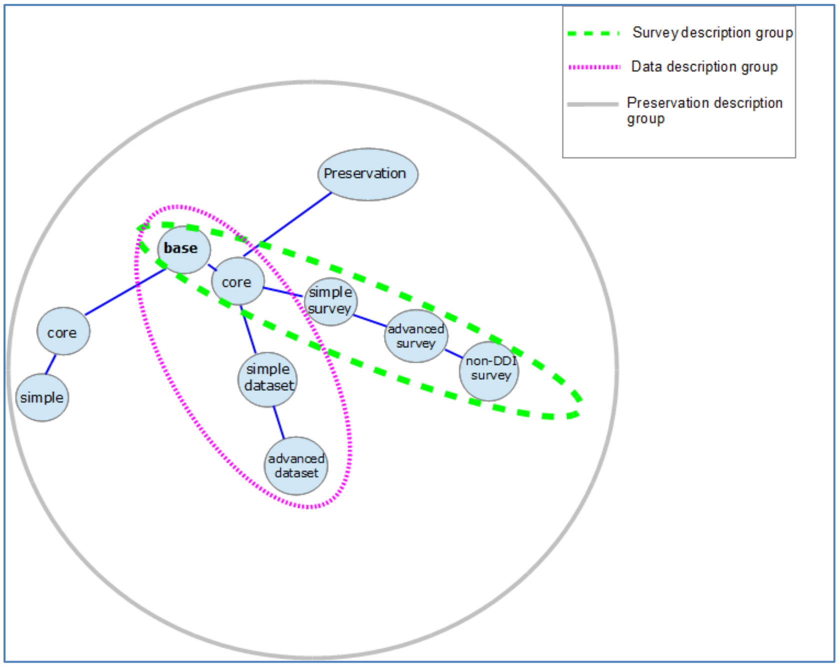


Figure 2 - The modularized DDI model

Bindings

The group defined these principles related to bindings:

- When represented in other forms, such as XML, the underlying DDI model should be bound into a collection of document types, each for a specific purpose, along with one document type encompassing the entire model, the latter to be used to handle unusual cases and perhaps for preservation purposes.
- A top level element of <Document> should be defined with a specific set of substitutables.
- There should also be a non-identified generic element <Collection> with specific substitutables. These will serve as containers for referenced items.
- Recursion should be avoided in the XML document types. Parent-child relationships (e.g., sub-categories) should be handled by references from parent to children.
- Bindings into document types should be able to be generated at least somewhat automatically.
- Bound artefacts should be designed to support "graceful degradation."[8] This reduces the need to build corner cases.
- Metadata should be modeled as items, and items should be reusable.

Binding to XML

Figure 3 shows the expression of the survey branch (green dashed ellipse in Figure 2 above) into XML.

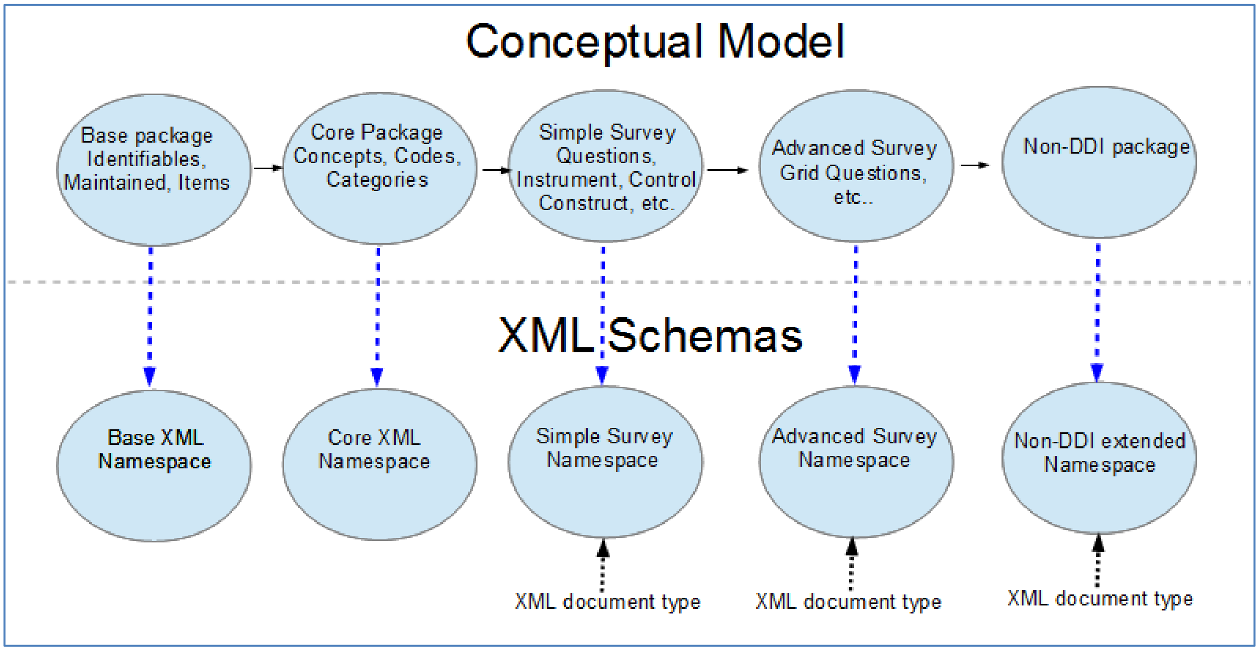

Conceptual Model
Base package Identifiables, Maintained, Items
Core Package Concepts, Codes, Categories
Simple Survey Questions, Instrument, Control Construct, etc.
Advanced Survey Grid Questions, etc..
Non-DDI package
XML Schemas
Base XML Namespace
Core XML Namespace
Simple Survey Namespace
Advanced Survey Namespace
Non-DDI extended Namespace
XML document type
XML document type
XML document type

Figure 3: The Survey branch of the DDI model expressed as XML document types

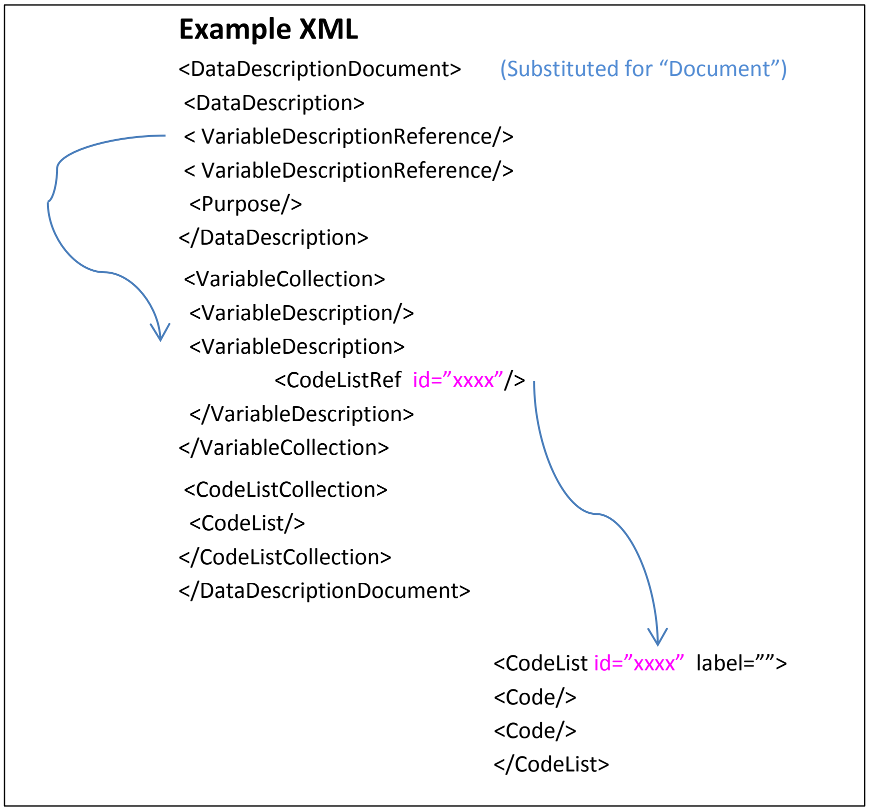
[8] For a discussion of graceful degradation see: http://en.wikipedia.org/wiki/Fault-tolerant_system

The group wrote the following description of how a binding to XML might be generated from the model.

1. *There is a FunctionalGroup object in the model, named DataDescriptionGroup, in its own module. It points to the DataDescription object in the Simple Data Description module. The goal is to support the function of describing a file of observational data in a basic fashion (enough metadata to produce an SPSS file, for example). The DataDescription object is the entry point which is learned from the processing of the DataDescriptionGroup object. The Core namespace is implicitly included.*
2. *Select the DataDescription object as the entry point*
3. *Generate a root element (doctype) based on the name of the function ("______Document") which is a sub-class of the abstract core:Document object. (This extension is reflected in the XML schema.)*
4. *Because DataDescription is identifiable (could be an item), we declare it as an element in the schema <DataDescription>. Each of its properties is expressed as an XML attribute, a contained element, or a Reference (if a non-composition relation to an identifiable object).*
5. *For each referenced item type, we create an artificial element "______Collection" to hold the metadata. Depending on how the dependencies cross modules, there could be an approach as follows: reference objects for modules outside of those being used with external references, indicated using fixed attributes (isExternal="true"). This is to be avoided if possible.*

Some of the issues involved in the above scenario include:

1) How do you tell from the model that related modules are related? A single item for an entry point needs to exist in the model.
2) The external references in 4 above represent problems for instances for single file data exchange or preservation



Example XML

```
<DataDescriptionDocument>      (Substituted for "Document")
 <DataDescription>
 < VariableDescriptionReference/>
 < VariableDescriptionReference/>
  <Purpose/>
</DataDescription>
 <VariableCollection>
  <VariableDescription/>
  <VariableDescription>
          <CodeListRef  id="xxxx"/>
  </VariableDescription>
</VariableCollection>
 <CodeListCollection>
  <CodeList/>
</CodeListCollection>
</DataDescriptionDocument>

                          <CodeList id="xxxx"  label="">
                          <Code/>
                          <Code/>
                          </CodeList>
```

Binding to RDF

At one level the process of binding to RDF may be fairly straightforward: For each object in the UML model, declare an OWL class and assign it a URL. Where there is a doctype in the XML, we would use named graphs in the RDF. We might also want to use named graphs for item graphs (an item and all referenced items). Graphs could have the same URI as their entry-point objects ("Basically, a Named Graph is a set of triples named by an URI. This URI can then be used outside or within the graph to refer to it."[9])

Data as a Service, Service Oriented Architecture (SOA), Web Services

The Dagstuhl group expressed a preference for the "data as a service[10]" approach, which is based on the concept that a product, data in this case, can be provided on demand to the user regardless of geographic or organizational separation of provider and consumer. Additionally, the emergence of service-oriented architecture (SOA)[11] has rendered the actual platform on which the data resides also irrelevant. In this "data as a service" scenario, if the user has the relevant infrastructure, the entire specification can be implemented, but it isn't necessary to have the full infrastructure.

With this data as a service principle in mind, the Alliance can build service-oriented architecture and Web services to implement the new specification – a useful approach for services that are public facing. One should be able to request bits and pieces of the specification – for example, a scheme that just lifts out concepts. We will need a simple set of document types to be seen by applications to obviate the need to carry the full payload.

In considering the typical functions involved in Data as a Service (*Create, Read, Update, and Delete*) it is likely that the important service for our use case is *Read,* which also may require a *Search* function. We are also most likely to be dealing with the ability to retrieve whole sets of metadata, not single elements at a time. There might be some concern about having an ability to limit the size of a set of metadata returned, but that capability need not be in the model. One suggestion was to think in terms of the concept of "banks" as in concept banks or question banks and to group services this way. Another suggestion was to treat metadata items uniformly to enable a common set of web service methods for all item types.

The group discussed whether there would be a need to retrieve things based on dependencies or to just retrieve lists of things. It was agreed that anything identifiable should be retrievable.

[9] See: http://www.w3.org/2004/03/trix/

[10] http://en.wikipedia.org/wiki/Data_as_a_service

[11] http://en.wikipedia.org/wiki/Service-oriented_architecture

The group listed a set of calls sufficient to support reading from a repository:

GetByIdentifier – Retrieve an object based on its identifier.
GetListOfVersions – Retrieve a list of all of the versions of an object.
GetListOfItemsByListOfIdentifiers – Given a list of identifiers, return a list of items.
GetListOfIdentifiersThatItemDependsOn – Given the identifier of a single item, return a list of all the items it depends on (recursively), i.e., get the item graph.
Search (a faceted search by itemtype, field, language, organization, etc.) -- Some items need to be searchable depending on content as for instance, text in a label, or description.

These functions might also need to be added:

GetListOfItemForType – List all of the items of a given type.
GetForFunctionalGroup – E.g., which "simple surveys" are in the repository. One issue with this is that a return type might not be defined. This could be fleshed out once the functional groups are defined.

There may also need to be services for introspection[12] – i.e., to allow querying a service regarding which modules and metadata are supported.

The possible need to support the SOA concept of a registry, where queries return pointers to objects (ID, label, and location), was also discussed.

Because all elements in the suggested model are declared as global elements, they can function as return types for Web services. This is because they can be valid XML instances. Any DDI element can potentially be a stand-alone element. Best practice might be to define a specific service's namespace that contains a wrapper element that has a content of "xs:Any" without the list of valid namespaces defined for its content. This would insulate service clients from changes in the XML schema.

The list of verbs defined above could be implemented as a set of simple http-based protocols.

Unresolved Issues

One issue on which the group did not reach consensus was that of how to represent relationships. While there may be a need to model relationships as abstract objects, the potential drawback is that this would add unneeded complexity to the model and in particular to the XML expressions of the model. Whether there are indeed cases for which having an abstract relationship object is useful should be explored more thoroughly as this project proceeds.

[12] http://en.wikipedia.org/wiki/Type_introspection

The Modeling Process Going Forward

The Dagstuhl group identified a methodology for moving forward to create the model. It will first be necessary to identify the substantive domain Core and the technical Base, along with extensions in the form of modules; each module may have a simple and advanced version.

To facilitate the substantive work, we identified three important user stories to describe: the archival use case, the longitudinal study use case, and the National Statistical Institute (NSI) use case of producing official statistics, which ties into the GSIM model. These use cases should surface key requirements for the DDI model.

Once the core and the modular structure are pinned down, the work can begin on the actual modeling. The general process used to produce the GSIM model was put forward as a good approach, and the Alliance is adopting a similar approach in creating a roadmap to define the stages of development for the DDI model.

The modeling will address how to structure and model the core, abstract objects, and relationships. It was acknowledged that there will likely be some back and forth and a tension between technology and expression. This is logical because while we want to be pragmatic – and this must be a model that produces implementable products – we also want to be creative and do this in a logical and efficient way.

Community-Driven Development

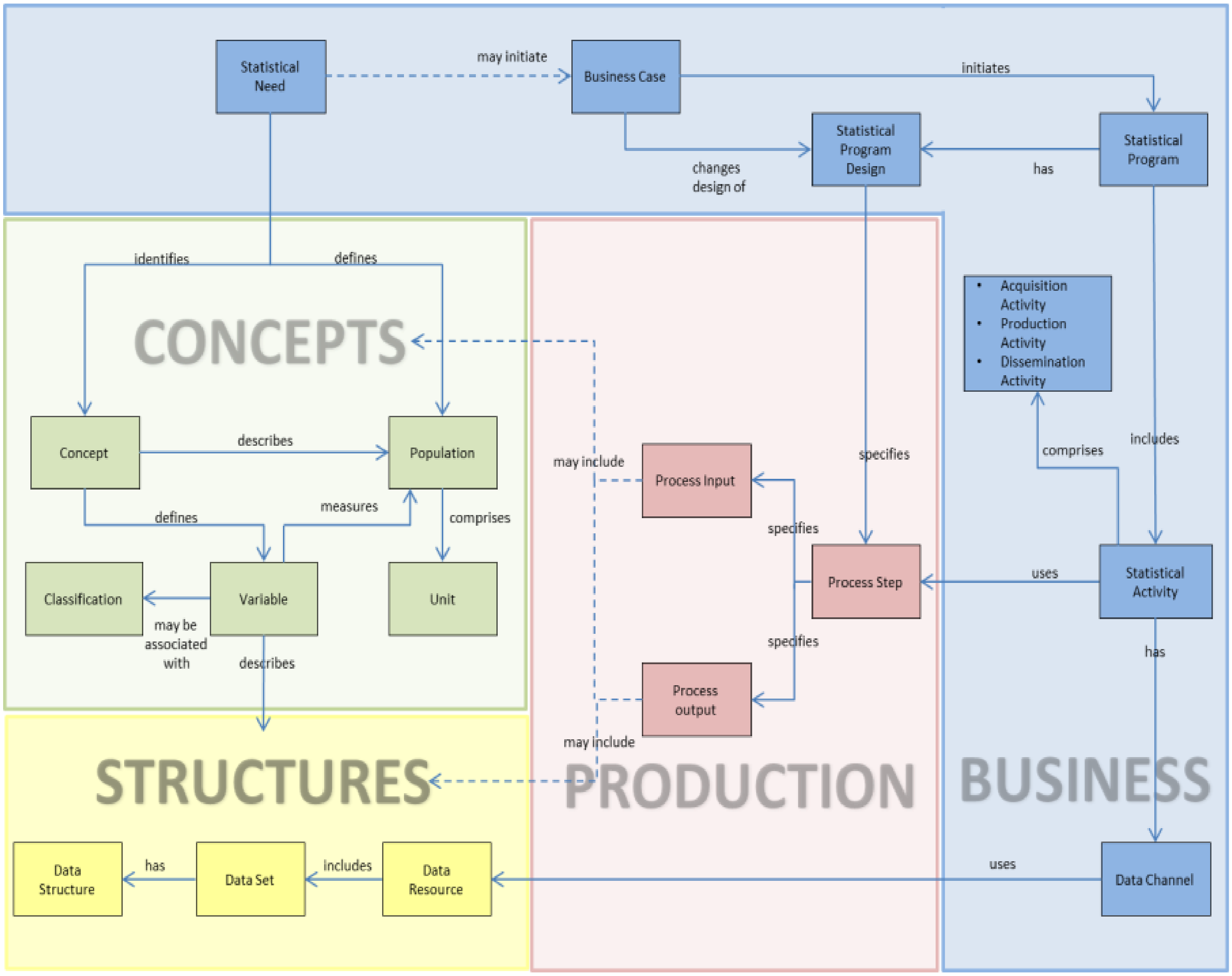
As mentioned, leveraging the community in moving forward to create a model-based specification is a key goal for the DDI Alliance, and the Dagstuhl group laid out a strategy and plan to reach this goal. The model should allow for a distributed development effort, with different groups able to work on parts of the model somewhat independently.

Domain experts are key players in the model development. They will be assisted and supported by technical experts in defining scope, structure, and documentation for different parts of the model. With this approach domain experts do not need to become professional modelers. The technical team is, of course, critical to successful outcomes in that they will ultimately produce the model and its bindings. The intention is to iteratively interact with developers along the way and then provide a technology preview, which will permit us to actively engage with other standards and groups and bring them into the community and the conversation.

Conclusion

Building a model is a complex process but one that is necessary for the DDI to continue to thrive. The DDI Alliance has made good progress in laying the foundation for a new model-driven specification, bringing diverse perspectives to the table so that the advantages and disadvantages of various approaches to modeling and to binding can be considered fully. DDI is a project of and for the community, so widening the effort to include new audiences and contributors is a natural outgrowth of the current structure for contributions. Work will continue as the roadmap develops, with additional face-to-face meetings envisioned, including another workshop at Schloss Dagstuhl in the fall of 2013.

Appendix A -- GSIM Model



A reference framework of information objects that enables generic descriptions of the management and use of data and metadata, GSIM sets out definitions, attributes, and relationships and provides common semantics to facilitate communication across organizations.

Appendix B

Acknowledgments

This paper is one of several papers that are the outcome of a workshop held at Schloss Dagstuhl - Leibniz Center for Informatics in Wadern, Germany (Dagstuhl Event 12432), on October 21-21, 2012.

Workshop Title

DDI Lifecycle: Moving Forward
Link: http://www.dagstuhl.de/en/program/calendar/evhp/?semnr=12432

Organizers

Arofan Gregory (Open Data Foundation - Tucson, US)
Wendy Thomas (Population Center, University of Minnesota, US)
Mary Vardigan (ICPSR, University of Michigan, US)
Joachim Wackerow (GESIS - Leibniz Institute for the Social Sciences, DE)

Participants

William (Bill) Block	Cornell Institute for Social and Economic Research (CISER), Cornell University
Thomas Bosch	GESIS - Leibniz Institute for the Social Sciences
Bryan Fitzpatrick	Australian Bureau of Statistics
Dan Gillman	U.S. Bureau of Labor Statistics
Jay Greenfield	Booz Allen Hamilton
Arofan Gregory	Open Data Foundation (ODaF)
Marcel Hebing	German Socio-Economic Panel Study (SOEP), DIW - Berlin - German Institute for Economic Research
Larry Hoyle	Institute for Policy & Social Research, University of Kansas
Chuck Humphrey	University of Alberta Libraries
Jon Johnson	Centre for Longitudinal Studies, Institute of Education, University of London
Jenny Linnerud	Statistics Norway
Brigitte Mathiak	GESIS - Leibniz Institute for the Social Sciences
Steven McEachern	Australian Data Archive
Olof Olsson	Swedish National Data Service (SND)
Barry Radler	University of Wisconsin
Ornulf Risnes	Norwegian Social Science Data Services (NSD)
Dan Smith	Colectica
Wendy Thomas	Minnesota Population Center (MPC)
Joachim Wackerow	GESIS - Leibniz Institute for the Social Sciences
Dennis Wegener	GESIS - Leibniz Institute for the Social Sciences
Wolfgang Zenk-Möltgen	GESIS - Leibniz Institute for the Social Sciences

Appendix C

You are free:

- to Share - to copy, distribute, display, and perform the work
- to Remix - to make derivative works

Under the following conditions:

- Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- Noncommercial. You may not use this work for commercial purposes.
- Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one. For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this Web page.
- Any of the above conditions can be waived if you get permission from the copyright holder.
- Apart from the remix rights granted under this license, nothing in this license impairs or restricts the author's moral rights.

Disclaimer

The Commons Deed is not a license. It is simply a handy reference for understanding the Legal Code (the full license) — it is a human-readable expression of some of its key terms. Think of it as the user-friendly interface to the Legal Code beneath. This Deed itself has no legal value, and its contents do not appear in the actual license.

Creative Commons is not a law firm and does not provide legal services. Distributing of, displaying of, or linking to this Commons Deed does not create an attorney-client relationship. Your fair use and other rights are in no way affected by the above.

Legal Code:
http://creativecommons.org/licenses/by-nc-sa/3.0/us/legalcode