A DESCRIPTION OF SPATIAL DATA INFRASTRUCTURES (SDIS) USING THE UNIFIED MODELLING LANGUAGE (UML)

Antony Cooper, Jan Hjelmager, Anders Nielsen and Petr Rapant
ICA Commission on Spatial Data Standards

CSIR ***i***comtek, PO Box 395, Pretoria, 0001, South Africa. acooper@csir.co.za
Kort & Matrikelstyreisen, Rentemestervej 8, Kobenhavn NV, DK-2400, Denmark. jnh@kms.min.dk
Kort & Matrikelstyreisen, Rentemestervej 8, Kobenhavn NV, DK-2400, Denmark. asn@kms.min.dk
Technical University of Ostrava, 17. Listopadu 15, 708 33 Ostrava-Poruba, Czech Republic. petr.rapant@vsb.cz

1. BACKGROUND

The International Cartographic Association (ICA) has a number of Commissions, including the Commission on Spatial Data Standards, and this presentation describes some of the work undertaken recently by the Commission.

A spatial data structure (SDI) is a mechanism for providing ready access to spatial data to as many users as possible, at a global, regional national or local level. The focus tends to be on providing a clearinghouse for framework or base data sets, with the provision of metadata and adherence to standards being crucial. The SDI provides services to allow the user to discover, evaluate and apply spatial data. An SDI cannot exist without strong partnerships to provide the data, technology and services, with these partnerships reducing duplication and the costs of collection, and leveraging the skills of the partners.

2. WHY UML?

The Unified Modelling Language™ (UML) is an object-oriented tool used widely for describing software systems, providing a graphical notation of the architecture of the system. Its use is not limited to software systems, and we felt that it might be useful to use UML to model (or describe) SDIs. For example, UML is being used within the International Organization for Standardization’s Technical Committee developing the international standards for geographical information and geomatics, namely ISO/TC 211, where it is used to encapsulate the essence of the standards, allowing their models to be harmonized.

Over the last few years, many papers and articles have been written, conference presentations made, round tables held, and projects executed dealing with SDIs at the international, regional, national and local levels. There are also a number of international organisations that deal with SDIs, such as the Global Spatial Data Infrastructure (GSDI). Thousands of pages have been written about SDIs, but what is lacking is a systematic approach to describing SDIs, and their contexts, users, providers, services and so on, necessary to establish them.

Hence, the ICA Commission on Spatial Data Standards proposes using UML for describing systematically all aspects of SDIs. It is clear that UML alone cannot serve this necessity: UML is only a tool, but it provides a starting point for a systematic approach to analyse SDIs. Modern approaches to analysing systems are based on object-oriented techniques, with UML often used as the tool to write down results of such analysis.

The first step in the analysis is focused on delineating the border between an SDI and its neighbourhood and we have used a conceptual diagram of the context for this purpose, as shown in Figure 1.

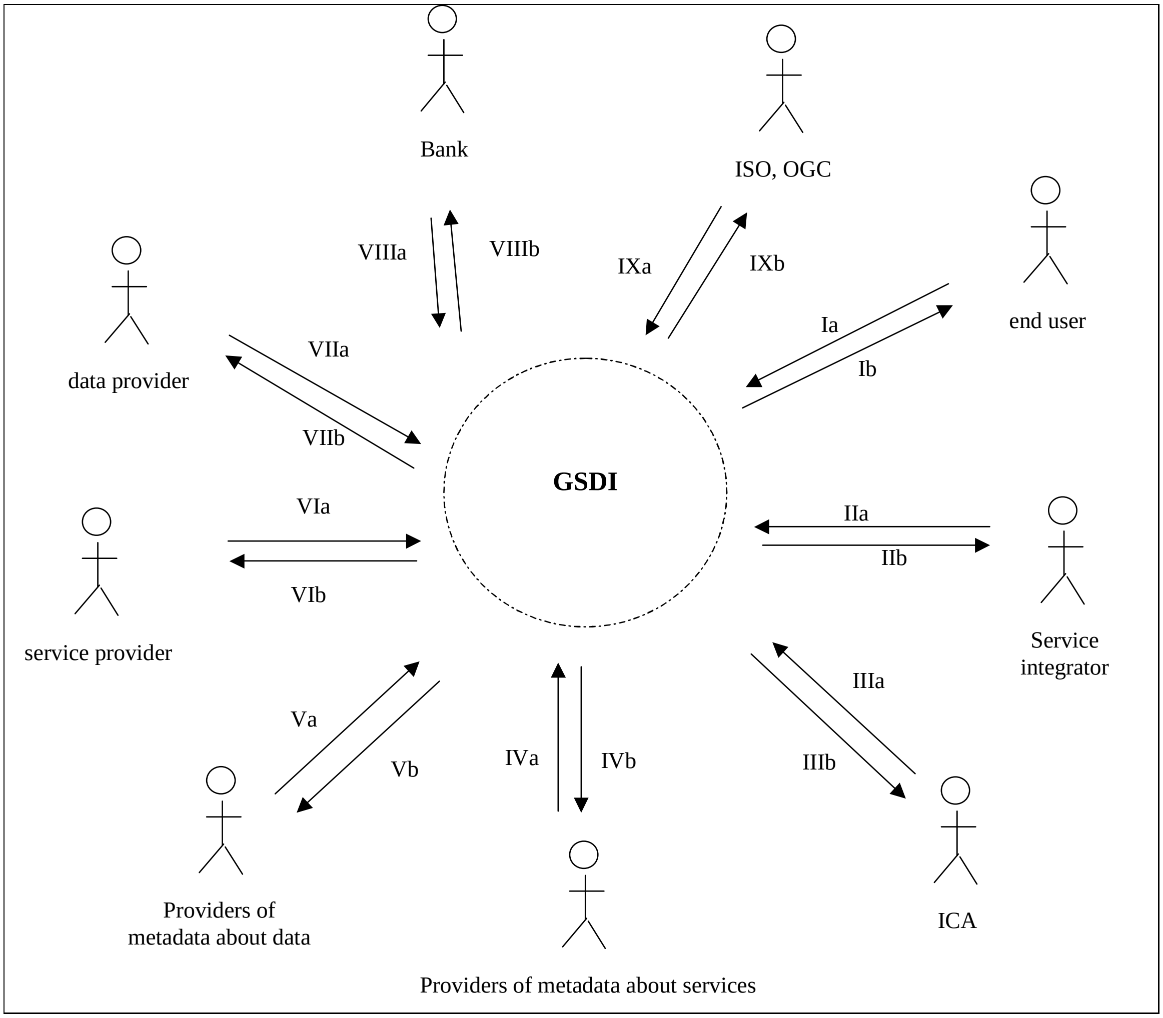

Figure 1: The context of an SDI

The central circle represents the SDI. We treat it here as a "black box", meaning we do not consider its internal structure at this stage. On the other hand, we focus here on objects (actors) that interact with the SDI and brief descriptions of those interactions. In this case, the brief descriptions are listed in Table 1 below. The Roman numerals provide the cross references between the figure and the table. An index "a" is added to the references for those interactions initiated by actors and an index "b" to those initiated by the SDI. Table 1 contains a quite exhaustive list of interactions. By analysing this list, we can develop a more generalised list of interactions, which can be realised between an SDI and its surrounding actors:

❑ Ask for something;
❑ Pay for something;
❑ Provide something;
❑ Invoice for something;
❑ Offer something;
❑ Service somebody; and
❑ Register something.

All interactions should be described in text form as scenarios of these interactions.

Table 1: Interactions between an SDI and actors.

No.	Name	Interaction
Ia	End user → SDI	- ask for metadata - ask for data - ask for services - ask for integrated services - ask for standards - pay for metadata - pay for data - pay for services - pay for integrated services - pay for standards
Ib	SDI → End user	- provide with metadata - provide with data - provide with services - provide with integrated services - provide with standards - invoicing for metadata use - invoicing for data use - invoicing for services use - invoicing for integrated services - invoicing for standards
IIa	Service Integrators → SDI	- offer integrated services - services end user (and the others) requirements or integrated services requirements - invoice end users (and the others) - ask for metadata - ask for data - ask for services - ask for integrated services - ask for standards - pay for metadata - pay for data - pay for services - pay for integrated services - pay for standards
IIb	SDI → Services Integrators	- ask for integrated services - provide with metadata - provide with data - provide with services - provide with integrated services - provision of standards - invoicing for metadata use - invoicing for data use - invoicing for services use - invoicing for integrated services - invoicing for standards
IIIa	ICA → SDI	- provide with cartographic standards - (invoicing for standards development and use)
IIIb	SDI → ICA	- ask for cartographic standards - (pay for standards development and use)
IVa	Metadata about services (MAS) providers → SDI	- services requirements for MAS - ask for standards
IVb	SDI → Metadata about services providers	- ask for metadata about services - provision of standards
Va	Metadata about data (MAD) providers → SDI	- services requirements for MAD - ask for standards
Vb	SDI → Metadata about data providers	- ask for metadata about data - provision of standards

Via	Service provider → SDI	- registering service in SDI - provide with service - ask for standards - invoice for service use
Vib	SDI → Service provider	- ask for service - provide with standards - pay for service use
VIIa	Data provider → SDI	- registering data (data set) in SDI - provide with data - ask for standards - invoice for data use
VIIb	SDI → Data provider	- ask for data - provide with standards - pay for data use
VIIIa	Bank → SDI	- financial services provision
VIIIb	SDI → Bank	- financial services use
Ixa	ISO, OGC → SDI	- provide with interoperability standards - (invoicing for standards development and use)
Ixb	SDI → ISO, OGC	- ask for interoperability standards - (pay for standards development and use)

Based on this diagram and scenario analyses, we can now develop a diagram of use cases. Figure 2 displays an example of such a diagram. We have prepared only a brief example, containing only some of all the actors and the interactions. Actors initiating use cases are situated in the left part of the diagram; actors consuming outputs from use cases are situated on the right side. We can see that different actors can address the same use case. The consumer of the output from a particular use case can be the same actor, but it could be another actor, as well.

Development of an object diagram can be a next step of our analysis. Figure 3 displays an example of such diagram that has not been fully developed.

This is only a very simple example. In the case of a fully developed diagram, one can meet the problem of dealing with a large number of classes with a huge number of associations. UML gives us very suitable tool for dealing with such complex systems: the modelled system can be divided up into blocks called packages (or from the other point of view – classes can be grouped to functionally and logically consistent groups called packages). So at first, we need to solve the problem on a more general level. We can do all the analysis on this level, develop a logical model of the packages, and develop all the associations between packages, etc. The next step is development of all the packages, which we can do one-by–one, or in a (semi) parallel way, depending on the complexity of the analysed system.

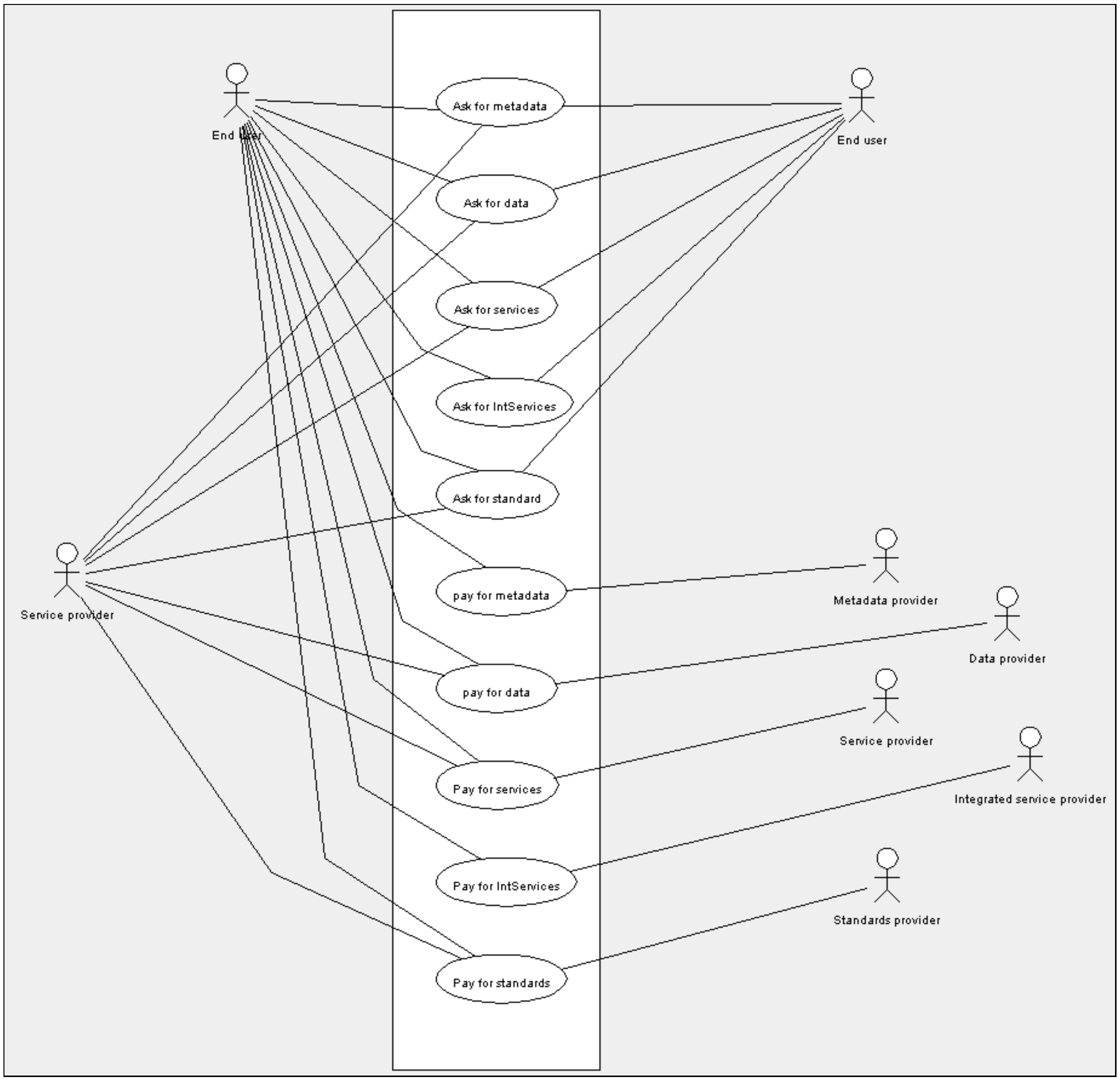

Figure 2: Brief example of the use case diagram

3. THE UML MODEL

We present here our first attempts at this modelling. We have attempted to use UML to describe the different types of SDIs and the different elements that make up an SDI, both physical and conceptual. The model will describe SDIs, not prescribe what should be in them – hence, the model could well be an aggregation of special cases, rather than a "clean" model of SDIs. It is also not a duplication of the efforts of others, but rather consolidates them into one model.

There are different ways to view or consider SDIs, for example:

- Organizational model, with three alternatives:
  - Committee: this is a community of interest which could be regulated or depend on the participants acting in the common interest, with independent participation (policed by the committee);
  - Common set of rules: here one has harmonized templates for operating which could be driven by regulations or by the mutual benefits, with autonomous participation (self policed); or
  - Organization: this would be a formally established body with employees and the like, providing a hierarchical coherence across the various levels (local, national, regional, etc).
- Logical model, viewed from two perspectives:
  - Business model: this is process driven, and should take precedence over the technology model; and
  - Technology model: this is input and output driven and depends on means of dissemination, and historically, has been considered to be more important than the business model.

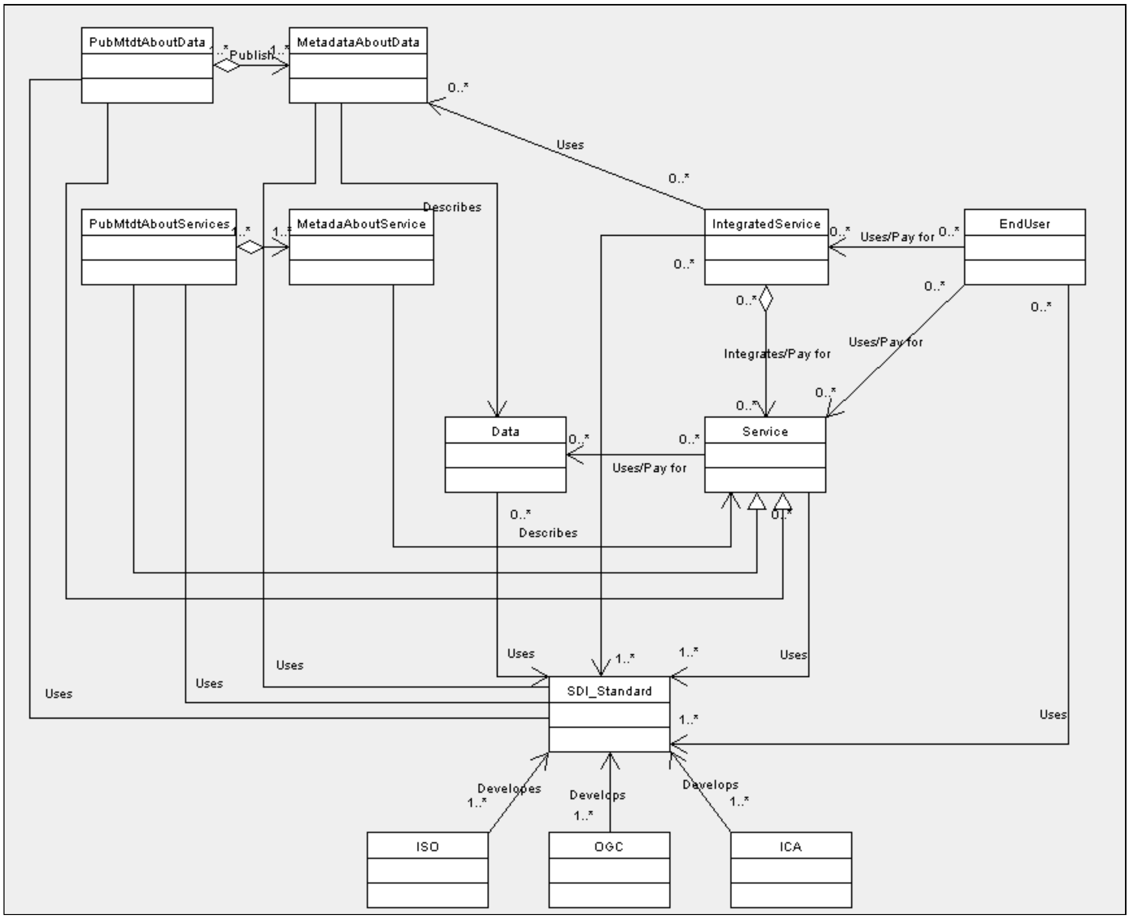

Figure 3: Example of an object diagram (not fully developed)

- Information model, with three components:
  - Physical enablers: these provide the “nuts and bolts” for the operation of the SDI, such as the standards for information exchange;
  - Logical enablers: these are the defined needs, rules, agreements, etc; and
  - Value: the outputs, especially the end-user benefits and new ways to address global issues.

Each perspective comprises a set of packages in the UML model, and together, they make up the “SDI concept”.

4. FURTHER RESEARCH

We have presented here a preliminary model of SDIs – we do not believe that it is the ultimate model of SDIs, but hope that it provides a useful starting point for developing a systematic model of SDIs. We would also suggest that the following issues should be considered when developing such models:

- There should possibly be a narrative for each grouping of classes (possibly grouped into one or more packages).
- Is technology (eg: networking, Web services) part of an SDI?
- Should we have a transactional model?
- How should one address policy, funding and territorial aspects?
- How should one address standards, protocols and processes?
- How should one address cultural and linguistic adaptability (CLA)?
- Such a model should probably be accompanied by a glossary of terms, especially for the classes and packages.

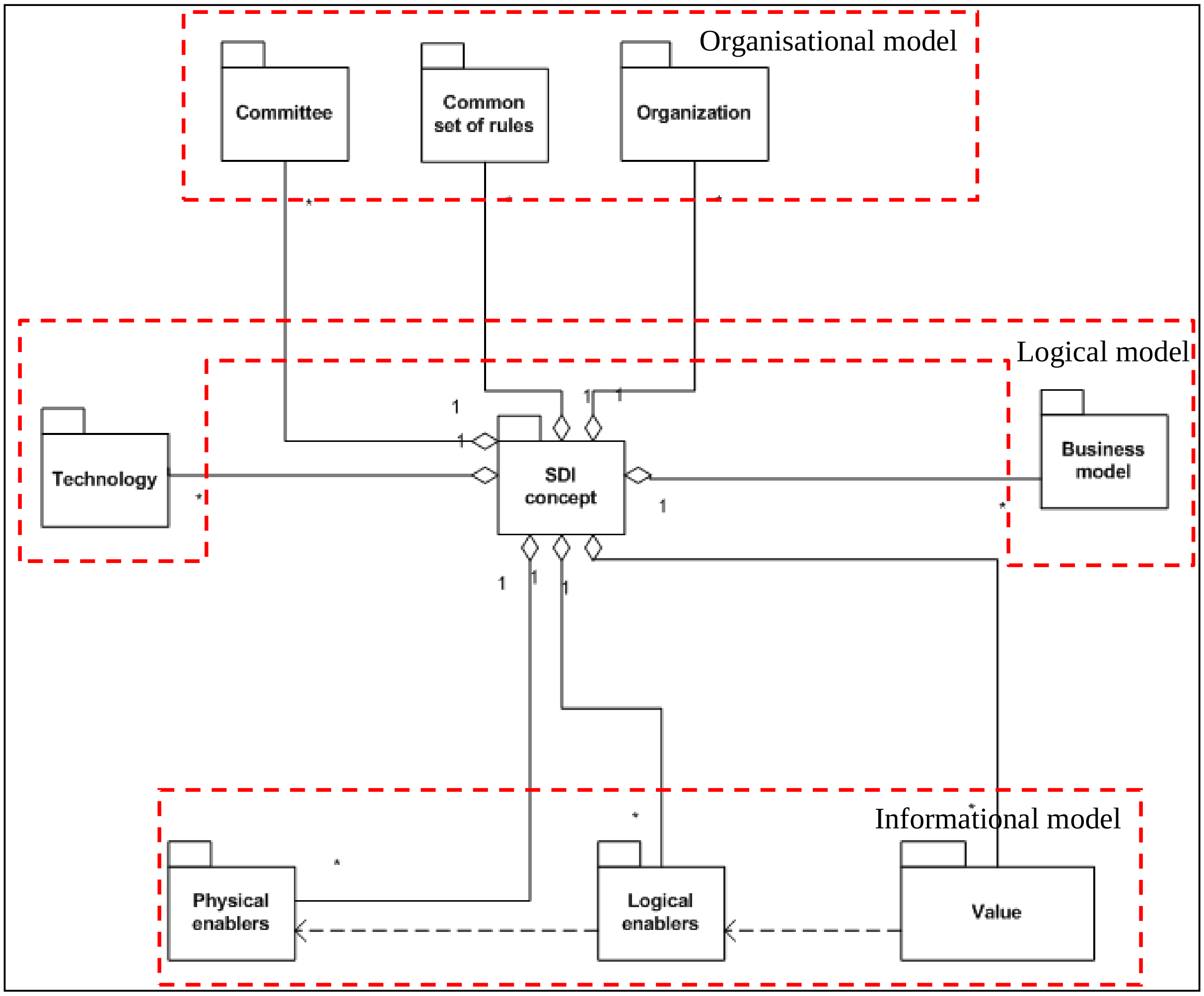

Figure 4: UML overview of the SDI concept

5. ACKNOWLEDGEMENTS

We would like to acknowledge the contributions made by the other members of the Commission, particularly at our meetings in Brno in July 2002 and Ithala in August 2003.

6. REFERENCES

Cooper AK & Nielsen AS, July 2002, UML model of SDI concepts, Summer 2002 meeting of the International Cartographic Association's Commission on Spatial Data Standards, Brno, Czech Republic.