---

1)
Metadata—Quo Vadis?

Metadata—Quo Vadis?

1. Introduction

Metadata is an important topic: many researchers produce enormous amounts of scientific publications—not only in geographic information science but generally in information sciences—and there is even legislation regarding geographic metadata [INSPIRE, FCDC (?)]. In contrast to all these activities, there is little sign of practical use of metadata, and the evidence there is, indicates that actual users eschew metadata and use other sources of information to make their decisions [Boin PhD, Boin and Hunter (?), Turk(?)].

---

2)

Goal of Talk:

Discrepancy

Metadata

- much research
- little metadata available
- hardly any use

---

3)

Some typical metadata entries:

Some typical comments from users

Take as an example the major tool to describe an ontology—the core part of metadata—is Protégé []; but Protégé has not become a household word like Google maps, or at least like “Oracle database”, “LAMP-server” or “KLM” describing other tools to organize and serve data, well known by web developers. In this keynote I will review the current state of metadata, concentrating on the ontology parts, which give the semantics of the data other aspects, like quality, spatial extent etc. add only more specific detail to the major point, namely what the data mean. I will argue that metadata, again especially the ontology, could go beyond a contribution to facilitate data exchange and interoperability, which are currently the major arguments and core cases OGC

---

4)

example screen for interoperability..

---

5)

Structure of talk

1.
2.
3.

Recent research of my group shows that additional use for metadata. Ontologies can be used to generate the user interface and automate graphical (map) presentation for data, at least. The requirements for such extensions of ontology use are derived and we used tools resulting from recent research in computer science to demonstrate the viability of these ideas.

Ontologies currently mostly describe the classes of things existing as taxonomic (is_a) and sometimes add mereological (part_of) relations. The key to more use is to include the operations. This is difficult in the logical framework currently dominant. An algebraic foundation for ontology description is necessary to make ontologies and metadata practical!

----

6)

Result:

Ontologies extended by definition of operations

-> further interoperability, permitting service discovery

-> GUI and visualization programmed automatically

----

7)

Necessary precondition.

Paradigm change in underlaying framework,

1st order logic

second order (algebraic) formalism

----

2. Semantic Matadata Today

There are several standards how to describe metadata, i.e., data describing the data. Some are specific for GIS [FDSD?] others are generic, independent of the application field. They invariably contain elements to describe the meaning (semantics) and the quality of the data.

Although I do not believe that a data quality description independent of the semantics is possible, I will concentrate today on the semantics and expect that operation semantics and expect that operation semantics in the metadata will help realistic fitness for user operation assessment [Frank Geog.Systems]. 1. Concentration

2) Data quality comes later

----

8)

overhead

Current (data exchanges) standards define the meaning of data through hierarchically structured terminologies often with some descriptive elements [FCDE].

2.1 Taxonomies (is_as)

----

9)

Critical reviews of such taxonomies have been published [] and it is accepted knowledge that pure taxonomies are insufficient for, e.g., helping with interoperability: different applications do structure reality differently and produce incompatible hierarchical taxonomies.

Generalizing to heterarchies, showing a general *is_a* relation; *X is_a Y* means that every element of the class *X* (e.g., dog) is also an element of the class *Y* (e.g., animal). Schemes allowing so-called "multiple inheritance" where one class is a subclass of more than one superclass makes this much more flexible.

Detail

order

is_a example

↓

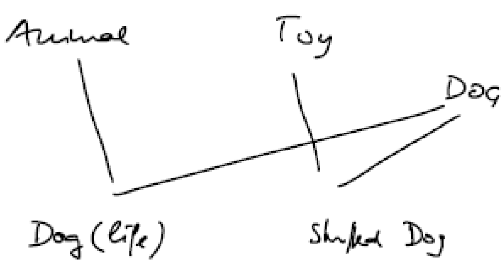
values not sufficient

↓

multiple

↓

example



----

10)

Overhead example waterway + stream

It has been pointed out decades ago that multiple inheritance is indispensable for describing geographic data [Frank-Egenhofer Toronto 1985?] but the logical complication with multiple inheritance—mainly caused by efficiency considerations in programming languages [C++]--did not advance widespread use.

----

11)

Overhead C++

----

12)

fundamental problem covariance

2.2 Mereology

Advanced tools include a *part_of* relation, to state that elements of *class P* are (necessary) part of an element of *class Q*. E.g., the heart is part of an animal body. [Smith mereology book medical ontologies]:

----

13)

example

The logical interaction between *is_a* and *part_of* relations are complex

----

14)

example

----

and tools to manage a description and to achieve consistency are crucial.

Protégé is the most often used ontology editor. It can produce output in various formats and operate different sets of restrictions.

----

15)

Protégé example waterways

2.3 Ontology Languages

The current debate in information science centers around ontology issues :

- formality

- expressive power languages—methods to describe an ontology.

The most popular language—with its own set of tools—is certainly the Unified Modeling Language (UML []) with tools like Rational Rose [] to enter data descriptions. UML is limited by an informally defined semantics of the constructs, which does not permit much logical inference.

----

16)

UML example waterway

The dominant ontology language development has produced a chain of languages OIL, DAML-OIL, to arrive today to OWL (...). It is based on description logic [Handbook], a subset of logic, specifically a well-defined set of subsets of logic with specific properties.

----

17)

List from Fallahi Figure 17)

The less restrictive a logic language is the more expensive it is, but also the more difficult are inferences. Relevant limitations are negation, generalization [Fallahi, articles].

----

18)

OWL example

These languages use all an XML method for encoding and are not meant for human readers but to be produced and consumed by programmed tools. Unfortunately, the most often used subset OWL-Light does not include XX ? and is therefore to limited for serious geographic applications because one cannot ...

----

19)

More advanced ontology languages, are used for example for the CYC project [] or linguistic methods as applied in wordnet are of [] only wordnet not CYC interest, but have been used—if ever—only in GIScience research.

----

20)

CYC example

2.4 Tools

----

21) Screen short

- Rat. Rose
- Protégé

The current ontology languages are not expressive enough to capture the complex geographic reality—they may be useful for limited administrative applications. The tools available show the limitations and are difficult to use, which probably explains why they are not popular among practitioners and they revert to the informally defined UML.

4. What Is Metadata Useful for?

The current argument for describing geographic data with metadata is an economic one. Often collected with substantial public funding, potential users can find the data and bring then to more use and economic savings—at least this is the politically accepted argument [INSPIRE, deutscher Bundestag]. Practically this argument may be flawed, because ontology descriptions including operations bring other benefits as I will show in the second subsection.

----

ags

21)

New GUI

Argument against this. Why not working? [Boin].

4.1 Data exclasse Interoperability

Metadata is collected and maintained to facilitate data exchange and interoperability: how to use datasets from multiple sources jointly, i.e., at least my own combined with data from others. Several steps can be identified to all:

----

22)

- discovery,
- usability decision,
- transfer or access to data,
- schema integration and alignment, or
- transformation of data formats.

A description of a dataset and its content in a standardized vocabulary, e.g., in a fixed taxonomy simplifies data discovery. A search for “gauge” data is (by fixing the vocabulary) effective because all data is described in this and not in equally probable alternative terms, e.g., “gauge” or languages “wasserstand”, “pegel”, etc. We expect then that the same term describes the same meaning—methods and measurement units [Fallahi PhD, Diss Münster].

----

23)

Dublin core metadata elements

To decide on usability of a discovered dataset, more information is necessary: included data elements, quality of the data and availability and prices. These are aspects of metadata not in focus today.

Schema integration is a productive research topic [Fousecca, Hornsby, Egenhofer, Kuhn, ...] but the solutions are very limited. They not always clarify what commonality between the datasets are assumed (e.g., [Lemmens PhD]) and may propose misleading equivalences between data from difficult sources. In my opinion only simple tools assuming a standardized vocabulary have the potential for limited practical use.

----

24)

Schema integration examples

expand for next

4.2 Ontology Descriptions for User Interfaces and Graphical Display

The user interfaces for input of data and commands by the user and the display of the data to make it visible to the user, is a generalized notions of interoperability. Interoperability here not between two computer systems, but between computer and human user.

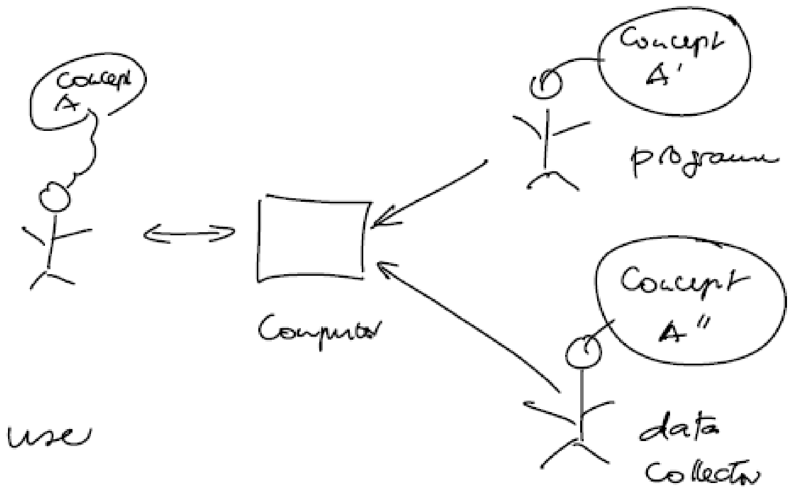
The ontology necessary describes a conceptualization of some parts of reality [Gruber, Guarino], of the reality of which the user but also the data collector has some conceptual knowledge.

----

25)

User interface is interoperability issues: human user

<->computer



Human users are very apt at aligning their conceptual (ontological) schema with the schema of their communication partner, as witnessed in everyday communication between foreigners the conceptualizations are certainly different.

Technically, the ontology contains all (or at least most of) the elements necessary for programming a user interface and a visualization. Confusion arises often when interfaces or visualization programmers do not properly understand the application ontology. Is it possible to produce user interfaces and the visualization programs automatically?

----

26)

The same happens at the GUI ... GUI shows more than

If two datasets are aligned, a consistent num interface could be ....

They watch for hints in the presentation about differences of the conceptualization used to help the user to align his conceptualization with the conceptualization used by the data collector. The conceptualization used by the data collector is described in the ontology and we must make it visible!

----

27)

Is this possible?

Some existing tools in the field of management of administrative (commercial) data give hope: the automatic production of reports based on database schema description with “Report Generators” is common for decades and (semi-) automatic production of input masks for data collection as well.

----

28)

Example WX Generic

The next section identifies what ontological information is necessary to produce more of the user interfaces.

5. GIS Are Computational Models of Reality

The data in a GIS—in general in every information system—is a computational model of reality. GIS give mostly static models and rely on application programs to add the operations to make “interesting” computational models.

----

29) How does this change our view of what a GIS is? (Reality Taxonomy)

Static GIS + static ontology (taxonomy)

->

dynamic as= computational geograhic model

+ontology for

data (static)

operations (dynamic)

The description of the semantics of the application operations is equally important for interoperability and some few suggestions for application ontologies and service discovery are made [].

6.1 Operations at GUI

The user sees reality as objects, where objects have state (data) and are involved in processes (operations).

----

30) OOview

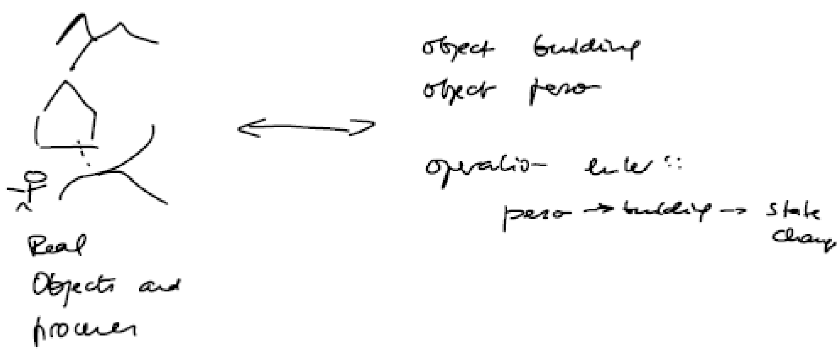
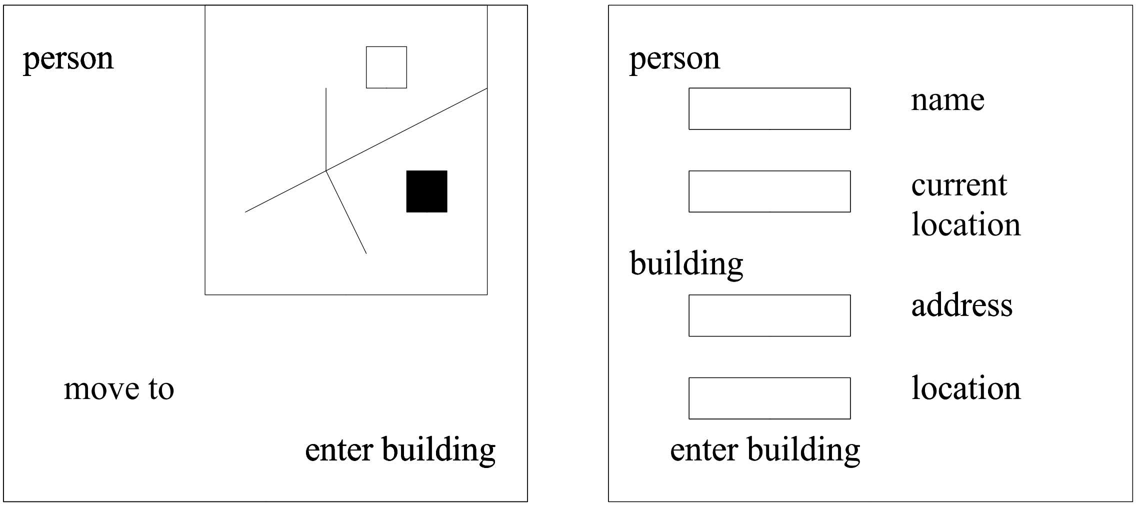


fig y

It is sufficient to map objects and operations to the interface. A screen like Fig X

----

31)



----

can in principle be prepared automatically from the ontology sketched in fig y.

6.2 Visualization

Similarly ontologies are useful when visualizing diversified data sets, which typically contain more detail a user can handle: the user must identify the processes he is interested in and aspects that are relevant to these processes will be visualized automatically and others can be left out.

----

32)

Example: A user interested in boat landing places, where he can bring his boat from his car trailer to the water is interested in

car driving: road -> place a -> place ->

place b

boating: water -> place a ->

place b

----

and this specifically where a place x has Affordance car driving and has Affordance boating (for a detailed discussion of the concept affordance see [Raubal], Gibson(?)]).

----

33)

6.3 What Is Missing?

The operations are missing in current ontology descriptions, and it is therefore not possible to make this step. Two questions will be addresses in the next two sections:

- How to describe operations in an ontology
- How to produce interfaces from ontology with operations

The used two sections remain sketchy as they discuss ongoing research; a move in-depth report would assume substantial technical detail not appropriate for a keynote.

6. How to Describe Operations in Ontologies?

Object-oriented software engineering has coined the slogan "objects are data with operations" [].

----

34)

Software engineering classes with ops C++ static example

Current ontologies describe static data structures and relations between data elements what to add ...dynamics but not operations with oo manifest.

Adding operations is not simple, because the ontology description methods are unlike programming languages which are hard to formalize and certainly not 1st order all based on (first order) logic, which does not allow variables ranging over relations or functions. This static view connects ontologies strongly to database schemes, but not to the object-oriented paradigm.

Object-orientation in software engineering is strongly associated with an algebraic view [Goguen, Yea, ...] Kuhn has recently [COSIT 2007] shown a GIS example of how an object + operation centered ontology can be described, using a second order language (in the form also Haskell [report]).

The difficulty is the "multiple inheritance"

Many details remain to be fixed before a usable tool is available, but the example is encouraging that a description, which is close to programming is possible and is flexible enough to capture the difference between two relatively close concepts, namely "houseboat" and "boat house".

----

35)

An alternative, using a verified view based on differences between entity classes, stressing aspects that can change and operations to change these, was published [Frank, FOIS]. A connection and translations between the two approaches seems possible.

7. Transformation of Ontology to User Interfaces

A systematic method to construct graphical user interfaces (GUI) is the result recent research and development [wxwidgets] and the integration with a powerful, second-order language to yield a “theory of user interface structure”, [xw haskell] numerous examples demonstrate the power to allow conceptual abstraction. Two aspects are relevant here.

----

36)

This gives not only a hierarchy of network classes that systematically inherits properties but also includes archive elements, events like buttons pressed by users and how the program reacts to these.

7.2 Data Type Driven User Interfaces

A description of data type yields automatically a user interface where data can be displayed and where users can enter data [wxgeneric] but not

----

37)

example

How to add operations

class

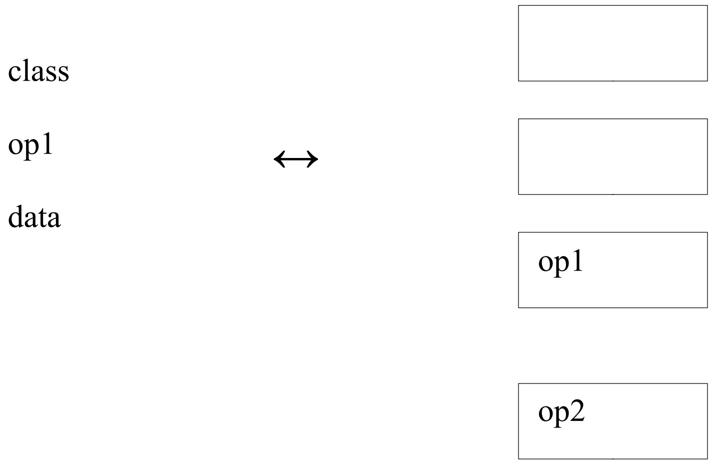
----

38)

class

op 1

data ↔



7.3 Composability of GUI Elements

---

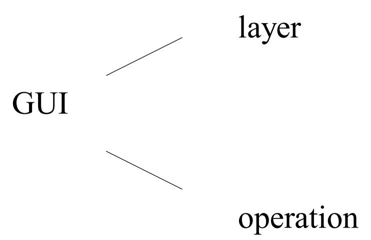
39)

Composability

The power of natural language is due to the possibility to compare freely elements to form new elements [Saussure]. Most user interface languages divide a user interface into a static description of the layout of the interface on the screen and of the logic of the processing behind it. This structure—with many tricky details—stands in the way of composing existing interfaces parts to new ones. The exceptions are web maskups, where composing existing web interface pieces works freely—and the rapidly gained popularity speaks once more for the power of composability.

---

40)



7.4 Actions & Events Manage Actions and Assign Them

The difficulty with composability is not only to compose the layout. Behind this more obvious problem, for which some practical solutions akin to cartographic label placement [Freeman are known, is the more obscure management of events: did the user move the mouse to here? Did he click here or there? What actions must be started in one or the other case?

7.1 Advances of

----

41)

The advantage of user interfaces produced automatically and from the composition of simpler parts is consistency. The same interactions are everywhere presented in the same form and with the same semantics, because it is the same code executed.

7.5 Current State

----

42) List state approach monad (a construction from category theory) specifically state monad

----

In the aforementioned wxHaskell framework, several research efforts have achieved composability [Phooney, Fruit, Composite]. They need to be tested and refined for practical examples of particular interest to see how they live up to work with map graphics and not only textual interfaces. Such efforts are part of an ongoing PhD thesis [Syed]. Geographic applications

8. Conclusion

----

42) Conclusion

43) give metadata a real use—producing a

44) Preconditions Paradigm change

45) A practical step up the ladder of abstraction

Extending metadata, especially the ontology to include not only a static description of the data, but the operations applicable as well, is a necessary next step. Ontologies with operations do not only better support data exchange and interoperability—especially discovery of datasets—and make discovery of services (which are operations) possible; ontologies with operations allow new users for ontology: namely, the automatic deduction of user interfaces and visualization. The current approaches are based on first order logic and cannot accommodate operations easily. To overcome the limitation in ontology description (as well as to make advances with temporal GIS [Chorochronos book] as well, second order languages with a more algebraic, object-oriented flavor) are necessary. The benefit from ontologies with operations will be the (mostly) automatic production of consistent user interfaces and visualizations. I expect this additional benefit from ontologies to be the step that is necessary to induce practitioners to describe the ontology of their data (in lieu of burying the same information in their code).