

# Representation of Continuous Fields

Andrej Včkovski

University of Zürich

Winterthurerstr. 190, CH-8057 Zürich, Switzerland

vckovski@gis.geogr.unizh.ch

## Abstract

This contribution discusses the representation of continuous fields within Geographical Information Systems. The fields considered are fields of physical properties (observables) defined for every point of a spatio-temporal region. The two major problems discussed are the *inherent* uncertainty present in the samples of the field (field values are measurements) and implications of the discretization when continuous fields are sampled. Fields are usually measured at a finite set of points or regions in space and time. The subsequent use of such data sets, however, often requires field values at other, unsampled locations, or with different temporal and/or spatial aggregation and unit systems. We present the concept of Virtual Data Sets (VDS) which helps to overcome such incompatibility problems. A VDS incorporates the necessary semantics of the data thereby allowing such transformations (i.e., interpolations, aggregations, unit transformations) to be performed transparently and within the domain of the data set. This is achieved by defining a common interface layer which lets the user query a VDS within the GIS and other client applications in a more general way than current GIS allow.

## 1 Introduction

Due to their strength in analysis and visualization of spatial data Geographical Information Systems (GIS) are frequently used for scientific information processing. Especially research in the broad field of the earth sciences (e.g., oceanography, atmospheric physics, geology, seismology) involves large spatial data sets and their integrated analysis. These data sets usually consist of a set of aggregated samplings (macro data) of natural phenomena (e.g., temperature), where every value is somehow related to a region or subset of space and time, i.e. related to a *spatio-temporal object* which we call *index* for a particular value. Those objects may be points (i.e., samplings at a given point in space and time), squares of a regular partition of a subspace (i.e., pixels of a satellite image) or other objects with a (temporal and spatial) geometry that is usually owing to the sampling and/or preprocessing stage. The samplings together with the related indices are an estimate of the corresponding field, i.e., an estimate of the physical property as a function of space and time.

There is a growing need to use and re-use such data sets for various applications, usually involving many different data sets for specific analyses, simulations and other scientific work. Previous work already dealt with issues of the representation of

continuous fields and appropriate data models (Kemp, 1993), (Laurini & Pariente, 1994).

As mentioned before, in most cases data sets entering such integrated analysis projects are already aggregated to some degree, i.e., they represent *macro data* as opposed to *micro data* (“raw” sampling values). The aggregation or postprocessing might be hidden within the sampling apparatus (e.g., temporal aggregation during the exposure time, and spatial aggregation over a pixel in a satellite sensor) or performed explicitly by the data collection organization (e.g., temporal averages over a time period). The aggregation leads to non-point indices<sup>1</sup> on the one hand, since the aggregation is some integral over space and/or time. On the other hand, the aggregation helps the data collectors to assess some information about the quality of the data, e.g., magnitude of measurement errors.

Therefore, a typical data set consists of a set of measurements being expressed as a set of mean values and standard deviations, for example, and a corresponding set of indices. When such data sets are used as an estimate of the corresponding field there are usually two major problems:

- Field values are needed for indices that are not available in the given set of indices. This means simply that one often needs values at “unsampled locations”.
- It is non-trivial to include the usually<sup>2</sup> available information about the measurement errors and other sources of uncertainties when using the data for further computations and analyses. This is especially a problem when data are used in multi-disciplinary projects by scientists from domains other than the data collectors’.

Here, an approach is presented to enhance reliability and usability of computer based scientific work when using data that represent fields. The next section will introduce some notation and give definitions of some important terms. Section 3 then gives a short overview of the digital representation of uncertainties (e.g., non-exact values), which is one of the basic building-stones of the Virtual Data Sets presented in section 4. This is followed by a short outlook for our future work.

## 2 Notation

A *field* is a physical property or observable  $z(s)$  that is spatially and temporally indexed (index  $s$ ). For the sake of simplicity we will just consider properties where the value domain  $\mathbb{B}$  is real, i.e.  $z(s) \in \mathbb{R}^N$ . The field values might be scalars ( $N = 1$ , e.g., temperature) or vectors ( $N > 1$ , e.g., wind speed). We will restrict the following discussion to scalars (or 1-dimensional vectors), but the generalization to  $N > 1$  should not pose any difficulties. An ideal (undisturbed and deterministic) field is a

---

<sup>1</sup>There are also cases where the indices can be approximated by points, e.g., measurements of stationary (in time) phenomena. The temporal dimension has not to be considered then. A digital terrain model (DTM) for example represents the (approximately stationary) terrain height field (THF) and its indices usually do not include temporal information.

<sup>2</sup>One might argue, that the “quality information” is exactly the type of information usually *missing*. I think, that information of data quality in general and measurement errors in particular are available in most cases, but this information often gets lost on the way from the data collectors to the data users. This is partly due to missing capabilities of the infrastructure for data exchange and data processing (e.g., data exchange formats, features of the GIS systems used) and partly to some user’s ignorance about the importance of error information.

function  $z(\cdot)$  which relates every value  $s$  of an index domain  $\mathbb{I} \subset \mathbb{R}^M$  to a value  $z(s)$  from the value domain  $\mathbb{B} \subset \mathbb{R}$ :

$$s \xrightarrow{z} z(s), \quad s \in \mathbb{I} \subset \mathbb{R}^M, \quad z(s) \in \mathbb{B} \subset \mathbb{R} \quad (1)$$

This model of a field has to be enhanced to allow for uncertainties owing to inherent indeterminism of the physical property and measurement-induced uncertainties. Let  $Z(s)$  be such an enhanced description of a field. If the uncertainties can be modeled using probabilistic concepts, then  $Z(s)$  is a random variable for every  $s$ .  $Z(s)$  is sometimes called a *random function* (Isaacs & Srivastava, 1993). Depending on the nature of the phenomenon under consideration  $Z(s)$  might also be an interval, fuzzy set, or some other means of modelling uncertainties. These objects usually can be described or approximated by a set of real numbers which are the (estimated) values of corresponding functionals (mappings to  $\mathbb{R}$ )  $\phi(\cdot)$ . For random variables, typical functionals are the *expectation value*<sup>3</sup>  $\phi_E$  and the *variance*  $\phi_V$ .

A data set describing such a field consists of metadata and a set of index-value tuples:

$$\mathcal{D} = \{M, \{s_i; z_{i,1}, \dots, z_{i,n}\}_1^n\}, \quad s_i \subset \mathbb{I}, \quad z_{i,j} \in \mathbb{R} \quad (2)$$

For every index  $s_i$  the values  $z_{i,1}, \dots, z_{i,n}$  describe the field at  $s_i$ . A typical example might be that  $n = 2$  and  $z_{i,1}$  is a mean value and  $z_{i,2}$  the standard deviation of the field over  $s_i$ .  $M$  is the metadata describing  $\mathbb{I}$  and  $\mathbb{B}$ , the corresponding unit systems (i.e., units and coordinate systems including appropriate metrics) and any other metadata necessary and available.

It is important to analyze and understand how the values  $z_{i,j}$  of a data set are related to the field  $Z(s)$ . This relation – which mathematically is the relation between  $s_i$  and  $z_{i,j}$  – consists of two parts:

- The relation between the index  $s_i$  and the field  $Z(s)$ : Since  $s_i$  is a non-point set in many cases (i.e., a spatio-temporal region) the values  $z_{i,1}, \dots, z_{i,n}$  in the data set corresponding to  $s_i$  are the result of some aggregation  $A_{s_i}$  of  $Z(s)$  over  $s_i$ :

$$\hat{Z}(s_i) = A_{s_i}(Z(s)) \quad (3)$$

$A_{s_i}$  might be an average over  $s_i$ , the value in the “center” of  $s_i$  etc.

- The relation between  $\hat{Z}(s_i)$  and the data set value  $z_{i,j}$ : This is basically the fore mentioned functional  $\phi_i$  yielding  $z_{i,j} = \phi_j(\hat{Z}(s_i))$ .

The  $s_i$ ’s are therefore related to the  $z_{i,j}$  through  $A_{s_i} \circ \phi_j$ :

$$s_i \xrightarrow{A_{s_i} \circ \phi_j} z_{i,j} \quad z_{i,j} = \phi_j(A_{s_i}(Z(s))) \quad (4)$$

In order to use a data set  $\mathcal{D}$  and to understand its values it is necessary to have an idea about the mappings  $\phi_j$  and  $A_{s_i}$ . It would be optimal to know its inverse since this would allow determination of the field values from the samplings.

Mathematical system theory calls the process  $A_{s_i} \circ \phi_j$  which the *behaviour* of the system under consideration (Kalman, 1982).  $A_{s_i} \circ \phi_j$  depends on the whole measurement process and contains all the transformations involved when measuring a field  $Z(s)$  (e.g., measurement apparatus, preprocessing). For the analysis of field measurements, i.e., the reconstruction of the field, the inverse  $(A_{s_i} \circ \phi_j)^{-1}$  has to be approximated, i.e., modelled.

---

<sup>3</sup>To be rigorous one would have to distinguish between those functionals and their estimates. The notation in the following will be a bit sloppy where the interpretation should be clear from the context.

### 3 Digital Representation of Uncertainties

The previous section has shown that physical properties affected by some uncertainty are usually characterized with a small set of real numbers, e.g., a mean value and a standard deviation. The way to describe uncertain values depends strongly on the nature of the sampling process and the phenomenon investigated. It is, therefore, impossible to define one single way to describe uncertain values applicable to all cases. A digital representation of uncertain information should meet the following requirements:

- Digital encoding means mapping to real numbers<sup>4</sup>. It is desirable that the representation uses a small set of numbers.
- Operators and operations for uncertain values should be defined. For instance, the arithmetic operators  $\diamond \in \{+, -, \times, \div\}$  should be available along with standard functions (e.g., trigonometric functions).
- It should be possible to convert one representation to another since different data sets and their values will often use different representations.
- A suitable set of functionals (mappings to  $\mathbb{R}$ ) should be available, e.g.,  $\inf(\cdot)$  (lower bound),  $\sup(\cdot)$  (upper bound),  $\alpha_i(\cdot)$  ( $i$ -th moment),  $\beta_i(\cdot)$  ( $i$ -th central moment).
- If the representation is based on probability theory it should support *Monte Carlo simulations* (Johnson, 1987). The representation of a random variable  $A$  should be able to generate pseudo-random realizations  $a_i$ . Actually, this is just another type of functional  $\phi(\cdot)$  which we call  $\text{rnd}(\cdot)$ .<sup>5</sup>

One of the basic decisions when choosing a suitable model for an uncertain value is whether it can be modelled with *probabilistic concepts*. In most cases it will be the primary choice to use probability theory to describe uncertainty within scientific results. Since its formalization (Kolmogorov, 1933) probability theory is quite well understood and a lot of methodology has been developed. Especially mathematical statistics has benefited from this framework and has produced many useful techniques and methods for the description and analysis of data. The random-ness of a property (random variable  $X$ ) is defined within probability theory with a corresponding probability distribution  $p(x)$ . Three variants were selected to describe a probability distribution  $p(x)$ :

**Empirical moments:**  $\alpha_k = \langle \int x^k p(x) dx \rangle$  and  $\beta_k = \langle \int (x - \alpha_1)^k p(x) dx \rangle$ , usually  $\alpha_1$  (mean value) and  $\beta_2$  (standard deviation).

**Empirical distribution function or histogram:** The distribution  $p(x)$  is described with a set of quantiles, i.e.,  $(x_i, p_i)$  with  $P(X \leq x_i) \approx p_i$ .

---

<sup>4</sup>In fact, digital encoding means mapping to integer numbers. There are, however, means to represent a finite, countable subset of the real numbers with integer numbers (i.e., *floating point numbers*).

<sup>5</sup>The functional  $\text{rnd}(\cdot)$  actually has another “hidden” parameter (sequence number) which identifies the realization requested. During a Monte Carlo simulation run, a specific realization might be requested several times so that it is necessary that the value is always the same during one simulation step. Consider for example an expression  $C = A + AB$ . The simulation would calculate  $c_i = \text{rnd}(A) + \text{rnd}(A)\text{rnd}(B)$ . The second  $\text{rnd}(A)$  must have the same value as the first one

Representation	para- meters	$\inf(\cdot)$ $\sup(\cdot)$	$\alpha_l(\cdot)$ $\beta_k(\cdot)$	$\text{rnd}(\cdot)$	$+, -,$ $\times, \div$	Standard- functions
Interval	2	●	○	○	●	●
Fuzzy Set	*	●	○	○	●	●
$\alpha_i, \beta_j, i \leq n, j \leq m$	$n + m$	●	*	●	*	*
Histogram ( $K$ classes)	3K	●	●	●	●	●
Uniform distribution	2	●	●	●	○	○
Normal distribution	2	○	●	●	$+, -$	○
Other distributions	*	*	●	●	*	*

● = available, ○ = not available, \* = variable.

Table 1: Properties of different representation types for uncertain values

**Parametric distributions:** A distribution type may be determined when the micro data are aggregated, i.e., the empirical distribution is approximated with a standard distribution and its parameters. Typical parametric distributions are the normal distribution (parameters  $\mu, \sigma^2$ ), uniform distribution ( $a, b$ ) or Weibull<sup>6</sup> distribution ( $p, \gamma$ ). Sometimes the parameters of a distribution correspond to some moments, e.g., for the normal distribution  $\mu = \alpha_1$  and  $\sigma^2 = \beta_2$ . It is, however, different to describe a distribution solely by moments than by distribution type and a set of parameters.

It is not always suitable to apply probability theory to describe uncertainty. Therefore, two other non-probabilistic ways to describe uncertain values were included:

**Intervals:** Intervals define lower and upper bounds for a value and are very convenient due to their simplicity. An important advantage are the simple rules for computations using interval values (e.g., (Moore, 1966), (Bauch *et al.*, 1987), (Mayer, 1989), (Moore, 1992), (Polyak *et al.*, 1992)).

**Fuzzy sets:** Fuzzy sets (Zadeh, 1965) may be seen as an extension to intervals. The basic idea is to define a “degree of membership” for a number in a (fuzzy) set. This is in contrast to intervals where a number is either within the interval or not.

While intervals are a simple yet powerful way to deal with uncertain real-valued data, fuzzy sets have not yet been used widely to describe scientific data<sup>7</sup>. despite the attention it had in the last decades. The major criticism of fuzzy sets is the usually subjective assignment of membership degrees (definition of set membership functions) which often is not acceptable in scientific work where objectivity is an important issue.

Table 1 summarizes the properties of these representations.

<sup>6</sup>Most often the special case with  $p = 1$  (exponential distribution) is used.

<sup>7</sup>However, for some applications see (Bandemer & Näther, 1992)

## 4 Virtual Data Set

### 4.1 Requirements

The previous section has introduced different methods to digitally represent uncertain values (i.e., physical properties from the real world). In this section an architecture is presented that is suitable for the representation of *fields* consisting of uncertain values.

Although the discussion here is rather theoretical, the final goal is to have an *actual* application. The properties of a representation, therefore, are strongly influenced (or even determined) by the user's requirements. The typical information the user wants from a data set  $\mathcal{D} = \{M, \{s_i; z_{i,1}, \dots, z_{i,m}\}_1^n\}$  describing a field are

**Query type A** Retrieve the various components of the data set, e.g. the metadata ( $M$ ), the set of indices  $\{s_i\}$  and for every index  $s_i$  the related set of parameters describing the field value  $z_{i,1}, \dots, z_{i,m}$ . This type of query is available in every system. It simply retrieves available data values.

**Query type B** Estimate  $\phi(Z(s))$ , where  $\phi$  is some functional (e.g., mean value) and  $s \in \mathbb{I}$  is an “unsampled location”, i.e.  $s \notin \{s_i\}$ .  $\phi$  might be one of the functionals  $\phi_1, \dots, \phi_m$  that define the set of parameters available for every  $s_i$  (i.e.,  $z_{i,j} = \phi_j(Z(s_i))$ ), so that it is not necessary to transform the representation as shown in the last section. This type is probably one of the most important queries when working with fields.

**Query type C** An advanced information request is to query the field for the spatial references where the field value has a certain value, i.e., for a specific  $z_0$  and  $\phi(\cdot)$ , solve the equation  $\phi(Z(s)) = z_0$  for  $s$ . Consider a digital terrain model, where  $\mathbb{I} \subset \mathbb{R}^2$ , all  $s_i$  are points, every field value (height of terrain) is given by one parameter which is the mean value (i.e.,  $m = 1$  and  $z_{i,1}$  is the mean height of the terrain at  $s_i$ ). Then, the solution would “compute the set of points  $s$  where the mean terrain height is  $z_0$ ”. This type of query is basically an inversion of the field function  $Z(s)$ .

In the context of this paper we will not discuss information requests of type C (inversion of field function). Instead, we will focus on queries of type A and B. In principle, type A queries can be answered by most of the current GIS and related systems.<sup>8</sup>

Type B queries typically need much more user interaction. The procedure to compute  $\phi(Z(s))$  at a location  $s$  is not trivial and involves sometimes very complicated computations. Even if  $\phi(\cdot)$  corresponds to one of the parameters available in  $\mathcal{D}$ , i.e.  $\phi(\cdot) = \phi_j(\cdot)$ , the estimation of the field value parameter  $\phi_j(Z(s))$  needs sophisticated inter- and extrapolation methods.

A common approach to handle type B queries is to transform them into type A queries. This means that a set of locations  $s'_i$  (and a set of functionals  $\phi'(\cdot)$  at which the field might be queried is determined beforehand. The data set  $\mathcal{D}$  is then transformed to a new data set  $\mathcal{D}' = \{M', \{s'_i; z'_{i,1}, \dots, z'_{i,m'}\}_1^{n'}\}$  so that every query for  $\phi'(Z(s'_i))$  will be a type A query. There are, however, many situations where this approach fails due to the long life cycle of some data sets. The transformation

<sup>8</sup>It is not always easy to obtain all of the metadata ( $M$ ) or the set of indices ( $\{s_i\}$ ) directly, e.g., the coordinates of the pixel cells of a satellite image

to a new data set  $\mathcal{D}'$  might not satisfy future queries, so that  $\mathcal{D}$  or  $\mathcal{D}'$  have to be transformed again into a new data set  $\mathcal{D}''$  that conforms to the new requirements. Sometimes the initial data set  $\mathcal{D}$  is not available anymore, so that  $\mathcal{D}''$  has to be computed from  $\mathcal{D}'$ . Every transformation  $\mathcal{D} \rightarrow \mathcal{D}'$  usually affects the quality of the data, the quality rarely ever increases.

Another problem that contributes to the loss of quality are the methods used to transform  $\mathcal{D}$  to  $\mathcal{D}'$ . Most often, these methods, i.e. the computation of  $\phi(Z(s))$  given  $\{s_i\}$  and the related  $\{z_{i,j}\}$ , are not trivial and need a lot of expert knowledge from the domain scientists and data collectors, respectively. When a field is sampled this expert knowledge is available in the data collecting organization. At a later stage, e.g., when transforming from  $\mathcal{D}^{(ix)} \rightarrow \mathcal{D}^{(x)}$ , this expert knowledge often is not available anymore. It is therefore the user's own choice and responsibility *how* to transform  $\mathcal{D}^{(ix)} \rightarrow \mathcal{D}^{(x)}$ .

## 4.2 Concept

The concept of the Virtual Data Set (VDS) described in (Stephan *et al.*, 1993) and (Bucher *et al.*, 1994) tries to minimize quality loss due to subsequent transformations, while maintaining usability of a data set at the same time. The main idea is to allow for queries of type B. A data set  $\mathcal{D}$  is therefore enhanced with information  $P$  needed to process queries of type B.

$$\mathcal{V} = \{P, \mathcal{D}\} = \{P, M, \{s_i; z_{i,1}, \dots, z_{i,m}\}_1^n\} \quad (5)$$

An approach would be to formalize the necessary information so that  $P$  includes standardized specifications for the answer of type B queries, e.g. interpolation method to use, parameters of the methods, uncertainty modelling methods used. For many data sets this approach would fail, for example because the interpolation method used has some very specific constraints which were not taken into consideration when  $P$ 's content was formalized.

VDS uses another approach.  $P$  is *procedural information* in the sense that it defines the entire method to process and answer queries of type A, type B (and probably type C). In the terminology of object-oriented design (Booch, 1991)  $P$  is the *behaviour* of object  $\mathcal{V}$  and  $\mathcal{D}$  its *state*. The term "virtual" emphasizes the VDS's capability to process queries of type B, i.e., queries that return values not stored on secondary storage.

This VDS-concept is similar to other approaches which try to enhance reusability of expensive data sets using object-oriented concepts (e.g., the OGIS project (Buehler, 1994)). A computer industry term for  $P$  could be *middle-ware*, giving a client standardized access to the data ( $\mathcal{D}$ ).

The next section outlines some implementation and design issues that have been considered as first experiments.

## 4.3 Implementation

The current design for the implementation of VDS uses a *client-server* architecture. Data requestors (e.g., a GIS application) are clients; a VDS  $\mathcal{V}$  or its procedural part  $P$ , respectively, is a server. Communication between the clients and servers is based on messages which are queries of type A or B and their corresponding replies. Whenever a client needs data it sends a request to a VDS (server), where the request is

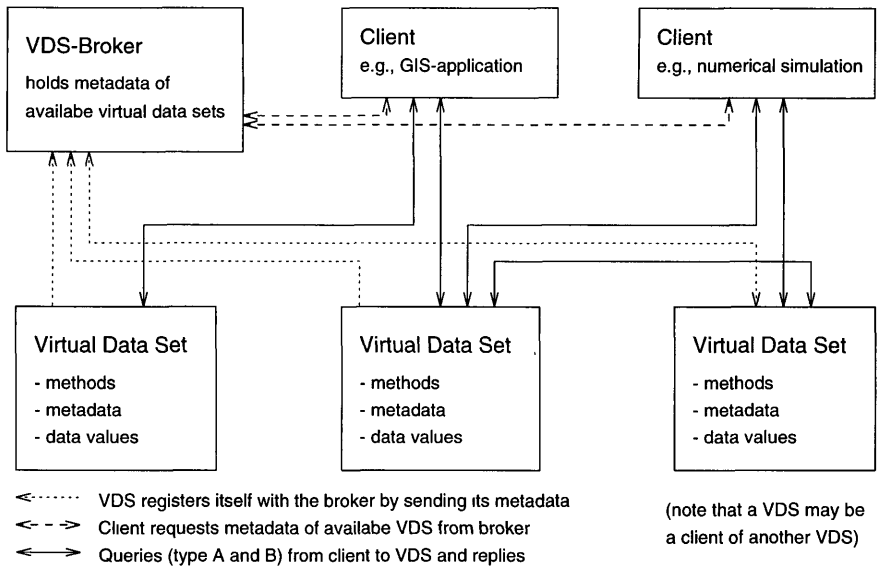


Figure 1: Communication between virtual data sets, broker and clients

processed and the results are returned to the client. The processing on the server-side (the component  $P$  of  $\mathcal{V}$ ) might just read the data requested from secondary storage (i.e., type A query), or read data and apply some interpolation methods (i.e., type B query). In addition to clients and VDS-servers there is a specialized server we call *VDS-broker*. The broker is a metadata-base that holds the information of the VDS available<sup>9</sup>. Figure 1 shows the different components involved and the communication between them.

As soon as a VDS becomes available (e.g., “runs” somewhere or is initialized) it sends its metadata  $M$  to the broker in a specified format, i.e., the VDS registers itself with the broker. Whenever a client needs some data it may look up if the requested VDS is available and how to access it (i.e., sending a corresponding request to the broker). Once the client knows the “address” of a VDS it sends future requests directly to the VDS. This structure allows the client and server parts to run on different computers, leading to a distributed computing environment. A message sent to a server might be a *local function call* or a “real” message sent over a *network*. Since the queries of type B might include a specific functional  $\phi(\cdot)$  a VDS needs to be able to handle different types for representing uncertainty. While the way a VDS handles the data  $\mathcal{D}$  internally is not relevant to a client, the results of queries processed by  $P$  are. Therefore, the creation of the  $P$ -part of a VDS is simplified by a common (class) library which covers the following functionality:

- low-level details for the communication to and from a client or server, respectively

<sup>9</sup>Its existence is technically motivated. It simplifies the communication from clients to servers because clients can easily look up existing servers through the broker.



- construction of uncertain values and conversion between different representations, evaluation of arithmetic expressions and standard functions involving uncertain values
- a collection of general purpose interpolation methods and associated helper functions (e.g., spatial neighbour search, etc.).

## 5 Conclusion And Outlook

This contribution has presented an overview on the questions involved when dealing with data sets that represent continuous fields. The two major issues were:

- The data available is *never* accurate. It is *inherently* necessary to use techniques for modelling uncertainty and errors.
- The field values often are queried for other locations, unit systems, types of aggregation, etc., than available within the data set.

The concept of the Virtual Data Set tries to approach those problems just by the definition of a clear interface for data access. In fact, it does not *solve* the problems but defines *where* the problems should be solved. It is nonetheless a concept that can be translated into a real system as first prototyping approaches have shown. We believe that the more complex systems and applications get, the more important interoperability questions become. The slowly emerging trend in recent years is towards systems built up from a custom set of cooperative modules or services. It has been shown that this is a way to reduce complexity and benefits both users and system providers. The former will have customized solutions instead of large and monolithic ones and the latter can increase the system quality delivered since they systems are more easily maintainable.

Our future work consists of more general prototype implementations and the selection and adoption of suitable industry or scientific standards for the implementation of distributed, modular systems (e.g., OMG CORBA (CORBA, 1992), OGIS (Buehler, 1994), COSIMA (De Lorenzi & Wolf, 1993)).

## References

- Bandemer, Hans, & Näther, Wolfgang. 1992. *Fuzzy Data Analysis*. Kluwer Academic Publishers.
- Bauch, H., Jahn, K.U., Oelschlägel, D., Süsse, H., & Wiebigke, V. 1987. *Intervall-mathematik*. BSB Teubner, Leipzig.
- Booch, Gary. 1991. *Object Oriented Design with Applications*. The Benjamin / Cummings Publishing Company, Inc.
- Bucher, Felix, Stephan, Eva-Maria, & Včkovski, Andrej. 1994. Integrated Analysis and Standardization in GIS. In: *Proceedings of the EGIS'94 Conference*.
- Buehler, K. A. 1994. *The Open Geodata Interoperability Specification: Draft Base Document*. Tech. rept. OGIS Project Document 94-025. OGIS, Ltd.

- CORBA. 1992. *The Common Object Request Broker: Architecture and Specification*. Object Management Group. OMG Document Number 91.12.1.
- De Lorenzi, Michele, & Wolf, Andreas. 1993. A Protocol for Cooperative Spatial Information Managers. In: *Workshop on Interoperability of Database Systems and Database Applications*.
- Isaacs, E.H., & Srivastava, M.R. 1993. *An Introduction to Applied Geostatistics*. Oxford University Press, New York.
- Johnson, Mark E. 1987. *Multivariate statistical simulation*. Wiley series in probability and mathematical statistics. Applied probability and statistics. John Wiley and Sons, New York.
- Kalman, Rudolf E. 1982. Identification from real data. *Pages 161–196 of: Hazewinkel, M., & Kan, A. H. G. Rinnoy (eds), Current Developments in the Interface: Economics, Econometrics, Mathematics*. D. Reidel Publishing Company, Holland.
- Kemp, Karen K. 1993. *Environmental modelling with GIS: A Strategy for Dealing With Spatial Continuity*. Ph.D. thesis, NCGIA.
- Kolmogorov, A. N. 1933. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Berlin: Springer Verlag.
- Laurini, R., & Pariente, D. 1994. Towards a Field-oriented language: first specifications. In: *Papers presented at the GISDATA Workshop on Spatial Objects with indetermined boundaries*.
- Mayer, G. 1989. Grundbegriffe der Intervallrechnung. In: Kulisch, U. (ed), *Wissenschaftliches Rechnen mit Ergebnisverifikation*. Akademieverlag.
- Moore, Ramon. 1966. *Interval Analysis*. Englewood Cliffs, N.J.: Prentice Hall.
- Moore, Ramon. 1992. Parameter sets for bounded-error data. *Mathematics and Computers in Simulation*, **34**, 113–119.
- Polyak, B., Scherbakov, P., & Shmulyian, S. 1992. Circular Arithemtic and Its Applications in Robustness Analysis. *Pages 229–243 of: Kurzhanski, Alexander B., & Veliov, Vladimir M. (eds), Modeling Techniques for Uncertain Systems*. Basel: Birkhäuser.
- Stephan, Eva-Maria, Včkovski, Andrej, & Bucher, Felix. 1993. Virtual Data Set: An Approach for the Integration of Incompatible Data. *Pages 93–102 of: Proceedings of the AUTOCARTO 11 Conference*.
- Zadeh, L. A. 1965. Fuzzy Sets. *Information and Control*, **8**, 338–353.