JOURNAL OF CHEMOMETRICS
J. Chemometrics 2003; **17**: 34–44
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cem.774

Multivariate data analysis: *quo vadis*? I. Object-oriented data modelling (OODM)†

Kim H. Esbensen[1] and Agnar Höskuldsson[2]*

[1]Institute of Chemistry and Applied Engineering, Aalborg University Esbjerg, DK-6700 Esbjerg, Denmark
[2]IPL, Building 358, Technical University of Denmark (DTU), DK-2800 Lyngby, Denmark

Received 15 January 2002; Accepted 16 October 2002

Industry and academe are characterized by steadily increasing huge amounts of data with very different data structures. Both static and dynamic data contexts need to be addressed. A new generic, flexible and comprehensive general data-modelling concept is needed to cope with these demands. During the past 20 years, *object-oriented programming* (OOP) has become a *de facto* industry standard of how programming tasks should be defined and carried out in the context of deterministic data modelling. We present here a first framework of *analogous ideas* for multivariate data analysis. A new strategy, *object-oriented data modelling* (OODM), is proposed which is invariant with respect to the specific data structures and the practical data context. We present a first delineation of meta-principles, ideas and stimulants for tomorrow's possible development paths of modelling, in which the fundamental data analysis unit is the generalized 'PLS object' in the OOP sense. The key novel aspect concerns *inter-object information transfer*, facilitated by 'root-sum-of-squares averaging' (RSSA), which uses w loading weights as between-object transfer agents. These features allow a powerful generalization beyond multiblock as well as hierarchical bilinear modelling to be laid out. The present part I outlines a first framework for the new data-modelling approach, while part II forms a complementing catalogue of specific options and possibilities when implementing the new principles. Copyright © 2003 John Wiley & Sons, Ltd.

KEYWORDS: latent data structures; object-oriented programming; object-oriented data modelling; H-principle; H-object; updated weighting; generalized bilinear data analysis; meta-principles

1. INTRODUCTION—*STATUS QUO* FOR DATA ANALYSIS

1.1. Statistical data modelling versus the chemometric approach

Statistical methods for data modelling have been well established over the last 100 years. The focus of the statistical sciences has been to establish methods describing the behaviour or properties of populations. Thus it is common to use separate notation for the population properties and the observed samples. The argument is that the population properties are fixed, while the samples vary from time to time (from one sample to another). Each internal data set is a rectangular table (matrix), where each row (observation) is specifically viewed as a representative for repeated sampling of the variables that represent the table columns. Very often the assumption is that of the multivariate normal distribution, from which the data values for each object of the data table are viewed specifically as but random fluctuations around a certain average or 'expected' level; the levels of course differ from column (variable) to column. The objective of most applied statistical methods is to identify the mean value structure of the variables. The motivation for this *modus operandi* is to generalize the laws of physics to other scientific or technological phenomena. Philosophically, as well as practically, there would appear to be a certain risk for *reductionism* here.

Opposing this is the well-known chemometric approach, which aims at modelling covariance data structures as presented by the available variables and with a distinctly different view on the role of the objects; these now represent individual measurements *per se* (which may be correlated or not, as the case may be). The sample–population aspect is specifically not carried over to multivariate data analysis in the chemometric context [1,2].

1.2. Industrial, practical requirements of data modelling

There is today a tendency towards a growing contradiction between statistical, properly researched methods and the immediate needs in practical, industrial data modelling. Industry views data-modelling procedures in a quite distinct way, looking for tasks which have certain directly identifi-

*Correspondence to: A. Höskuldsson, IPL, Building 358, Technical University of Denmark (DTU), DK-2800 Lyngby, Denmark.
E-mail: ah@ipl.dtu.dk
†Dedicated to Professor John F. MacGregor: a pioneer of multivariate statistical process control and recipient of the fourth Herman Wold medal.

able—and practically useful—purposes. For example, we may need to calibrate an on-line measurement instrument, predict a quality parameter, evaluate a new sample or monitor/control a complex production process, etc. If the data-modelling procedure used fulfils the practical task at hand, the concern is not focused on the statistical assumptions of the procedure. If for example the procedure used requires that assumptions of multivariate normality be fulfilled, we can of course test for multivariate normality (and statistical data analysis program packages are flush with tests of this kind); but what if the requirements are *not* fulfilled? In spite of the results of such tests, very often the specific procedure is used anyway, because *it works* in the industrial context.

1.3. On-line measurements—increasing responsibilities

As a generic example from the process industries, the Toyota automotive company has the motto '*all components* are to be *measured*'. Car makers have had great success in securing that all components have absolutely correct dimensional tolerances—for obvious reasons. Very many other types of companies are now also investing in similar on-line measurement equipment. Another important aspect is that experience has shown that it is important to detect as soon as possible when a process has gone off-spec or become defective. Industry is increasingly interested in methods that can be used on-line for *optimal* real-time process monitoring and control. Clearly such approaches are both data-intensive as well as heavily computer-demanding. From the point of view of today's computer facilities it is fully possible to set up the necessary advanced procedures for monitoring and control of industrial productions. Thus many companies within—and outside—the process industry sectors have made large investments with the purpose of automatic process control. Industry is demanding more from the present monitoring and control technologies. When a construction engineer computes the strength of a specific construction, his/her company relies upon the results of the method employed. Similarly, when decisions are made on the basis of data modelling, one needs to be able to rely on the recommendations of the software engineer or data analyst who has carried out the analysis.

1.4. Latent data structures

Today's industrial data are always multivariate, not only representing many variables related to each instrument (sensor), but also in the sense that it is necessary that a wide series of measurement points cover the entire process. An emergent concept of whole-plant coverage can be clearly seen. Typically this kind of variable set is invariably highly correlated. It will not be possible to construct effective monitoring systems which are based only on specific, selective signals. On the other hand, there will always be a massive redundancy in these types of measurements. This implies that we usually cannot find a simplistic data model which includes all the measured variables or where we can work with each variable separately. Instead we must identify the *latent data structures* and explain changes in the monitoring and control signals in relation to this structure. One of the most important guiding principles of chemometrics is the imperative of visual inspection/understanding of the complex data structures involved (scores, loadings, residual variances, etc.). Clearly the latent variable concept comes to the fore with particular force here. This concept is of course the very essence of chemometric bilinear data modelling, which needs no further introduction in the present context.

2. A CURRENT TECHNOLOGY CHALLENGE

It will be instructive to look closer at how companies are using, or planning for, on-line data collection and administration. We shall look at a current example from one specific company and at the tasks facing the responsible process engineers and data analysts. The organization of data collection is shown in Figure 1 (the illustration has been stripped of proprietary specifics).

At any given time there are around 30 000 units in production or in the inventory. For each unit the value of the raw material and labour costs comes to between 500 and 5000. Each unit is to be furnished with an MSM mobile phone. The status of each of the 30 000 units is reported through the GPRS net and the Internet to the M2M host. The status and production information is stored in the M2M database. In the SAP R/3 database there are approximately 12 000 data tables, necessary for administrating the production. These describe the production conditions at each step. The 'users' can supervise the production and inventory at any stage, from any point. The customers can see the status of any-and-all orders, etc. Workstations are designed for the floor-workers.

Each of the 12 000 tables is a possible $\mathbf{X}$ data matrix. The corresponding $\mathbf{Y}$ matrices relate to various quality indices which are reported by the MSM clients. Besides analysing each product, there is a need for numerous supervising tasks that can be translated into data analysis tasks of the type presented in the present and the sequel paper. The major issue here is not only the massive data volume relative to ordinary chemometric standards, but also that the data structures met with take on a completely new complexity. There is not only a massive number of on-line measurements from a concomitant large number of variables, there is also a vast number of individual simple PLS regression relationships, or between overlying block or hierarchical relationships—all of which is fairly standard except for the sheer size and volume. However, there can also be seen principally whole new types of more complex block, hierarchical and recurrent looping relationships. The need for more general data structure-modelling tools should be clear. It could perhaps be expected that whole new data-modelling principles would also be called for in this context, but we have chosen to stay within the broad confines of chemometrics and bilinear modelling in what follows. It would appear that it is fully possible to augment the current data analysis potentials—almost at no end—without straying from these familiar territories.

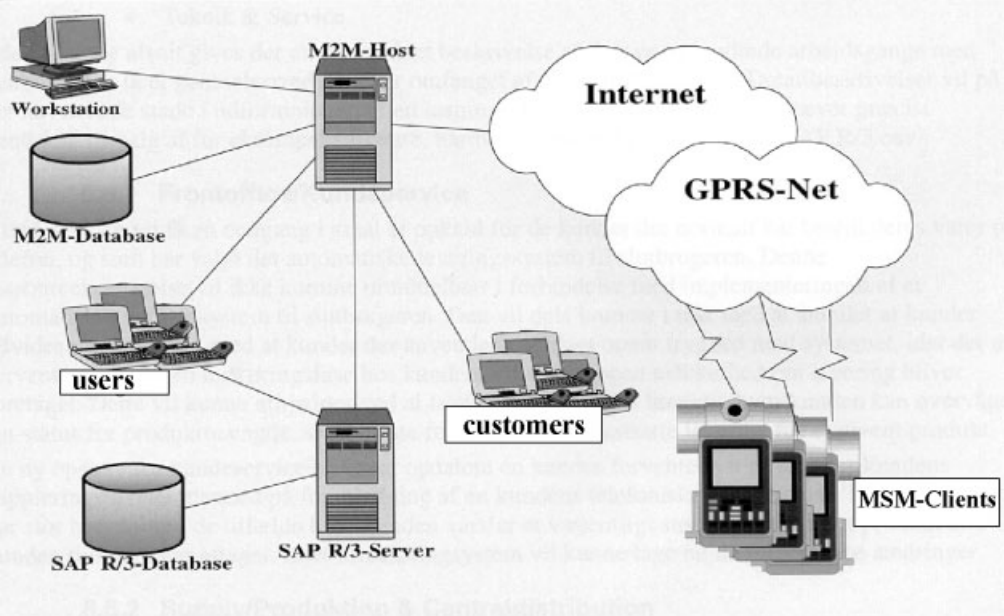

Figure 1. Large-scale data collection and organization. A real-world example showing the need for more evolved and comprehensive data modelling principles and methods.

3. NEW REQUIREMENTS FOR DATA ANALYSIS—AND OLD INVARIANTS

In developing such new data analysis approaches, there are a few prerequisites which are invariant and which need only be briefly commented upon.

It is important to distinguish between *redundancy* in data and *no information*. Redundancy in data appears as variables, or measurements (objects), which are highly correlated. It is often advantageous to compute *score vectors* on the basis of such highly correlated variables. Score vectors have very useful averaging properties over the set of many correlated variables and consequently acquire large 'sums-of-parts' sizes in contrast to the individual variables. They are thus *reliable* for predictions. No information in data means that there is no correlation between the specific parts of data we are modelling and/or the so-called residual complements.

It is critically important to detect *outliers* in calibration data as well as in new measurements, because of the intrinsic least squares properties of bilinear modelling. Therefore it may be necessary to store more data than actually used in the contemporary modelling procedure in order to be able to evaluate new measurements in relation to the entire, relevant data history. New evolving features in the data structures often imply that modelling fit and/or prediction errors change in quantity or quality. It will always be necessary to *update* models—it is only a matter of the time periods involved in which any particular model will remain stable.

Similarly, when there is an underlying trend (drift) in data which has not been picked up in the model, it will of course be necessary to revise the model, etc. We can detect changes in model parameters by many means, but usually some form of validation is employed, e.g. test set validation, cross-validation, etc. For process modelling, an often-invoked *scheme* is that of estimating the parameters for the first, say, 50% of the data and then for each new sample of similar size. Alternatively we may wish to re-estimate the model parameters at regular intervals (as measured in *proper process time*, not in chronological time), or we can detect changes in the model structure by examining the residuals. If the residuals e.g. stop displaying the signature of 'random noise', this would indicate important changes in the model structure, etc.

One of the most important tasks in modern industrial process analysis is to be able to detect changes in the mean levels of key informative variables. Univariate statistical process control (SPC) used to be the industry standard; today, MSPC (multivariate SPC) is increasingly substituting for more involved tasks. *Process analytical chemometrics* looks to be a good front-line prospect for many years ahead.

Even when employing latent variable descriptors, it is often still a challenging task to detect changes in the levels of the individual variables. Recent developments within process chemometrics have resulted in useful, very user-friendly graphic solutions and software, however, which are now typically developed in collaboration with the responsible process and production engineers. It is more and more accepted by all parties that it is distinctly advantageous to look *both* at the task-related score vectors and at the residuals involved, as these two descriptors reflect the complementary process-related phenomena (model fit) as well as the errors (model misfit) involved.

3.1. New approaches to process data modelling

In chemometric data modelling, we often either focus on one block of data (**X**) and carry out a typical PCA type of analysis, or we are interested in two data blocks (**X**,**Y**), most often in the context of multivariate regression analysis. Chemometrics must be fully aware of the existing body of very accomplished and powerful systems analysis, process-monitoring and process control disciplines. Recently an overview was published taking on the entire established

statistical theoretical and practical context, in a comprehensive reinterpretation of the essentials of the chemometric approach, constructed around a novel compound optimization concept, the *H-principle* [1]. This *unified* approach may be carried over to dynamic data modelling only to some extent, however. A comprehensive attempt at bridging the gap between these fields and chemometrics was published recently, termed 'dynamic systems multivariate calibration' [3], written from the dynamic systems point of view. These theoretical developments are aimed directly towards practical industrial usefulness, however, and lie in this fashion also squarely within the chemometrics playing-field.

By contrast, a theoretical scientist often describes changes in form of one or more differential equations, signifying a simplifying law-like relationship, often expected to be valid over the whole range of possible measurement values. In industrial environments it is usually unrealistic to expect the data to follow a set of differential equations, because the situation is usually much more complex than in the typically controlled experimental laboratory set-ups. However, one would still like to be able to carry out an analogous data modelling, typically described by a number of 'process stages', and to study the *changes* from stage to stage. One powerful approach is *path modelling*.

There is thus an emerging challenge in trying to encompass all of the above with just one set of overlying meta-principles for data modelling, which is the precise task of the present paper.

4. THE CONCEPT OF *OBJECT ORIENTATION*

The software industry has adapted the ideas of *object-oriented programming* with enthusiasm. The basic concept of object orientation is to look at a programming task as an independent *object*. The technical details of the task are implemented in the object itself. When the object is used ('called') in a program, the object is initiated by a declaration. Thereafter we can freely use the different functions belonging to the object. The idea of object orientation is to put together *everything* that is needed (data values, parameters, functions, links to other objects or functions) to complete the specific task. When an object is initiated, the program typically uses only a few of the totality of available features of the object. When an object-oriented program is executed, it will often take up a lot of RAM, because many objects may have to be initiated, although the task itself may be small. Several related objects may be organized or derived from a single *class*. A class typically describes an overall given task. It contains data and shows how the individual objects are organized. There are many important features included in object-oriented programming. An example would be *polymorphism*, where methods and functions are *context-sensitive* in the sense that their behaviour may depend on their specific usage. Different functions may have the same name, but the one chosen in an actual case is dependent on the contextual usage. *Object-oriented programming* (OOP) is now a generally widely accepted programming methodology. By utilizing objects, the programmer has to ask questions of the type: what should be *included* in this object if it is to function according to its primary as well as its secondary, tertiary objectives in this and that *context*, etc.

There is much more to be said about object-oriented programming [4,5], but suffice here to state that the results have been very successful in the form of well-tested and more *reliable* software applications and programs, both because of the ingenious object orientation itself and because of the inherent obligations to think deeply about the complex sets of tasks to be faced, indeed *well before* any programming can start.

The above OOP paradigm has inspired us to conceive of a new concept of the 'data analytical object'. We are looking for a data analytical *unit operation* which ideally should be so general that all that is needed in order to cover any data analysis task is the way in which these unit operations (*OOP objects*) are to be interrelated. While this object analogy is to be developed as completely as possible, the data analytical, task-specific interrelations are always closely *problem-dependent*. Thus the analogy with object-oriented programming cannot necessarily be expected to be fully complete.

4.1. Contents of an 'object'

- Start the object with the relevant parameters.
- Initialize the object (initialize internal parameters, carry out consistency checks, etc.).
- Parameters as an integral part of the object.
- Inheritance and 'friendship'.
- Local functions.
- Visibility of parameters/functions.

5. *TOWARDS* A NEW DATA ANALYSIS OBJECT-ORIENTED APPROACH

Classical statistical methods are all based on the *strategem* 'model → compute → result'. This is a direct, deterministic approach which presupposes that one knows that the model will be more or less in principal correspondence with one's data analysis problem. Classical statistics works on the premise of a set of well-understood and well-characterized distribution models. One also needs to know the task flow sheet thoroughly before one can program the task, etc. The only 'problem' is that there would appear to be a virtual plethora of 'methods' pertaining to each of the seemingly isolated, non-overlapping different data-modelling *objectives*, e.g. discrimination, classification, regression, time series analysis, forecasting, etc. [6].

If we look at data modelling in a similar way to object-oriented programming, there are still many familiar traditional issues involved, but they may now also be appreciated in a distinctly new fashion. We shall consider some of these central issues below in a bird's-eye perspective in order to establish a first framework for this 'new way' to view data modelling.

5.1. Validation

Whenever data modelling is carried out, a proper performance validation is *mandatory*. Validation should always, as far as practically possible, reflect the *future situation* in which the data model is supposed to function—often termed the

'test set' context. This is termed proper external validation ('external' in relation to the training modelling activities). In many situations in which automatic process control is on the agenda, there may be no time or no possibility for this approach, however, in which case some form of internal validation (cross-validation, etc.) must be invoked. The validation issue is covered in more detail elsewhere [2,7].

5.2. Warnings

Special features in the actual data structures (as opposed to the model assumptions) that might *invalidate* the modelling results *must* be detected. There is a need for proven and completely reliable outlier and upset detection methods. It is important to work out completely reliable procedures for detecting outlier cases, as these will wreak havoc with any multivariate modelling. On the other hand, it is equally disastrous to base faulty outlier detection on an invalid data analysis rather than on an actual outlier control signal.

5.3. Expert validation

When a domain expert looks at the results of a particular data analysis, he/she may have some important 'ex-data analysis' opinions as to the quality of the results. It would be very useful for the user if such expert opinions could be formalized as concrete measures, e.g. correlation coefficients, scores in sensory profiling, etc. There are many ways to go about this endeavour, but all are by necessity closely problem-specific.

5.4. Problem-specific model reliability

In industrial implementations it is often a critical issue whether the data model is appropriate for all parts of the data. For example, if a model is established from a random half of the data, a *reliable* model should produce closely the same model parameters if it is applied to the other half. This 50% sampling may relate to e.g. chronologically sampled data or spatially sampled data, or it may be expressed e.g. with respect to *proper process time* or some other relevant process reference. The simple 50/50 internal division is of course only one of many possible cross-validation *model reliability scenarios* which can be envisioned, but all again hinges on the specific, practical situation context.

6. THE H-PRINCIPLE

The H-principle, or the *Heisenberg principle of data modelling*, has been formulated as a basis for general prediction data modelling [1]. The basic idea is to carry out the modelling in sequential steps or stages. At each step a balance is sought between the estimation and prediction optimizations of the modelling procedure being used, as prescribed by a suitably chosen optimality criterion [1]. In practical terms this means that we should scale data appropriately in relation to the specific data modelling being used, and for the scaled data we should optimize the covariance at each step. We stop modelling when we can no longer detect any significant covariance. A series of exhibitions of these general data-modelling principles and their specific manifestations have been published in the literature [1,8–10].

6.1. One data block

Suppose that we are given a singular data block $\mathbf{X}$. In general we want to find a *score vector* $\mathbf{t}$, given by $\mathbf{t} = \mathbf{X}\mathbf{w}$, subject to some pertinent optimization criterion. If there are no special preferences, the consensus recommendation is to choose $\mathbf{w}$ such that the score vector has maximal variance, i.e. to maximize $(\mathbf{t}^T\mathbf{t})^2$. In other words, we are to find a weight vector $\mathbf{w}$ of unit length that maximizes the (squared) length of $\mathbf{t}$. This procedure leads to the well-known principal component analysis (PCA). For the sake of a unifying systematic terminology, to be used below, we may also choose to speak of this as maximizing the 'covariance' $(\mathbf{t}^T\mathbf{t})^2$, and thus include PCA in the context of 'prediction data modelling' without risk of confusion; see further below.

6.2. Two data blocks

Suppose that we want to describe a data block $\mathbf{Y}$ by another data block $\mathbf{X}$, e.g. as a regression analysis, $\mathbf{X} \rightarrow \mathbf{Y}$. One approach would be to ask for an X-weight vector $\mathbf{w}$ and a corresponding Y-weight vector $\mathbf{q}$ such that the associated ($\mathbf{X}$, $\mathbf{Y}$) score vectors ($\mathbf{t}$,$\mathbf{u}$) have maximal covariance:

$$\text{maximize } (\mathbf{t}^T\mathbf{u})^2; \qquad \mathbf{t} = \mathbf{X}\mathbf{w}$$
$$\text{and } \mathbf{u} = \mathbf{Y}\mathbf{q}, \text{ with } |\mathbf{w}| = |\mathbf{q}| = 1$$

This result is of course the well-known PLS regression (PLS-R). This result may easily be generalized.

6.3. Three data blocks

The need for generalizations of the basic two-block PLS layout occurs often, as soon as more complex data structure models appear on the agenda. References [1,8–10] describe the necessary technical background. Suffice here to focus on three data blocks, each carrying specific information (in the $\mathbf{X}$ vs $\mathbf{Y}$ regression sense, etc.), which cannot be concatenated (e.g. $(\mathbf{X}, \mathbf{Y}) \rightarrow \mathbf{Z}$).

Thus suppose that we are given three data blocks $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$, each having the same number of rows (objects). Consider three potential score vectors $\mathbf{t} = \mathbf{X}\mathbf{w}$, $\mathbf{u} = \mathbf{Y}\mathbf{q}$ and $\mathbf{v} = \mathbf{Z}\mathbf{r}$, one from each data block. If for example we want to know how the $\mathbf{X}$ *and* the $\mathbf{Y}$ block describe the $\mathbf{Z}$ block *simultaneously*, one possible optimization criterion would be to find weight vectors $\mathbf{w}$, $\mathbf{q}$ and $\mathbf{r}$ of unit length such that $(\mathbf{t}^T\mathbf{v})^2 + (\mathbf{u}^T\mathbf{v})^2$ is maximized. However, one could also have chosen alternative criteria. It is clear that the *choice* of which optimization criterion to use is intimately related to the known or expected 'data path' characterizing the three-block set-up. With three blocks we face for the first time this multiple-choice alternative regarding which compound optimization criterion to use.

6.4. Multiblock data modelling

For multiblock (>2 blocks) data scenarios there is no trivial solution to the data-structuring problem; it is not always appreciated that even in comparatively simple situations there is no single, solitary 'solution' which only needs to be implemented—indeed, here we meet the all-important 'problem-specific' characterization at full force. Thus multi-block relationships *must* have very problem-specific prescriptions concerning which score vectors are related to

which across the block boundaries, etc. In this situation we may thus use similar sums of squared covariances to measure the strength of the modelled relationship, as was indicated above. If at some stage in this procedure there is no covariance left between any two data blocks, we must stop modelling this particular relationship, etc. This procedure was the one adopted by Herman Wold in his modelling of causal structures—also known as *path modelling*.

7. WEIGHTING PROCEDURES

7.1. Internal weights

A score vector $\mathbf{t}$ can be computed as $\mathbf{t} = \mathbf{X}\mathbf{w}$, where $\mathbf{w}$ is a *weight vector* of length one in a specific data-modelling context. In PCA, for example, $\mathbf{w} := \mathbf{p}$. In the regression context it is important that the computed score vector describes *both* $\mathbf{Y}$ and $\mathbf{X}$. Although the primary interest is often $\mathbf{Y}$ (e.g. $\mathbf{Y}$ is desired to be predicted), the score vector should still be *reliable* in the sense that it should also reflect a maximal part of $\mathbf{X}$ for stability and reliability reasons.

Similarly, a corresponding loading vector $\mathbf{p}$ is typically computed as $\mathbf{p} = \mathbf{X}^{\mathrm{T}}\mathbf{v}$, where $\mathbf{v}$ can also be viewed as a similar weight vector of length one, or—for the most well-known chemometric methods (PCR, PLS-R)—the lengths of these score vectors are not normalized, e.g. $\mathbf{v} := \mathbf{t}$ or $\mathbf{u}$, etc. In general we may look at these $\mathbf{w}$ and $\mathbf{v}$ variables as *internal* weighting operands, in the sense that they are wholly derived internally as part of the algorithms performing data modelling of one particular data structure pertaining to the corresponding matrices $\mathbf{X}$ and $\mathbf{Y}$.

In the case of one data block the stepwise computational scheme makes for the so-called *updating* (or *deflation*) stepping-stone to the next stage: $\mathbf{X}$ is reduced by rank one in the relation $\mathbf{X} \longleftarrow \mathbf{X} - d\mathbf{t}\mathbf{p}^{\mathrm{T}}$, where d is computed as $d = 1/\mathbf{t}^{\mathrm{T}}\mathbf{v}$. Similar deflation is usually also employed in the context of regression modelling, although there are contemporary theoretical discussions within chemometrics as to the advantages of this. This difference of opinion is of no substantive relevance for the present developments, as it pertains only to the technical computation issues.

7.2. External weights

In practical implementations (in industrial or other contexts) it is often relevant to allow for differences in the 'quality' of different samples. Such a quality notation will be closely problem-dependent. In many situations, domain experts can in fact assign very meaningful quality measures (absolute or relative), e.g. in the sense that samples which are characteristically close to a specified optimal product quality are more 'important' than other samples. This type of external weight is tentatively very 'subjective', but also a very relevant feature in many situations. It may in fact be quantified in an *external* object weight vector $\mathbf{r}$. As will become apparent shortly, there are many situations in which a problem-relevant *updating* of such an external weight vector may be required, whether automatically or as a result of operator intervention, etc.

The first option, automatic updating of external weights as part of the data modelling, will play an integral role in the new data-modelling approach we present below.

8. H-OBJECTS

The introduction of the *H-principle* [1] was the first time that chemometric bilinear data modelling was augmented by this optional external object weight concept. We shall now propose another related feature, the *H-object*, which we shall introduce by way of an illustrative example.

H-object. Together with matrix $\mathbf{X}$ there are now always associated a *pair* of external weights ($\mathbf{r}$, $\mathbf{s}$) ($\mathbf{r}$, object weight; $\mathbf{s}$, variable weight), prescribing how the $\mathbf{X}$ rows (objects) and the $\mathbf{X}$ columns (variables) should be weighted respectively.

$$\text{The H-object is the } \textit{triplet } (\mathbf{r}, \mathbf{s}, \mathbf{X})$$

The H-object is the weight-augmented data matrix $\mathbf{X}$, or some similar data matrix $\mathbf{Y}$, $\mathbf{Z}$, etc. Note, however, that it is not always needed that these external weights be used—it is simply their *potential* use which allows for (far) more complex data modelling.

Illustration. In on-line environments a data model per force has to be calibrated on the basis of historical data. At any one time there may only be the accumulated historical array of data upon which to calibrate the data model in question. As more and more new data arrive, however, in principle, better contemporary knowledge about the system is obtained. As time passes, there may now be new requirements to the model, which may or may no longer be satisfied for the data in question. Specifically, there may now be information as to which of the objects carry the most valid, relevant, important information, etc. This is precisely the kind of information which may be encoded into an external, *dynamically updated* $\mathbf{r}$ vector.

The corresponding variable weight vector, the $\mathbf{s}$ vector, is well-known in chemometrics. In a typical static context, $\mathbf{s}$ can take any number of *preprocessing forms*, e.g. the all-important standardization (1/std), or a transformation form, e.g. log($\mathbf{X}$), etc. Note that for such simple applications the *action* of the external weighting operands is carried out once and for all; that is, once the $\mathbf{X}$ matrix has been scaled or transformed, this action need not be repeated, since the otherwise standard chemometric methods will simply be performed on the transformed ($\mathbf{X}$, $\mathbf{Y}$) matrices. This global multiplicative weighting is well known to the data analysis communities, and we shall not be interested in these aspects in the discussions further below. Instead we shall devise a more dynamic 'soft' approach to invoke the same type of 'external' weighting information, but in a distinctly new fashion, the RSSA information exchange agent (root-sum-of-squares averaging).

9. H_0: THE *GENERALIZED* PLS OBJECT

9.1. External H-object weights (s) as data path 'gluons'

We shall now present a self-contained and comprehensive context in which to conceptualize general data structure models in the form of interrelated, or hierarchical, H-objects. This concept is general enough to encompass PCA, PLS, multiblock PLS and hierarchical multiblock PLS but is mainly aimed at extending general data modelling beyond these established methods. It also encompasses some three-

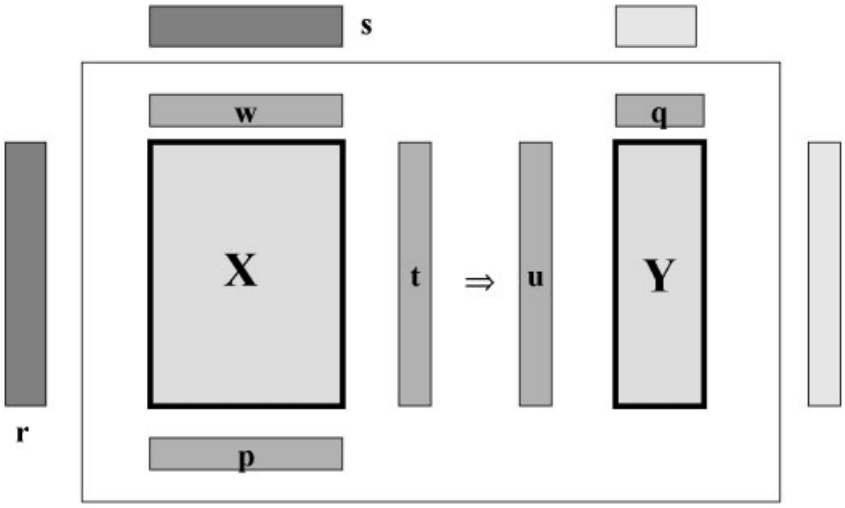

Figure 2. The generalized H_0 object (the full PLS/PCA object). H_0 is *always* in principle assigned a pair of *external weights* (**r**, **s**), whether to be used actively or not in a specific situation; see text.

way and multiway data-modelling aspects [11]. The general concepts of the new H-object approach may—*presumably*—be generalized also into these multiway regimes, along the same principles as presented by Esbensen *et al.* [6] in which the data array organization was generalized. See also the seminal parallel development overview on multiway modelling by Alsberg [12], from which we have also been greatly inspired.

Reference [1] described systematic bilinear data modelling for two-way matrices. Specifically, it was explained how PCA may also be viewed as but a marginal case of two-block PLS regression modelling. From a purely data-modelling point of view we may opt to view PCA as a special $\mathbf{X} \rightarrow \mathbf{X}$ regression modelling case in which the covariance maximization takes the special form of maximizing the 'covariance' $(\mathbf{t}^{\mathrm{T}}\mathbf{t})^2$. Thus, without loss of generality, below we shall speak mostly in general terms of 'PLS regression modelling', keeping in mind this special 'PCA-R' as a specific possibility if the data-modelling objective so demands.

The new 'unit operation' of general data modelling will thus be the standard PLS regression data model, but *augmented* as the complete H-object triplet (**r,s,X**)—indeed, also the corresponding (r,s,Y) if need be. This basic unit is termed the H_0 *object* (Figure 2).

The above situation in which was illustrated the need for updating of these weight vectors can now be illustrated in terms of a *data path graph* with the corresponding terminology 'external updating of the H_0 **r, s** vectors' (Figure 3). In this particular case the updating could be automatically performed, or it could be related to some monitoring and/or user-defined activity. Figure 3 particularly lays the foundation for *external updating* of the H_0 **s** vector, which is going to play a central role in what follows. Note how the usual company of PLS weights and scores **w**,**t**,**p**,**u**,**q** all reside *inside* the H_0 object at all times, i.e. the conventional PLS algorithm runs according to all standards. The only difference—so far—concerns how the external weight information is to be invoked in the internal H-object algorithm.

The objective of inducting external information, e.g. in the form of a particular **s** vector, is to establish a basis for *guiding* the internal data modelling—guided by information from other objects. In its simplest form, such guidance could be in a similar fashion as pertaining to the **w** vector in ordinary PLS regression, e.g. $\mathbf{t} = \mathbf{X}\mathbf{w}$. Thus one possibility would be that **s** (or **r**) may represent the internal weight **w** (or **t**) from 'external' H-objects, i.e. from a separate bilinear H-object pertaining to *another object* (which in its simplest manifestation could be just another variable *block*). It is easy to envisage how one may build up *interrelated sets* of such H-objects, 'pathed' by a specific data analysis objective. This data path always needs to be defined *before* it is possible to consider how to interrelate the blocks involved.

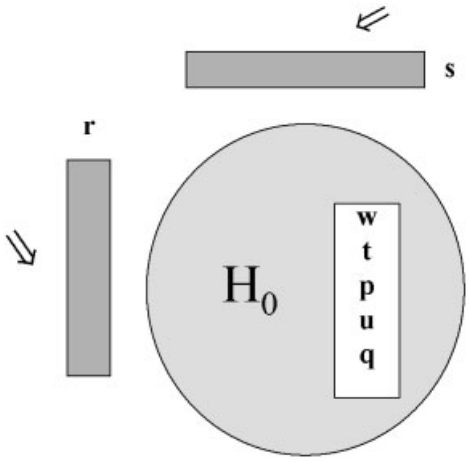

Figure 3. Illustration of *internal* versus *external* H-object parameters. The external weights (**r**, **s**) are the *only* carriers of information to be inducted into the object. In OODM this can *only* happen via the 'soft' RSSA procedure (see Box 1 and text for details).

9.2. Constraints in the standard multiblock concepts (path modelling)

Consider the situation in which we have a data path consisting of two H_0 objects (two simple PLS-R as it were), each complete with their respective internal scores (**t**, **u**) and loadings (**w**, **p**, **q**) as well as their corresponding external variable weights **s** (Figure 4). In the chemometrics literature so far, all interrelations between 'data blocks' (*path modelling*) have *exclusively* taken place as direct *interchanges of score vectors* (**t**, **u**), or loading vectors (**w**), between blocks. The point here is that in both cases of multiblock PLS generalizations the operative interchanges occur *only* via *direct* score/loading vector interchanges from one internal algorithm to the other, and *vice versa*. After one of these types of 'external' weight vectors is received and substituted for the pertinent 'internal' counterpart, the particular internal algorithm continues according to the standard PLS *scheme*.

In these previous endeavours, however, there are two characteristics which would appear to constrain more complex data modelling rather severely: (1) *only* direct score/loading vector interchanges are invoked; (2) all generalizations presented address only a very specific and limited data-modelling objective, e.g. multiblock PLS (interrelated via object scores) or the 'symmetric' variable multiblock PLS (interrelated via variable loadings). All these direct vector exchange multiblock schemes were recently presented in one *unifying* theoretical context [13]. There have also been presented attempts at dealing with what has been termed 'hierarchical PLS', employing a multilayer, multiblock organization, but again relying on interchanged or concatenated score or loading vectors upon which a 'meta-PCA' is carried out, or a 'meta PLS'. Thus in all these generalizations the information flow between blocks only takes place via interchanged weight vectors (scores/loadings).

However, if there is no objective data analytical correspondence between any two or more blocks, the entire compound data path model breaks down, because the empirical data in fact constrain the specified meta-model in a very restrictive fashion. The inheritance from earlier statistical data modelling, 'model → compute → result', is clear. We are still in this particular sense in a 'fixed model' situation. Here we do not wish to pursue this avenue further, however.

In the new object-oriented approach we now offer the alternative possibility of *external information exchange* between blocks via the external H-weights (**r**,**s**). This is crucial for understanding what comes e.g. in relation to hierarchical H-object loops.

H-objects can be organized freely in any pertinent fashion, path modelling and hierarchical organization being only the two most obvious. *Any* (more complex) data path interrelationship can be modelled by a *connecting graph* of the type indicated in Figure 4, either via direct score vector information transfer (as before) or via the new *dynamic* information transfer concept RSSA.

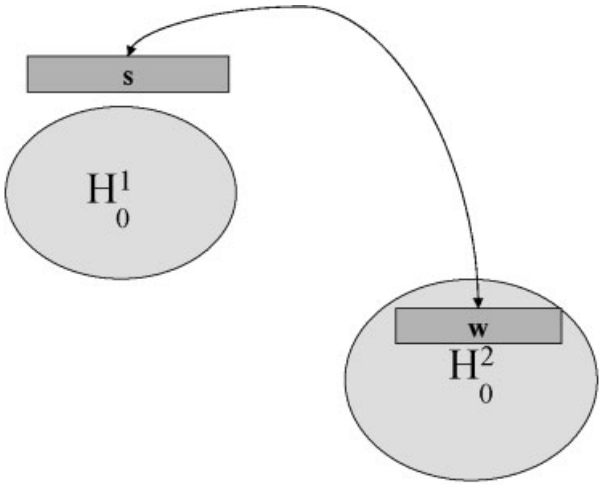

Figure 4. Illustration of a specific *data path* consisting of two objects. In the situation depicted, an internal weight **w** from H^2_0 is *connected to* H^1_0, where it is functioning as an external **s** weight (in contrast to the alternative *direct exchange* as the equivalent **w** vector in the H^1_0 object).

9.3. RSSA: root-sum-of-squares averaging

Box 1 illustrates the inner workings of RSSA. The point of departure for RSSA is where two vectors (of equal dimensionalities) exist which are to be 'combined'. The combination is based on the familiar RMS concept, but in a slightly modified fashion. In Box 1, **w** represents the internal weight, while **s** represents the external counterpart. We only develop the RSSA facility for **w**-like weight vectors for reasons which will become clear immediately.

Case 1. Identical signs, s_i and w_i, $\forall i$:

$$W_i(*) = \mathrm{sign}(W_i)\sqrt{(S_i^2 + W_i^2)}$$

Case 2. Mixed signs, s_i and w_i:

$$W_i(*) = \mathrm{sign}(W_i)\sqrt{\max(S_i^2, W_i^2) - \min(S_i^2, W_i^2)}$$

w:(*) is *normalized* to length one.

Box 1. RSSA 'soft' combination of two weight vectors, one 'external', **s**, and one 'internal', **w**. The new combined **w(*)** weight vector is *normalized* to length one before it is returned to the two parent H-objects. Compare Figure.

RSSA is designed to allow the larger (absolute sense) **w** elements to play a more dominating role than their smaller counterparts; hence averaging of the sum-of-squares expressions is not pre-divided by two, but is carried out by taking the square root directly of the sum of squares—followed by normalization to length one. This new 'soft' combination objective allows that the *derived* RSSA vector is substituted for *both* the external and internal weight vectors in their next respective iterations in their separate H-object contexts. Hence that which is returned to both H-objects for one more internal iterative run is *the same* **w**(*). This updated **w** weight vector will now guide the resulting data decompositions differently according to the individual empirical data structures in these two separate H-objects, i.e. the RSSA **w**(*) will take part in the otherwise 'standard' internal H-

object next iterations. Another iteration results in a new **w** vector, which is again *returned* for a new RSSA combination iteration, etc. This next RSSA pass will reflect a more covariant interconnection between these two H-object data structures, depending on the objective between-block data structure (correlations) present in the two H-objects.

RMSA is designed to open up a 'soft' information transfer capability, which is impossible with direct ('hard') vector substitutions, in which an internal vector is substituted *in toto* by the corresponding external counterpart, and *vice versa*. With the RSSA concept the external and internal vectors both contribute on an *equal footing*. The 'star-vector' represents the RSS-averaged go-between between an internal and an external loading weight. This serves as a new way to derive a compromise between the two data structures. *If* the two data structures correspond well to each other, if indeed the expected data path corresponds to the objective correlation to be observed between the H-objects, the RSS averaging will *converge* to a stable **w**(*), which will then represent both data structures equally well. If/when convergence is achieved, i.e. when stability of the RSSA star vector is obtained, this vector performs much as if an ordinary direct weight vector interchange is taking place. This stable end-result of the inter-object RSSA calibration may then be used as the pertinent **w** loading weight for both objects, etc. There is thus no problem with the RSSA concept *if* a specific simple multiblock or hierarchical data model assumption holds up to the data reality.

However, the benefit of the RSSA transfer function capability perhaps shows itself even more clearly in the situation(s) in which there is less direct transferability of the between-block information to be found. For the sake of argument we shall assume that the well-known NIPALS algorithm is being used for all internal H-object PLS. For each iterative loop in the *separate* H-object algorithms, both the internal and external weight vectors are *dynamically updated* by the RSSA procedure. This means that neither of them passes back in their respective next NIPALS iterations without this updating. They may be greatly or less affected or they may be only insignificantly affected (i.e. close to or at convergence/stability). This ensures that the internal data structures as well as the 'between-block' data structure are now *equally* respected in each iterative loop in *all* objects (two or more; see below), provided that there indeed is a common intercorrelated between-block data structure. However, in the event of a significant degree of *mismatch*, this is immediately reflected by the RSSA procedure, which will simply grind to a halt in the case of no objective compatibility between the inner and outer vectors (there will be no convergence no matter how many iterations).

Thus whether the RSSA process is succeeding or not is a good measure of the *objective compatibility* of the two (or more) data structures involved. This means that the simple RSSA information transfer facility will suffice to reveal whether there is agreement between the empirical data structures involved and the *a priori* specified path model. In either event (success or failure) this will furnish a very important consistency check of the *a priori* established path model. Indeed, it is precisely this degree of possible mismatch which is the quality left out by all the previous 'fixed' vector exchange block specifications described above.

9.4. OODM—a new data-modelling pathing platform

By invoking the RSSA procedure for 'soft' between-block information transfer, facilitated by (**s**,**w**) weight vectors, while making full use of augmented generalized PLS H-objects as the data analytical unit operation, it is now possible to depict a wider scope of data model complexities in which the pertinent data paths can be expressed by RSSA **s** vector *interconnections*.

So far we have only conceived of these three components in the data path schemes in object-oriented data modelling (OODM) (Figures 2–4). *All* types of complex two-way interconnected data modelling can be *graphed* within this new concept. It is highly likely, however, that only seldom (very seldom indeed) will all the (**r**,**s**) weights, potentially available for all (**X**,**Y**) matrices, have to be invoked *simultaneously*—i.e. many will take the 'neutral form' of suitably dimensioned 1-vectors in many specific situations—but the possibility of using the full apparatus of fully (**r**,**s**)-armed multiblock interrelated and/or hierarchical H-object data modelling is potentially very powerful.

9.5. r weight information exchange—an asymmetric issue

In the present new OODM context it is still required that N, P, Q be equal between all OOP objects which are to be interrelated (N: = number of measurements in **X**, **Y**; P: = number of variables in **X**; Q *ditto* for **Y**). It is well known that it is only within e.g. the NIPALS algorithmic calculation context there would *appear* to be a symmetric relationship between measurements (objects) and variables (carried over to scores and loadings/weights). Philosophically, as well as in practice, there is a world of difference between these categories, however [1,2,6]. This is e.g. manifested in the different transformation, scaling and normalization schemes used for objects and variables respectively. This fundamental distinction actually demands that the information transfer mechanism for **s** and **r** weight vectors also be different. It would be distinctly incorrect and futile to even try to apply the RSSA formalism on an external **r** vector and a pertinent **t** vector (internal score vector). Indeed, this would not even be possible, the reason being that scores (and score weights) are not similarly normalized. In fact, scores are *never* normalized to length one—at least in all standard chemometric bilinear data modelling (disregarding *analyse correspondence*, etc. for the moment). Normalizing would eliminate the entire information carried by the length of a score vector, something which would be akin to projecting the entire (measurement, object) data space onto a P-dimensional hypersphere—resulting in a tremendous loss of information [1]. For this reason we do not foresee RSSA being applied symmetrically in this fashion in OODM. Rather, the **r** weight information exchange issue needs to find its very own solution. Once-and-for-all multiplication is certainly one option, as is direct score substitution, but what form could one imagine for the pertinent **r** weight RSSA equivalent—if it even exists? This fascinating teaser we leave to the first

fellow chemometricians taking up the present gauntlet—it will not detract from the feasibility of the overall OODM proposal if such minor side-issues are not all worked out in every detail.

We can summarize the definition of PLS H-object data modelling as a new generic multipurpose concept for *graphic* data path modelling, in which the elemental building blocks can be expressed by two new elements and one traditional element: the unit operator is the individual, generalized PLS H_0 objects, *interconnected* by transfer functions in the form of dynamically updated external (**r**,**s**) weighting vectors. Specification of a data model always begins with the pertinent data path graph. This specification is very much analogous to using a visually oriented programming language, e.g. 'G' in the LabView context, only here we are not depicting the algorithm(s) at the programming level—we are depicting the overlying data path directly as a set of (two-way) matrices interconnected by RSSA (**s**) or other information transfer/substitution ties (**r**).

There also remains the interesting issue of exactly how to include the direct between-block vector substitutions in the simpler cases with *proven* data structure connections of the OODM concept. Likewise it may turn out that other 'soft' between-object information transfer relationships than our first foray, RSSA, will have to be studied. We shall further explore these concepts in more depth elsewhere. Our main thrust in this paper is only to point attention in the new ODDM directions, RSSA only being the first landfall in these new uncharted waters. We do not claim full conceptual, far less implementation, completeness at this early stage—much fascinating work remains. We believe, however, that chemometrics needs a thorough discussion as to the future data modelling *quo vadis*?—and thus present these reflections as possible stimulants for this debate.

10. *PATHING* COMPLEX DATA MODELS IN THE OODM CONTEXT

As an example of the ease of using this new graphic OODM, consider the intricacies involved in *generalizing* a hierarchical PLS objective. In the OODM context we may easily delineate any number of iterative calls of the basic H_0 object, successively termed H_1, H_2, etc. Note how each H_0 object can be outfitted with a full or reduced complement of the associated (**r**,**s**) weight vectors, fully as determined by the *specific* data analysis objective(s). Figure 5 shows this *complete* generalized two-level hierarchical facility: easy using OODM—not an easy thing to delineate in the case of a hierarchical nested set of direct substitutions.

The above is but one example of how it will be relatively simple to *graph* a complex data analysis objective by a data path. If for example there also is a need for an additional, *external* multiblock part, one possible solution will be by graphing in a 'direct connection' to another H-object at the appropriate level in the hierarchy, in this case as an **r** weight (perhaps acting as a **t** vector in direct substitution, perhaps used in a more elaborate way); see Figure 3. This combination of hierarchical and multiblock objectives may be delineated in any degree of interrelated complexity using this *OODM meta-programming strategy*. This is what we mean by stipulating that complex data analysis objectives will be far more easily grasped and mastered using this new *graphic meta-programming blackboard*.

Observe how there apparently is no limit to the complexity, i.e. the number of hierarchical levels, as well as the number of 'lateral' objects or blocks involved, which may be invoked using this systematic OOP-inspired approach; there is truly no upper complexity limit. Rather, there will (probably very quickly) crop up limitations more to do with

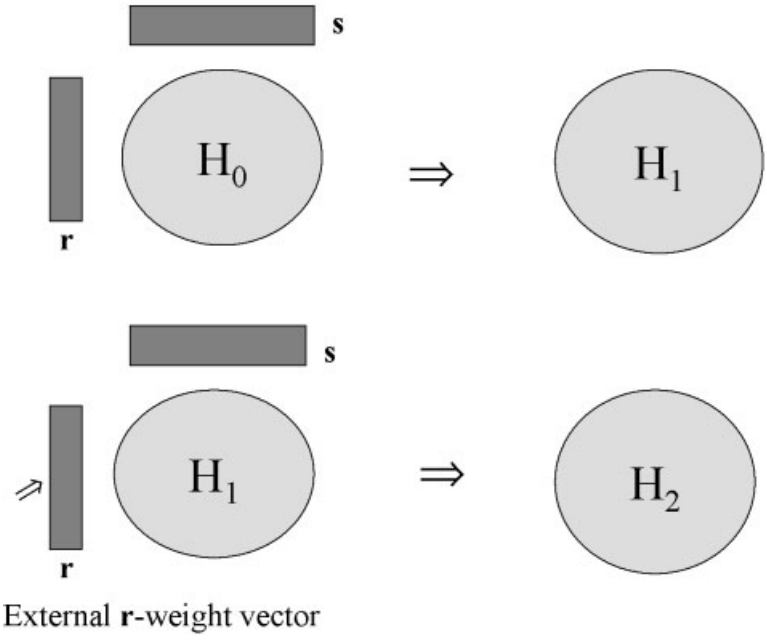

Figure 5. Hierarchical PLS illustration, in which is also shown one option for score information influx in the form of an external **r** weight (see text for details).

the conceptual faculties of the data analyst, or, for sure, with respect to specific software (and hardware) implementation constraints, etc. However, these are 'only' technical issues that sooner or later will be solved/solvable.

10.1. Resulting 'behind screen' programming tasks

Ideally, once the H-objects and their specific data analytical interconnections have been graphed properly (there are of course certain rules governing how these interconnections will have to be specified, etc., more of which will appear elsewhere; see again also Reference [12]), we envisage an off-screen *programming engine* which will translate the graph into executable code in some suitable language, C++ or similar. A good comparison would be the 'G' graphic programming language in the LabView system, which encompasses such a C++ translation facility, but the needs in the present scenario are probably somewhat more complex. Of course, it is not necessarily an easy task to translate pathing graphs to the executive programming level, but at least this can always be reduced to the relevant *interconnection tasks* only, because of the pre-existence of the basic H-objects, the *fully complete* H_0 objects in the fully established OOP sense. We fully expect that the analogy to object-oriented programming, while not entirely complete, will allow this task to be(come) doable in a very short time only.

It was noted by one of the referees that the analogy between objected-oriented programming and object-oriented data modelling is halting in some respects. In object-oriented programming there is typically a hierarchy of classes that make up the tasks that are available. In object-oriented data modelling, we are proposing new procedures for data analysis using concepts and ideas borrowed freely from object-oriented programming. However, it is beyond the scope of this paper to explore further the differences in detail—neither is it necessary. Much detailed programming work remains in working out exactly what can (and what cannot) be achieved within this new context. If the accumulated development history of chemometrics is anything to go by, these and related issues will first see their final solution when the software implementation is swinging into action.

10.2. Epilogue—a new data analysis freedom

The new OODM approach will offer the data analyst a new "graphic programmers' interface"—an *OODM blackboard* as it were—which is *invariant* w.r.t. updating, or even substitution, of the underlying programming language—at least from the point of view of the meta-programmer, i.e. the data analyst, who is the entity responsible for setting up the data path in the above sense, which corresponds to a particular data-modelling problem. This is exactly the answer to the emerging demands on data analytical modelling from industry or from academe which were specified in the introduction to this part I paper.

While it will be relatively easy to depict a multitude of such relationships in a graph environment, it is emphasized that what is presented is *not* a data-modelling amusement park to be used as a 'connector set' *without* a specific data analysis objective. What is presented here is a first description of a new generic, multipurpose concept for data path modelling, in which the elemental building blocks all are (**r, s**)-weighted H_0 objects, interconnected by 'soft' RSSA transfer functions (augmented by standard vector substitutions/exchanges and perhaps also yet-to-be-developed further **r** vector transfer protocols). We have here only been concerned with presenting the new conceptual object-oriented pathing *framework*—hopefully pointing towards a fruitful, continuously developing data analysis future.

In part II we shall give a thorough overview of the necessary further ingredients of object-oriented data analysis/statistical data modelling, together with objectives and methods, organized as a series of 'levels of modelling'.

Acknowledgements

Many were the fellow chemometricians whom we contacted in order to present nascent versions of the above ideas during several years—but very few were those who could see *any* benefit (at all) in discussing such 'esoteric matters'. It is therefore easy to acknowledge *Jim Burger*—a third partner on the path onwards—for carefully reading the whole manuscript and providing valuable improvements.

REFERENCES

1. Höskuldsson A. *Prediction Methods in Science and Engineering*, vol. 1, Basic Theory. Thor Publishing: Copenhagen, 1996.
2. Esbensen KH. *Multivariate Data Analysis—in Practise* (5th edn). CAMO ASA, Trondheim, Norge, 2001.
3. Ergon R. *Dynamic system multivariate calibration for optimal primary output estimation*. *DrIng Thesis*, HIT: Højskolen I Telemark (Telemark University College) & NTNU: Norges Teknisk-Naturvitenskapelige Universitat (Technical University of Norway), 1999.
4. Binck RV. *Testing Object Oriented Systems*. Addison-Wesley: Reading, MA, 2000.
5. Jacobsen I, Christenson M, Jonsson P and Övergaard G. *Object Oriented Software Engineering*. Addison-Wesley: Reading, MA, 1998.
6. Esbensen K, Wold S and Geladi P. Relationships between higher-order data array configurations and problem formulations in multivariate data analysis. *J. Chemometrics* 1998; **3**: 33–48.
7. Esbensen KH. Principles of proper validation and relationships with proper sampling—a tutorial. (*in prep.)*
8. Höskuldsson A. Experimental design and priority PLS regression. *J. Chemometrics* 1997; **10**: 637–668.
9. Höskuldsson A. The Heisenberg modelling procedure and applications to process control. *Appl. Math. Comput. Sci.* 1998; **8**: 755–773.
10. Höskuldsson A. The H-principle and non-linear models. *Chemometrics Intell. Lab. Syst.* 1998; **44**: 15–30.
11. Höskuldsson A. Weighting schemes in data analysis. *J. Chemometrics* 2001; **3**: 371–396.
12. Alsberg BK. A diagram notation for N-mode array equations. *J. Chemometrics* 1997; **11**: 251–266.
13. Höskuldsson A. Causal and path modelling. *Chemometrics Intell. Lab. Syst.* 2001; **58**: 287–311.

