Commentary

Impact of Data Base Structure in a Successful *In Vitro-In Vivo* Correlation for Pharmaceutical Products

B. Roudier,^{1,2} B. Davit,³ H. Schütz,⁴ and J-M. Cardot^{2,5}

Received 30 July 2014; accepted 30 September 2014; published online 12 November 2014

Abstract. The *in vitro-in vivo* correlation (IVIVC) (Food and Drug Administration 1997) aims to predict performances *in vivo* of a pharmaceutical formulation based on its *in vitro* characteristics. It is a complex process that (i) incorporates in a gradual and incremental way a large amount of information and (ii) requires information from different properties (formulation, analytical, clinical) and associated dedicated treatments (statistics, modeling, simulation). These results in many studies that are initiated and integrated into the specifications (quality target product profile, QTPP). This latter defines the appropriate experimental designs (quality by design, QbD) (Food and Drug Administration 2011, 2012) whose main objectives are determination (i) of key factors of development and manufacturing (critical process parameters, CPPs) and (ii) of critical points of physicochemical nature relating to active ingredients (API) and critical quality attribute (CQA) which may have implications in terms of efficiency, safety, and inoffensiveness for the patient, due to their non-inclusion. These processes generate a very large amount of data that is necessary to structure. In this context, the storage of information in a database (DB) and the management of this database (database management system, DBMS) become an important issue for the management of projects and IVIVC and more generally for development of new pharmaceutical forms. This article describes the implementation of a prototype object-oriented database (OODB) considered as a tool, which is helpful for decision taking, responding in a structured and consistent way to the issues of project management of IVIVC (including bioequivalence and bioavailability) (Food and Drug Administration 2003) necessary for the implementation of QTPP.

KEY WORDS: CPP; CQA; IVIVC; object-oriented database; QbD.

INTRODUCTION

The objective of *in vitro-in vivo* correlation (IVIVC) (1) is to evaluate *in vivo* performance of a pharmaceutical formulation from its *in vitro* dissolution characteristics by establishing a predictive model. Although definition may seem simple, the underlying processes necessary for establishment of IVIVC are complex (2–5); they generate a large amount of data of various natures, forms, and from various sources (6,7). In this context, the storage, treatment, and management of information in a database become an important issue for development of new pharmaceutical formulations. Indeed, IVIVC projects appeal to areas that transverse many departments (pharmacokinetic, formulation, analytics, bio-analytics, clinics, production...). These characteristics must structure the database and integrate

a multidisciplinary, transversal, dynamic, and incremental approach subjected to specific recommendations and regulations.

Even if attempts to structure and cross-analyze pharmacokinetic (PK) data were developed in the past (8), commercial software presenting IVIVC modules are originally dedicated mainly to PK or *in silico* prediction. Their various data analysis modules offer compatibility and a possibility of data exchange without real structuring and coding of the data within or across studies. This across-study structuring is important in case of external predictability and comparison of forecasted profiles with observed values. This data compiling is a way to assess robustness of IVIVC. Even with population PK modeling, a similar structuring and coding of the various studies is mandatory.

Multidisciplinary aspect is obvious as IVIVC requires diverse areas of knowledge (6,9) that refer to different professions, each bringing crucial information for the validation of the whole process (see Fig. 1). This multidisciplinary emphasizes the importance of the structure which could manage the heterogeneity of data source and nature. The need for early implementation of either statistical or quality tools in development increases the need for secured structures (2,3).

Final role of IVIVC, as *in vivo* surrogate, is to assess the consequences of any modification of the entry function

¹ ESIEE Cités Descartes/BP 99, 93162 Noisy Le Grand Cedex, Paris, France.

² UFR Pharmacie, EA4678, Biopharmaceutical Department, Auvergne University, 28 Place H. Dunant, BP 3863001, Clermont-Ferrand, France.

³ Biopharmaceutics, Merck Research Laboratories, One Merck Drive, P.O. Box 100 Whitehouse Station, NJ 08889-0100, USA.

⁴ BEBAC, Neubaugasse 36/11, 1070, Vienna, Austria.

⁵ To whom correspondence should be addressed. (e-mail: j-michel.cardot@udamail.fr)

Fig. 1. Multidisciplinary nature of IVIVC

(pharmaceutical formulations) on *in vivo* pharmacokinetic profiles and then assess potential therapeutic impact on patient. That must be done in optimal security conditions and associated in theory with quality by design (QBD) in order to determine critical process parameters (CPPs) and critical quality attributes (CQAs). IVIVC imposes a multidisciplinary approach and a global vision of the processes based on a *vivo* and *vitro* comprehensive and controlled body of information.

Given diversity of the fields necessary for establishment of an IVIVC, the generated information can be considered dependent on particular domains. Interpretation of domain's related information greatly depends on working field (called here "profession") point of view but should be shared with other professions. As presented in Fig. 2, information considered to be key factors of IVIVC could have significant repercussions on other domains. Consideration of these characteristics by all actors concerned requires (i) agreement upon a common reference system (dictionary), and (ii) possibility of evaluating consequences on all affected domains of all modifications/evolutions of these key factors with a systemic and transversal approach (see Fig. 2).

IVIVC's development is part of a dynamic and incremental process (prospective analysis) based on continuous utilization of existing data to support hypotheses and optimization. Initial analysis of data is recommended to be carried out using a two-stage approach (deconvolution-convolution) method to set up critical parameters. This development includes several steps (10). The first step is the implementation of initial *vitro* tests in parallel to initial formulation development. The second step requires *in vivo* pilot study to test the initial formulation in step one. The third step is initial establishment of *in vivo-in vitro* relationship. The fourth corresponds to optimization of dissolution (if needed) and then establishment of finalized IVIVC. The fifth step allows optimization of formulations, of the

manufacturing procedures and of product's validation characteristics from pre-acquired knowledge. The final step is the transfer to manufacturing scale of pivotal batches and their validation leading to registration.

Given the development process of an IVIVC, as well as its regulatory and normative aspects, database's (DB) structure must take into account those prerequisites. In addition, DB's structure must take into account the diversity of the sources of information required for investigation of an IVIVC: analytical, chemical, formulation, and clinical, each with specific methodological approaches and type of data. The following items (items 1 to 4) are considered of main importance:

1. Data are of different natures depending upon their sources: spectroscopic, numerical, text, image, *etc.*...
2. Approaches incorporate a sequence of procedures (data validation, numerical calculations) and thereby generate an important volume of data that must be stored in a dynamic and structured way.
3. Representation of domains for users must be pertinent and perfectly demarcated to ensure strong cohesiveness (formulation, analytical, clinical...).
4. DB must enable a global vision of project's parameters required for management of ongoing studies and for proper development of the process (quality and regulatory).

Following a reminder about DBs, we describe the implementation of a prototype of an object-oriented database (OODB) in three steps: (i) architecture of the developed model and its coherence regarding DB's principle objectives, (ii) processing and data manipulating tools, and (iii) creation of domain-specific fields of use and management of processes. The prototype is developed under MATLAB® 2011b (MathWorks, Natick, Massachusetts, USA).

Fig. 2. Transversality of the IVVC

DATABASES

A database can be considered as an ensemble that structures information and that can be used for one or several calculation applications. It must (i) be accessible with any criteria, (ii) allow user to find all information pertaining to a certain criteria, and (iii) link different criteria together (11).

A database management system (DBMS) can be perceived as a group of application software enabling users to insert, modify, search, and efficiently format specific data within a significant amount of information. This system must allow the use of this data for scientific calculations and ensure efficient sharing of data with an acceptable performance rate.

Prerequisite of a Database Dedicated to IVVC

Implementation of a DBMS requires several levels of description. Among these, two are particularly important because they enable progressive abstractions of data to gradually approach each individual user's point of view. The first level is the conceptual one; it expresses transversality and fits description of all characteristics of the field use in IVVC. To build it, all users need a work method and must take into account transversality between users and domains. The second level is the external one (express modularity); it concentrates on each domain's and users' needs. It must

consider that each profession (i) has a specific work method, (ii) generates specific information, and (iii) uses data in a specific way. DB implementation requires different criteria along two complementary approaches: (i) specialization by profession or field due to multidisciplinary and (ii) securing of information that has a particular significance in pharmaceutical field and that is submitted to a strict regulation.

Regarding the profession criteria, a DBMS must enable:

- Implementation by field of specialty that responds to typical user profiles and modularity
- Sharing and integration of different fields required for transversality
- Facilitated administration of data (basic development and conception)
- Processing of data. Users must be able to update and interrogate data and easily perform calculations. Users should be able to describe data without having to describe the way they found it; this aspect is related to dynamic and incremental process.

Regarding the securing of information, a DBMS must enable (12):

- Protection of data. This should be against non-authorized access and coherent restoration of data (following constraints of integrity and coherence).

- Coherence of data. A DBMS must ensure compliance with data-modification regulations that guarantee coherence, monitoring, and traceability of data (*i.e.*, integrity constraints).
- Controlled redundancy of data. Integrating different information can lead to redundancy and duplication of certain data within DB. DBMS must control this redundancy and render it implicit, thus non-visible for the user. This criterion refers to DBMS's robustness.
- Traceability of data. This refers to constant surveillance of data use as well as freezing of experimental data once these have been implemented and confirmed into the base.

Choice of Database Model

Historically, relational models (RDBMS) were used. These are very structured formal databases. These relational models optimize the storage of profession data by distributing it to many files to optimize the speed of access and ulterior use from an information technology (IT) point of view. Despite the fact that RDBMS are powerful, they are not truly dynamic. Indeed, data storage structure is fragmented into numerous static tables (files) whose links are predefined. These characteristics make difficult management of modularity and transversality as well as dynamic aspect described in the introduction of this article. Due to their construct (fragmentation of information into numerous tables) and due to their inflexibility (static tables with predefined links), these are the most secure but least modular databases. Furthermore, these language queries require a learning process that can put off certain users, specifically in a context of complex interrogation of the base. However, the processing and interrogation of the base is complex and fastidious.

Emergence of object-oriented databases (OODBs) is due to the convergence of two fields: (i) databases and (ii) object-oriented modeling (OOM) that allows for adaptive and dynamic usage. These are supported by object-oriented programming language (OOP) and rest on two fundamental notions: object and class.

- An object can be defined as an entity with its own proper meaning in a specific field. For example, an analysis (size of API particulates) will be an "object" that will regroup all of the results (attributes) as well as all of the ways to obtain and confirm them (API code, lot identifier, date of trial, operating methods, control types, norms, alerts...). Each object is unique by its attribution of its own and constant identifier.
- A class is a structure that regroups objects by field of similarity. For example, we can compare the periodic table of elements to the "class" representing real world of chemistry. Each atom that constitutes this class represents an "object," each composed of electrons, neutrons, and protons in different quantities (called properties or attributes). The class regroups similar elements. Each object (for example, carbon), depending on technique used, can result in appearance of new objects 11C, 12C, 13C, and 14C "inheriting" characteristics of the father object (*i.e.*,

same number of electrons and protons but a different number of neutrons). In OOM, transmission of the parent object's elements (attributes and methods processing attributes) to "children" objects is called specialization or inheritance (Fig. 3). Moreover, one class can be contained in another supra-class (called metaclass).

OODBs are feature-rich because they enable (i) representation of complex data structures of any nature (audio, image, text), (ii) dynamic addition of objects or attributes (13,14), and (iii) regrouping of data (attributes or properties) and of methods to process the said data (methods) (15).

Therefore, OODBs are more flexible and adaptive than RDBMS and respond better to our requirements (multidisciplinarity, transversality, data processing).

ESTABLISHMENT OF IVIVC DATABASE

Database Architecture

The database is built following a hierarchical model (see Fig. 4) with four levels, each composed of a set of objects (see Fig. 3). First level is the superior level (or root) that fits general description of study and active pharmaceutical ingredient used (API). Second level (formulation) fits the description and statement of tested formulations. Each formulation generates new sublevel (third level) that describes *in vivo* and *vitro* methods (method) tested on each formulation to be used in IVIVC (absorption algorithm, modeling of experimental data...). Fourth level (dataset) proceeds from validation of experimental data (detection of aberrant values, missing data, lower limit of quantitation (LOQ), below limit of quantitation (BLQ),...) to all corresponding data of methods described in third level.

The design (object-oriented DB) allows, from a structural point of view, a dynamic implementation through addition of sublevels, each of them being an object. Addition of these sublevels inherits from the characteristics of the upper levels and ensures non-redundancy of data. That approach preserves the DB coherence. Furthermore, the design sublevel also ensures an *in vitro* as well as an *in vivo* implementation.

Hierarchization requires that declaration of study must be executed before declaration of formulations (formulation) composing the said study. Similarly, declaration of formulations must be executed before declaration of methods (design) (method) and resulting data (method of validation of confirmed raw data; results obtained with a calculation of absorption method; results obtained with a modeling method...) (see Fig. 3). This approach allows securing of data and information by step and by level. Indeed, we cannot add a branch without securing the parent level beforehand. For example, it is not possible to add a calculation method without securing beforehand information relating to formulation referring to the said calculation. It also allows dynamic and flexible addition of branches (formulations, methods) depending on the study's needs.

The database allows one, under permanent monitoring with log file, to create a study, to add and delete levels, and to access and extract data. This last function could be made *via* a simple "search" function (recursive function), which does not

Fig. 3. Architecture of database

slow down the system but is by nature slow itself. Alternatively, last function can be made *via* index tables which are incremented during the input process of data, thereby slowing down the system somewhat during input, but guaranteeing rapid access afterwards. Depending upon final aim, either one or other method should be chosen. Extraction of data can be combined with functions carrying out logical operations (for example, OR, AND, NOT) (16).

Securing of Data

All objects, regardless of their level, must have a unique object identifier (OID). OID constitutes a constant indexing key through time that enables verification of the uniqueness of object by comparing it with objects in the database located at the same sublevel (12,17). The verification and integrity process (absence of duplicates) of indexation key is automatic and transparent for the user.

For certain fields, use of a dictionary to code the entry is mandatory. In the present case, the dictionary guarantees coherence between various departments of a company, as well as between studies and across time. It should be obvious to note that it is better to use existing recognized dictionaries (like MEDRA for pharmacovigilance) than to create a new one.

Practical Database Establishment

Necessary data for an IVIVC study originates from two different sources: *in vivo* and *in vitro*, both of which share data in common. These origins define specific fields of use

and implementation within the database for specific user objectives, referring to a dedicated body of information and professions.

Common Data

Higher level (level 1) is link with development of one API. Common data regroups all qualitative and quantitative data relating to (i) physicochemical and biological characteristics (such as solubility, granularity, density, stability...) of API contained in the formulations, and (ii) description of formulations and production procedures (formulation fields). All of this information allows us to evaluate the key factors of production (CQAs).

Some methods such as bioanalytical methods could also in some cases be considered as common data across studies. However, bioanalytical method improvement leads to dynamic modifications of their characteristics across time. For this reason, those methods were not included up to now in our design, and in addition, they do not behave to CQA parameters. They could be considered as non-critical items in term of quality by design (QbD) and could be then included in the corresponding levels of the methods.

As described above and presented in Fig. 5, item (i) is linked to API characteristics. Characteristics are grouped and registered in a class (Class_CDP: *Component_drug_product*) and linked to the upper (first) level. It integrates all information related to biological, physical, and chemical properties of the API. For example, methods linked to this class could sum up the experimental results obtained such as solubility based on pH, hygroscopicity, density, flowability...

Fig. 4. Organization of *in vitro* data

Methods allow the user to implement warnings which are going to highlight out of specifications values (OOS). This step corresponds to an initial implementation of risk evaluation and finding of CQA. That is called *cdp_risk_management* class for the CDP's information level and allows one to implement corrective actions to maintain the constant quality.

Item (ii) is linked to formulation level (level 2 just below the API level). The formulation integrates different types of information through composition links. These types include as example:

- Formulation procedures including description of the production procedure and the referring steps. The *process_risk_management* class carries out management and evaluation of the risks associated with production procedure.
- Formulation composition: selection of excipients that is made up of (i) list of excipients (function, quantities...),

(ii) compatibility studies justifying their choice, and (iii) excipients' specification and tests carried out to justify their choice. The *excipient_risk_management* class carries out the management and evaluation of the risks associated with excipient selection.

In Vitro Data

In vitro information concerns dissolution characteristics of the tested formulation and corresponds to level 3 (method) of Fig. 3: method. *In vitro* information (Fig. 5) is composed of two types of data: (i) description of protocol and dissolution methods (analytical fields) and (ii) test results obtained from the dissolution methods which are then created in level 4 (dataset).

- Description of protocol and dissolution methods is recorded in the *test_diss* class. *Test_diss* class is

Fig. 5. *In vivo* data structure and connection

related by an inheritance link to method level (level 3 of the base). Through composition links, method class integrates different types of information (recorded in classes) pertaining to (i) description of dissolution method, (ii) description of technology used and associated equipment, and (iii) dissolution medium and dissolution protocol (norms). *Diss_risk_management* class carries out management and evaluation of risks associated with dissolution.

- Test results are recorded in level 4 of the base (dataset). This data matches experimental data that needs to be confirmed and secured because it is important for IVIVC calculations.

In Vivo Data

In vivo information, like *in vitro* information, starts in level 3 (method) of Fig. 3. Description of this level's organization is presented in Fig. 6 and fits clinical part of the IVIVC.

Depending on the levels, we can differentiate the following:

1. Data regarding the randomization of the study and associated methods (update, creation...). This includes all of necessary information (in index table form) to access a patient's or a group of patients' data according to randomization criteria (for example, administration sequence, group...). This class is linked to third level (method) since its role is to manage the methods of access to all vivo data in the study by active ingredient.

2. Data regarding study's units (unit) of measure (concentrations, mass, and primary and secondary pharmacokinetic parameters...) as well as its associated methods (unit conversion, creation...). This class is linked to the third level (method) since units are common to the whole study.
3. Data regarding scheduled time of sampling (schedule), stored within a class linked to second level (formulation) as the formulation, which inherits from API characteristics, drives, in IVIVC, the *in vivo* release. These data pass to method and dataset levels (level 4) and enable the comparison between the scheduled times of sampling and the actual times of sampling. This allows detecting inconsistencies or deviations from the administration protocol.
4. All other data such as analysis of biological fluids (dates of analysis, equipment, methods, biological fluid(s)...) and pharmacokinetic/pharmacodynamic data are lodged in the fourth level (dataset).
5. Patient class belongs also to level 4 (dataset) and contains general information (age, height, weight...). It is possible to record, in a dynamic manner, other information with three classes that describe (i) clinical history of a patient and/or clinical events that can occur during the study (*clin_event*), (ii) associated treatments (*clin_treat*), and (iii) biochemical data (BIOCH). Given that the patient's clinical data can evolve during the study, previous classes are included in the lists that store information based on occurrence date of an event.

Fig. 6. Specialized classes for computational approach

The Processing of IVIVC Data: a Computational Approach

Data processing is executed by specialized classes (see Fig. 7) that add to each other in a dynamic manner to create a fifth level (see Fig. 3) that inherits all information from level 1 to 4. This level allows performing calculations by combining different data stored at the fourth level (DATASET).

Estimation of an IVIVC and its predictability requires a processing of incremental data that uses vivo and vitro data and configurable computational procedures. Some procedures could require the use of covariates which are stored in dataset level. As an example of procedures:

- OPTIM block makes adjustment from predefined structural models (STRUC_MOD class) following several methods (Nelder and Mead method, Levenberg and Marquardt method, and conjugated gradients method). The inheritance method lets us consider the dataset class as the “child” class that will regroup adjustment method and associated parameters as well as adjustment results.
- NCA block makes a non-compartmental analysis of profiles (AUCt, AUCinf, Cmax, tmax, MRT, VRT, Vd, CLz, ke, $t_{1/2}$, MRT, and peeling calculations).
- Convdecon block uses convolution and deconvolution methods according to different

numerical approaches (Vaughan and Denis, Cuttler, Langenbucher, Loo-Riegelman, and Wagner-Nelson methods) and analytical approaches (direct convolution and deconvolution from structural models).

- RNPART block uses non-configurational regression of vivo and vitro profiles through different approaches (cubic splines method, smoothed splines, kernel regression).
- STAT block uses descriptive statistics (medium, standard deviation, percentiles...) and inferential statistics (ANOVA, t test, confidence intervals, adjustment tests, power estimation, and two one-sided t tests...) as well as tests dedicated to bioequivalence.
- IVIVC block uses (i) linking of vivo and vitro profiles according to directly linear models or by statistical adjustments (Hazard model, Odd model, time scaling) and (ii) internal and external predictability calculations.
- Simulation performs predictive determination and *in vivo* surrogate or de-risking support.

Data Import and Export

Data import can be automatic or manual from text files (ASCII) or spreadsheets or dedicated software. It is achieved in two steps: (i) design import then (ii) data import. Import is subjected to a systematic verification of (i) authorization access keys to the base, and (ii) validity of data based on

Fig. 7. Diagram of hierarchical data base

attributes' possible values (character chain, numerical values, aberrant values...).

Data export is carried out *via* text files or spreadsheets. It applies to all levels of the base by describing the results and associated methods that generated these results. Information regarding CQA and CPA is regrouped by field of application (formulation, analytical, clinical to risk factors), which facilitates legibility and formatting of quality target product profile (QTPP) file.

Access Control and Traceability

The securing of data is a crucial element. Every user is given a unique identifier that determines his or her rights of access. In the current state, four levels have been determined.

- Administrator level has access to all administration procedures of the database (management of user clearance, management of access rights, and addition of methods...).
- Study design level enables, in each field, (i) definition of content of levels in relation to the study design and its view, (ii) access to data in read-only and write-in, (iii) access to all procedures of calculation methods (derivation), results of which are stored in the base in a dynamic and automatic way, and (iv) data export toward different interfaces and data backup.
- Data entry level allows the person to enter data in the frame of the study designed in the previous level.
- Consultation level confers no right except that of data consultation on screen and data export.

For each user, a unique key is created by the administrator, the content of which is encrypted. When a user logs into the base, system systematically and transparently checks user's access rights by confirming adequacy of his or her key with reference tables that authorize or deny the processing of data according to user's clearance level.

REMARKS AND CONCLUSION

This database is based on concepts developed in the 1970s by Wagner (18) who posited that the understanding of *in vivo* future of an active ingredient (and its therapeutic activity) implies concomitant knowledge of (i) physiological factors, (ii) the active ingredient's physicochemical characteristics, (iii) and formulation properties of pharmaceutical form. Thanks to the simple techniques available today, the possibility exists for each scientist to follow in a secure way the approach preconized in the 70s!

This database is articulated around three main points: (i) understanding of dissolution and absorption mechanisms (formulation, API) and of their variability that validate *vivo-vitro* relation; (ii) determination, confirmation, and validation of key development and control factors (CQA, CPP); and (iii) predictability calculations, bioequivalence calculations, and dissolution limit calculations. It is an incremental process integrating QbD, which is considered necessary for validation of IVIVC. Given the diversity of data, design of this database requires all of the functionalities of object approach (dynamic integration of new parameters not initially identified, profession-and-disciplinary-fields axed structure, possible reuse of code, integration into the base of

data processing: computational approaches...). Its modelization, developed from a profession-axed approach and from key associated factors, is relatively simple and allows all actors to work with common references. The database is currently in the testing phase using studies that have been executed and validated beforehand with other IT tools (relational databases and general pharmacokinetic software). From an IT point of view, object development authorizes a modular validation of the database and proceeds in two steps: (i) validation of each module specific to fields and underlying domains; and (ii) integrated validation of modules in database. This process enables an incremental validation concomitant to development, which optimizes management of the process.

Numerous software packages, dedicated to pharmacokinetic analysis, are available on the market. All of them are capable of analyzing data from each study individually. They all contain modules for data entry and storage; however, no preexisting structure is proposed and raw data are stored in different but unique files. For example, one file can be created by the user to enter time-concentration data, a second for the dissolution data, and a third for derived parameters (such as absorption kinetics or PK parameters), *etc.*.... In some software packages, the derived parameters are obviously in the workflow section and not in the dataset section. In case of copying from results (*i.e.*, absorption kinetic) to the dataset section, the link with the initial raw data (time concentrations) is broken, and a modification of those initial raw data does not impact the copied file. This can lead to possible incoherency between data (time-concentration and absorption kinetics). All the files are often independent, one from the other, it is expected that their will be either poor links or no links between the files and the studies. This architecture restrains the maximum capacity of data storage and does not impose common keys within and between studies. Each study could be entered in a different way with different data structure depending of the scientist's willingness, and even within a study, each dataset could be entered without common keys. For example, between *in vivo* and *in vitro* datasets formulations', coding can be different between files within the same study: formulation "A" in the first dataset can be renamed as "REF" in the second dataset. Units (dissolution in percent *vs* absorption in mass unit) can be different as well, without any warning from the software. Only units (mass, volume, time but not percent) are often proposed in form of a preset list with conversion possibilities. To highlight the above problems, for external predictability, two or more *in vivo* studies are performed. The first one establishes the correlation and second validates *via* external predictability the IVIVC. This requires a link or pooling data from the two studies in order to perform the calculation of the external predictability based on data of the second study and on the IVIVC established in the first study. This approach is not an easy programming task using the existing software packages, and does not ensure that the data are coherently entered between the two studies. Some suppliers propose modules (often called connect or data access) which allow to import from multiple sources or from an online repository the relevant information to be included in the analysis. However, this import is the responsibility of the user and no validation of coherency is performed by the software. This means that

the sources must have the structure and coherency close to what is described in the present paper to avoid any mistake. In the approach proposed in the current paper, use of the hash table, implementing an associative array, will resolve this problem, assuring that coherence of the structure, and coding of the key data will be fulfilled *a priori* in the implementation of the data base.

All above described possible problems emphasize that, currently, the scientist must currently carefully handle data, continuously check for possible incoherencies, and note in a log file all the correspondences and links between data to ensure quality and tradability of all operations.

IVIVC can be considered as a determining and central point to the new and generic drug development that requires a rigorous project management and that must take into account (i) multidisciplinary and transversality of the fields of knowledge necessary to its realization, and (ii) integration of information from different domains (professions) that often happens in a progressive and asynchronized way but actually requires a dynamic and incremental approach.

Development of an IVIVC is a strongly structuring project. Its realization requires a strong implication of different actors and departments of companies (direction, quality, regulations, analysts, formulators, clinicians, production...). Design of a database resolves a problematic situation by supplying a system that helps decision-making, thereby providing a tool that can effectively enable management of complex projects.

REFERENCES

1. Food and Drug Administration. Guidance for industry: extended release oral dosage forms: development, evaluation, and application of *in vitro/in vivo* correlations US. Washington, DC: Department of Health and Human Services, Food and Drug Administration Center for Drug Evaluation and Research (CDER); 1997.
2. Food and Drug Administration. Quality by design for ANDAs: an example for immediate-release dosage forms. Washington, DC: Department of Health and Human Services, Food and Drug Administration Center for Drug Evaluation and Research (CDER); 2012.
3. Food and Drug Administration. Quality by design for ANDAs: an example for modified dosage forms. Washington, DC: Department of Health and Human Services, Food and Drug Administration Center for Drug Evaluation and Research (CDER); 2011.
4. Food and Drug Administration. Guidance for industry: bioavailability and bioequivalence studies for orally administered drug products—general considerations. Washington, DC: Department of Health and Human Services, Food and Drug Administration Center for Drug Evaluation and Research (CDER); 2003.
5. Dunne A. Approaches to developing *in vitro* – *in vitro* correlation models. In: Chiluriki DM, Sunkara G, Young D, editors. Drugs and the pharmaceutical sciences, vol 165, pharmaceutical product development *in vivo in vitro* correlation, Informa. 2007. pp. 47–69.
6. Emami J. *In vitro-in vivo* correlation: from theory to application. J Pharm Pharm Sci. 2009;9:169–89.
7. Devane JG. Impact of IVIVR In product development. In: Young D, Devane GJ, Buttler J, editors. Advances in experimental medicine and biology, vol 423. *In vivo in vitro* correlation. New York: New York Plenum Press; 1997. p. 241–59.
8. Cardot JM, Jaouen A, Godbillon J. PKC a new pharmacokinetic software using SAS®. Eur J Pharm Biopharm. 1997;43:197–9.

9. Sirisuth N, Eddington ND. *In vivo* – *in vitro* correlation definitions and regulatory guidance. Int J Generic Drugs. 2002. <http://www.iagim.org>. Accessed 5 Aug 2013.
10. Devane JG. Impact of IVIVR in product development. In: Young D, Devane GJ, Buttlar J, editors. Advances in experimental medicine and biology, vol 423. *In vivo in vitro* correlation. New York: New York Plenum Press; 1997. p. 241–59.
11. Garardin G. Les bases de données. 2^{ème} ed. Paris: Eyrolles; 2003.
12. Petmul G, Tjoa AM, Winiwarter W. Modelling data secrecy and integrity. Data Knowl Eng. 1998;26:291–308.
13. Hines ML. Conceptual object-oriented database: a theoretical model. Inf Sci. 1998;105:31–68.
14. Lee S-W, Kim H-J, Ahn J-H. A schema version model for complex object in object-oriented databases. J Syst Archit. 2006;52:563–77.
15. Beraha S, Su J. Support for modeling relationship in object-oriented databases. Data Knowl Eng. 1999;29:227–57.
16. Niemi T, Christensen M, Jarvelin K. Query language approach based on the deductive object-oriented database paradigm. Inf Softw Technol. 2000;42:777–92.
17. Huang Y-F, Chen J-M. The study of indexing techniques on object oriented databases. Inf Sci. 2000;130:109–31.
18. Wagner JG. Biopharmaceutics and relevant pharmacokinetics. Hamilton: Drug Intelligence Publications; 1971.