Revista Electrónica de Metodología Aplicada
2008, Vol. 13 nº 2, pp. 1-14

Modeling item banking: Analysis and design of a computerized system.

J. Gabriel Molina, Ignacio Pareja y Jaime Sanmartín[1]
Universidad de Valencia

RESUMEN

Los tests adaptativos informatizados dependen de la existencia de bancos de ítems de los que seleccionar los ítems más adecuados a aplicar a lo largo de la administración del test. Ahora bien, la disponibilidad de bancos de ítems es bastante limitada y, por otra parte, el desarrollo de bancos de ítems suele representar un esfuerzo considerable en la práctica. Ello motivó el desarrollo de este trabajo, centrado en el análisis y diseño de un sistema orientado a dar soporte informatizado a todo el proceso de construcción y mantenimiento de bancos de ítems. Así, el modelo de desarrollo de bancos de ítems propuesto provee las bases para implementar un programa orientado a tal fin y, también, promover la discusión acerca del modelo más conveniente en el desarrollo de tal programa. Es de esperar que la disponibilidad de software para desarrollar y trabajar con bancos de ítems ayude a extender la aplicación de los tests basados en bancos de ítems, como es el caso de los tests adaptativos informatizados.

Palabras clave: Tests adaptativos, Tests computerizados, Bancos de ítems, Diseño de Software.

ABSTRACT

Computerized adaptive tests rely on item banks from which to select the more suited items to be applied along the test administration; however, the availability of item banks is rather limited and, on the other hand, the development of an item bank can involve a great deal of effort for test users. The aim of this work concerned the analysis and design of a computerized system aimed to support the whole process of developing and maintaining item banks. Resulting from it, an item banking model is proposed which provides the basis to implement such a computerized system and, also, to encourage the discussion about the more suitable model for this goal. It is to be expected that the availability of sound item banking software could help to extend the application of item-bank based tests, like computerized adaptive tests, beyond the boundaries of large-scale testing programs.

Keywords: Adaptive Testing; Computerized Tests; Item Banking; Software Design.

[1] Dirección de contacto:
J. Gabriel Molina
Universitat de València, Dpto. de Metodología de las CC. del Comportamiento
Av. Blasco Ibáñez, 21 - 46010, Valencia.
Telf.: 963864420 - FAX: 963864697
e-mail: Gabriel.Molina@uv.es

1.- Introduction

The chance of using computers in the design and application of tests as well as in the collection and subsequent processing of the responses has enabled relevant advances in the field of psychological testing (Navas, 1999). This has been the case of Computerized Adaptive Tests (CATs) which have as its main point that as long as the computerized administration of the test takes place, the test contents are tailored to the responses given by the subject taking the test. Thus, the sequence of items administered to the subject depends on his/her responses to the earlier items in the test. The basic goal of CATs is to administer a set of items that are maximally informative for each subject, and its major advantage is efficiency: research has demonstrated that on average a CAT is 50% shorter than a paper-and-pencil test with equal or better measurement precision (Embretson & Reise, 2000).

Different CAT designs has been raised which mainly differ according to the answers given to the next three key questions (Wainer, 1990): (1) how do we choose an item to start the test; (2) how do we chose the next item to be administered after we know the examinee’s response to the previous ones; (3) when do we stop test administration. In this context, Item Response Theory has offered a theoretical framework that is particularly suitable for adaptive testing since it provides some psychometric models that allows obtaining subject scores which are independent of the particular set of items taken (Lord, 1980). Some specific works about computerized adaptive testing can be found in Embretson & Raise (2000), Olea, Ponsoda & Prieto (1999), Renom (1993, 1997), Rojas (2001), Sands, Waters & McBride (1997), van der Linden & Glass (2000), Wainer & Steinberg (2000) or the special issue about CATs in the Psicologica journal (Ponsoda, 2000), among others.

2.- Computerized adaptive testing and item banks

A CAT rests on having an item bank (IB), that is, a pool of items whose psychometric proprieties are known so that the more appropriate items to be applied to the subject taking a CAT can be selected at each step of the test application (Lunz, Bergstrom, & Gershon, 1994). As stated by Wainer & Mislevy (1990, pg. 82): “A CAT cannot begin until it has an extensive and calibrated item pool (item bank)”. Once an IB oriented to measure a specific construct is available, it will be possible to develop from it as many tailored tests as required for the testing situation. However, it is not an easy task for test users to obtain already built IBs, as there are rather few available in the software market; moreover, prices of the available ones may be excessively high for the budget of most psychologists. This situation can explain, in part, the limited application of IB-based tests (e.g. CATs) beyond the limits of large-scale testing programs.

Another option for test users interested in applying IB-based tests is to actually build one; furthermore, this building process will result in an IB specifically tailored to the developer's testing goals (e.g., Navas, 1993, Olea et al., 2004). The development of IBs, however, involves a substantial amount of work, which ranges from collecting a sufficiently large number of items (Wise & Kingsbury, 2000), to finally obtaining indicators of the psychometric properties of these items, that is, the IB calibration. The latter requires the IB developer to collect item response data for the items in the IB, since this empirical information is going to enable the calibration of the IB (Barbero, 1996). But gathering response data for an IB raises some difficulties: the most significant in practice is that, given the usually large number of items in an IB, it is neither possible nor reasonable to administer all of the items to just a sample of subjects. A common strategy around this problem is to split the IB in several reasonably sized parts, the administration of which, usually in a traditional non-adaptive way, is going to provide the necessary response data for calibrating the IB (Flaugher, 1990).

The splitting administration strategy of an IB must follow some design which finally allows putting the parameters of all the items in the IB on a common scale (Vale, 1986). This is the case of some of the usually applied calibration designs represented in Figure 1 which shows, in a schematic form, four designs in the simplistic case of considering only two IB subsets and two subject subgroups. Figure 1(a) illustrates a calibration design where the IB is split into 2 tests with no overlap, which are taken by two sample subgroups with no overlap either. In this case, linking of the item parameters onto the same metric can be achieved by making equivalent either the two tests or the two groups. Figure 1(b) represents the anchor-item design, where a few items are taken by both subject subgroups; these anchor items provide the key for linking the item parameters obtained from different subject groups onto a common scale. Figure 1(c) illustrates the anchor-subject design, that is, a common group of subjects take all the items in the IB and provide the key for linking the item parameters on a common scale. Like the anchor-item design, this calibration design does not assume equivalence of the items/subjects subgroups. An important practical disadvantage of the anchor-subject design is that a group of subjects (the anchor subjects) should take all the items, perhaps a task too exhausting to be asked for. Finally, Figure 1(d) illustrates a double-anchor design in which there are both common items and common subjects in the administration of the IB items.

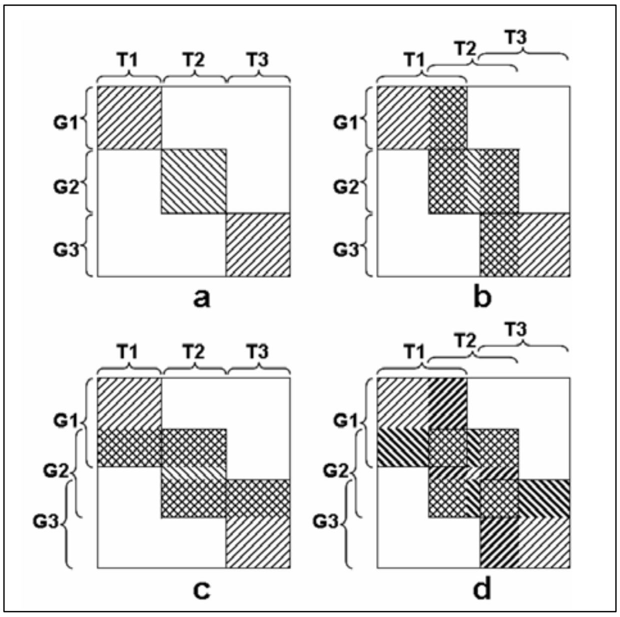


Figure 1: Four standard calibration designs oriented to link the parameters of the IB items on a common scale.

Other more complex anchoring designs have been proposed which can provide a more efficient calibration of the parameters describing the psychometric properties of the items (Berger, 1994; Dorans, 1990; Jones & Jin, 1994; Vale, 1986). The basic point, though, is that all of them must make feasible to place the item parameters of all the items in the IB on a common scale. In practice, available programs for IRT parameter estimation that can handle data matrix with missing data (e.g. LOGIST and BILOG) make possible the estimation on a common scale of the ability and the items parameters for all the calibration designs considered above.

3.- Computerized support to item banking

Potential IB developers can find computerized support in the development of an IB from different sources. Some programs include a number of functions oriented to support the storage and management of already calibrated IBs, such as CONTEST (Timminga & van der Linden, 1996), FastTEST (Weiss, 2005), and LXR-TEST (Logic eXtension Resources, 2006). Some programs are suited to check the appropriateness of several item response models to our empirical data as it is the case of the applications EO-Fit (Ferrando & Lorenzo-Seva, 2001) and IMINCE (Lorenzo-Seva & Ferrando, 2003). EQUATE 2 (Baker, 1993) is a useful program to put the results of one group into the metric of another if there are common items in the two (o more) tests. Finally, some programs make feasible the calibration of the IB according to several psychometric models, for example, BILOG (Mislevy & Bock, 1990), MULTILOG (Thissen, Chen & Bock, 2004), and XCALIBRE (Assessment Systems Corporation, 1996), among others.

Other computer programs supporting some specific tasks involved in the building of IBs could have been listed above. It is noteworthy, though, that none of them is providing comprehensive support to the development of an IB. This is especially true for the management of all the information associated to the calibration of an IB, commonly consisting of sparse response data sets corresponding to the answers of subject samples to item subsets. Actually, this lack of computerized support to the whole process of item banking can be pointed as a limitation to the availability of IBs and, consequently, to the option of applying IB-based tests like CATs.

The analysis of a computerized system covering this limitation in the support to the development of IBs was precisely the main focus of this work. The item banking model resulting of this analysis should provide the basis to implement a comprehensive system oriented to build and maintain IBs. The design specifications of a computer program based on this item banking model are also provided in order to depict this model in a more intuitive way than it would be with just the results of the analysis of the system.

4.-System Analysis

We chose a database model as the programming framework more suited for a computerized system oriented to build IBs. This option was preferred taking into account, on the one hand, that the results of the analysis of the information associated to the development of IBs has showed a well-structured underlying data scheme (Molina, Sanmartín & Pareja, 1997, 1999) and, on the other hand, that the database programming model excels at processing data which are, in the main, highly structured (Willits, 1992).

According to the software development methodology, we analyzed first the requirements that potential IB developers can have when involved in this task. This analysis of requirements provided the structure of the database underlying the program model to be set. Figure 2 offers a global view of the requirements involved in the process of developing an IB. Five main requirements are differentiated (some technical terms introduced below are italicized the first time they are used):

(1) Input of all the information describing the items (item format, item statement, etc.). It is not the case of the information related to the psychometric proprieties of the items, which would be obtained after proceeding with the calibration stage.

(2) Selection of item subsets from the IB (*IB subsets*), the administration of which is going to provide the item response data required to calibrate the IB. It must be noted that, before the IB calibration is accomplished, the administration of IB subsets uses to be in a non-adaptive way, either by computerized or by paper-and-pencil means.

(3) Integration in the IB database of the *administration datasets*, that is, the item response datasets collected through the administration of IB subsets to samples of subjects. These administration datasets represent pieces of empirical data that are to be integrated into the IB response data matrix.

(4) Assembly of the *calibration dataset* by merging of the administration datasets separately collected in the IB building process. A calibration dataset will integrate the response data that make feasible the IB calibration. It should be an option of the IB developer to decide which administration datasets are to be merged into it. On other hand, given that data gathering in the development of IBs uses to rely on some data collection design like these described above (Figure 1), it will be a common practice to deal with calibration datasets which consist of incomplete, sparse data matrices.

(5) Psychometric analysis of the calibration dataset as the way to obtain the psychometric proprieties of the IB items. It must be noted that different calibration datasets can be set along an IB life as new administration datasets are collected or new items are integrated into the IB. This fact allows updating and keeping alive the IB after its initial calibration.

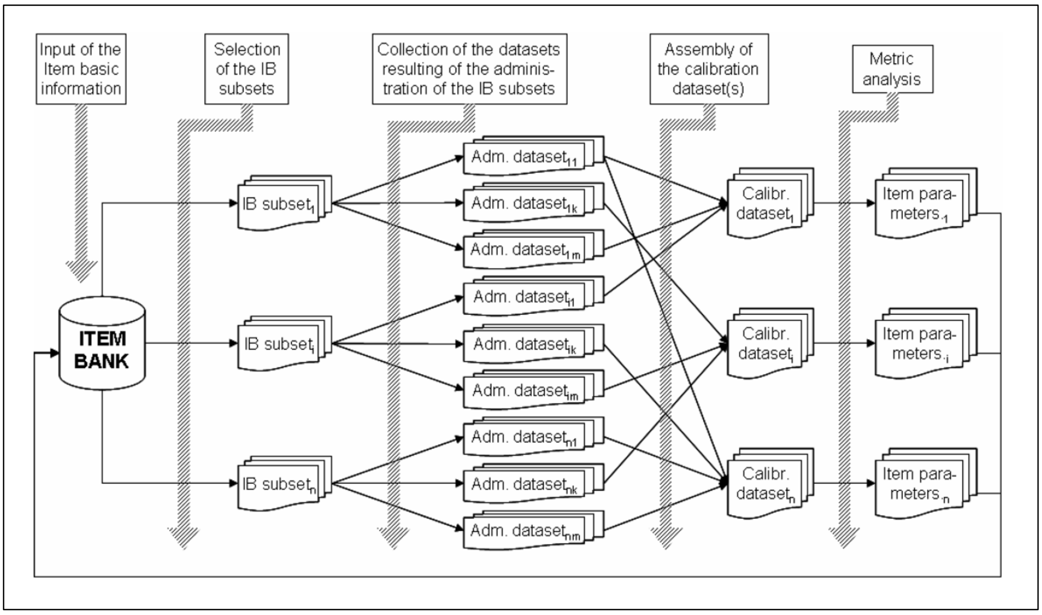


Figure 2: Main requirements involved in the development of an IB.

According to the results of the analysis of requirements to be satisfied, we considered an item banking model based on a relational database set by four tables. These are described below through the two basic dimensions of any database table: its entities and its attributes. The entities of these 4 tables appear represented in Figure 2 as the objects connected with arrows.

(1) Item table. Entities: the items in the IB. Attributes: item format; item statement; descriptive keywords (e.g. keywords about the item content); answer key(s), when suited; item source (when the item was entered in the IB, who did it, item origin/author); item administration constraints (time limit in the computerized presentation of the item, conflicts with other items, conditions related to the presentation of the item before or after another specific item, or together with other items); item administration record (item exposure, identifiers of the tests which the item was included into, when they took place, how many subjects took the item). The attributes of this database table should integrate all the pieces of information about the items that could be relevant in the development and subsequent application of an IB.

(2) IB-subset table. Entities: the IB subsets selected by the IB developer(s) to be administered as tests. Attributes: name assigned to the IB subset; size (number of items in the IB subset); when was it set; list with the identifiers of the specific items included in the IB subset; IB-subset exposure (number of times it has been administered as test); constraints considered in the selection of the items that integrate the IB subset (e.g. time limit for the test, content balancing).

(3) Administration-dataset table. Entities: the datasets collected through the application of the IB subsets defined in the IB. Attributes: identifier of the IB subset associated to the administration dataset; raw response data matrix, that is, the subjects' responses to the items in the corresponding IB subset; number of rows (subjects) and columns (items) in the dataset; some keywords oriented to describe the administration of the test (e.g.

when, where, who, to whom). It should be noted that the raw response data matrix could consist of a 3-way table if more than just one subject's response for every item is collected (e.g. response time, specific answer, solving strategy, etc.). Additionally, as it is illustrated in Figure 2, the proposed item banking model provides for different administration datasets associated to the same IB-subset, what corresponds to the practical situation where an IB subset is applied to two or more groups of subjects.

(4) Calibration-dataset table. Entities: the calibration datasets defined by the IB developer as result of merging some of the administration datasets available in the administration-dataset table. Attributes: name assigned to the calibration dataset; size (number of rows and columns); percentage of missing data; identifiers of the specific administration datasets merged; the item psychometric proprieties obtained through the analysis of the calibration dataset with a suitable item response model (e.g. location, discrimination, information function). The option to define different calibration datasets allows to the IB users to set which calibration dataset is providing the information to fill the fields related to the psychometric proprieties of the items. Anyway, it could happen that this fourth database table had just one record, for example, in the case of merging all the available administration datasets to calibrate the IB once and for all. The latter would be confirmed if, furthermore, no more administration datasets were integrated into the IB in order to achieve new calibrations oriented to update the IB.

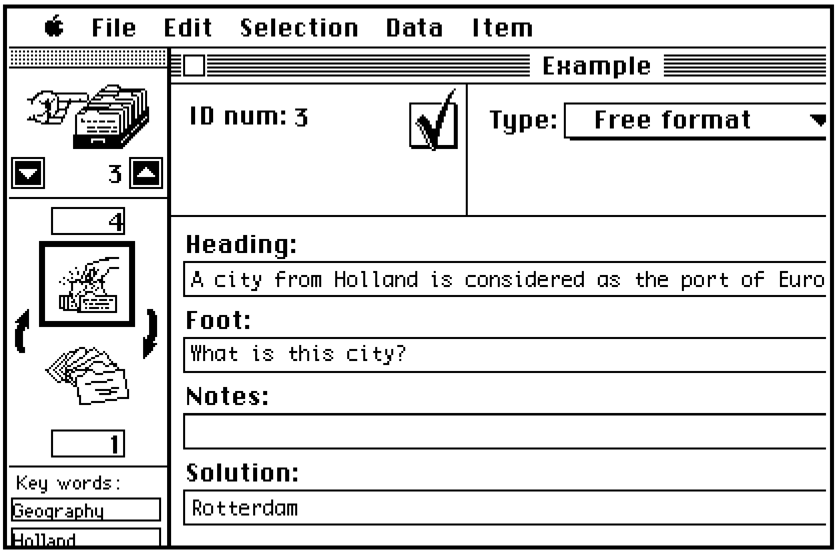
5.- System Design

The description of the design specifications of an item banking program was achieved for the four main tasks involved when building and maintaining an IB: (1) edition and visualization of the information for the IB items; (2) selection of IB subsets; (3) collection of item response data; and (4) calibration dataset assembly and analysis.

A prototype matching these design specifications was developed by the authors in order to overcome some communication difficulties associated to the report of design specifications through just a writing description. Prototyping is a common practice in software engineering as a way to assist in the validation of the program requirements as well as to help in the description to the final program users of the design specifications proposed for a program (Sommerville, 1985). Our program prototype was implemented with the object-oriented language AppWare 1.2, formerly named Visual AppBuilder, from Novell Inc. (Novell, 1995). This language provides a database object which was especially useful in the development of the relational database underlying our prototype.

(1) Edition and visualization of the information for the IB items. A database system has to offer the resources to edit and visualize the information that the system is designed to manage. In our case, most of this information is this related to the IB items, that is, the attributes described above when the *item table* was introduced. These pieces of item information were arranged in our prototype into six interface windows that correspond to six blocks of related information: the basic item information (item format, item statement, answer key…); the item descriptors (a number of user-defined fields for key words describing the item); the item source information; the item administration constrains; the item administration record; and, finally, the item psychometric proprieties. The last window should be

automatically edited after choosing a calibration dataset, of course, if this already were calibrated with some psychometric model. Figure 3 shows the prototype's window oriented to edit and visualize the information of the first of these blocks of information, this of the basic information of the IB items. The graphical tools on the left are to navigate trough the records with the item information.



ID num: 3 ☑

Type: Free format

Heading: A city from Holland is considered as the port of Euro

Foot: What is this city?

Notes:

Solution: Rotterdam

Key words: Geography
Holland

Figure 3: Prototype's window oriented to edit and visualize the basic information of the IB items.

(2) Selection of IB subsets. The selection of IB subsets from an IB has as main goal, first, to set the tests to be administered in order to obtain the data required to achieve the IB calibration and, then, to use the IB in order to apply IB-based tests. Two complementary selection strategies could be considered in order to support a powerful item selection facility:

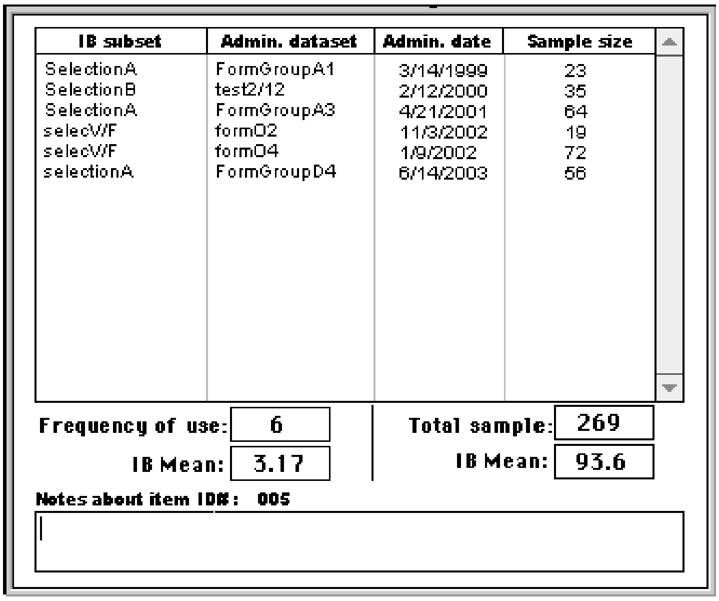
(a) One of these selection strategies should enable the program user to browse through the IB and handpick the individual items he or she wishes to select. This item-by-item selection should be complemented with a second item selection strategy which enables automated selection according to certain criteria that the user can establish. For this purpose, relevant criteria to be considered include the item format (e.g. multiple-choice items only), the item content (e.g. item statements containing some specific terms), the item exposure (e.g. items that have been administered less than 6 times), and the item descriptive key words (e.g. items described with the key word *Form perception*), among other potential criteria.

(b) In addition, successive selection criteria should be enabled by using the logical operators AND, OR, and NOT (e.g. items with a sentence-completion item format AND with an item exposure below the item bank average item exposure). This feature provides power and versatility in the automated selection of items, especially when working with a large number of them.

Both manual and automated item selection strategies are not mutually exclusive, since there could be items finally selected by any of them. The handpicking strategy enables the program user to make selections by inspecting each item, while the automated strategy only requires establishing item selection criteria. A practical approach can result from making a first automated selection of the items and then polishing the resulting selection through an individual item review.

(3) Collection of item response data: A set of functions in our prototype are oriented to support, on the one hand, the edition of IB subsets as ready-to-use tests and, on the other hand, the importation and storage of the empirical response data resulting from the administration of these IB subsets. These functions are oriented to facilitate the hard work of collecting the empirical data required to calibrate an IB. With regard to the test edition task, it can be considered to link the item banking system with other available programs specifically oriented to this task, as it was the case in our prototype with the TECA program (Sanmartin & Vidal, 1988), a free distribution program used in conventional test assembly, which supports editing and printing of paper-and-pencil tests. By providing this external link to our prototype, we avoided the implementation of some complex functions already available in TECA, saving development time and program size. This strategy should be considered in the implementation of an item baking system as it is not difficult to find shareware and freeware programs supporting test edition and administration.

After the information from an administration dataset is integrated into the IB database, it is relevant to consider the resources to update the item administration records, given that this information about the item usage is going to enable the optimal definition of new IB subsets to be administered as tests. Likely, IB items with the lowest exposure levels are the more convenient candidates to be administered in the ongoing process of developing and using an IB. Figure 4 shows our prototype's window with the information related to the item administration record of a specific IB item: the names of the IB subsets in which the item has been picked; the names of the tests the item was part of (administration datasets); the date in which these administrations took place; the sizes of the samples taking the item; and four summary indexes oriented to assess the item exposure. The information in some of these fields can be especially useful as criteria to be applied in the automated item selection described above.



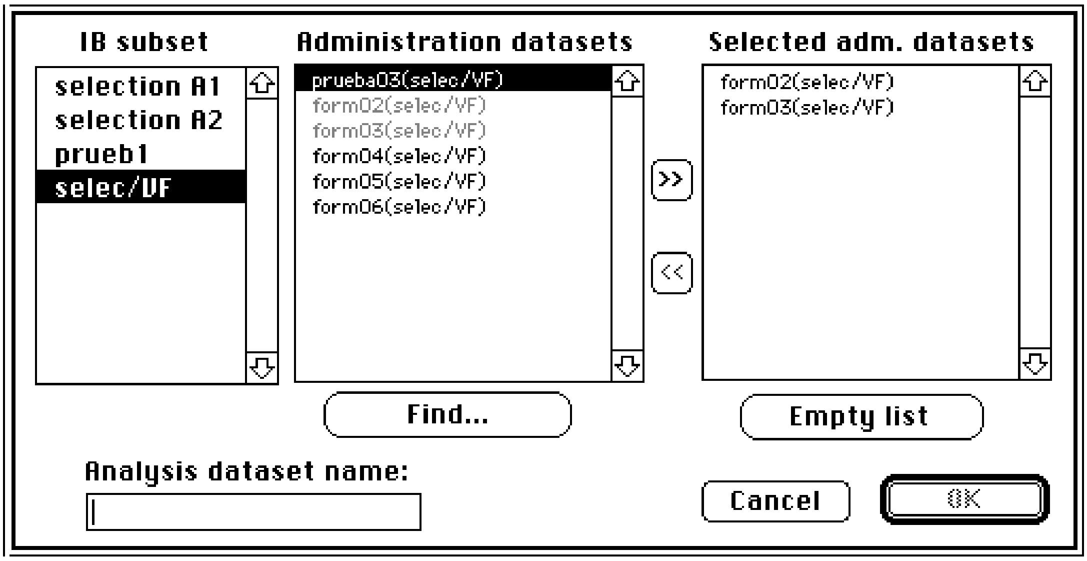
IB subset	Admin. dataset	Admin. date	Sample size
SelectionA	FormGroupA1	3/14/1999	23
SelectionB	test2/12	2/12/2000	35
SelectionA	FormGroupA3	4/21/2001	64
selecV/F	form02	11/3/2002	19
selecV/F	form04	1/9/2002	72
selectionA	FormGroupD4	6/14/2003	56

Frequency of use: 6
IB Mean: 3.17
Total sample: 269
IB Mean: 93.6
Notes about item ID#: 005

Figure 4: Prototype's window with the administration record for an IB item.

(4) Calibration dataset assembly and analysis: To obtain the psychometric proprieties of the items in the IB implies to merge the dataset to be analyzed from the pieces of data obtained through the administration of IB subsets (the administration datasets). This calibration dataset may be integrated by all the available administration datasets in the IB database; however, it will sometimes become more desirable to perform psychometric analysis on a part of the data collected, for example, when the IB developer is not confident with some administration datasets. Taking the administration datasets as the unit of work makes it feasible for the IB developer to assemble the calibration dataset in a more flexible way, since it does not limit the process of carrying out the IB psychometric analysis to just one global data file. On other hand, it must be noted that different calibration datasets can be set along an IB life as new administration datasets are collected or new items are integrated into the IB. This is important to update the IB after its initial calibration, making feasible the necessary task of removing flawed, obsolete, and overly used items, and adding new items to the IB (Wainer & Mislevy, 1990).

Some of these specifications are reflected in the prototype's interface window oriented to achieve the assembly of the calibration dataset (see Figure 5). The list in the center of this window shows the administration datasets associated to the specific IB subset selected on the left list ('selec/VF' in this example). These administration datasets can be moved by the IB developer to the list on the right, which contains the administration datasets will be finally merged into the calibration dataset. This manually-oriented selection of the administration datasets is complemented with the 'Find…' function, which allows automatically selecting the administration datasets matching some criteria defined by the user, somewhat can be specially useful with large IBs. Selection criteria for the administration datasets could include: the administration date (e.g. administration datasets which have been collected before a specific date), the size (e.g. administration datasets which contain response data for more than 25 items and/or more than two hundred subjects), and the satisfaction of some item-related constraints (for example, administration datasets which contain items #23 and #73).



IB subset: selection A1, selection A2, prueb1, selec/VF

Administration datasets: prueba03(selec/VF), form02(selec/VF), form03(selec/VF), form04(selec/VF), form05(selec/VF), form06(selec/VF)

Selected adm. datasets: form02(selec/VF), form03(selec/VF)

Figure 5: Prototype's window enabling the assembly of the dataset to be analyzed.

With regard to the psychometric analysis of the calibration dataset, the design strategy achieved in our prototype was the same as this applied for test edition, that is, to take advantage of consolidated programs performing this task. Obviously, this external-link strategy involves enabling the IB system to import the output files saved by those programs as well as the automatic edition of the information in these output files into the IB fields arranged to it. The choice of this strategy in the design of the IB calibration function has two main advantages: a reduction in the programming resources required to develop the system; and the chance of applying a wide repository of tested software performing the latest in item response modeling.

6.- Discussion

The item banking model proposed by the authors tries to integrate all the requirements involved when developing and maintaining IBs. Thus, in the analysis of the system was considered not only the data structure associated to the whole process of item banking, but also the relationship between the database tables and the relationship of these with the steps involved in the development of an IB. Figure 2 summarizes this model, which sets the basis for further discussion about the sense and viability of such specifications and, ultimately, for the design and implementation of an item banking computerized system.

The item banking model derived from this analysis served to raise the design specifications of a program supporting item banking, from the storage and organization of the IB information, until the use of the IB to build tailored tests by being able to set the criteria for the automated selection of the items that achieve a desired test profile. However, it should be noted that different design specifications can be set based on the results of the analysis of a computerized system. Thus, the program prototype presented by the authors was implemented only as a means to reach two main objectives: communication of some design specifications for the system in a quicker, intuitive way; testing of the viability of the proposed item banking model when implemented in practice. Other formal programming developments in this line would be desirable as a way to cover the lack of software supporting the entire process of developing and maintaining IBs. This development could serve as one of the catalysts to extend the application of IB-based tests, like the widely highlighted CATs, in other settings apart from the large-scale testing programs applied by some testing companies in some countries.

7.- References

Assessment Systems Corporation (1996). XCALIBRE (version 3.5) [Computer program]. St Paul, MN: Author. Available: http://www.assess.com

Baker, F. B. (1993). EQUATE 2.0: A computer program for the characteristic curve method of IRT equating. *Applied Psychological Measurement, 17,* 20.

Barbero, M. I. (1996). Bancos de items. En J. Muñiz (Ed.): *Psicometría*. Madrid: Editorial Universitas.

Berger, M. P. F. (1994). D-optimal sequential sampling designs for IRT models. *Journal of Educational Statistics, 19*, 43-56.

Dorans, N. J. (1990). Equating methods and sampling designs. *Applied Measurement in Education, 3,* 3-17.

Embretson, S. E. & Reise, S. P. (2000). *Item response theory for psychologists.* Mahwah, NJ: Lawrence Erlbaum Associates.

Ferrando, P. J., & Lorenza-Seva, U. (2001). Checking the appropriateness of item response theory models by predicting the distribution of observed scores: The program EO-Fit. *Educational and Psychological Measurement, 61,* 895-902.

Flaugher, R. (1990). Item pools. In H. Wainer (Ed.), *Computerized adaptive testing: A primer* (pp. 41-64). Hillsdale, NJ: Lawrence Erlbaum.

Jones, D. H., & Jin, Z. (1994). Optimal sequential designs for on-line item estimation. *Psychometrika, 59,* 59-75.

Logic eXtension Resources (2006). LXR-TEST (version 6) [Computer program]. Rancho Cucamonga, CA: Author. Available: http://www.lxrtest.com

Lord, F. M. (1980). *Applications of item response theory to practical testing problems.* Hillsdale, NJ: Erlbaum.

Lorenzo-Seva, U., & Ferrando, P. J. (2003). IMINCE: An unrestricted factor-analysis-based program for assessing measurement invariance. *Behavior Research Methods, Instruments, & Computers, 35,* 2, 318-321.

Lunz, M. E., Bergstrom, B. A., & Gershon, R. C. (1994). Computer adaptive testing. *International Journal of Educational Research, 21*, 623-634.

Mislevy, R. J., & Bock, R. D. (1990). BILOG (version 3) [Computer program]. Chicago: SSI Scientific Software. Available: http://www.ssicentral.com

Molina, J. G., Sanmartín, J., & Pareja, I. (1997). Item banks information storage and management: Analysis of a computer system. Paper presented at the *10th European Meeting of the Psychometric Society*. Santiago de Compostela, Spain.

Molina, J. G., Sanmartín, J., & Pareja, I. (1999). Desarrollo de un sistema informático orientado a la construcción y gestión de bancos de items [Development of a computerized system oriented to the development and management of item banks]. In J. Olea, V. Ponsoda & G. Prieto (Eds.), *Tests informatizados: Fundamentos y aplicaciones*. Madrid: Pirámide.

Navas, M. J. (1993) *Aplicación de la TRI al campo de la medida: Creación de un banco de items para evaluar la capacidad matemática*. Tesis doctoral, Madrid: Universidad Nacional de Educación a Distancia (UNED).

Navas, M. J. (1999) Un siglo utilizando tests. *Revista Electrónica de Metodología Aplicada, 4,* 1-11.

Novell Inc. (1995). AppWare (version 1.2) [Computer language]. Provo, Utah: Author.

Olea, J., Abad, F. J., Ponsoda, V., & Ximénez, M. C. (2004) Un test adaptativo informatizado para evaluar el conocimiento de inglés escrito: diseño y comprobaciones psicométricas. *Psicothema, 16,* 519-525.

Olea, J., Ponsoda, V., & Prieto, G. (1999). *Tests informatizados: Fundamentos y aplicaciones.* [Computerized tests: Foundations and applications]. Madrid: Pirámide.

Ponsoda, V. (2000). Overview of the computerized adaptive testing special section. *Psicológica, 21,* 115-120.

Renom, J. (1993). *Tests adaptativos computerizados. Fundamentos y aplicaciones.* Barcelona: PPU.

Renom, J. (1997). Retos y perspectivas de los tests adaptativos informatizados. En A. Cordero (Coordinador). *La evaluación psicológica en el año 2000.* Madrid: TEA Ediciones.

Rojas, A. J. (2001). Pasado, presente y futuro de los test adaptativos informatizados: Entrevista con Isaac I. Béjar. *Psicothema, 13,* 685-690.

Sands, W. A., Waters, B. K., & McBride, J. R. (Eds.) (1997). *Computerized adaptive testing: From inquiry to operation.* Washington: American Psychological Association.

Sanmartín, J., & Vidal, A. (1988). Generation, correction and analysis of computerized objective tests. *Wheels for the Mind Europe, 3,* 23-28.

Sommerville, I. (1985). *Software engineering.* Wokingham: Addison-Wesley.

Stocking, M. L., & Lord, F. M. (1983). Developing a common metric in item response theory. *Applied Psychological Measurement, 7,* 201-210.

Thissen, D., Chen, W., & Bock, D. (2004). MULTILOG (version 7) [Computer program]. Chicago: SSI Scientific Software. Available: http://www.ssicentral.com

Timminga, E., & van-der-Linden, W. J. (1996). CONTEST (version 1.0) [Computer program]. Groningen 9700AV: The Netherlands. P. O. BOX 841.

Vale, C. D. (1986). Linking item parameters onto a common scale. *Applied Psychological Measurement, 10,* 333-344.

van der Linden, W. C., & Glass, C. A. W. (Eds.) (2000). *Computerized adaptive testing: Theory and practice.* Boston, MA: Kluver Academic Publishers.

Wainer, H., & Mislevy, R. J. (1990). Item response theory, item calibration and proficiency estimation. In H. Wainer (Ed.), *Computerized adaptive testing: A primer* (pp. 65-102). Hillsdale, NJ: Lawrence Erlbaum.

Wainer, H., & Steinberg, L. (2000). *Computerized adaptive testing: A primer* (2nd edition). Hillsdale, NJ: Lawrence Erlbaum.

Weiss, D. J. (2005). *Manual for the FastTEST Test Development System* (version 2.0) [Computer program]. St. Paul, MN: Assessment Systems Corporation. Available: http://www.assess.com

Willits, J. (1992), *Database design and construction.* London: Library Association Publishing.

Wise, S. L., & Kingsbury, G. G. (2000). Practical issues in developing and maintaining a computerized adaptive test. *Psicológica, 21,* 135-155.