Using Cognitive Design to Generate Employment Selection Tests

Damon U. Bryant **Sandra Alexander** **Kristophor Canali** **Tanya Delany** **Ashley K. Smith-Carlson** **Kathlea Vaughn**

IBM Global Selection Team, Global Talent Policy and Programs, Corporate Headquarters

Abstract

This presentation demonstrates the design, development and integration of an item generation software application into a human resource function, employment selection testing. The software is able to create multiple versions of IBM's Information Processing Aptitude Test (IPAT). The use of the software reduced IPAT form development time by 80%, reduced test development costs by over 90%, and enabled IPAT deployment into emerging markets on demand.

Introduction

One of IBM's core values is innovation that matters not only for the company, but also for the world. The Global Selection Team (GST) within CHQ Global Talent Policy and Programs was faced with a challenge that involved quickly supplying internal customers with numerous employment tests. The purpose of this presentation is to summarize the process that GST used to turn a problem into an innovation opportunity, which will have widespread implications across the employment testing industry.

Today, test development is quite expensive and time-consuming based on traditional design principles. In emerging markets where time and costs are important factors in the HR staffing process, meeting the demands of customers for new tests is quite challenging using standard approaches. This challenge motivated the GST to develop a scalable, cost-effective method to generate tests for internal IBM customers in emerging markets where hiring volumes are high and the potential for breaches in test security is a concern.

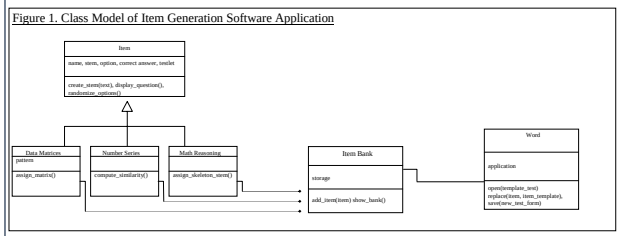
In January 2005, the GST began a project to develop software to generate versions of the Information Processing Aptitude Test (IPAT) for use in Asia Pacific. There was an initial planning session with GST team members who collaborated to determine the specific functionality of the item generator application. A timeline was set for the development of the prototype and the development team began the Object-Oriented Analysis and Design phase of the project.

Class Model

In order to develop the class model, GST members had to understand the current real-world object - in this case, the Information Processing Aptitude Test (IPAT). This is a proprietary IBM-approved selection tool that assesses reasoning skill and the ability to learn; it does not measure specific knowledge. Research studies in 1999, 2000, and 2004 indicate that the test is a valid predictor of performance in entry-level technical positions at IBM. In the US, the test is currently administered online; however, in Asia Pacific (AP), it is administered in paper and pencil format. Because there is an increase in hiring efforts in AP, the paper and pencil version of the IPAT was the object of focus.

Currently in AP, IPAT test forms are created in Microsoft Word, so it was necessary to create a class to write test items to a word document. Also each IPAT form has a set of questions covering three parts. The three parts are Data Matrices (DM), Number Series (NS), and Math Reasoning (MR). DM contains 15 items, NS has 20 items, and MR includes 12 items. Each part contains item types that are quite different from each other, which prompted the development team to subclass items belonging to each of the three parts. Based upon this initial analysis, the following class model was produced.

Figure 1. Class Model of Item Generation Software Application



Language Implementation: Python

The classes in the model were implemented in Python, which was chosen for the following reasons:

(1) Dynamically-typed (2) Fast prototyping capabilities

(3) Clean syntax (4) Easy to learn

Other advantages of this language can be obtained at http://www.python.org.

The GST used cognitive science and artificial intelligence to model existing test questions across the three parts in Python to generate thousands of cloned questions. The total application includes less than 5,000 lines of code. Alpha testing of the code began in March 2005. To show the simplicity of the language and the implementation of the class model in Python, two example classes are provided below. In Figure 2, there is an example of a Math Reasoning Class that inherits properties of the abstract Item Class. An instance of this class can be generated and stored in an Item Bank instance along with other item instances from Number Series and Data Matrices. Once an instance of the Item Bank Class has 15 Data Matrices, 20 Number Series, and 12 Math Reasoning instances stored, then the Word Class (Figure 3) is used to produce the IPAT version along with the associated answer sheet.

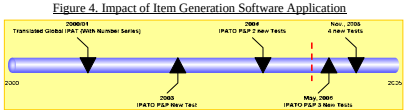
Figure 2. Math Reasoning Item with Inherited Item Class

Figure 3 Word Class Used to Write Items in MS Word

Use of the Item Generator in China

In March 2005, the GST received a request from internal customers in China to produce three new versions of the IPAT for a testing session of university hires in several locations. Because alpha testing was nearing completion, GST took this opportunity to beta test the item generator in producing IPAT forms. After collaborating with contacts in China about the logistics of using multiple test forms in one testing session, GST generated several forms that were used in May 2005 to test approximately 4,000 job applicants. GST incorporated experimental test parts into several forms. The experimental parts were produced by the software. Analyses indicated that the test questions produced by the item generator were comparable to the operational test forms. Based on these findings, the decision was made to use tests constructed wholly from the item generator during the next IPAT testing session.

Figure 4. Impact of Item Generation Software Application



Results of Generated Test Versions: China

In November of 2005, the software application was used to generate four versions of the IPAT: A, B, C, and D. The tests were administered to over 1400 university applicants applying to IBM technical and non-technical positions in China. The four versions were administered at the same time in different testing locations across China, i.e., Beijing, Guangzhou, Hangzhou, Harbin, Nanjing, Shanghai, and Wuhan.

One of the objectives in this second use of the item generator was to ensure that the application produced test forms that were similar to each other in terms of overall test scores and item difficulty. When controlling for location, results indicated that the generated test versions do not differ from each other. (See overall means in Table 1). Moreover, an assessment of the difficulty (measured as the proportion of examinees who answered the items correctly) was completed. Findings showed that the generator performed reasonably well in cloning base classes of items. See Figures 5 through 7 for cloned item comparisons.

Table 1: Summary Statistics for Generated IPAT Versions

Test Version	Average	Standard Deviation	Number of IBM Applicants
A	62.33	7.89	674
B	63.07	6.77	316
C	62.61	6.50	312
D	64.71	5.18	105
All versions	62.74	7.20	1407

Figure 5. Data Matrix Item Difficulties

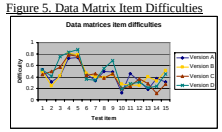


Figure 6. Math Reasoning Item Difficulties

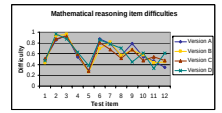
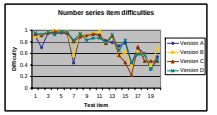


Figure 7. Number Series Item Difficulties



Benefits of Item Generation Software

By incorporating recent advances in cognitive science, artificial intelligence and psychological measurement, the GST is now more efficient in creating IPAT forms for emerging markets across the world. Efficiency is recognized in terms of the following:

- *Time*
  In the past when IPAT forms were developed manually, at least 6 months were needed to produce an operational version of a test. With the item generator, IPAT forms can be produced in approximately 2 weeks.

- *Cost*
  When traditional test development procedures are used, (i.e., with a trained psychometrician writing test questions manually), the cost of development for one psychometrically sound test question is approximately $150-$250. However, the cost of developing one question using the item generator is approximately $2.

- *Supply*
  When internal customers in emerging markets needed IPAT forms in the past, few unique tests were available to them, which raised issues of test security and validity after a certain number of test administrations. The item generator software now enables a new test to be used for each testing session, thus reducing the chances of test over-exposure and breaches in test security (e.g., item/content sharing, cheating).

Summary

The use of this innovative application in an on demand era has proven to be very successful in supplying new versions of the Information Processing Aptitude Test (IPAT) to internal IBM customers in countries where hiring/applicant volumes are large and the potential for breaches in test security is a concern. Due to the success with the development of new versions of the IPAT with the item generator, this innovative process is being expanded to other IBM-sponsored testing programs, including the Non-Verbal Processing Aptitude Test (NPAT) and the Administrative Clerical Aptitude Test (ACAT). From an industry perspective, as more and more companies move to online testing solutions, which will require constant refreshment of test questions, it will be imperative to have an innovative approach such as this one to support HR staffing needs worldwide.