SUBJECT DIDACTICAL CRITERIA AND GUIDELINES FOR THE SELECTION AND IMPLEMENTATION OF A FIRST PROGRAMMING LANGUAGE IN HIGH SCHOOLS

LEILA GOOSEN
School for Teacher Training, University of Pretoria, South Africa
lgoosen@gk.up.ac.za

Programming instruction is used as an environment to promote meta-cognitive and higher order thinking skills, including problem analysis and design of solutions, and the programming language as a medium for developing these skills. The programming language used until recently in schools for various reasons cannot stay on. This study establishes criteria what should be considered when selecting a first programming language to teach in high schools, and introduces guidelines for the implementation of such a language. The validity of selection criteria and implementation guidelines identified are then empirically verified within the South African context.

INTRODUCTION
Computer Studies is an important subject at school level, because it offers a new approach to problem solving by which the thought processes of learners can be drastically enriched (Von Solms 1979). The programming taught in Computer Studies is linked to the development of practical abilities in problem analysis, design of solutions, practical implementation (which includes coding in a computer programming language) and evaluation (Kirkwood 2000). Computer programming also provides a very rich and opportune environment for meta-cognitive development (Thomas and Upah 1996).

Programming instruction is used as an instructional environment to promote the above-mentioned higher order thinking skills (Palumbo 1990), and the programming language as a medium for developing these skills. However, research on how students learn beginning (first) programming languages has been lacking in the past (cf. Mayer and Dyck 1989), and the choice of which first language to use remains one of the concerns of programming language instruction (cf. Palumbo 1990).

RESEARCH BACKGROUND
The programming language used until recently in schools, Turbo Pascal, for various reasons, cannot stay on. The adoption of Pascal as programming language in schools was an easy one, as there were not too many contenders and it presented something that other languages may not have offered at that stage, e.g. it was structured (Gibson 2000). Thus, when it was realised that a new language was needed, no selection criteria existed against which different language candidates could be assessed, nor guidelines for the way in which the selected language(s) could be implemented successfully. Very little evidence of specific research into this area can be found in either national or international searches. In the light of these, the following aims were set for this study:

1. To establish criteria what should be considered when selecting a first programming language to teach in high schools;
2. To introduce guidelines for the implementation of a first programming language in high schools;
3. To verify the validity of selection criteria and implementation guidelines identified empirically within the South African context.
The first two of these aims were addressed by analysing and applying relevant related literature to the selection and implementation of a first programming language in high schools, while the third was accomplished by undertaking a field survey.

**SELECTION CRITERIA**

*Within a universal context, selection criteria should be set that will have worldwide relevancy in all circumstances, as well as still be relevant in a decade.*

The purpose of the subject Computer Studies is the development and use of problem solving approaches and skills, with a focus “on activities that deal with the solution of problems” (Department of Education 2003a).

With regard to the level and nature of learners of Computer Studies, the language adopted should adequately match the abilities of its users, the learners (Mendelsohn et al. 1990). The majority of programming takes place at high school level, where learners’ cognitive development levels are growing towards the formal operational stage. The chosen language should therefore provide an instructional environment that promotes the development of higher order thinking and problem solving skills, as well as critical thinking.

The role of programming in Computer Studies instruction is justified in that:

- Programming is an important skill
- Knowledge of programming provides a needed background for dealing with issues of modem society.
- Learners develop general cognitive skills that are identifiable and transferable to other situations
- Learning to program promotes the development of metacognitive skills, the knowledge of which can be applied to other fields.

The outcomes envisioned when learning a programming language, in conjunction with the role that problem solving plays in instruction, make it necessary that any programming language selected should be appropriate in the sense that it can be used as a medium for developing problem-solving skills (Palumbo 1990) by being as variously problem and solution oriented as possible (Richfield 2000).

The language should encourage a self-regulated approach to solving problems by promoting strategies for analysing programming problems and formulating solutions to them (van den Berg 1990). As the acquisition of cognitive skills is facilitated by high degrees of metacognition, the language should facilitate the use of the metacognitive components of problem solving during instruction, so that learners’ understanding of their own thought processes and their involvement in metacognitive behaviour would be developed.

As errors in learners’ programs are commonly related to deficiencies in problem-solving strategies and insufficient planning (Deek and McHugh 1998), an overall understanding of effective program development, including analysis methodologies, planning and design strategies and problem-solving skills and techniques should be encouraged. The role of algorithms in problem solving further emphasises the necessity to focus on underlying skills for the design and analysis of algorithms.

Due to the didactical principles involved in the teaching and learning of the subject, a first programming course should use a language that is beneficial for shaping problem solving (Palumbo and Reed 1991). Programming principles such as procedure and data abstraction, top-down design with steps-wise refinement, good programming style, testing and debugging should form an integrated part of the language.

As the compilation and coordination of the components of a program represent some of the greatest stumbling blocks for beginner programmers, learners should be provided with a safe, stable, structured and controlled programming language and environment that does not frustrate them because of unnecessary difficulties with the language and environment. A suitable software development
environment is required, with compiler tools that simplify learners’ task, such as debuggers to locate and correct errors and supply well-defined and easily understandable error messages.

The language and programming environment should be suitable for novice programmers in that it is easy to learn. The programming environment shouldn't be too expansive, but instead should be sufficiently simple to work with and offer relative simplicity of commands. Awkward syntax and complex language semantics should be avoided.

Learning about the programming language should produce learners who are life-long learners, who had been provided with the necessary tools and skills to adapt to a lifetime of change and will prosper over the length of long careers.

*Even though criteria should not date, topics with current relevance and regarding new tendencies in programming should nonetheless be considered.*

**OOP** is currently one of the accepted problem solving and programming paradigms being used. The language should therefore offer possibilities for object-oriented design, encapsulation, inheritance and polymorphism.

International trends with regard to the programming languages used in high-school education in other parts of the world should be given consideration.

In terms of the software development process, the language should be standardised and have reasonable prospects for continued development.

The language should be suitable and have capabilities for working on and with the Internet and "facilitate electronic communication" (DoE 2003b).

As a lot of programming revolves around programming for databases, the language should have sufficient capacity for database connectivity.

Visual programming languages present learners with an environment that is easy to use, gives them concrete feedback, and they can also see what is happening in terms of execution behaviour.

*Although criteria should be general in terms of relevancy, these should be specifically suited to the South African context.*

The South African subject statement for Information Technology has four learning outcomes. Of these, learning outcome 4, on programming and software development, is more heavily weighted, "because it is the crux of the subject" (DoE 2003b). It is also most relevant to this study in terms of the selection of a programming language. According to this outcome, the language should provide access to "a set of core development tools" (DoE 2003b) so that learners can practically experience the design and implementation of solutions using said tools.

The financial situation of schools make it necessary to consider whether schools themselves or education departments will carry the cost for the upgrade of hardware and software. The costs associated with the acquisition of the software required, including the programming language, IDE and database, should be within reasonable reach (Sanz-Casado et al. 2002). In this regard Open Source software can be considered. Costs with regard to sufficient in-service training of current CS teachers are also necessary.

Local tertiary establishments expect learners to be able to solve problems in any programming language. If the languages offered at these institutions are the same as those used at school level, students will be slightly advantaged; if it is different, students will at least know programming principles. Teachers new to the subject should be able to learn the language used in school during their training at local tertiary institutions. Due to the training that they provide in different programming languages, both tertiary and private computer training institutions could offer teachers training in, and support for teaching, the language.
It is imperative that learning and teaching support materials and other resources, particularly textbooks, for the language should be available to teachers. Specifically, resources appropriate to an OBE approach to the subject should be selected.

A general purpose programming language that supports many academic and commercial tools should be used. Many of the languages used for object-oriented programming in industry involve significantly more complexity than more traditional languages (SIGCSE 2001). This, together with the fact that the purpose of the subject is not to provide training, makes it unnecessary to consider the popularity and/or demand for specific languages in industry when selecting a first programming language for high schools.

The new language should provide adequate avenues for assessment (Scordilis 2000). In terms of external assessment, options for a language-independent exam (Walker 2000) need to be considered if provinces/schools implement different languages. Problem solving questions could however still be completed by using a common pseudo-code. Another option is for the programming section to no longer be assessed externally, but instead to use only projects for this purpose.

IMPLEMENTATION GUIDELINES

Instituting sufficient, efficient guidelines for curriculum implementation (Pratt 1999) may facilitate the successful implementation of a new curriculum. Planning should be started by doing an introductory investigation, followed by a situation analysis in conjunction with various other methods of gathering information about the process. All role players in the curriculum process ought to come and work together for a common outcome, with the needed appreciation for the role of teachers in implementation. Outcomes that need to be achieved (Mostert 1986) have to be identified and formulated. It is of great importance that these outcomes are stated clearly, and in a concise way.

Specifying and classifying learning content that is of consequence to learners can initiate the development of a curriculum. Teachers’ concerns regarding content and conceptual knowledge have to be addressed by providing practical advice about teaching the new curriculum. Documentation in this regard must however not be overly prescriptive, but should allow teachers enough leeway to implement according to their specific circumstances. Materials, including, but not limited to, textbooks, to accompany and support the new curriculum are supposed to be developed.

Curriculum assessment benefit from the use of internal assessment, expert appraisal and confidential review. Preliminary testing of the curriculum could take place in the form of pilot testing, using volunteer, experienced teachers. It is important to emphasise that this part of a reform program constitutes a temporary system. Program assessment is supposed to be carried out in order to revise and improve the curriculum to a standard suited to implementation. At the macro-implementation level of curricular change, cost can be an important factor. Micro-implementation takes place at classroom level, and granting teachers the necessary freedom facilitates this local adoption. The dissemination of information regarding the new curriculum is required to be done in a manner that will overcome resistance to change and convince participants that their implementation efforts will be supported. Especially teachers are supposed to be made aware of the differences between old and new practice, the relative advantage of reform must be made clear to them and their expectations for implementation have to be gauged.

Factors and variables that will either further or impede implementation need to be identified. Some influence is to be expected in terms of

- the quality of dissemination done before implementation
- effective leadership
- in-service training
- discussion of implementation
- reform campaign managers need to be able to obtain sufficient resources and teaching materials
- time available for implementation
- obstacles regarding teachers' working conditions.
Sources of resistance to change and obstacles to implementation identified should be handled in such a manner that implementation can proceed smoothly.

Curriculum documents and materials are required to be distributed to schools. Teachers must be given the time and resources to be retrained properly (Aiken and Snelbecker 1991). They would also benefit from guidance and support provided during user implementation, and this support ought to be continued as implementation becomes institutionalised.

The curriculum can finally be assessed in terms of the extent to which outcomes had been realised, as well as the quality of implementation (Mostert 1986). Program management needs to focus on outcomes rather than inputs.

Reform in teaching and learning, with the underlying need to empower teachers and learners, is neither simple, nor straightforward (Manouchehri and Goodman 1998). The process needs to be accompanied by changes in teacher preparation and development, as it requires teachers to enter a very uncertain world: They must create and test solutions to education problems by researching and using new curriculum materials and systematically integrating new instruction techniques that impose a drastically different structure on learning environments. The competence, motivation and experience of teachers will therefore ultimately govern the success of all such reform programs.

“We need to find ways to select for this endeavour those professionals who have a passion for the undertaking in their hearts. And we need to find ways continually to revive, maintain and augment that passion in all educators and their learners.”

(Pratt 1999:192.)

EMPIRICAL STUDY

In order to ascertain to what extent various role players within the South African context substantiate the authority of selection criteria and implementation guidelines identified, a field survey was undertaken, using a questionnaire containing Likert-type responses and shorter answers to open questions.

The population for the study comprises role players involved at different levels in various structures connected to Computer Studies, including:

- Team members of the Writing Committee for the National Curriculum Statement (NCS) for Information Technology [N=7]
- Subject co-ordinators representing their provinces with regard to the finalisation of the NCS and implementation issues [N=8] (A number of people are involved in Computer Studies in various capacities. Numbers presented here are such that persons already included in previous structures are not included again).
- Examiners and moderators for Matriculation papers in the provinces [N=18]
- Curriculum planners and advisors in the provinces [N=7]
- Other people involved with Computer Studies at provincial level in an administrative capacity [N=7]
- Teachers from all schools offering Computer Studies (Higher Grade and/or programming) [N=445]
- Persons involved in the training of Computer Studies teachers at tertiary institutions [N=10]

The procedure followed subsequently:

- The necessary permission to have the questionnaire completed was obtained from the subject advisors (Computer Studies) in the various provinces.
- Role players were informed of the study, questionnaires were distributed, completed and gathered (most questionnaires were completed via electronic communication).
- The gathered data was processed statistically, using mainly descriptive statistics, but inferential statistics were also employed to establish the significance of similarities and differences.
Results of the empirical study were compared to selection criteria and implementation guidelines developed to establish the implications of the study for the selection and implementation of a first programming language in high schools.

**RELEVANCE**

Because of the rapid changes in computer technology and programming languages, Mayers (2000) is of the opinion that in 5 years' time the language issue will have to be re-evaluated and languages changed again. Criteria and guidelines established in this study could prove indispensable in such a case.

**REFERENCES**


Walker, H.M. (walker@cs.grinnell.edu) (2000). Survey for AP CS. [E-mail to:] SIGCSE.MEMBERS@acm.org. August 1.