2.1 The Lack of Development Methods

ITS are complex to be developed. Adaptations of software development traditional methods, such as software engineering methods and tools, do not support intelligent systems development, as traditional systems lead with data, while ITS handle knowledge, mainly heuristics.

ITS basic architecture consists of a student model, a teacher model, a domain model, composed of a knowledge base and inference engine and an interface. The domain model is a knowledge based system (KBS). This way, the development methods of KBS could be adapted and extended to support the ITS development. Some experiences were already done [Costa, Santos & Rocha, 1997]. However, experiences emphasize the lack of support offered by KADS, as well as by extended-KADS [Werneck, 1995] for modeling teacher and student.

ITS has been developed in an empirical way, without any support of specific methodology and any accurate evaluation measures. It is observable that the references about development methods of ITS have a much closer approach to IA and engineering knowledge than to software engineering, where the experiences have been done ad-hoc, through prototyping and interactions with the domain experts.

Maybe the lacking of an ITS development method, supported by the concepts of Software Engineering, is caused by the ITS own structure. Many difficulties met in ITS building are concerned to the pretension of implementing the complete shaping of the domain model as well as of the modeling student. The knowledge bases are very specific and hard to reuse in other domains. The student model is one of the biggest challenges, because the measurement of his knowledge level in each moment of the tutoring process is yet complex. The incorporation of ITS intelligence should be done gradually, in such way that it would be possible to see the system operation before obtaining all the requirements for teaching a new topic. Elliott & Wolf [1996] recommend the iterative development approach, where is established a straight dependency between the ITS planning and its implementation.

Beyond doubt, there is a gap between the ITS conception phase up to an operational prototype. It is fundamental to search methodologies that turn ITS development process cheaper and possible to be evaluated by a set of well-defined criterion, creating reliable high quality products. The perception that KBS development methods do not hold potential to be directly applied on the ITS, points to the need of intensifying the research in this area. The adaptation of Software Engineering techniques to ITS area, could bring life cycle models, development methods, cost forecast, quality assurance and documentation. It can normalize the building of ITS and decrease costs and complexity of assignments.

2.2 The Lack of Evaluation Methods

The complexity of building intelligent educational software reflects also in the quality evaluation process. Educational software evaluation involves not only technical issues but also learning concepts. Some approaches concerning ITS evaluation are found in literature and that they are highly dependent to the development process adopted. Because of the lack of specific evaluation methods, experiences try to verify the aspects that influence the ITS development, offering preliminary criterion.

Mark & Greer [1993] associated traditional software evaluation techniques, like inspections and tests, to ITS. However, they do not describe how these techniques should be
applied. To Geissman & Schultz [1992], formal verification and validation are essential to ITS be accepted in critical areas. They describe an interactive development methodology, similar to the spiral model. Fast prototyping is used to permit verification and validation. The authors present the evaluation stages without demonstrating a practical description. Region & Shute [1994] emphasize that ITS should be evaluated during and after its complete development, checking the achievement of technical objectives and the performance in real environments. However, the authors do not clarify how that can be done.

The immaturity of the questions concerning ITS, the complexity of its architecture and the few available experiences, do not benefit the outstanding of a specific methodology of evaluation.

2.3 THE LACK OF PEDAGOGICAL MODELS

Modeling student is also a hard task, embracing student’s cognitive style, the tutoring process and the domain knowledge. Educational software cannot only favor technical aspects but mainly, its pedagogical adjustment to the context in which it is inserted. Corredor [1993] notices the importance of assuring the student’s individual style. She considers that the instruction must be individualized, to facilitate the arising of conceptual structures, accordant to student’s capacity and interest. It points to the need of a reflection on the teaching program and learning styles. In particular literature, there are few references to the use of pedagogical theories upholding student model’s cognitive conception. The exceptions are found in Akhras & Self [1995] works, related to the constructivist theory and in systems based on agents, supported by Vygotsky theory [Frasson, 1996].

However, the value of pedagogical theories on ITS is not an unanimity. Regian & Shute [1994] believe that theories do not assure the mechanisms of information acquisition or real knowledge representation. They also consider that the benefit of pedagogical theories can only be determined by empirical tests. Then, a question remains: how can the ITS interact with the student in an individual way, respecting his cognitive style, tutoring effectively his study session, keeping his constant interest, and presenting the topics in an attractive, creative and suitable manner to his knowledge level?

2.4 THE INTEGRATION OF ITS TO WWW

The World Wide Web is becoming the most popular technology of the Internet and an important device for educational software distributing. This fast outgrowing has done the educational software developers to reconsider the current ITS architectures. The integration of ITS to WWW has aroused many discussions and created promising experiences. The WWW offers the hardware platform independence, but the ITS supports flexibility and intelligence guidance [Gold, 1997]. This integration offers to users the accessing to ITS from remote places in different periods. However, new challenges emerges:

- the adjustment of existing systems to the WWW environment is not trivial [Stern, 1997],
- the WWW time required for the answers is still very long, and this could unmotivated the student [Okazati et al, 1997],
- the control of student navigation becomes more difficult, because the WWW browser has its own control buttons [Stern, 1997; Okazati et al, 1997], and
- the time of multimedia resources transmission, used by some ITS applications, is long [Gold, 1997].