reformulate the diagnosis or the accessing for medical tutors built in Sisautor. It is possible since HyperDoctor can access the Sisautor external medical tutors, triggering the necessary tutoring. The system also warns the student when he/she is firing important and primary steps, along the search of the diagnosis. The HyperDoctor main goal is not only help students to diagnose correctly patient heart diseases, but help them to do it in optimal time, considering their stereotypes, and without spent unnecessary resources.

Fig. 2 - Interface Design - HyperDoctor

The student architecture proposed on Vassileva [1994] is being studied to be possibly used in the system. It that permits an adaptive navigation, through a layer structure, which handles the result of associating the student classification in one of the stereotypes added to his/her task's performance.

4. CONCLUSIONS

The authoring factory HyperDoctor was designed for building intelligent hypermedia tutors in medical education - cardiology area. The main advantage of an authoring factory
seems to be the reusability, and as result, higher development productivity. The hypermedia web shaping, the student and teacher models can easily be reusable. Therefore, some impasses must be solved. The theoretical foundations of our work are based on the software engineering. Then, the development of the authoring factory for building intelligent hypermedia tutors must be supported by software engineering development methods.

The lack of a particular development method guided us to analyze over and again the effectiveness of Extended-KADS. Extended-KADS supports modeling and specification of expert systems, usually met in the ITS, however expert systems do not handle with student model or teacher model. We is facing this challenge. **HYPERDOCTOR** is a work-in-progress and many implementation questions related to the system are opened. The development scheme, where intelligence is gradually inserted on the ITS, was being adopted. In one hand, it can make the process surer and reliable, but in other hand, the system and the tutors generate with it can lose power, even remaining more flexible than conventional tutors.

OpenScript, the ToolBook event-oriented programming language, and its authoring system have been considered as the potential environment for system developing. In previous implementation of intelligent educational environments [Crespo et al, 1995; Glanzmann et al, 1995], OpenScript plus C++ libraries (for the inference engine construction), added to commercial databases, such as FoxPro and Access, (for handling student model bases) was used as the software environment. In these cases, the mix of software performed well. So, in **HYPERDOCTOR**, the same strategy is being considered. In addition, we are aware of the WWW spread and importance. Moreover, now it is feasible developing plug-ins and applet Java to promote the communication between the medical tutors built in **HYPERDOCTOR** and WWW sites.

**HYPERDOCTOR**'s pendent questions require a hard work and some solutions must be refined, mainly those related to knowledge representing and handling. Suggestions, technical comments and recommendations are welcome.

**References**