

Database Requirements of Knowledge-based Production Scheduling and Control: A CIM Perspective

Dr. Ulf Schreier

AHP Havermann & Partner GmbH
Moostraße 5
Starnberg, Germany

A leitstand system is a scheduling and control station that plays the following role in computer integrated manufacturing. It receives information about production orders and resources from a MRP system (carrying out long and middle term production planning) and performs interactive or automatic finite capacity scheduling for the considered production resources (like machines, operators, tools, etc.). The exact schedule will be forwarded to the shop floor. During and after execution the resources report back to the leitstand data about the actual performance. The leitstand can use then the up-to-date information about the actual status for rescheduling. An essential technical goal of KBL is to support easy customization for various production and system environments. This requires a flexible knowledge representation.

A leitstand is therefore a scheduling, control and information system based on knowledge and complex graphical information. Frequently, it is also a distributed system, when several leitstands have to cooperate. This results in many different requirements for a database system:

- Tight integration of a knowledge base and its interpreter with the database: Scheduling and processing of control information has to be fast and flexible. Smooth integration of core and shell is required, especially a database meta schema that is not only readable, but also writeable and executable.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, requires a fee and/or special permission from the Endowment.

Proceedings of the 19th VLDB Conference
Dublin, Ireland, 1993

- Automatic scheduling and the graphical presentation of a schedule requires a fast traversing and processing of many data objects.
- In order to carry out "what-if" analysis users will create several schedule versions. A special notion and implementation of version management is therefore a requirement, too. Especially refreshing of versions is needed in order to keep plan versions up-to-date with information of the actual shop floor performance.
- As a control system the database has to be active in order to react (even in real time) on data events. Conventional trigger systems do not support very well data streams having many applications interested only in some selections of arriving data (the events) by obeying response time restrictions.
- A leitstand is also a kind of a classical information system with information entered in forms and retrieved by queries.
- If several shop floors cooperate, the leitstands have to cooperate. The information flows through the various shop floors and leitstands. The overall information system has to keep track of the current state in order to answer questions like "What shop floor processes currently a certain order?". Furthermore, a leitstand wants to access information of "neighbored" leitstands by keeping the site autonomy. In case of network failure the leitstand should continue to run.
- Integration into enterprise-wide information systems is another issue. Companies start to develop enterprise-wide data models and data dictionaries

integrating the various existing databases. How can a dedicated sub-system as a leitstand using a special database be integrated?

Most of the single requirements in this list are known in database research. From the CIM perspective, some of the proposed solutions go into the right directions, others do not. Some of the problems are supported better by relational database systems, some are better supported by object-oriented databases. But, how can all these different and conflicting requirements be integrated into one system? In my opinion, this require a multi-language access to the data. In our application, the same data require tight integration with

- a compiled language like C+ in order to implement fast automatic scheduling,
- an interpreted language like Smalltalk in order to implement flexible graphical interaction,
- a 4th generation language for simple data maintenance programs,
- SQL for queries,
- a problem-oriented language to implement knowledge-based features and
- an event language to interpret new data.

Relational database systems however support only a relational language dialect, and object-oriented database systems usually just one object-oriented programming language.

AHP develops together with partners a knowledge-based leitstand (KBL) for production scheduling and control using an object-oriented database. KBL provides a simple knowledge representation language with a core of predefined functions (written in C++) and a shell of user-definable functions. The development is funded by the European Commission as a Esprit project. KBL will be the successor product of the current AHP leitstand system.