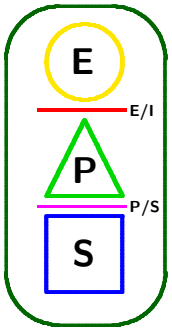
# MODELING DATA PROCESSING IMPLEMENTATIONS

## Operation-Centric, Structure-Dependent, & XML Data Models

In the 1970s there was a revolution in the technology of modeling data processing systems. The current XML-revolution could either enhance or threaten the last thirty years of data processing progress. To avoid the later, definitive action needs to be taken soon. Both revolutions relate how users view data external to a computer and how data is actually processed and stored internally to the computer. The difference lies in the *operation-centric* nature of one, and the *structure-dependent* nature of the other.

**Structure-Dependent Data Models:** A *UPS* schema depicts the relationship between *User Friendly* (*U*) data which is *brain-oriented*, *Processor-Friendly* (*P*) data which is byte-oriented, and *Storage-Friendly* (*S*) data which is also byte-oriented. The *U/P* interface between brains and bytes is accomplished by: keyboards, printers, video & audio devices, mice, joysticks, etc. The byte to byte *P/S* interface is commonly known as the I/O interface, supported by data access methods. The challenge of every implementation is to provide a user-friendly view of data, map it to a processor-friendly representation, store it in a storage-friendly form, and then reverse the process. Prior to the early 1970s, the data modeling problem was solved very simplistically: present data to the user in its native form, let the processor process data in its native form, and then store data in its native form. The data representations are all the same: user-data is processor-data is storage-data. However, this *structure-dependent* modeling does not provide the most convenient data representation for the users, nor the most efficient form of data for processing, nor even the most economical way to store data, nor is it easy to implement, and it is almost impossible to model formally; but it is very easy for developers to conceptualize.

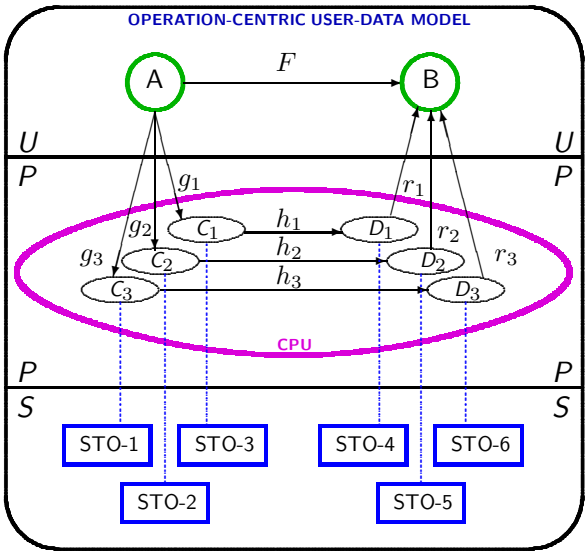


*IAS PLATFORM*

**Operation-Centric Data Models:** In 1970 the Relational Data Model (RDM) introduced an *operation-centric* model for *user-data*. As illustrated in the *UPS* diagram, $F$ operates on A to produce B, where A and B represent user-data expressed as *RDM-tables*; and $F$ represents a sequence of *RDM-operations*. The power of the RDM comes from its mathematical foundations. All *RDM-tables* and *RDM-operations* are mathematically well-defined, thus assuring consistent and reliable results. Unlike structure dependent models, the RDM does not required any knowledge of how A and B are represented as *processor data*:

$$\mathsf{B} = F(\mathsf{A}) = r_i(h_i(g_i(\mathsf{A}))).$$

This *data independence* at the *U/P* interface is unquestionably the most significant contribution of the RDM. It provides functionality, convenience and performance not possible with structure dependent implementations. Data independence is not yet common at the *P/S* interface, since most data access methods still rely on structure dependent technologies. Today, the best systems are hybrids with operation-centric *U/P* interfaces and structure-dependent *P/S* interfaces. If progress continues, entire systems will eventually become operation-centric, providing data independence at both *U/P* and *P/S* interfaces.




**XML Data Models:** Unfortunately, emerging XML systems may not support data independence at any level. Any implementation of XML data using a native XML data representation for user-data, processor-data, and storage-data would ensure all the structure-dependent disadvantages of pre-RDM systems, providing no data independence of any kind! Unlike the RDM, there is no widely accepted operation-centric model for describing and manipulating XML data. Yet, if the potential advantages of XML data are to be incorporated with the accumulated progress of the last thirty years, an operation-centric XML data model needs to be adopted, and at least one such model already exists. Extended set processing (XSP) technology provides a formal foundation for the integration and implementation of XML and RDM data processing systems with full data independence.