# Evolution of an Intelligent Information Fusion System

*William J. Campbell* and *Robert F. Cromp*
National Space Science Data Center, NASA/Goddard Space Flight Center, Greenbelt, MD 20771

ABSTRACT: In this paper we argue that, without radical new thinking in data and information management hardware and software designs, there is little hope of managing the enormous amount and complexity of data that the next generation of space-borne sensors will provide. An anthology is presented illustrating a logical evolutionary path that has taken place in artificial intelligence, science data processing, and management from the 1960s through a futuristic look into the first decade of 2000. Problems and limitations of technologies, data structures, data standards, and conceptual thinking are addressed. A new approach proposed by the authors is presented on the development of an end-to-end Intelligent Information Fusion System (IIFS) that embodies knowledge of the user's domain-specific goals as well as automatically generating meta knowledge about objects or features of interest in near-real-time.

## THE EVOLUTION OF HANDLING SCIENTIFIC DIGITAL DATA

IF WE TAKE A CURSORY LOOK at the recent past, say *circa* 1970, of available technologies and methodologies for handling scientific digital data, a logical evolutionary path has taken place. Space-borne sensor systems had moderate data rates and resolution and most ancillary data were analog with greatly varied formats. Computers and storage systems were mainly comprised of mainframes and batch processing and we saw the first generation of super computers in operation. The software area had no data standards; primitive, no-frills compilers (FORTRAN, BASIC, PL/I, etc.); and minimal spatial data handling capabilities. Workstation technology was not available, image display and analysis tools were expensive, and performance was limited. User interfacing was usually done with cumbersome JCL cards or paper tapes and information management was nothing more than simple record management systems. Communication data rates were usually limited to TTY performance (300 to 1200 baud). The 1970s saw the realization of time-sharing environments and the prevalence of line editors.

"A typical best-case scenario for the management of digital scientific data has rarely been anything more than the archiving of a computer-compatible magnetic tape with an analog catalog briefly describing the tape identification number, data generation source, and time and location of acquisition. If a researcher wished to locate and browse the data, he had to know where the data set was archived, the specific data sets of interest, and who to contact to order the data. Once this effort was completed, the archiving agency retrieved the tape(s) from the archive, copied the data, mailed the tapes to the user, and returned the original tape(s) to the archive. The user had to verify the shipped data to determine that it was indeed the most appropriate in supporting the particular scientific research. If the data set was not what was requested or it was incomplete, then the entire process was repeated, often taking months to complete" (Campbell, 1989). If the data were what was needed, the user had to ensure that the system at his location was compatible (the software was appropriate for the format) to load, review, and finally permit the pursuit of the science. Figure 1 is a cartoon that illustrates the 1970's environment.

## SIGNIFICANT PROGRESS

In the 1980s, sensor systems had higher data rates and resolution, and some ancillary data was in digital format. Mainframe computers still with batch processing were commonplace and second generation super computers were crunching away. Affordable super minicomputers arrived with mainframe performance run by interactive operating systems (e.g., VMS and UNIX). We also saw interactive development environments, robust compiling and debugging tools, full-screen editors, and the useful emergence of expert systems, natural language query processors, and three-dimensional (3D) graphics. User interface and analysis tools were established on personal computers that also supported image analysis and display. Local area networking (baseband, broadband) telephone communications were humming along. Commercially supported spatial data handling systems made a big impact and data base management systems (DBMS) and machines were deeply embedded not only in the commercial and business sector but also in the science community.

Such systems provide a capability to "find, sort, merge, organize, update, and output diverse data types. However, most DBMSs have been designed and developed specifically for archiving and managing data for a specific domain and, in many instances, by developers with a background in computer science or related fields" [Bic, 1986]. The result is that these systems suffer from the intrinsic flaw of not effectively providing the capabilities needed by a casual or new user. In addition to this disadvantage, these databases limit the capabilities for managing the syntax of a domain as part of its data structure, have limited data structures which cannot represent explicit relationships between data classes, demand precise mathematical query formulation for database interactions, exclude many of the data objects used in the application domain, and, finally, do not efficiently store, index, or retrieve (i.e., in relational systems) image or spatial data (Bic, 1986; Yao, 1985). The significance of this is that the users of existing database systems require an in-depth understanding of the database architecture, data content, location, and query language in order to use the database effectively. When data structures exist in a large database (i.e., schemata), this problem is exacerbated, often making the database unusable in an operational environment" (Campbell, 1989). Figure 2 represents the state of information systems in the 1980s.

## LAYERED APPROACH

One possible 'fix' to this problem is the intelligent layering of a variety of available technologies that adds syntax, semantics, and pragmatics to these systems, thereby increasing performance while at the same time simplifying and facilitating the understanding of the various complexities of the system for the user. We have built several prototypes of this concept. One such system is a large scale domain-independent spatial data management expert system that serves as a front-end to data-

PHOTOGRAMMETRIC ENGINEERING AND REMOTE SENSING,
Vol. 56, No. 6, June 1990, pp. 867–870.

0099-1112/90/5606–867$02.25/0

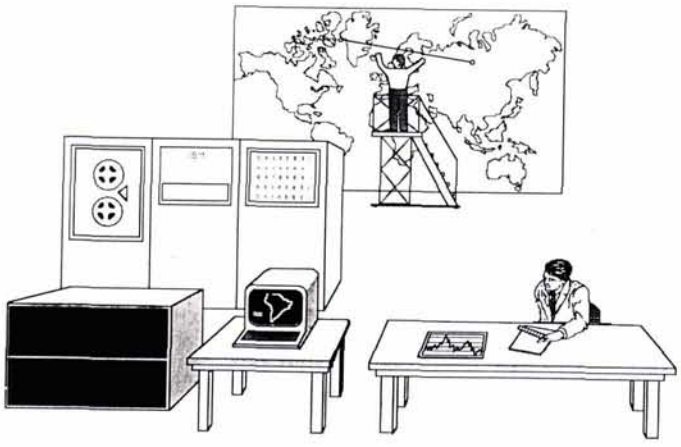

FIG. 1. The state of information systems, 1975.

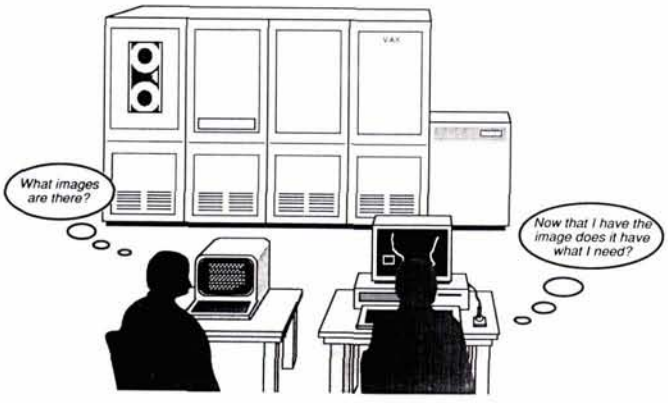


FIG. 2. The state of information systems, 1985.

bases containing spatial data. This system, described in Cromp (1989), uses spatial search techniques to generate a list of all the primary keys that fall within a user's spatial constraints prior to sending the query to the remotely distributed DBMS. This process significantly decreases the time required to answer a user's query. In addition, a second domain-independent query expert system uses a domain-specific rule base to preprocess the user's English query, effectively mapping a broad class of queries into a smaller subset that can then be handled by a commercially supported natural language processing system. The integration of natural language, expert systems, and 3D graphics provides a very powerful interface to a relational DBMS. It can link different worlds (e.g., database, graphics, satellite imagery, etc.) over a network by reducing user goals into a plan based on processor/network loads, the type of application, and specific syntax.

It should now be obvious that, even with the layering concept described above, performance limitations still exist specifically in the underlying data structures and relational data models. The solution is also not in applying a 'brute force' approach of bigger and faster hardware. The enormous amount and complexity of new data that space missions planned for the 1990s will generate (e.g., the Earth Observing System alone will generate over 4 trillion bytes of new data per day) will simply overwhelm the most sophisticated hardware architectures envisioned.

## INTELLIGENT INFORMATION FUSION SYSTEM

So what is the answer to the data glut for the 1990s? A new approach proposed by the authors is the development of an end-to-end Intelligent Information Fusion System (IIFS) that embodies knowledge of the user's domain-specific goals as well as automatically generating meta knowledge about objects or features of interest in near-real-time. We feel that this is the key to surmounting the impending data chaos problem and providing the users with useful information. Intelligent interactive development environments for planning, scheduling, and knowledge-base creation will be needed. English-like programming languages that support parallel computer system architectures and intelligent systems with 3D and 4D graphics (including time through animation) for interactive browse and display is crucial. In the 1990s, we will see mainframes and super computers merged with intelligent robotic mass storage support in the teraflop range. Parallel system hardware and software will be in vogue, and workstation environments for development and analysis will have performance that rivals today's mainframes (50 to 100 MIPS). Communications will be uniform and standardized with high speed fiber optics and global coverage with multiple tracking and relay satellites.

The user interface will be dominated by icons, graphs and supporting visualization cues, high definition color displays, and voice activation. We will see knowledge-based spatial data handling systems and database management systems that are object-oriented. Combinations of artificial neural networks, expert systems, and object-oriented environments will provide the capability to intercept automatically and in near real time a satellite data stream, identify objects of interest (e.g., boreal forest cover in an image), and then characterize and label the objects [Campbell, 1989]. Then, by using a rich and robust domain-specific knowledge base, the system will be able to verify and combine other attributes such as location, time sensor, etc., to the object(s) and ingest this into an object-oriented DBMS. A remote user will communicate with the system using natural language queries drawn from the domain, interacting through a combination of typing and voice. The user will be able to interrogate the system as to what new data have been ingested since his last interaction and to retrieve data of interest in near real time for further manipulation at his own site.

The advantage of this approach is that the relevant contents and meanings (i.e., relationships) of the data as well as the relation between objects (i.e., actual database objects or clusters of objects) within the data structure all work together in a seamless environment, optimizing system and user performance. This approach also supports approximate reasoning to infer conclusions that are not explicitly stated by the user. Approximate reasoning may include, but not be limited to, the posing of imprecise queries which can be stated without mentioning data names or operations. Furthermore, it can provide the casual user with a logical representation of the database architecture, the stored data, and the analysis procedures. In other words, the IIFS concept accommodates the needs of the user at a level appropriate to his skills for interacting between the database and his specific scientific discipline. Figure 3 is a top level architectural view of this concept.

Scientists will also have the ability to change the local knowledge base (that is, a subset of the central knowledge base), the database schema (e.g., into views), and the data specific to the user's needs. Value-added knowledge and data results can then be returned to the central source if the user so desires, providing a distributed knowledge-based interactive blackboard. Figure 4 illustrates the 1990s environment.

If the concepts we presented for the 1990s proved effective, then if we let our imagination wander just a bit, the scientific computing environment for the 2000s would look like that described in the next section.

## CEREBRAL SAVANT MODEL

The start of the twenty-first century will introduce intelligent on-board satellite data processing and real-time episodic monitoring with built-in alert mechanisms that automatically capitalize on the most effective combination of sensors to capture data of interest. These systems will also have the ability to monitor the health and safety of the platforms as well as the individual instruments. They will be able to correct errors automatically and even point at special features or objects that are new or that they have been programmed to look for. Robotics will play a critical role in these systems in performing experiments and even doing on-board repairs or additions to modules. Such a system will alert the ground monitoring system of any change, either on-board or in what is being sensed, and the users will have the capability to make corrections and even control the satellite-based experiments or sensors as they desire. All ancillary data will be in digital format and hyperlinked to the real-time monitoring and data ingest module.

There will be four primary types of computers: super, database, workstations, and special purpose. The super computer will be modeled after the human brain (hence the name Cerebral Savant), and have extensive distributed parallel system architectures in the mega trillions of floating point operations per second range (teraflop). The 1990s personal computers and workstations will have merged and perform in the multi-billions of floating point operations per second (gigaflop) range and be interconnected to the super computing and database systems in a total transparent fashion. Intelligent operating systems will give expert advice in English (or whatever language is required) and have interactive learning. This will be accomplished by having a system-wide 'curiosity' sub-module running in the background that monitors all system functions and user interactions and even suggests more efficient user methods or new but different information about a given query. The input/output will have much higher bandwidths because the system will provide multiple ways of interacting with the user. When a user has finished a dialog with the system, the system would then automatically decompose the query into chunks of domain specific data and send each chunk to the most appropriate device for processing. For example, an intensive numerical computation problem would be sent to a pipelined architecture machine, data that are appropriate for a parallel machine would go there, data queries could be sent to a database machine, etc. Once each job has been completed, the system would reassemble the data chunks for verification by the user or for further processing locally or centralized.

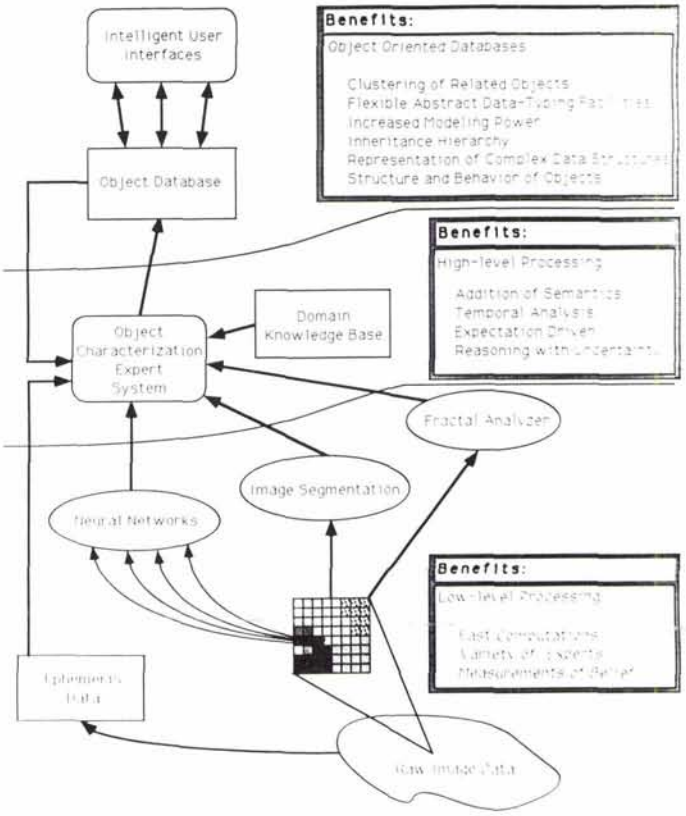


FIG. 3. Top level architectural view of an intelligent information fusion system.

For example, the user can suggest phrases or words that have similar meanings to other words or phrases, draw icons interatively (and of course in color), and tell the system "it looks

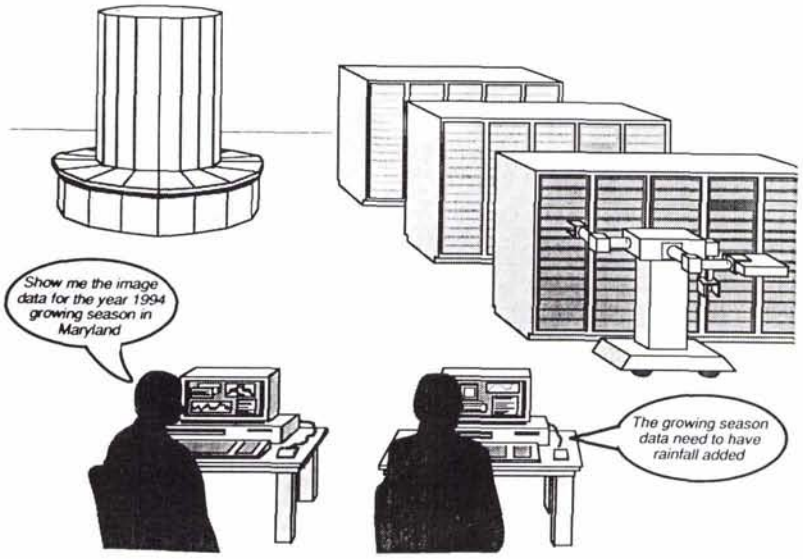


FIG. 4. Future directions, 1995.

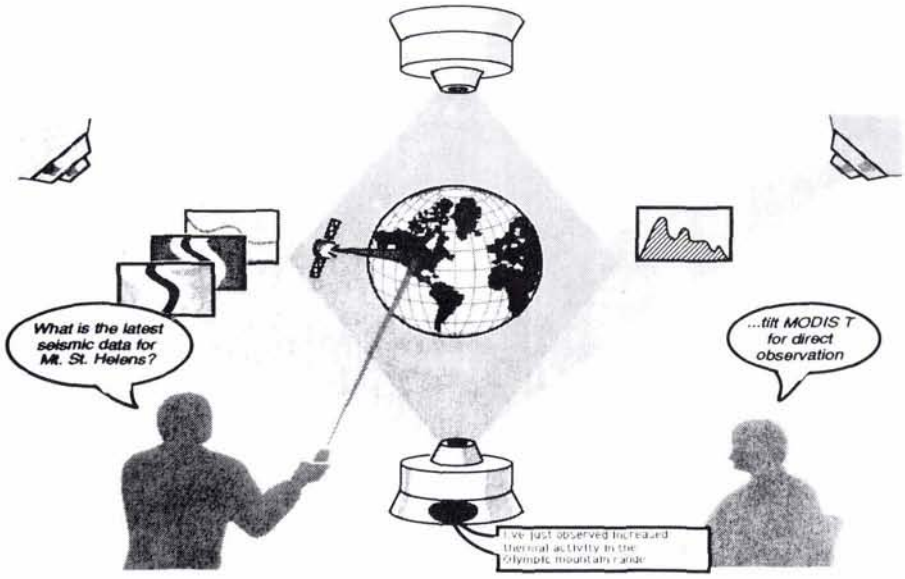


FIG. 5. Future directions, 2005.

sort of like this, go find it." Visualization and sound will play a major role in determining the optimal method for determining the best approach to the problem. Most of the software throughout the system will have intelligent, creative environments with human level capabilities, and automatic programming and learning will be based on problem discretion and interactive resolution. Critical to the understanding of data will be data representation and displays with real-time 3D holography, very high resolution, distributed electronic notebooks, and, of course, high speed digital communications that support real-time computer interactions. Figure 5 illustrates this concept.

## CONCLUSION

Satellite remote sensing systems now waiting for launch or being planned (e.g., Space Telescope, Space Station, EOS), will generate more data in one year than all previous systems combined. For example, the National Space Science Data Center (NSSDC) has been in existence for over 20 years with a mandate to store and disseminate remotely sensed data to the user community. As of this writing, the NSSDC has accumulated a little over 6 terabytes of data in those 20 years. The Earth Observing System alone will generate approximately that much data in 32 hours of operation. In an era of potential environmental threats of global consequence such as ozone depletion, greenhouse effect, deforestation, desertification, acid rain, and toxic waste, getting relevant data to the scientist in a timely and useful fashion will not only be helpful but could be critical to humanity. Hopefully, the technologies and architectures we've described will overcome the problems in data management of the past. We realize that these innovative ideas will require significant risks; however, if we don't take those risks, we believe there is little hope of intelligently managing our planet.

## REFERENCES

Bic, L., and Gilbert, J., 1986. Learning from AI: New trends in database technology, *IEEE Computer*, Vol. 19, No. 3, pp. 44–54.

Campbell, W. J., N. M. Short, Jr., and L. A. Treinish, 1989. Adding intelligence to scientific data management, *Computers in Physics*, Vol. 3, No. 3, pp. 26–32.

Campbell, W. J., S. Hill, and R. F. Cromp, 1989. Automatic object labeling and characterization using artificial neural networks, *Telematics and Informatics*, Vol. 6, No. 3/4, pp. 259–271.

Cromp, R. F., and S. Crook, 1989. An intelligent user interface for browsing satellite data catalogs, *Telematics and Informatics*, Vol. 6, No. 3/4 pp. 299–312.

Yao, S. B. (ed.), 1985. *Principles of Database Design*, Prentice-Hall, Inc.

(Accepted 18 January 1990)

## Erratum

Equation 2 on page 177 of the article, "Application of SPOT Data for Regional Growth Analysis and Local Planning," by Ehlers *et al.* (*PE&RS*, February 1990) should read

$$\pi_L(S\%) = N_c / [N_c + (N - N_c + 1)F_{n_1, n_2}].$$