A microcomputer-based farm management/operating system

LAURENT GAUTHIER and ROBERT KOK

Agricultural Engineering Department, Macdonald College, McGill University, 21,111 Lakeshore Rd., Ste. Anne de Bellevue, Quebec H9X 1C0. *Received 3 April 1987, accepted 24 September 1987.*

Gauthier, L. and Kok, R. 1988. **A microcomputer-based farm management/operating system.** Can. Agric. Eng. **30**: 69–76. The intelligent agricultural decision support and management/operating systems of the near future will require very flexible architectures in order to support the high levels of functionality and effectiveness which are prerequisite to their adoption by the farming community. The number and diversity of the tasks that such systems must embrace is discussed. The implementation of these tasks will require open, nonrestrictive data/knowledge bases capable of expressing all possible relationships between farm entities and events. The authors assess the type and contents of the data structures required to build intelligent computer-aided farming systems and propose a conceptual data model to support the development of such systems. Their conclusions and proposals are based on a 4-yr effort at designing, implementing and using a general farm record keeping system.

INTRODUCTION

Recent technological breakthroughs in the fields of microelectronics and artificial intelligence have the potential to greatly increase the amount and quality of information available on farms and consequently aid farm managers to rationalize the use of farm resources. VLSI circuits, knowledge bases, expert systems, computer graphics, adaptive control and robotics are all part of the new generation of intelligent and flexible tools which will affect production agriculture within the next few decades (Holt 1985). These technologies are voracious users of information. They require exhaustive, detailed, accurate and up-to-date data to operate effectively. The design and creation of systems to acquire, structure and deliver this information poses an interesting and original challenge to professionals providing support and services to agricultural producers.

The envisaged intelligent agricultural decision support and information systems will require very flexible architectures to support the high levels of functionality and effectiveness which are prerequisite to their adoption by the farming community. The number and diversity of the tasks that such systems must embrace is considerable. Open, nonrestrictive data/knowledge bases capable of expressing all possible relationships between farm entities and events will be required. The knowledge bases must use logical and consistent data structures to efficiently and reliably "remember" or locate events and resources in time and space. For comparative analysis and decision optimization purposes historic records must be kept and tagged with a temporal attribute. There is obviously a high cost, at today's standard, in terms of hardware and software resources required for such systems. The past and continuing increase in performance and capabilities of computer-based technologies allows, however, the design and prototyping of systems which may be out of reach of today's average microcomputer but will be well within the capabilities of tomorrow's machines and devices. In this paper we have assessed the type and contents of the data structures required to build the intelligent computer-aided farming systems of the near future. Our conclusions are based on a 4-yr effort to design, implement and use a general farm record keeping system (Kok and Gauthier 1986; Gauthier and Kok 1985).

FUNCTIONAL REQUIREMENTS

The design and implementation of a comprehensive, integrated and intelligent farm-operating/management system represents a formidable challenge. The diversity of both the environment in which it must operate and the functions it must accomplish requires the use of innovative methodologies and architectures. Computer technologies in the area of information systems, simulation, expert systems, robotics and man-machine interfaces will have to be integrated into a single, logically consistent and open framework (Deering 1985). The aforementioned technologies are evolving side by side at a rapid pace and are exerting considerable influence on agricultural research and production (Riddle 1985; Sonka 1985). This trend will most likely continue. The authors share with other observers (Devlaeminck 1985; Holt 1985; Schueller et al. 1985) the opinion that the potential of the various technologies will not express itself fully nor reach widespread adoption in production agriculture until functional integration is achieved. A list is presented in Table I of the desirable functional requirements for a farm-operating/management system in order of increasing intelligence levels.

The effectiveness of a given system will depend on the ability of the various functions and components to interact and share common data structures. The relationship between functions is illustrated in Fig. 1 grouped around three main poles: (1) cognition and communication, (2) data and knowledge management, and (3) information processing. The information and control flow throughout the system is handled by an "intelligent arbitrator." In a truly intelligent system, this arbitrator will make the strategic decisions affecting system behavior and performance, thus constituting the high-level "reasoning ability" of the system. High priority, direct functional links between poles will be necessary as safeguards against hazardous arbitrations. As well, each subsystem (e.g., database, robot, irrigation system) will require a minimum level of resident or local intelligence and operating limits to enable its operation in a standalone mode and to ensure self-preservation (e.g., overheating in a greenhouse, division by zero, information destruction etc.), thus implementing a hierarchical, adaptive control system (Kok and Desmarais 1985).

Cognition and communication

The data acquisition function is fundamental to any intelligent decision support system. The latter must be able to determine

Table I. System functions and technologies

Function	Examples	Technology
Data entry	Inventory of farm resources, production records, accounting data	Interactive computing
Financial management	Accounting, budgeting, production cost calculation, tax forms preparation	Computer algorithms, procedures
Resource management	Machinery selection/ calibration, crop rotation, ration formulation	Flexible manufacturing systems
Automatic data acquisition	Meteorological data, milk/egg production, stock levels, drainage rates	Analog/digital conversion, sensors, transponders
Communication	Electronic mail, remote database access, buyers, suppliers, consultants	Modems, telemetry, local area networks, Videotex
Data management	Classification/storage/ retrieval of information, archiving, reporting	Relational/network databases, query languages
Process/ environment monitoring	Milking, feeding, seeding, irrigation, greenhouse, barn, crops, soil structure	Multitasking O/S, distributed processing
Process/ environment control	Heating, irrigation, ventilation, storage chambers, grain drying	Digital/analog conversion, real-time computing
Simulation	Crop/livestock growth, water use, pest populations, market behavior	Mathematical modeling, real time graphics
Scheduling/timing	Field work, harvest, routine maintenance	On-line wear management
Model calibration/ validation	Hydrological, meteorological, biological, financial, economic	
Knowledge management/ expert systems	Pest/disease diagnostic, marketing, teaching, strategic planning	Symbolic computing, logical inference, knowledge bases
Mechanical/ electronic aids	Tilling, seeding, harvesting, cleaning, milking, shearing, feeding, surveillance	Robots, vision systems, legged machinery
Natural language processing	Audio interaction, text analysis/synthesis, friendly interfaces	Speech synthesis/ recognition, context sensitivity
Cognition/learning/ reasoning	Knowledge generation/ evaluation, pattern recognition, behavior acquisition	Fifth generation computers, parallel processing
Creation/ imagination	Design of equipment, processes, models, adaptive behavior, scenario generation and evaluation	Artificial intelligence

and evaluate the response to any action it undertakes (controlled variables) and to assess the current state of the environment in which it evolves (climate, market conditions, crop and livestock health, etc). This implies the use of sensors, data links to other systems and interfaces with human operators. At the same time, "instructions" must be transmitted to robots, process controllers and suppliers/buyers of goods and services.

Data and knowledge management

The knowledge and information acquired by the system must be "memorized." It must be structured, classified and processed. The basic data structure used must be flexible enough to accommodate a changing environment and an evolution in the number and types of needs and functions supported. As the system "learns" and increases its "intelligence," the memory function will grow in scope and sophistication. The distinction made in traditional databases between the data structure and its contents is somewhat artificial and must be relaxed. Does the container define the contents or vice versa? It is important to find ways to subjugate real-world objects and events to a high-level, generic data structure and to locate them within a continuous time frame. In fact, it can be argued that information deletion or removal should not be allowed and that the capacity to handle conflicting information must be built into the system. Integration of data and knowledge is essential and even inevitable within intelligent structures. Thus, the information storage architecture must accommodate descriptive and historic records of the farm environment as well as production rules (expert systems) and mathematical models (simulations). The resulting knowledge base can then be used in decision making or support and strategy generation and evaluation.

Information processing

The information acquired and memorized by the system must be processed in order to generate actions or decisions and to enhance the existing knowledge. The different processes include (1) calibration and evaluation of models and validation of knowledge bases, (2) simulation of physical phenomena for decision support and forward planning, (3) analysis and synthesis for budgeting, scheduling, reporting and optimizing, and (4) inference drawing by consultation systems for diagnosis, monitoring and control (McKinnion and Lemmon 1985a). These processes can be triggered by scheduled tasks or queries from users and remote systems. The required inputs must be supplied either by the outside world (e.g., directives from the user) or from the system memory. The resulting output is then routed back to the prompting agent or to the internal data/ knowledge structure for future use.

DESCRIPTION OF EXISTING PROTOTYPE

The functional description presented above has emerged from a 4-yr effort to implement a prototype farm and crop record keeping system. This prototype was installed on the Macdonald Research Farm on an IBM XT with 512 Kb of memory, a 10-Mb hard disk, a monochrome screen, a color graphics adapter and a graphic dot matrix printer. The package has been under evaluation for the past 2 yr and is being used on a daily basis by farm personnel for record keeping and report preparation. It has been described in detail by Kok and Gauthier (1986).

The database management product selected (MDBS III from Micro Data Base Systems, Inc.) uses an extended network architecture (Holsapple 1980) supporting a variety of data and set types as well as range and access checking clauses. The data

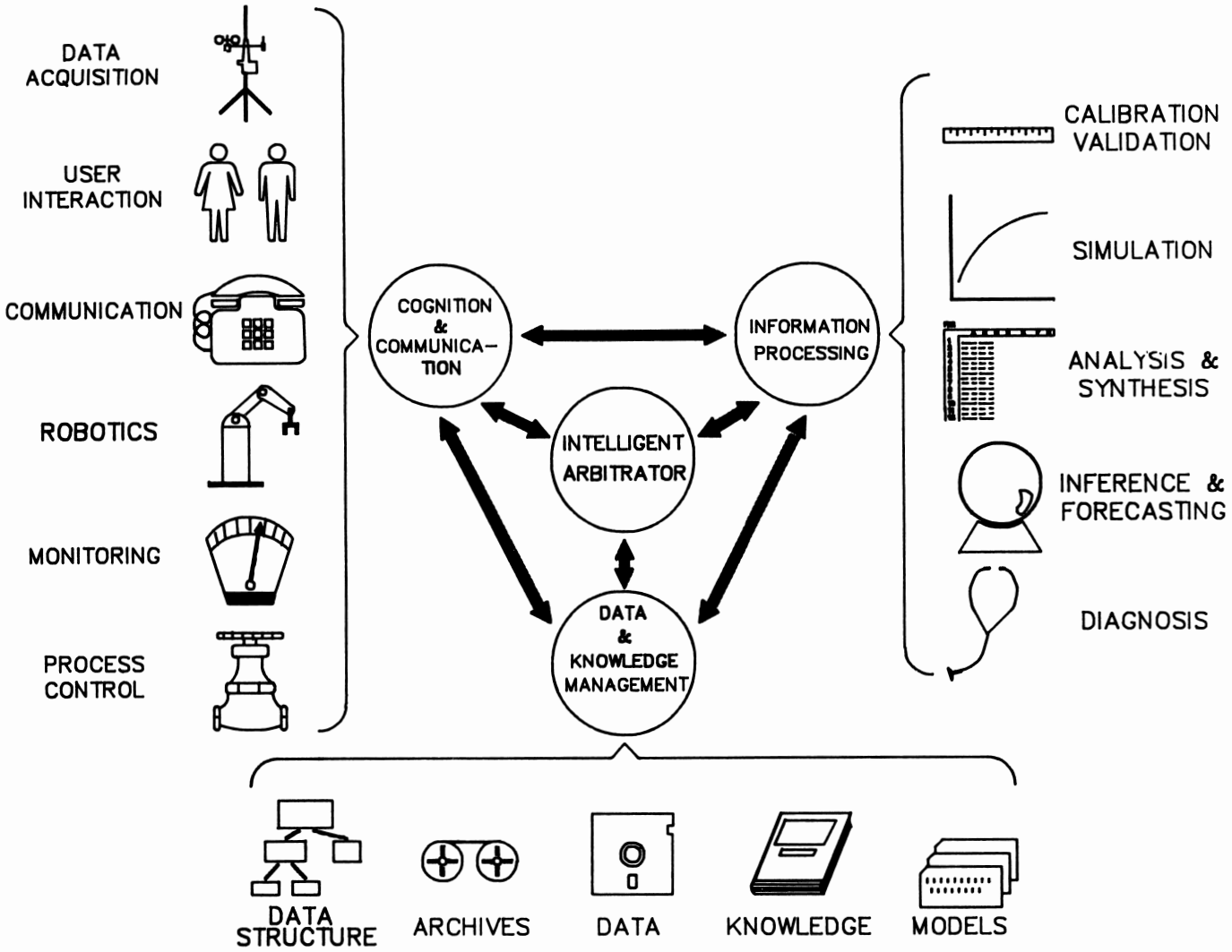

Figure 1. Functions and components of an intelligent and integrated farm-management/operating system.

schema devised for the prototype makes use of 20 record types and of several kinds of relationships and sort clauses. Four categories of information are distinguished: (1) attributive (information describing resources), (2) seasonal or annual (crops, rotations, yields, soil analysis etc.), (3) inventory (operating inputs and outputs) and (4) farm operations (tilling, seeding, spraying, repairs, etc). The program package was written in compiled BASIC and consists of a hierarchy of menu-driven modules and submodules for data input and editing, report preparation and printing, chart/map generation and output. Extensive error checking and input validation is used to ensure data integrity and enhance user friendliness. The program package has proven to be friendly and easy to learn, requiring very little user supervision by system implementers. The reliability and performance of the overall system is considered adequate given the size and complexity of the database and software.

The database schema was designed to integrate all the information required for crop production management on a large dairy farm. It supports the recording and archiving of complete field, machinery and product histories (field operations, crop rotations, treatments, yields, maintenance, use, etc). The representation of such information structures was found to be problematic due to the complex and variable nature of real-world operations. The use of operation-specific record and relation types did not allow the design of general data entry and editing procedures. The fixed and deterministic nature of the data schema thus restricts the application developer and consequently the user to a single information representation model for each type of event. For example, a harvesting operation for a given field can yield more than one product, it can extend over several days or weeks and use a variety of tractors, implements and inputs. The data schema used in the prototype can handle the variability of such an information structure but only at the price of extensive programming and system complexity. These observations have prompted a search for alternative data models which can represent all farm objects and events in a logical and consistent manner.

In the existing system, day-to-day farm operations are manually recorded on worksheets by employees and subsequently entered in the database by clerical staff. This procedure works well despite the inevitable delays, misinterpretations and omissions. The forthcoming use of mechanical/electronic devices to directly measure and record the various operating parameters (e.g., yields, inputs, time, distance, etc.) will undoubtedly increase the accuracy, integrity and completeness of the collected data and augment the effectiveness of the overall system considerably.

The information retrieval modules can supply the user with a variety of device-independent outputs in the form of tables, charts or maps. Reports are customized by the user through various menu-driven procedures. The ease of use and efficiency of this approach is counterbalanced by its rigidity: the designer must arbitrarily limit the number and types of user-specified parameters and search criteria for report generation. The development of interactive graphics and the advent of natural language interfaces capable of processing ad hoc queries will enhance the flexibility and friendliness of human-machine dialogs. Such interfaces will require the use of data structures which allow the creation and manipulation of generic concepts.

INTEGRATED MANAGEMENT/OPERATION IMPLEMENTATION CONSIDERATIONS

A partial list of the physical and virtual technologies which must be applied or developed to implement an integrated farm operating/management system is present in Table I. The physical technologies are readily identified and have been discussed at length in the literature. The requirements for complete or partial automation of field machinery (tractors, combines, implements) have been described by Johnson et al. (1983) and Pejsa and Orrock (1983). Gerrish and Surbrook (1981) have discussed the types of sensors and functions required for the construction of autonomous robots needed for agricultural production. The required connections, interfaces and links between physical components on an integrated farm were reviewed by Moncaster and Harries (1983). McKinnion and Lemmon (1985b) discussed the capabilities of LISP computers as artificial intelligence workstations while the hardware architectures and devices needed to support logical reasoning, natural language processing and symbolic computing were described by Hewitt (1985).

Virtual technologies include the software, databases, knowledge bases, models, protocols and simulation techniques used to implement the functions of Table I. Sonka (1985) discussed the implications of changing from industrial-stage to information-stage farms. Schueller et al. (1985) described how physical and virtual technologies can be combined to implement and use flexible manufacturing systems in crop production. Smith et al. (1985) have discussed the potential applications of expert systems in crop production management and described the virtual components needed to construct such decision support tools. In general, however, few descriptions of the virtual structures required to assemble integrated farming systems can be found in the literature. The authors have worked on the development and definition of a conceptual data model to hold descriptive and historic information on farm resources and operations. This model is described below.

The various characteristics to be considered in the design of agricultural production systems are listed in Table II. The farming environment has unique and stringent requirements in terms of system performance and cost. Intelligent systems in the form of autonomous, self-guided machinery operating in open fields have to guarantee the security and integrity of the biological machines (humans, other animals) they interact with. Designers must seek to achieve intrinsically safe designs which prevent loss of or damage to farm resources (physical and virtual). The reliability and predictability of system components are also important aspects of safety. The harsh, corrosive and demanding agricultural environment requires components which can withstand severe abuse. The effectiveness of the overall design is particularly important. Appropriate and efficient designs

Table II. Desirable system characteristics

Characteristics	Aspects	Means
Safety	Protection or resources	Intrinsically safe designs
	Reliability	Hierarchically distributed safeguards
	Predictability	Multiple and redundant sensing
	Robustness	High-performance design specs
Effectiveness	Ease of use and friendliness	Human factors engineering
	Appropriateness	Systems analysis
	Efficiency	Performance assessment and optimization
	Increased benefits	
Modularity	Universality	Standards
	Serviceability	Field engineering
	Expandability	Open architecture
Affordability	Capitalization	
	Operating costs	
	Return on investment	

contribute to perceived friendliness and special attention must be devoted to human factors considerations to ensure that people do not feel alienated in any way by the technology. The latter must promise and deliver accrued benefits to its users, otherwise it will not be adopted. System modularity at both hardware and software levels is a prerequisite for the definition and adoption of industrial standards. It also allows the adaptation of systems to various physical, economic and cultural environments. Moreover, modularity has a bearing on reliability (failure of one module should not hinder the operation of the whole system), serviceability (ease of repair and replacement of parts) and expandability (configuring a given system to the needs of an enterprise). The cost of the final system has to be controlled since agricultural production enterprises have limited access to capital and must minimize operating costs. Such considerations apply to both the physical and virtual components in the system.

A CONCEPTUAL DATA MODEL

Requirements

During the development and evaluation of the prototype farm-record-keeping system it was found that the structure of the underlying data model was a determinant factor with respect to the modularity and effectiveness of the overall system. The insight gained in the course of this research program has helped identify the nature and scope of the information needed to support traditional computer-based farm management functions (accounting, budgeting, resource management, process/environment control, etc). Furthermore, it has assisted in the definition, at the conceptual level, of a comprehensive and "open" data model which can host all the data required for record keeping while providing the foundation for intelligent, knowledge-based functions such as natural language processing, robotics and expert systems. The data structure must be designed to allow the implementation of these various functions. This may be achieved with the use of a dictionary-driven generic data model. The data structure must allow the modification and

restructuring of the information held. It should also permit the definition and manipulation, at the end-user level, of generic entities while providing support for the identification and differentiation of individual, "real-world" occurrences of these entities. The advantage of using such an approach can be shown by examining the complexity of the information associated with cropping data as an example. A crop can be grown on many fields, it can be harvested several times a year (e.g., grass) or only once (e.g., grain) and can be either perennial or annual. Type and quantity of yield could be recorded per crop, harvest or field. The date and duration of all field operations should be recorded together with the equipment, products and labor used. Thus, a seeding operation might involve a tractor, one or more implements, seed, fertilizer and a hired laborer. Alternatively, it might be performed by a custom operator. Application rates for each resource (seed, fertilizer, water, labor) must be recorded and linked to the appropriate operation. In a polyculture system, a crop can be connected to a different set of fields each year. In such a management information system crops, resources and operations should be identified by generic names (e.g., corn, wheat, pesticide, nitrate, harrowing, seeding, etc.) corresponding to unique data records in the database. These records must be linked to the specific date, rates and labels associated with a cropping operation or resource. The resulting network of records and links must be structured according to a logical and consistent data model in which the relations between data records carry as much information as do the record contents.

Although it can be argued that information cannot exist without structure, the imposition of a rigid framework on the data limits the amount of "available" information. Data acquired and memorized by the system must, by necessity, fit into some sort of predetermined and confining pattern. However, the data model used must help and support the entry and retrieval of information rather than hinder it. It must accommodate new and unforeseen information structures. In order to create the open architecture databases needed for intelligent farm management/operating functions, system designers must find ways to represent the intricate and unpredictable structure of the information described above. The following is an attempt to define a data model which will allow the construction of such systems.

Proposed data model

By restricting the types of basic information elements used to represent the real world it should be possible to implement a simple, yet flexible structure which minimizes the constraints imposed on the data stream. In the data model, information elements are separated into two basic types: data records and links (see Fig. 2), each of these being in turn subdivided into several types. Data records are used to hold the values and labels which constitute the nodes of the information network. Links are used to connect data records and thus provide the paths and hierarchy in the network. The term "entity" is used below for generic names which may represent concepts, objects, events, qualities or quantities. A data record of the type entity contains a label referring to a specific concept. The term "value" is used to refer to records which hold numeric or character information corresponding to the occurrence of an entity. The use of only two basic types of data records implies a network of links relating generic labels (entities) and data items (values). Such links must be used to (1) specify the relationships existing between entities, entity-to-entity links, (2) establish the correspondence between a data item and its generic entity name,

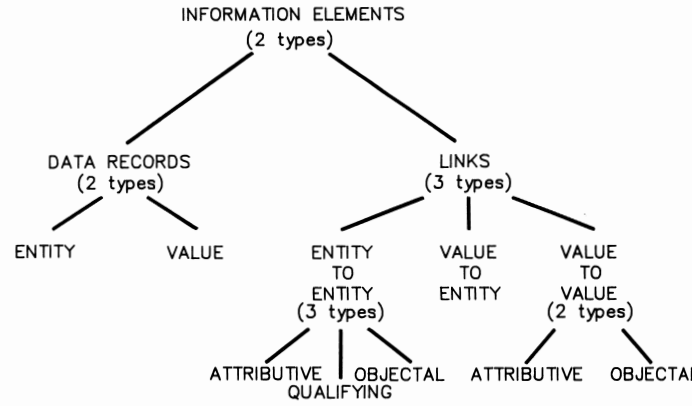

Figure 2. Types of information elements used in the conceptual data model.

value-to-entity links and (3) associate related values, value-to-value links. Thus, information is carried by the type and contents of data records and by the links existing between them. The structure of the data related to a specific operation, transaction or resource can be described in terms of "information templates."

In general, an entity identifies a group or class of things which can be further subdivided into groups identified by a more specific entity name, thus implementing a hierarchy of entities starting from the more general to the more specific. In the data model, entities have three dimensions: attributive, qualifying and objectal. Attributes, qualifiers and objects are themselves entities which in turn can be described by attributes, qualifiers and objects. Dimensions are specified by the existence of links relating entities with other entities. This arrangement is shown in Fig. 3 by the various levels of entities and their associated dimensions.

The qualifying dimension is used to specify the relationship between a higher level entity representing a group or class of things and the subgroups identified by entities at the next lower level. In Fig. 3 the entity E1.1 is "qualified" by the entities E2.4, E2.5 and E2.6. The latter three represent members of the large family of entities identified by E1.1. Similarly, E3.2 is a qualifier of E2.2. The entity-qualifier chain can start at any level in the template and is terminated by a nonqualified entity which, however, can have attributes or objects (e.g., E2.7 is nonqualified but has the attribute E3.5). The shortest possible chain consists of a single entity (i.e., with no qualifiers).

An attribute represents a quality, characteristic or aspect of an entity. One entity can be linked by attributive relationships to several other entities (e.g., in Fig. 3 E1.1 has three attributes: E2.1, E2.2 and E2.3) themselves described by attributes (E3.1, E3.3 and E3.4). The first attribute of an entity usually specifies the data type of the value associated with it. In the data model, an attribute specifying a data type is not itself described by other attributes. An entity transmits to its progeny (qualifiers) all of its attributes. Thus, a particular entity is described by the attributes it inherits from its "ancestor" entities and by the attributes to which it is explicitly linked. A qualifying entity such as E2.4 inherits the attributes E2.1, E2.2 and E2.3 of its parent entity E1.1. Because it is an object and not a qualifier, the entity E2.7 does not inherit from E1.1 but it has attribute E3.5 explicitly linked to it.

An object is that which is affected or acted upon by an entity. E1.1 has two object entities: E2.7 and E2.8. An object is usually described by a data type attribute and can constitute the

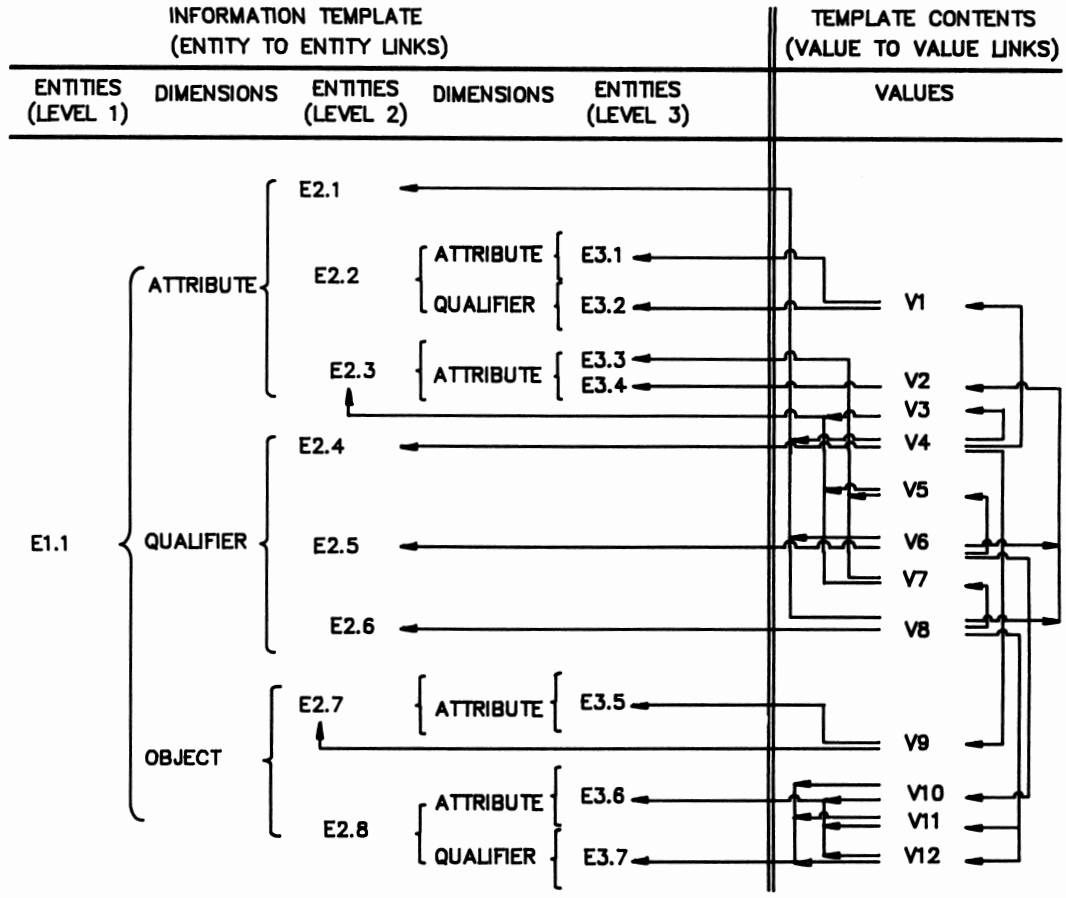

Figure 3. Generic schema with types of data records and links.

head of a chain of qualifiers. Qualifying entities (such as E2.4, E2.5 and E2.6) inherit objects from the ancestor entities in their dynasty as they inherit attributes. This hierarchical arrangement can be implemented across several "generations" of qualified entities where each new generation inherits the attributes and objects of its ancestry.

Entities at the end of a chain of qualification are nonqualified and are represented by occurrences of data records of the value type. Several records of the value type can be linked to the same nonqualified entity since each data item represents an occurrence of this entity. In Fig. 3, the data item V4 represents an occurrence of entity E2.4. Simultaneously, V4 is linked to the attribute E2.1 and is also an occurrence of that entity. Thus, value-to-entity links are used to connect a data record of the value type (located on the right hand side of Fig. 3) to a nonqualified entity record (located on the left hand side). Any instance of a qualifier is also regarded as an instance of the entities at the higher levels in the qualifying chain (i.e., V4 is an instance of E2.4 and of E1.1). During the creation of a value type record, the type and number of links which are needed to fully describe this record are implied by the "template" (i.e., entities, their associated dimensions and links). Specific links can be used later on to retrieve the occurrences of a particular entity.

The first attribute of an entity is used to specify the type or nature of that entity and consequently data items are generally involved in two value-to-entity relationships; the first a direct occurrence of an entity and the second the occurrence of an attribute entity describing the type of the data item. For example, in Fig. 3 the values V4, V6, V8 which are, respectively, direct occurrences of the nonqualified entities E2.4, E2.5, E2.6 are also linked to the first attribute of these entities (E2.1). Similarly, V1 is linked to the entity E3.2 and to its attribute E3.1. This arrangement allows the definition and use of many data types and facilitates interaction with humans and foreign data structures. A database could make use of data types such as label, ID, quantity, date, duration, amount, number, etc. Thus, it is possible to infer the meaning and nature of a given data item by examining the entity names to which it is linked.

Links between data items, i.e., value-to-value links, are of two types: objectal and attributive (see Fig. 2). Such links echo the existence of entity-to-entity relationships of the same type. Qualifying relationships are not needed between value type records since these are implied by the dynasty structure. In Fig. 3, E1.1 has two object entities: E2.7 and E2.8. Hence, data items representing E1.1 (such as V4, V6 and V8) are linked to one or more data items representing occurrences of E2.7 or E2.8 (right hand set of links). The same is true for attributive relationships i.e., an instance of an entity is linked to the instances of its attributes (e.g., in Fig. 3, V4 is connected to V1 and V3 by links on the right-hand side of the figure). The hierarchy on the left hand side of Fig. 3 provides a template which defines and structures the network of data items on the right-hand side. Generally, a template will use only the entities and dimensions needed to describe the structure and contents of a generic entity. Hence, it represents a very small and limited part of the overall data network. Templates are distributed throughout the network of entity names, each entity name participating in any number of templates. In fact, a template is

merely a means of representing the structure and composition of a specific fragment of the total database.

A sample template that could be used to maintain cropping data is presented in Fig. 4. The objective is to record the field number and yield associated with each crop on a yearly basis. This can be achieved by associating two attributes (DURATION and YIELD) and one object (FIELD) to an entity called CROP. The latter is qualified by the two entities SOYBEAN and CORN which are connected to data items of the type DURATION (first attribute of CROP). Thus, any instance of SOYBEAN (e.g., 01/85-365/85) is linked to the entity DURATION. CROP instances (i.e., instances of any qualifier of CROP namely SOYBEAN or CORN in this case) are connected by an attributive relationship to YIELD instances which are described in turn by the attributes QUANTITY and UNITS. CROP also has an object called FIELD. Hence, CROP instances are linked by objectal relationships to one or more instances of FIELD. The CROP template shown in Fig. 4 does not show all possible or actual relationships between entities in the total database; the dimensions and qualifying chains not relevant to the representation of a CROP entity have been omitted or left out of the diagram. For example, the entity SOYBEAN could participate as an attribute in a template describing seeds and the entity DURATION could be used to specify the type of a data item representing field operations, pest infestations, etc. Also, the CROP template could be augmented or modified through link editing and/or the addition of entities (e.g., qualifiers such as WHEAT, OATS or APPLES; attributes such as PESTS, SEEDING DATE, POPULATION, HEIGHT OF STALKS, etc).

Data model implementation considerations

In general and in a strict sense, entity names do not carry meaning by themselves; they are chosen solely on the basis of their significance in interactions with humans or foreign data structures. For example, when entering cropping data the user would be presented with the existing list of qualifiers from which the appropriate one might be selected or alternatively, into which a new one might be inserted. The user would then be prompted for the duration, yield quantity and units and field identification. The answers or selections would be registered through the creation or selection of entity names, data items and links.

The set of entity names constitutes the vocabulary of the system and could be simply translated or adapted to a new human language without loss of meaning or effect on the data structure. Ultimately, the low-level procedure and algorithms which manipulate entities should be able to operate with any human language and with any network of entities since entity names are merely nodes in a network of hierarchies. This permits the design of general procedures (software) to infer the composition and structure of database transactions.

In a farming environment information can be classified into four general categories, each having a unique function: farm resources, farming operations, farm products and environmental factors. Resources (e.g., equipment, operating inputs) are generally characterized by a number of descriptive attributes and a temporal attribute (life span within the enterprise). Resources constitute the means of production on the enterprise and must be accounted for in budgeting and cost analysis. Operations affect farm resources or products and occur within a spe-

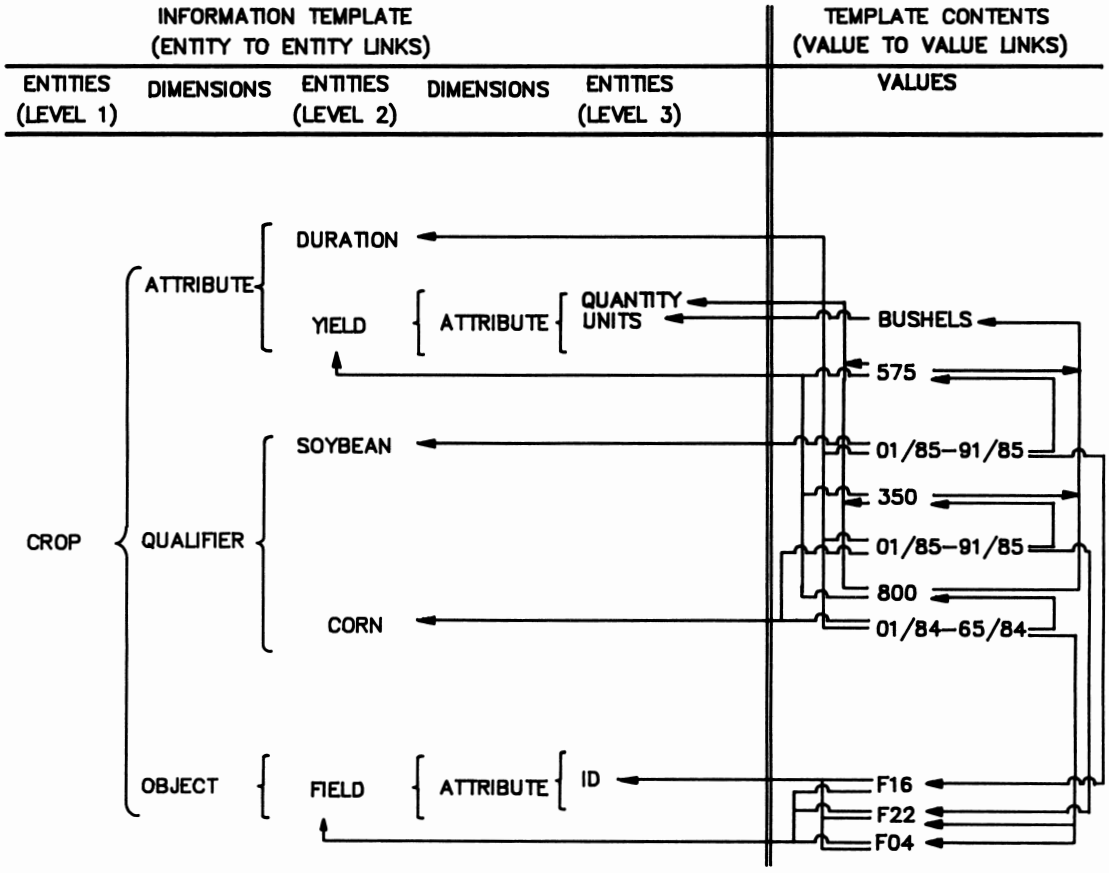

Figure 4. Sample information template with entity names and data items for crop data.

cific time frame (e.g., seeding). They are characterized by the use of one or more resources. Thus, operations are defined by (1) a temporal attribute, (2) physical and virtual attributes (equipment used, cost) and (3) the object of the operation (e.g., a field). Products include all material, food or fiber produced, harvested or transformed on the farm (e.g., silage). Products are accumulated and disposed of within a given time period (temporal attribute) and can be described by a variety of descriptive attributes. Environmental factors (e.g., climate, diseases, pests) are recorded or calculated for a specific time frame and similarly to operations, affect a farm resource or product. Thus, they are also defined by descriptive and temporal attributes and one or more objects. The inclusion of temporal attributes for all four categories of information permits the "memorization" of enterprise history. The existence of an inclusive historic database is prerequisite to the implementation of high-level, intelligent, management/operating functions.

SUMMARY AND CONCLUSIONS

Functional integration of the various management and operating tasks associated with production agriculture is needed to increase the effectiveness and productivity of farms. The advent of sophisticated computer and artificial-intelligence-based tools such as parallel processors, robotics, expert systems, knowledge bases and natural language processors has the potential to greatly help the design and implementation of integrated systems. The emerging information stage farm will require and use considerable amounts of data. Delivering this information to the farmer poses an original challenge to designers because the underlying data model must be very flexible, adaptable and comprehensive. It must convey the structure as well as the nature of the real world being represented. The data model must accommodate all physical and virtual relationships implied by agricultural production and provide the context for data interpretation. A model based on two conceptual record types (entity and value) and several generic relationships or dimensions between records (attributive, qualifying and objectal) was proposed. Information is stored in a hierarchical network of entities where the context of an entity is described by a template. These templates are distributed throughout the entity network and can be used to infer the composition and structure of a database transaction.

REFERENCES

DEERING, M. F. 1985. Architecture for AI. Byte Mag. 10(4): 193–206.

DEVLAEMINCK, R. M. 1985. Computer integrated farming. Proceedings of the Agri-Mation 1 conference. Chicago, Ill. 25–28 Feb. 1985. Am. Soc. Agric. Eng. pp. 356–366.

GAUTHIER, L. and R. KOK. 1985. Design criteria for a field management system. Am. Soc. Agric. Eng. paper no. 85-5523. 12 pp.

GERRISH, J. B. and T. C. SURBROCK. 1983. Mobile robots in agriculture. Robotics and intelligent machines in agriculture. Proceedings of the first international conference on robotics and intelligent machines in agriculture. Tampa, Fla. 2–4 Oct. 1983. Am. Soc. Agric. Eng. pp. 30–41.

HEWITT, CARL 1985. The challenge of open systems. Byte Mag. 10(4): 223–242.

HOLSAPPLE, C. 1980. A primer on data base management systems. Micro Data Base Systems, Inc., Lafayette, Ind. 54 pp.

HOLT, D. A. 1985. Computers in production agriculture. Science 228(4698): 422–427.

JOHNSON, C. E., R. L. SCHAFER, and S. C. YOUNG. 1983. Controlling agricultural machinery intelligently. Robotics and intelligent machines in agriculture. Proceedings of the first international conference on robotics and intelligent machines in agriculture. Tampa, Fla. 2–4 Oct. 1983. Am. Soc. Agric. Eng. pp. 30–41.

KOK, R. and G. DESMARAIS. 1985. An integrated hierachical control system for an intelligent greenhouse. Am. Soc. Agric. Eng. paper no. NAR85-403. 12 pp.

KOK, R. and L. GAUTHIER. 1986. The development of a prototype farm information management system. Comp. Electron. Agric. 1: 125–141.

McKINNION, J. M. and H. E. LEMMON. 1985a. Expert systems for agriculture. Comp. Electron. Agric. 1(1): 31–40.

McKINNION, J. M. and H. E. LEMMON. 1985b. Symbolic computers and AI tools for a cotton expert system. Am. Soc. Agric. Eng. paper no. 85-5520. 15 pp.

MONCASTER, M. E. and G. O. HARRIES. 1983. Automatic control of tractors and field machines. Proceedings of the first international conference on robotics and intelligent machines in agriculture. Tampa, Fla. 2–4 Oct. 1983. Am. Soc. Agric. Eng. pp. 120–127.

PEJSA, J. H. and J. E. ORROCK. 1983. Intelligent robot systems: potential agricultural applications. Robotics and intelligent machines in agriculture. Proceedings of the first international conference on robotics and intelligent machines in agriculture. Tampa, Fla. 2–4 Oct. 1983. Am. Soc. Agric. Eng. pp. 104–111.

RIDDLE, W. E. 1985. Agriculture 2000 — A time of technological change. Proceedings of the Agri-Mation 1 conference. Chicago, Ill. 25–28 Feb. 1985. Am. Soc. Agric. Eng. pp. 21–26.

SCHUELLER, J. K., R. E. YOUNG, and D. A. BENDER. 1985. Potential applications of recent advances in manufacturing technology to crop production agricultural machinery. Proceedings of the Agri-Mation 1 conference. Chicago, Ill. 25–28 Feb. 1985. Am. Soc. Agric. Eng. pp. 286–296.

SMITH, R. D., J. R. BARRETT, and R. M. PEART. 1985. Crop production management with expert systems. Am. Soc. Agric. Eng. paper no. 85-5521. 11 pp.

SONKA, S. T. 1985. Information management in farm production. Comp. Electron. Agric. 1(1): 59–85.