for ISMIS-91

The Roles of Artificial Intelligence in Information Systems

Gio Wiederhold
Stanford University

November 1, 1998

Abstract
By determining classes of information processing tasks which are suitable for artificial intelligence approaches we outline an architectural structure for large systems. In that structure processing modules become specialized. We argue that artificial intelligence programs should remain simple and constrained to a single ontology. Tasks are typed as focusing, pruning, fusion, and reduction. There are also control tasks associated with effective resource management. Their results are then composable by higher level applications, which have to solve problems involving multiple subtasks.

The artificial intelligence modules will need a machine-friendly interface to support those applications. This interface must provide good communication, while encapsulating the artificial intelligence tasks, so that the complexity of the composed system is not much greater than that of the individual subtasks.

The corresponding architecture is a generalization of a server-client model. The partitioning enhances maintainability, but questions of effectiveness and efficiency do arise.

1. Introduction

We define information systems as computer-based systems which can access a variety of computer-stored or generated base data, and select and process that data to provide specific information to aid planners and management in their decision-making. While the source data are often voluminous, a modest amount of alternative choices should be presented, in a clear and concise form. Presentation to the end-user of alternatives that are not reasonable, are subsumed by other alternatives, or are largely similar only create overload, forcing the decision-maker to delegate analysis to an assistant.

In the computing tasks needed to support the higher levels of such processing we will find that artificial intelligence techniques are essential, since the regularity appropriate to algorithmic approaches is typically lacking. However, large databases and single-focus information retrieval can be effectively handled by algorithmic approaches. In information retrieval heuristics appear when precision and completeness of search must be balanced with relevance and avoidance of reader's overload. Complexity abounds when there are multiple information sources. There will be mismatches of data representation, of time and space, of confidence, and of level of abstraction. However, the large volume of data needed for many support tasks means that algorithmic approaches, when appropriate, must be exploited [Wiederhold:91].

The need for a multiple-paradigm approach leads to a partitioned architecture, which is actually well suited to the modern multiprocessor networked environment, as described in the next section. In the approach we advocate we assign tasks which often require artificial intelligence to the intermediate layer. The core of this paper is in Section 3, where we enumerate some of the tasks foreseen here. These mediating tasks are substantial, and require capabilities that are similar in scope to those attained by todays successful free-standing applications of artificial intelligence.

In Section 4 we discuss the effectivenes of the approach. We illustrate our notions with some examples. Any multi-paradigm approach has to deal with the problems of the inefficiency of the interface. We discuss how this problem can be mitigated by careful matching of the partitioning to the proposed architecture.

In the final section we enumerate problems to be addressed, with some hints for their solution.

2. Architectural Issues

Artificial Intelligence techniques deal well with problems of greater complexity than algorithmic approaches, but are in practice limited to relatively small amounts of information. Whereas simulations, databases, and design-systems routinely deal with millions of elements, major artificial intelligence tasks, as XCON are becoming very difficult to manage when the number of rules exceeds, say 10 000. These projects may then look for guidance from algorithmic approaches [Barker:89], or develop a partitioning, as *micro-theories*[Lenat:90], to make maintenance more feasible.

This difficulty faced in scaling artificial intelligence approaches is due to their need for alternative processing paradigms, and not surprising. In algorithmic approached the logic complexity is in the processing algorithms, and this logic is applied to regular structures, as numeric matrices, trees and other structures for hierarchically decomposed data, time-series with fixed steps, and database tables. For the artificial intelligence tasks, the complexity is represented within the structures which represent the information: rules can have arbitrary linkages; frames have variable numbers of slots; and instances can be members of multiple classes [Cohen:82].

We will hence assume a *horizontal* partitioning, where complex processing is allocated to a mediating layer. The mediators will draw upon resources from a voluminous base layer. Applications will have to invoke multiple mediators, as argued below. The overall architectural concept is sketched below [Wiederhold:91]. In this presentation we focus on dealing with the complexity of the the problems we find in the processing of artificial intelligence representations.

---

result $\rightarrow$ decision making

Layer 3 Independent **decision makers** — managed on workstations

- - - - network services to information servers - - - -

Layer 2 Multiple **mediators** — managed by domain specialists

- - - - network services to data servers - - - - -

Layer 1 Multiple **resources** — managed in server systems

input $\leftarrow$ real-world changes and simulations

Figure 1: The three layers of this architecture.

2.1 Interpretation and representation

As stated above, the problem statement in artificial intelligence approaches is mainly represented in their knowledge structures. The logical interpreters of these structures are typically intended to be domain independent, and must deal with the complexity presented by these representations. We note that the issue of representation is in fact so pervasive that *knowledge representation* is often regarded as a separate scientific effort. But representation without a defined interpreter is not wise. For a minimal task common first-order logic may be an adequate interpreter; but a pure implementation will be inadequate to deal with structured representations. The use of higher-order concepts, as classes, partition knowledge into groupings so that the computational effort is structured and reduced. Although the classification concepts may be describable in terms of the first-order logic primitives, it is not practical to always reduce their interpretation of the representation to that level.

Higher level concepts are hence essential. For execution these concepts require explicit semantic description and direct interpretation. This abstraction principle is common to all computer languages. Algorithmic languages can also be viewed theoretically as abstractions over Turing machine concepts, but in practice always have algebraic or English definitions of their actual semantics. Practical implementations of LISP define the semantics of many available functions explicitly, even though they are always reducible to the simple primitives which made up the foundations of LISP in the early sixties. We are hence kidding ourselves when we attempt to study complex representations independently of interpretation. With representation comes a whole implied ontology, and with the ontologies a variety of paradigms to address information processing tasks.

2.2 Partitioning the mediator layer

The diversity of artificial intelligence ontologies makes a *vertical* partitioning also desirable in that layer. We reject the notion then that a single knowledge-based layer can deal with a wide variety of resources and serve multiple applications [Sacca:86]. The proposed partitioning matches the concept of having specialized experts to assist in complex tasks. The experts will consult and maintain their own knowlegde bases, but must also access a shared base of resource information describing the problem being addressed, and its environment [McIntyre:87].

Once we have such a partitioning we need not be limited to artificial intelligence approaches. Some expert tasks might be best served by linear programming, and others may extract information from bibliographic files.

2.3 Resources

To clarify the tasks that are needed in the mediator layer we will summarize the types of resources which are available. Many of the resources are likely to be massive. They must use representations that are effective for algorithmic approaches. Typical are databases, large bodies of text, and simulations of physical processes using spatial or four-dimensional (space and time) representations. These types of base data do not require initially very complex processing, so that algorithmic processes will be effective. However, the results of this basic algorithmic processing are rarely directly suitable at the level that we envisage.

We do see a need for the artificial intelligence processes to acquire from them modest amounts of selected, focused, but possibly complex results. These results must be easy to integrate into the representations used in intelligent processing. We see obvious matches here:

- Object representation ≡ frames
- Aggregation results ≡ rules
- Simulation results ≡ event lists

The interface issues here are critical. But, again, not only the representation must be considered. The resources are active engines, invoked by modules in the mediator layer.

Active interface management encompasses selection of input parameters, control of execution and distribution, as well as the further processing of results.

Interface issues for databases, simulations, and the like are also being addressed by researchers concerned with direct user interfaces. Although our mediating modules do not need traditional *user-friendly* interfaces, there is sharable technology, due to the preponderance of the client-server models in modern architectures. In that systems interface it is important that the developing standards be *machine-friendly*, since they will rely on client-based interpretation. Unfortunately that architectural insight comes slowly, and corresponding standards lag even more. The SQL standard for databases, for instance, is a compromise of a not-so-*user-friendly* language, with an awkward embedding convention into programs.

2.4 Resource modules

We expand now on the types of resources which the mediating routines can exploit. The use of these resources requires a level of understanding in terms of parameter setting and execution control which poses a challenge to artificial intelligence techniques. We will cite some of these as we survey those resources.

2.4.1 Databases

A simple resource is a database. It provides fairly rapid selection of factual information, especially when the volume of data or considerations of shared access demand the use of secondary storage. The database paradigm is to return closures, that is all valid results. Further computation is needed to reduce valid results to relevant ones.

It is desirable to reduce as much as possible the valid output so that the subsequent computations will be small, and also so that bandwidth requirements between database servers and analysis machines are minimized. For instance, practical database systems provide facilities to aggregate results by group classifications. A value as the `average_delay` at a port may be all that is needed in a transportation analysis, rather than the detailed history. But the computational cost of aggregation over large data volumes is high.

Further effectiveness can be introduced by intelligent control. For instance, for long histories the average value may also be estimated effectively from a sample of the database [Ozsoyoglu:91], [Roy:91], [Lipton:90]. More difficult to estimate are maximum or minimum values [Rowe:83], but the important concept of variance of the result values is estimatable from a sample. Now the bandwidth requirements for remote access are also minimized. Some intelligence is needed to phrase queries that give sufficiently accurate results at low cost.

Much information for effective minimization resides in the databases. An important optimization supported by databases processing is in scheduling join-operations among multiple relations [Ullman:88]. For a list of `aircraft` and `pilots` only those combinations of qualified `pilots` and pilotable `planes` will be considered. Further constraints as `range` of aircraft can be integrally imposed. The declarative nature permits the database to optimize access schedules. It will be hard to balance the use of external intelligence and internal optimization without good insights into database processing [Smith:85].

Optimal transfer of database results into an object representation for expert systems has only recently been considered [Lee:90]. The cost of the impedance mismatch is reduced by letting the client-server interface also be the representation interface [Shoshani:85].

2.4.2 Simulations

The results of simulations provide another resource for artificial intelligence processes. If the applications involve future planning, then data about, say, weather forecasts may be critical. The simulations execute continuously, refining short-range predicting with recent observations, and pushing the bounds of long-range predictions as the time progresses.

The trade-off of precision and depth of projection is not easy, and requires intelligent processing. For instance, stability for long-term prediction tends to be reduced when many parameters are used, but for short-term predictions more parameters will increase the precision.

2.4.3 High Performance Computing

Whereas traditional simulations prepare and store information for later use, there are also instances where a particular situation has to be evaluated. In those cases the computational demands to evaluate the case may be immense. The use of high performance computing engines is warranted when the cost of transmitting input specifications, starting and executing a computation, and transmitting the result is less than local computation.

An intermediate agent may make that assessment. In a resource-rich environment both local and remote computations may be initiated, and the alternative abandoned when measures of computational progress indicate that one approach is better. This model easily extends to use of multiple asynchronous processors in parallel. A concern will again be to minimize bandwidth and demands on the controlling mediator.

2.4.4 Reference systems

Databases, simulations, and the like, rarely try to justify their decisions. Bibliographic-style references remain an important part of back-up for decision making. However these systems are well-developed on support software that is typically wholly independent of DBMS or AI technology. As indicated in the introduction, bibliographic systems are served by their own heuristics. However, for many decision-making tasks it is useful to select relevant backup references.

We expect that many of these references will not only be available as citations, but that abstract and full-text will be available [McCune:85]. Now hypertext paradigms for establishing conceptual linkages become effective, but they might be driven by an intelligent mediator, rather than by the end-user, who is likely to ignore even useful justifications if the browsing effort is made tedious by lack of linkages to the problem analysis at hand.

2.5 A hardware mapping

With the functional divisions we have established, a corresponding hardware partitioning makes sense, as was implied in Figure 1. When using foreign processes, it becomes natural to also use remote processors for substantial tasks. The reduction of cost for powerful hardware makes the use of specialized processors more attractive. This specialization may merely be in terms of software, specialized knowledge, and data resources. Reduction of task-switching in processors, longer real-memory residence, full exploitation of caching makes software specialization attractive, even while the hardware platforms are interchangeable. The benefits are gained through improved response times within the individual modules, and parallel operation overall.

Since, when applying artificial intelligence, we are looking for minimal data quantities, with relatively high information content per representation unit to be transferred at the module interface, we believe that the same interface will be good for the network communication boundaries. That assumption is evident in Figure 1 as well.

3. Tasks

Since we are getting access to so much more data, from a variety of sources, arriving at ever higher rates, automated intermediate processing will be essential. The processing tasks needed within the mediators include: selection, focusing, pruning, fusion, reduction, abstraction, and generalization.

The tasks required from the mediating routines must create information for decision making or actual action-taking. This requires a further reduction of choices, either by further abstraction, ranking, or mutual selection of alternatives. Often multiple resources will be used. Investment choices for instance involve assessments of cost and profits, the competitive situation, and alternative investments. Logistics decisions involve availability of land, air, and sea transportation, readiness of the goods to be shipped, availability of storage at the destination, and prediction of weather conditions. All of these are likely to be on distinct resources, as outlined above.

3.1 Focusing

A primary task for a mediator is to transform the information obtained from sources into a more focused representation. That involves first of all reorganizing the data representation. Databases, for instance, because of their goal of universality, produce large tables with no indication of what entries are important and what is not. A typical object-focused representation is a tree structure, where the objects of concern form the root.

Alternative or complementary representations are ranked list, which order the results by some, possibly multi-variable, measure of relevance.

In decision-making applications, the best ranking considers the benefits and the costs. Risk may also be a concern. If solutions are subsumed on all criteria, they should be deleted. If solutions have similar balance, then they should be presented as group, and then entries within a group ranked as secondary choices.

The primary choices - good but different in balance, are to become the initial focus. Once a balance has been selected, then other members of the group can pop up for further refinement.

3.2 Pruning

Implied above was that the decision would be made over the closure of possible alternatives. But in many cases generating closure fast is not computationally feasable. We do not want all routes to fly from San Francisco to Knoxville. There are many valid choices, but an infinity of invalid ones. Stopping in Singapore may be fun, but is not valid.

To manage pruning requires intelligence in dealing with the resources. A simple heuristic is that any trip with a leg longer or more costly than the best trip now in the result set should be avoided.

An initial assessment may consider hubs: Denver, Chicago, Salt Lake City, Atlanta, Dallas, as well as Dulles. Limiting search to a depth three makes also sense here, but the depth will vary based on itinerary. Erik Sandewall might not be able to attend ISMIS with that limit.

In general breadth-first search is preferable over depth-first when pruning is an issue. But breadth-first is a poor strategy when pruning rules are not effective and the solution set is large.

3.3 Fusion

New discoveries are made and insights are gained when information from distinct sources are combined, or even when the combination is attempted and fails. Historical information from references can be combined with current detail from databases, and used to compare with simulation results. Further simulations can project into the future.

The fusion of results from such widely differing sources is complex. To access historical references text has to be processed. Databases contain a mix of textual and numeric data, and simulations procude numeric results.

3.3.1 Mismatch resolution

Data obtained from remote and autonomous sources will often not match in terms of naming, scope, granularity of abstractions, temporal bases, and domain definitions. The differences shown in the examples must be resolved before automatic processing can join these values [deMichiel:89].

When data match semantically, but not precisely, as for instance town-names versus zipcodes, uncertainty in the match is generated. Ignoring the result when the match is uncertain can involve great risk — there *may* be dangerous material in this town, and people have to be notified.

Processing with mismatched data also creates null values. For items where only a partial match exists, as exemplified by outerjoin results in relation models, null values are created. Mediating modules will, in general, need to deal intelligently with missing data, as well as with partial values and sets of alternative values. [Winslett:87]

3.4 Reduction

Aggregation of detail into higher level classes is a primary means of reduction. When classes are well defined, the computation may be delegated to one of the resource modules. But classifications for a given objective may be vague, and in fact deduced by inspection of data and their correlations. Classification is already a traditional task for AI programs [Cohen:82]. Linking these programs with existing resources, and making them responsive to a variety of applications can provide major benefits.

3.4.1 Temporal reduction

A frequent task will be to reduce voluminous time series data to a simple predictive value at some point in time, or perhaps instantiate a function to permit assessments at multiple points in the future.

The mediator task will have to deal with cyclic effects, perhaps due to seasonal patterns, long term effects, and correlations with concurrent events.

3.4.2 Exception seeking

Another variety of reduction of information volume is in exception reporting. An example is a search for places where money appears to be laundered. The banking regulations now require the reporting of large cash transactions to the treasury, but further assumptions need to be made to identify those transactions that need further investigation. When a hypothesis is known the controlling system can select and analyze source data to prove or disprove the hypothesis.

A generalization of exception seeking is automated knowledge acquisition, or as refered to by some database folk, *data dredging* or *data mining*. This activity is also a component of future information systems. It does depend critically on having deep databases: i.e., databases with enough attributes so that unexpected hypotheses are possible, and enough instances to validate findings even in the presence of noise and many covariates [Blum:82].

3.4.3 Missing Data

Missing data, whether created by computation or in the source information, is a major problem in simple tabular processing. Many statistical systems deal poorly with it. However, if values are intelligently aggregated, then at the higher conceptual levels missing data will be reduced, although uncertainly will remain [DeZegher:88].

3.5 Examples

We do not now have a single system which contains a general composition of these tasks. We can however cite a large number of examples where intermediate and intelligent transformations of source data take place.

1 Database view definitions and object templates [Chamberlin:75] [Lai:88]

2 Multi-database schemas to integrate data [Smith:81] [Dayal:83]

3 Methods to achieve closure [Beeri:87] [Ramakrishnan:89]

4 Library catalogs and indexing aids [Humphrey:87]

5 Thesaurus structures for access to bibliographic systems [McCarthy:88]

6 Materialized views and view indexes [Hanson:87] [Roussopoulos:86]

7 Rule bases for semantic query optimization [King:84] [Chakravarthy:85]

8 Object-generators [Wiederhold:86] [Barsalou:88]

9 Knowledge-bases to access multiple databases [Sacca:86]

10 Computations on derived data [Hammer:78] [Adiba:81]

11 Graphic tools to help in hypothesis generation [deZegher:88] [Downs:86]

12 Statistical summarizations [Ozsoyoglu:84] [Olken:86]

13 Interpolation functions for interpreting incomplete observed data [Callahan:81]

14 Programs that fuse data arriving from sources that differ in terms encoding and accuracy [Litwin:86] [deMichiel:89]

15 Programs that relate information, say, factory production to budgets, compensating for differences in reporting periods [Chiang:82]

16 The FAST component ordering system [Cohen:89]

17 Modules to provide privacy protection for sensitive data [Cohen:88]

This list is based on direct observation. Most readers will be able to augment the list from their experience.

4. Effectiveness

The complexity of the resources and the variety of applications may make the mediation concept unattainable. We can in fact point to research where the complexity caused a drastic scaling-down of applications.

Furthermore, to be effective, the mediators must be useful in many applications, and we are all aware of the high cost of generalization. In this section we present arguments why we believe that the concept of partitioning is managable, and what the criteria are to keep mediation effective.

4.1 Limiting Mediators

The tasks presented in the previous section, if integrated into a single application, would be complex, deal with multiple computational paradigms, and access a wide variety of base resources. Their complexity would cause problems in maintenance and hence rapid obsolescence. At the same time, the subtasks would not be separatable for reuse, although many provide sharable functions.

For these reasons we do not envision large mediators; but rather a variety of small, maintainable and specialized ones. Each mediator would be characterized by a single ontology: knowledge representation and its corresponding interpreter. A single mediator would only access a few base resources, so that it can be adapted to changes in resources. In operational databases, we observe that schemas tend to change about every two years.

Having a single ontology, and a specific knowledge domain permits maintenance by a single individual or focused group. No costly committee work is needed to decide on compromises.

4.2 Composing Mediators

Since mediators are specialized, applications will need to compose results from multiple mediators. Since only results are to be composed, the application does not now have a complexity that is based on the union of all the mediators it invokes. For instance, the mediator that uses temporal reasoning will produce results valid for the instance of time specified. The application which uses these results does not have to understand temporal reasoning.

Similarly, many computations of alternatives require pruning. In practice the closure of all choices on how to perform a task, say to transport some good from A to B is immense. But it becomes composable once a basic means of primary transportation is selected, based on approximate information. If air transport is the choice, then we have the tasks of getting to A's airport, getting from there to B's airport, and then getting to B proper. These tasks use distinct information, and can execute relatively independently. However, their dependencies are crucial.

If the time-of-arrival at B is the constraint, then we must perform a goal-driven search. Conditions at B determine the desired aircraft arrival time at B's airport, and so on. Other constraints, leading to alternate computations, might be to ship the goods as soon as they are ready or to select the cheapest flight. The same mediating modules will be invoked, but in different sequences, if optimal computation is desirable. When computational requirements are heavy, some modules may be invoked in parallel.

To perform optimization of module use, estimates of expected performance and result cardinalities are needed. If autonomy of modules is to be preserved, then these estimates must be produced in a canonical form by the modules, obviating the need for the optimizers to model the modules. Such a proposal appears in [WWC:90].

4.3 Sharing of mediator modules

Diverse mediator modules will use these functions in varying extents to provide the support for user applications at the decision making layer above.

The mediator modules will be most effective if they can serve a variety of applications [HayesRoth:84]. The applications will compose their tasks as much as possible by acquiring information from the set of available mediators. Unavailable information may motivate the creation of new mediators.

Sharing reinforces the need for two types of partitioning: one, into horizontal layers for end-users, mediators, and databases, respectively, and two, vertically into multiple user applications, each using various configurations of mediators. A mediator, in turn, will use distinct views over one or several databases. Just as databases are justified by the shared usage they receive, mediators should be sharable. Note that today's expert systems are rarely modular and sharable, so that their development and maintenance cost is harder to amortize. The same is true for the examples listed in Section 3.5.

For instance, the mediation module which can deal with inflation adjustment can be used by many applications. The mediator which understands postal codes and town names can be used by the post office, express delivery services, and corporate mail rooms.

We foresee here an incentive for a variety of specialists to develop powerful, but generally useful mediators, which can be used by multiple customers. Placing one's knowledge into a mediator can be more rapidly effective, and perhaps more rewarding, than writing a book on the topic.

We can now summarize these observations in Figure 2.

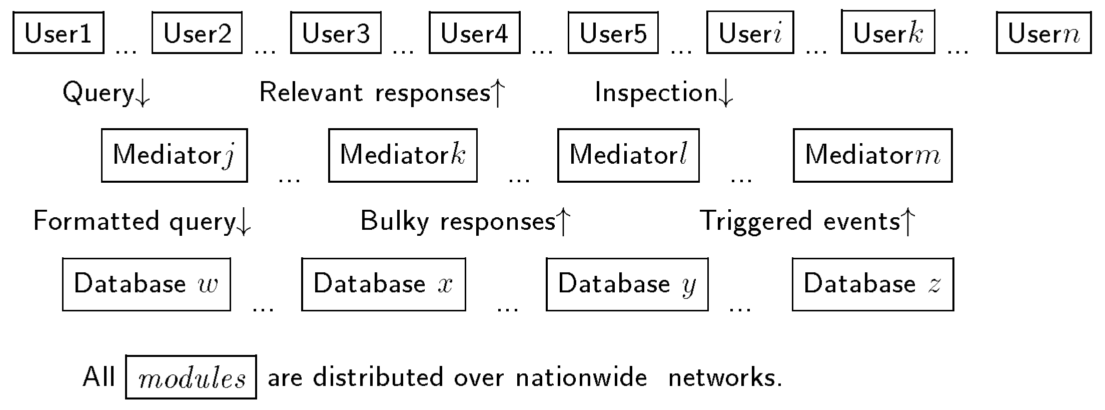
User1 ... User2 ... User3 ... User4 ... User5 ... User$i$ ... User$k$ ... User$n$
Query↓
Relevant responses↑
Inspection↓
Mediator$j$ ... Mediator$k$ ... Mediator$l$ ... Mediator$m$
Formatted query↓
Bulky responses↑
Triggered events↑
Database w ... Database x ... Database y ... Database z
All $modules$ are distributed over nationwide networks.

Figure 2. Interfaces for information flow.

Further layering of mediators is reasonable. The tasks that we presented can be grouped into distinct conceptual levels. Since we do not have a generalized language proposal for the composition of mediators, and too little experience as well, we do not speculate further on this topic. A further generalization, as for instance seen in the ACTORS model [Hewitt:73], seems unwise, because then we have to endow our mediators with motivations to serve others, learn, and survive, as suggested for the BEINGS hypothesized by [Litwin:89]. We have, as of now, no insight into a control structure adequate to simulate such an environment.

5. Conclusion

We envisage a variety of information processing modules residing in nodes along the data highways that advances in communication technology can now provide. A conceptual layering distinguishes nodes by function: decision-making exploration, intelligent information support by mediation of data by knowledge, and base resources.

The mediation function, now seen in a variety of programs, is placed into explicit, sharable mediation modules, or *mediators*. For clarity we place all the mediators into one horizontal layer. These mediators are to be limited in scope and size to enable maintenance by an expert as well as inspection and selection by the end-user. Mediators are associated with the domain expert, but may be replicated and shipped to other network nodes to increase their effectiveness. Specialization increases the power and maintainability of the mediators and provides choices for the users.

In addition to performing information-processing tasks, as outlined in Section 3, we see that management of the operation of the resources, in support of the tasks, is equally important, as shown in Section 2. These latter functions can range from traditional semantic query optimization [King:84] to making tradeoffs in estimation, relevance, and precision.

Applications obtain information by dealing with abstractions supported by the mediators, and not by accessing base resources data directly. A language will be needed to provide flexibility in the interaction between the end-users' workstation and the mediators. We discuss the partitioning of artificial intelligence paradigms into pragmatics (at the user-workstation layer) and the formal infrastructure (in the mediation layer) further in [Wiederhold:91]. The functions in the mediation layer itself may be structured. Perhaps the controlling functions are performed by a lower layer of mediators than the primary tasks. Multiple layers will demand more capabilities from the module interfaces and the languages used to drive them. It will be difficult to define the needed language concepts completely until more experience is gathered in the operation of such systems [WW*:89].

References

[Adiba:81] Michel E. Adiba: "Derived Relations: A Unified Mechanism for Views, Snapshots and Distributed Data"; *VLDB* 7, Zaniolo and Delobel(eds), Sep.1981, pp.293–305.

[Barker:89] V.E. Barker and D.E. OConnor: "Expert Systems for Configuration at Digital: XCON and Beyond"; CACM, Vol.32 No.3, Mar.1989, pp.298–318.

[Barsalou:88] Thierry Barsalou: "An Object-based Architecture for Biomedical Expert Database Systems"; *SCAMC* 12, IEEE CS Press, Washington DC, November 1988.

[Beeri:87] C. Beeri and R. Ramakishnan: "On the Power of Magic"; *ACM-PODS*, San Diego, Mar.1987.

[Blum:82] Robert L. Blum: *Discovery and Representation of Causal Relationships from a Large Time-Oriented Clinical Database: The RX Project*; Springer Verlag, Lecture Notes in Medical Informatics, no.19, 1982.

[Callahan:81] M.V. Callahan and P.F. Rusch: "Online implementation of the CA SEARCH file and the CAS Registry Nomenclature File"; *Online Rev.*, Vol.5 No.5, Oct.1981, pp.377-393.

[Chamberlin:75] D.D. Chamberlin, J.N. Gray, and I.L. Traiger: "Views, Authorization, and Locking in a Relational Data Base System"; *Proc.1975 NCC*, AFIPS Vol.44, AFIPS Press, pp.425–430.

[Chakravarthy:85] U.S. Chakravarthy, D. Fishmann and J. Minker; "Semantic Query Optimization in Expert Systems and Database Systems"; *Expert Databases*, Kerschberg(ed), Benjamin Cummins, 1985.

[Chiang:82] T.C. Chiang and G.R. Rose: "Design and Implementation of a Production Database Management System (DBM-2)"; *Bell System Technical Journal*, Vol.61 No.9, Nov.1982, pp.2511–2528.

[Cohen:82] P.R. Cohen and E. Feigenbaum (eds.): *The Handbook of Artificial Intelligence*; Morgan Kaufman, 1982.

[Cohen:88] H. Cohen and S. Layne (editors) *Future Data Management and Access, Workshop to Develop Recommendations for the National Scientific Effort on AIDS Modeling and Epidemiology*; sponsored by the White House Domestic Policy Council, 1988.

[Cohen:89] Danny Cohen: "Computerized Commerce"; *Information Processing* 89, Ritter (ed), IFIP North-Holland 1989, pp.1095–1100.

[Dayal:83] U. Dayal and H.Y. Hwang: "View Definition and Generalization for Database Integration in Multibase: A System for Heterogeneous Databases"; *IEEE Trans. Software Eng.*, Vol.SE-10 No.6, Nov.1983, pp.628–645.

[DeMichiel:89] Linda DeMichiel: "Performing Operations over Mismatched Domains"; *IEEE Transactions on Knowledge and Data Engineering*, Vol.1 No.4, Dec. 1989.

[DeZegher:88] I. DeZegher-Geets., A.G. Freeman, M.G. Walker, R.L. Blum and G. Wiederhold: "Summarization and Display of On-line Medical Records"; *M.D. Computing*, Vol.5 no.3, March 1988, pp.38-46.

[Downs:86] S.M. Downs, M.G. Walker, and R.L. Blum: "Automated summarization of on-line medical records"; IFIP *Medinfo'86*, North-Holland 1986, pp.800-804.

[Hammer:78] M. Hammer and D. McLeod: "The Semantic Data Model: A Modelling Machanism for Data Base Applications"; *Proc.. ACM SIGMOD* 78, Lowenthal and Dale(eds), 1978, pp.26–36.

[Hanson:87] Eric Hanson: "A Performance Analysis of View Materialization Strategies"; *Proc. ACM-SIGMOD* 87, May 1987.

[HayesRoth:84] Frederick Hayes-Roth: "The Knowledge-based Expert System, A tutorial"; *IEEE Computer*, Sep.1984, pp.11-28.

[Hewitt:73] C. Hewitt, P. Bishop, and R. Steiger: "A Universal Modular ACTOR Formalism for Artificial Intelligence"; *IJCAI* 3, SRI, Aug.1973, pp.235-245.

[Humphrey:87] S.M. Humphrey, A. Kapoor, D. Mendez, and M. Dorsey: "The Indexing Aid Project: Knowledge-based Indexing of the Medical Literature"; NLM, LHNCBC 87-1, Mar.1987.

[King:84] Jonathan J. King: *Query Optimization by Semantic Reasoning*; Univ.of Michigan Press, 1984.

[Lai:88] K-Y. Lai, T.W. Malone, and K-C. Yu: "Object Lens: A Spreadsheet for Cooperative Work"; *ACM Ttans. on Office Inf. Systems*, Vol.6 No.4, Oct.1988, pp.332–353.

[Lee:90] Byung Suk Lee: "Efficiency in Instantiating Objects from Relational Databases Through Views", Report STAN-CS-90-1346, Department of Computer Science, Stanford University, 1990.

[Lenat:90] Douglas Lenat and R.V. Guha: "Building Large Knowledge-Based Systems"; Addison-Wesley, 1990.

[Lipton:90] R.Lipton and J.Naughton: "Query Size Estimation by Adaptive Sampling"; *Proc. ACM-PODS 90*, Nashville, Apr.1990.

[Litwin:86] W. Litwin and A. Abdellatif: "Multidatabase Interoperability"; *IEEE Computer*, Vol.19 No.12, Dec.1986, pp.10–18.

[Litwin:89] W. Litwin and N. Roussopolous: "A Model for Computer Life"; University of Maryland, Institute for Advanced Computer Studies, UMIACS-TR-89-76, July 1989.

[McCarthy:88] John L. McCarthy: "Knowledge Engineering or Engineering Information: Do We Need New Tools?"; *IEEE Data Engineering Conference* 4, Feb.1988, Los Angeles.

[McCune:85] B.P. McCune, R.M. Tong, J.S. Dean, and D.G. Shapiro: "RUBRIC: A System for Rule-based Information Retrieval"; *IEEE Trans. Software Eng.*, Vol.SE-11 no,9, Sep.1985, pp.939-945.

[McIntyre:87] S.C. McIntyre and L.F. Higgins: "Knowledge Base Partitioning For Local Expertise: Experience In A Knowledge Based Marketing DSS "; *Hawaii Conf. on Inf, Systems* 20, Feb.1987.

[Olken:86] F. Olken and D. Rotem: "Simple Random Sampling from Relational Databases"; *VLDB* 12, Kyoto, Aug.1986

[Ozsoyoglu:84] Z.M. Ozsoyoglu and G. Ozsoyoglu: "Summary-Table-By-Example: A Database Query Language for Manipulating Summary Data"; *IEEE Data Engineering Conf.* 1, Los Angeles, Apr.1984.

[Ozsoyoglu:91] G. Ozsoyoglu, K. Du, A. Tjahjana, W-C. Hou, and D.Y. Rowland: "On Estimating COUNT, SUM, and AVERAGE Relational Algebra Queries"; Proc. DEXA'91, Berlin FRG, Springer Verlag, 1991.

[Ramakrishnan:89] Raghu Ramakrishnan; "Conlog: Logic + Control"; Un. Wisconsin-Madison, CSD, 1989.

[Roussopoulos:86] N. Roussopoulos and H. Kang: "Principles and Techniques in the Design of ADMS"; *IEEE Computer*, Vol.19 No.12, Dec.1986 pp.19–25.

[Rowe:83] Neil Rowe: "An Expert System for Statistical Estimates on Databases"; *Proc. AAAI*, Stanford, Mar.1983.

[Roy:91] Shaibal Roy: "Semantic Complexity of Classes of Relational Queries"; *Proc. ACM-PODS 91*, Denver CO, May.1991.

[Sacca:86] D. Saccà, D. Vermeir, A. d'Atri, A. Liso, S.G. Pedersen, J.J. Snijders, and N. Spyratos: "Description of the Overall Architecture of the KIWI System"; *ESPRIT'85*, EEC, Elseviers, 1986, pp. 685–700.

[Shoshani:85] A. Shoshani and H.K.T. Wong: "Statistical and Scientific Database Issues"; *IEEE Trans.Software Eng.*, Vol.SE-11 No.10, Oct.1985, pp.1040–1047.

[Smith:81] J.M. Smith et al: "MULTIBASE — Integrating Heterogeneous Distributed Database Systems"; *Proc.NCC*, AFIPS Vol.50, Mar.1981, pp.487–499.

[Smith:85] D.E. Smith and M.R. Genesereth: "Ordering Conjunctive Queries"; *Artificial Intelligence*, Vol.26, 1985, pp.171–215.

[Ullman:88] Jeffrey D. Ullman: *Principles of Database and Knowledge-based Systems*; Computer Science Press, 1988.

[Wiederhold:86] Gio Wiederhold: "Views, Objects, and Databases"; *IEEE Computer*, Vol.19 No.12, December 1986, Pages 37-44.

[Wiederhold:90] G. Wiederhold, P. Rathmann, T. Barsalou, B-S. Lee, and D. Quass: "Partitioning and Combining Knowledge"; *Information Systems*, Vol.15 no.1, 1990, pp.61-72.

[Wiederhold:91] Gio Wiederhold: "Mediators in the Architecture of Future Information Systems"; to appear in *IEEE Computer*, 1991.

[WW*:89] G.CM Wiederhold, M.G. Walker, W. Hasan, S. Chaudhuri, A. Swami, S.K. Cha, X-L. Qian, M. Winslett, L. DeMichiel, and P.K. Rathmann: "KSYS: An Architecture for Integrating Databases and Knowledge Bases"; in Amar Gupta (ed.) *Heterogenous Integrated Information Systems*, IEEE Press, 1989.

[WWC:90] G. Wiederhold., P. Wegner, and S. Ceri: "Towards Megaprogramming"; Stanford Un. report STAN-CS-90-1341 and Brown Un. report 90-20, October 1990.

[Winslett:87] Marianne Winslett: "Updating Databases with Incomplete Information", Report No. STAN-CS-87-1143, Department of Computer Science, Stanford University, 1987.