Research report 92

AN OVERVIEW OF CASSANDRA-II

Iain D Craig

(RR92)

Abstract

In this short paper, a brief non-technical overview of the CASSANDRA-II system is presented. The system performs knowledge-based monitoring of aircraft movements in controlled airspace, and reports violations of air traffic control rules to the operator. The system is based on the blackboard model of problem solving, but differs in the relative autonomy of its abstraction levels. Abstraction levels, in CASSANDRA-II, are equipped with control components which are responsible for scheduling all decisions based on the level contents. The system is designed for continuous real-time operation: in this context, it is important that computer memory does not become full and halt the execution of the system. CASSANDRA-II contains special-purpose, knowledge-based mechanisms for the removal of out-dated blackboard information.

Department of Computer Science
University of Warwick
Coventry
CV4 7AL, UK

Jan 1987

AN OVERVIEW OF CASSANDRA-II

Iain D. Craig

Department of Computer Science

University of Warwick

Coventry CV4 7AL, UK

ABSTRACT

In this short paper, a brief non-technical overview of the CASSANDRA-II system is presented. The system performs knowledge-based monitoring of aircraft movements in controlled airspace, and reports violations of air traffic control rules to the operator. The system is based on the blackboard model of problem solving, but differs in the relative autonomy of its abstraction levels. Abstraction levels, in CASSANDRA-II, are equipped with control components which are responsible for scheduling all decisions based on the level contents. The system is designed for continuous real-time operation: in this context, it is important that computer memory does not become full and halt the execution of the system. CASSANDRA-II contains special-purpose, knowledge-based mechanisms for the removal of out-dated blackboard information.

January 21, 1987

AN OVERVIEW OF CASSANDRA-II

Iain D. Craig

Department of Computer Science

University of Warwick

Coventry CV4 7AL, UK

1. INTRODUCTION

In this short paper, a brief non-technical overview of CASSANDRA-II is presented. CASSANDRA-II is a system based on the blackboard model of problem solving[2,3,4,5,6,7] . CASSANDRA-II differs from many other blackboard systems in its distribution of levels and knowledge sources. Each level in CASSANDRA-II represents a completely autonomous object which contains a collection of entries, a set of knowledge sources (KSs) and a local scheduler. The local scheduler is charged with the selection of tasks to perform in order to solve the current problem. Levels communicate by sending messages between each other and to the global scheduler (which is responsible for controlling the entire system).

CASSANDRA-II was developed as an intelligent airspace monitor for civilian air traffic control. Its task is to warn the user of any events in the monitored airspace which violate the statutory requirements for aircraft separation.

The paper is organised as follows. In the next section, the domain is briefly presented. In section three, the system is compared to the traditional blackboard systems. In the following sections, CASSANDRA-II is described in moderate detail. The final section is by way of a conclusion.

This work was done while the author was with the Department of Computing, University of Lancaster, Lancaster, UK.

A more detailed description of CASSANDRA-II can be found elsewhere[8].

2. TASK DOMAIN

CASSANDRA-II's task is to monitor the movements of aircraft in a form of controlled airspace called an airway and report possible conflicts between aircraft. This section outlines the domain.

2.1. Separation

All of the Knowledge Sources in the system are concerned with *separation rules.* Two aircraft in the air are separated in the horizontal both laterally and longitudinally, and in the vertical. If the distance between aircraft is zero, then the aircraft have collided. The purpose of separation rules is to provide minimum distances between aircraft: these minima are designed so that aircraft do not collide or interfere with each other. Interference can be caused by the passage of one aircraft through the wake of another, for example, or can be caused by the need for an immediate course change to avoid another aircraft. The imposition of separation minima increases the time taken for an aircraft to move from its current position to the position at which conflicts of these kinds occur.

3. CASSANDRA-II AS A BLACKBOARD SYSTEM

CASSANDRA-II is based on the blackboard architecture, although it differs from many other implementations in significant respects. The differences were introduced in order to increase the runtime performance of the system and to simplify the garbage collection and control problems.

4. THE STRUCTURE OF CASSANDRA-II

CASSANDRA-II is based upon a multi-panel variant of the standard blackboard architecture. The nearest equivalent in the literature is the Hayes-Roth's OPM planning system[4]. It contains panels for

- input of data from a simulation of the airspace under consideration;
- vertical separation checks;
- longitudinal separation checks;

- lateral separation checks;
- cruising level allocation checks.

The panels are divided into levels according to the function of the panel. The input panel has only one level: this is because the tasks performed in the panel are essentially quite uniform. All the other panels have an upper and lower level: in most cases (the cruise level allocation panel is the exception), the upper panel removes all references to aircraft which cannot possibly participate in a violation of the separation rules.

Figure 1 shows the global structure of CASSANDRA-II. Note that each panel has only one input (from the Input Panel) and one output (to the user).

Each level in CASSANDRA-II is considered to be a separate entity. Each level operates separately from all the others in the system and has its own Knowledge Sources and control regime. This design has the advantage of permitting local decisions to be made without any reference to a global plan. It also entails that the levels in the system can operate independently of each other. Levels communicate by placing entries on other levels as is the norm. In the normal case, only the input panel can send messages to other panels: the remaining panels send messages only to a special process which is concerned with reporting rule violations to the user. Control aspects are described in the next section.

The input panel in CASSANDRA-II is concerned with receiving information from the airspace (either real or a projection into the future -- currently this information comes from a simulator). The input panel first schedules a KS which removes all references to aircraft which are outside the airspace being monitored. Then a KS is activated to send entries to the cruise level allocation panel. All the panels in the system apart from the cruise level check panel need to work on pairs of aircraft references, so the input panel forms all possible pairs of aircraft and places them in a list in a blackboard entry. This list is then ordered by vertical and horizontal distance -- two ordered lists are created. The list ordered by vertical separation is sent to the vertical separation checking panel. The list ordered by horizontal distance is sent both to the longitudinal and lateral separation checking panels.

Upon receipt of a new list the panels prune away all those agenda entries which cannot possibly represent violations of the separation rules. In the case of vertical separation, this entails removing all entries which are separated by 5000 ft. or more (the minimum vertical separation for supersonic aircraft above FL450 is 4000ft.). For lateral separation, all aircraft which are more than 100 nm (nautical miles) from a beacon are removed (it is stipulated in MATS[9] that the particular rules in CASSANDRA-II only apply to (pairs of) aircraft which are within 100 nm of a common VOR beacon). For the longitudinal separation panel, all aircraft which are separated by more than 50 nm are removed from the list (this is because 50 nm is greater than the greatest minimum separation specified by the rules). The input pruning task is performed by a KS which is scheduled whenever a new entry is placed on the upper level of each panel.

In the case of cruise level checks, all the aircraft have to be checked. This entails the application of two KSs -- one to check that each aircraft is on the correct level for its heading, the other to check that no aircraft has been allocated FL245 as a cruising level (this is the boundary between upper and lower air-space and may *never* be allocated as a cruising level).

Once the pruning operations have been performed (where necessary), a KS is scheduled to take each pair of aircraft in the input structure and create an entry on the lower level for the individual checks to be made. For cruise level checks, the upper level contains one KS which places the items on the lower level when a new entry is placed on the upper level of the panel.

Once lower level entries have been created and placed on the blackboard, the control mechanism associated with the level is used to determine which checks are to be scheduled and when. The entries on the lower level are subjected to a matching process in which all the KSs on the level are matched information in the agenda associated with the panel. The matcher contains a check that the entry currently being matched has not been matched by the same KS as the one currently trying to match it: this is to prevent loops to occur, and the consequent waste of processor time. The basic matching rule is that each entry can match a particular KS only once -- thereafter the entry is not available to that KS for matching. When a KS has fired, the results of the action part are passed to the output module (in the case of checking KSs on the lower levels of the checking panels) or to the upper level of the appropriate panel in the case of input panel KSs.

5. CONTROL IN CASSANDRA-II

CASSANDRA-II provides an intelligent and distributed control mechanism. The control mechanism is divided into two main components: a hard-wired scheduler and an agenda-based scheduler. There are two control problems to be solved: a local problem and a global one. This section begins with a description of the local problem and its solution in CASSANDRA-II, then the global problems are addressed.

A control mechanism is associated with each level in the system. For each level in the system, there is a completely local scheduler which forms part of the level's level manager.

The scheduling problem is complicated by the presence of tasks from both real and projected airspace. The selection of tasks must reflect the needs and capacilties of the system at any one time and also reflect the urgency of dealing with tasks related to real airspace and the increasing priority of tasks concerned with a projection of airspace into the future. Projection tasks become increasingly important as the time by which the projection is needed becomes closer. Despite this requirement, the schedulers in the current version of the system always select real airspace tasks: this is because it is always possible to re-request a projection, whereas it is not possible to request that real airspace stop while projections are being performed.

With each level, there is associated a scheduler and a control database. These items occupy slots on the level data structure (the level manager). Decisions are made on a level-wide rather than a panel-wide basis. The scheduling regimes in different levels may be very different, so there is a separate scheduler associated with each level. The schedulers associated with the vertical separation levels consider the distance between aircraft pairs when making decisions, whereas the schedulers associated with the cruise level allocation levels do not have distance information available, and so work on expiry time only.

In addition to agenda-based scheduling, there is an element of hard-wired scheduling. The hard-wired scheduler in the current version of CASSANDRA-II always prefers real airspace tasks over projection tasks. When all real airspace tasks have been executed on a given cycle, the scheduler executes projection tasks in order of the proximity of the expiry time of the task to the current system clock value.

In the input panel and the cruise level allocation panels there is little scheduling data to be had: only the task completion time and the type of airspace to which the task refers. The input panel has a number of tasks which must be performed in a given order: this ordering is, in part, performed by applying the matcher constraint mentioned above, it is complicated by the fact that unless the requisite entry is available, there will be no KS which can be matched. Thus, the initial pruning operation tends to be scheduled first (this is an element of non-determinism in the current version): the pruning KS releases an entry to other KSs on the input panel for matching on subsequent cycles. Pruning tasks cannot be rescheduled on a given entry because the matcher used in the system tags entries as having been matched successfully when the matcher succeeds on an entry.

In the cruise level allocation panel, there are two tasks which have to be performed per entry -- one to check that FL245 has not been allocated as a cruising level for the aircraft, and one to check that the aircraft has been allocated the correct level for its heading. Here again, there is only time information and no distance information upon which to make a scheduling decision, so the scheduler operates on time order and the number of tasks to be scheduled for each completion time window with a bias to real airspace.

When entries have been placed on the lower level of a panel, the agenda-based scheduler takes over. In the case of tasks which involve distance components (i.e., vertical, longitudinal and lateral separation), the agenda-based scheduler selects tasks first on the basis of real airspace versus projections and then on the separation distance. In one version of the scheduler (the one currently used), only real airspace tasks are selected until they have all been completed, at which point it considers projections in time order and then by distance. This gives a hard-and-fast scheduling rule which runs the risk that not all projection tasks will be completed: this is because real airspace tasks are presented to a panel at regular intervals (in the limit, once per radar sweep), whereas projections occur randomly -- thus a projection may never complete because it always has too low a priority.

The information flow through a panel in CASSANDRA-II is depicted in fig. 2. Information flows from the input panel (not shown) and entries are placed on the upper level. The upper level schedules tasks according to the directions issued by its local scheduler. Entries are then passed to the lower level and warning messages (where appropriate) are sent to the user.

CASSANDRA-II has a global scheduling problem as well as a local one. The behaviours of the individual panels are controlled by local scheduling decisions. The global behaviour of the system depends upon global control decisions.

In the current version of CASSANDRA-II, the global scheduler activates each level in some kind of order. The global control regime, which is quite effective, is to run the input panel first, then the cruise level allocation panel, then the vertical separation panel and then the other panels in some arbitrary order. The rationale for this is that the input panel has to run first in order to provide input for the remaining panels and that the cruise level checks are usually quite fast, as are vertical separation tests: the other panels require a certain amount of numerical computation (some of it trigonometric) which tends to be slow. Although this regime is good for testing the system, it suffers the fault that it slows the system down considerably by focussing on a panel at a time with the consequence that on a cycle, some tasks may remain which should have been executed (it also has the consequence that there is a longer garbage collection period at the end of each cycle due to the presence of unexecuted tasks on the blackboard which have to be removed).

6. GARBAGE COLLECTION

CASSANDRA-II is intended for continuous operation which entails that garbage collection is a major influence on the design. After every task execution, there are entries on the blackboard which eventually become useless and so have to be removed. In CASSANDRA-II, there is a completion time associated with every blackboard entry: if an entry is still on the blackboard after its completion time has expired it can be removed as garbage. If the garbage collection process operated in this fashion, there would be a period in the operating cycle in which the system concentrated on garbage collection: this would eventually slow the system down and send it out of synchronization with the external environment. As a result of this consideration, a continuous process has been built into the system.

The collection process is further complicated by the fact that entries in any level are part of a hierarchy. Higher level items cause lower level items to come into existence. Entries are related by two kinds of link -- a parent link which points to the upper level and a descendent link which points to a set of entries on the lower level. These links are present so that the system can provide snapshots of its state for diagnostic and, perhaps later, for explanation purposes. Any parent slot in an entry can point to at most one entry on the upper level; descendent slots can point to many entries on the lower level of the panel. No inter-panel links are permitted.

There are two basic ways in which garbage collection occurs in CASSANDRA-II. On each cycle, the KS tests to see whether or not each entry has a completion time which is greater than the current value of the system clock. If the entry has expired, the garbage collection KS removes it from the blackboard, removes all entries in the agenda which refer to the recently collected blackboard entry; finally, it removes the entry from the descendent slot of the upper level entry which was responsible for its existence. If the parent entry has an empty descendents slot, it is collected as garbage. This scheme allows gradual garbage collection of both levels of a panel. It also ensures that all agenda entries referring to a blackboard entry are removed as required.

The second method is to examine the tags left by the matcher. If an entry has been tagged by all the KSs which can possibly match it, it is collected as garbage. This, second, method of garbage collection deletes entries as soon as they have reached their minimum match potential (an entry, upon creation, has maximum match potential): it acts as a short-cut to deletion, whereas the other method waits for expiry. The two methods are included because the first may delay collecting an entry belonging to a projection which has been matched by all possible KSs well before the entry expires, and the second may never collect an entry because it has not been successfully matched by any KS on the level.

The input panel has a slightly different regime. Entries on the single level of the input panel are not chained, so link propagation is not possible. The input panel has time-based and match-based garbage collection KSs which work in the usual way. The fact that an entry has been collected on the input panel has no impact on the entries which are derived from it on other panels: this is because panels are independent of each other.

7. CONCLUSIONS

The CASSANDRA-II system has been extensively tested in a variety of simulated airspace conditions. It has met its performance criteria quite well: measurements indicate that the current implementation (in FranzLISP, running on a DEC VAX 11/750)* can handle about twice as much traffic as West Drayton

* DEC and VAX are registered trademarks of Digital Equipment Corp.

control centre presently handles. The performance of the system could be increased still further by recoding in a language with a better compiler and by distributing the system over a communications network.

The architecture of CASSANDRA-II is highly modular. Additional components of the air traffic control domain could be added with few problems. The most interesting aspect of the system is that it appears to combine some aspects of object-oriented programming with the blackboard architecture, but in a way which does not compromise the architecture's modularity constraints too much.

REFERENCES

1 L.D. Erman and V.R. Lesser, A Multi-level Organisation for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge. Proc. IJCAI 4, Vol. 2, pp. 483 - 490 (1975)

2 L.D. Erman, P. London, and S. Fickas, The Design and an Example Use of HEARSAY-III. Proc. IJCAI 7, Vol. 1, pp. 409 - 415 (1981)

3 F. Hayes-Roth and V.R. Lesser, Focus of Attention in the Hearsay-II speech understanding system Proc. IJCAI 5, pp. 27 - 35 (1977)

4 B. Hayes-Roth and F. Hayes-Roth, A Cognitive Model of Planning, *Cognitive Science,* Vol. 3., pp. 275 - 310 (1979)

5 B. Hayes-Roth, The Blackboard Architecture: A General Framework for Problem Solving? Report No. HPP-83-30, Heuristic Programming Project, Computer Science Dept., Stanford University, Palo Alto, CA (1983)

6 B. Hayes-Roth, A Blackboard Model of Control. Report No. HPP-83-38, Heuristic Programming Project, Computer Science Department, Stanford University, Palo Alto, CA., (June 1983 : Revised December, 1984)

7 B. Hayes-Roth, BB-1: An Architecture for blackboard systems that control, explain, and learn about their own behavior. Technical Report HPP-84-16, Stanford University (1984)

8 I.D. Craig, Decentralised Control in a Blackboard System, Ph.D. Dissertation, Dept. of Computing, University of Lancaster, April, 1987

9 Civil Aviation Authority, Manual of Air Traffic Services, Publication CAP 493, London

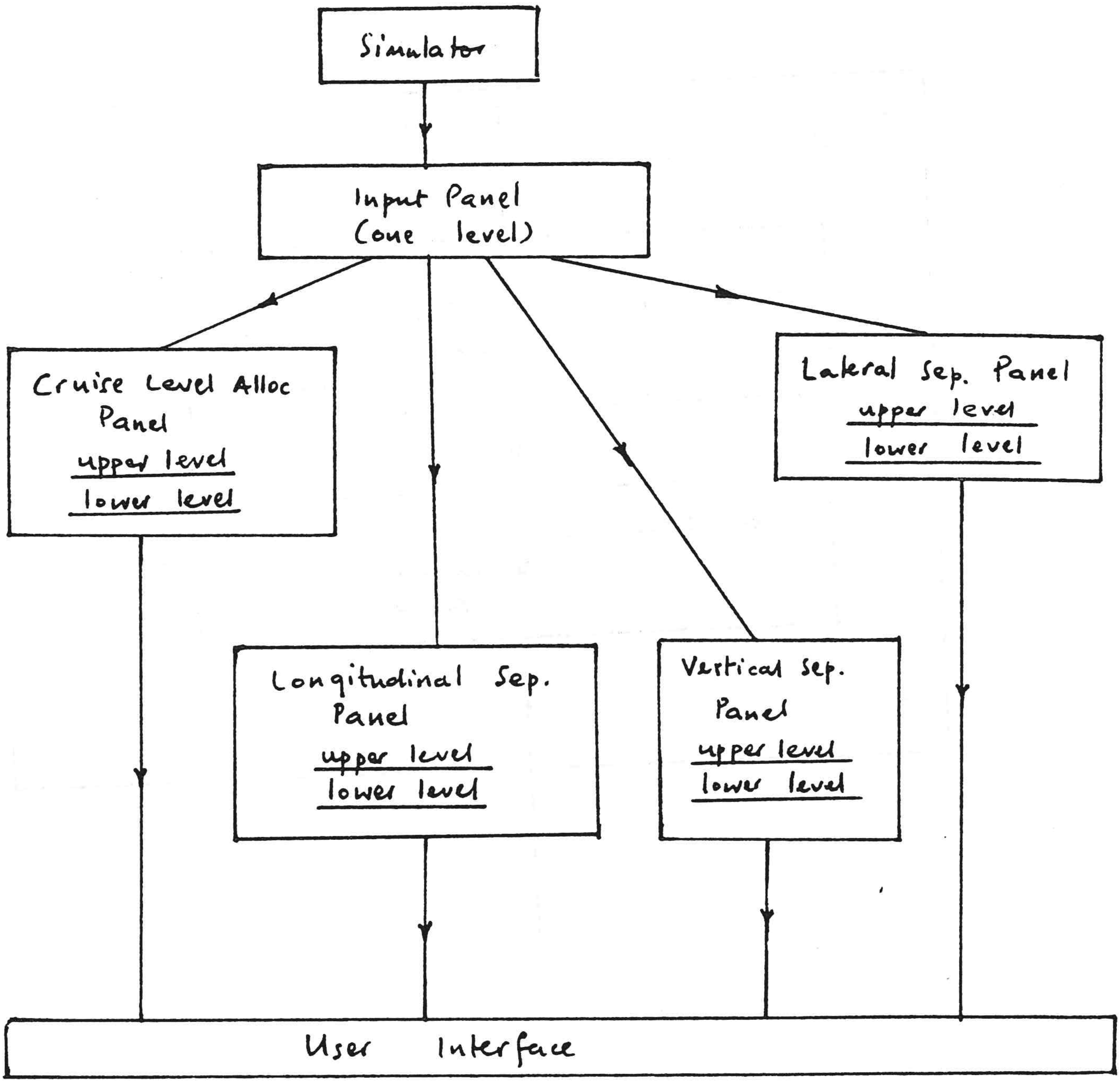

Figure 1

CASSANDRA-II global structure

(level names underlined)

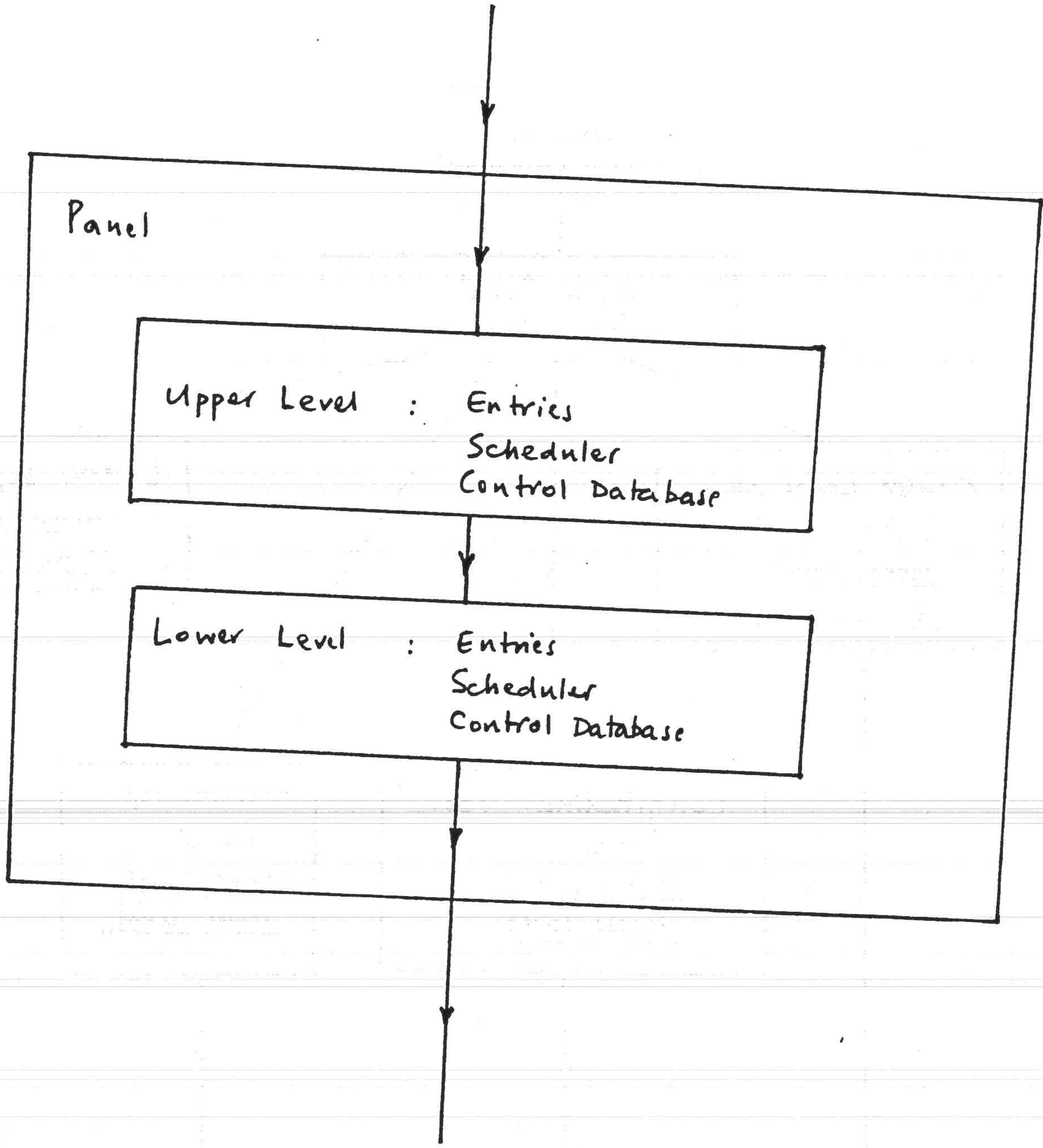

Figure 2
Information flow through a CASSANDRA-II panel