A General Framework for Slow Intelligence Systems

Shi-Kuo Chang
Department of Computer Science
University of Pittsburgh, Pittsburgh, PA 15260 USA

chang@cs.pitt.edu

Abstract: Contrary to popular belief, not all intelligent systems have Quick Intelligence. There are a surprisingly large number of intelligent systems, quasi-intelligent systems and semi-intelligent systems that have Slow Intelligence. Such Slow Intelligence Systems are often neglected in mainstream research on intelligent systems, but they are really worthy of our attention and emulation. I will describe the characteristics of Slow Intelligence Systems and present a general framework for Slow Intelligence Systems. I will then discuss evolutionary query processing and mission control in emergency management systems as two examples of Artificial Slow Intelligence Systems. Researchers and practitioners are both invited to explore the applications of Slow Intelligence Systems in software engineering and knowledge engineering, and publish their findings in IJSEKE.

Keywords: Slow intelligence systems, intelligent systems, evolutionary computing, knowledge engineering, software engineering.

(International Journal of Software Engineering and Knowledge Engineering, Volume 20, Number 1, February 2010, 1-16.)

Shi-Kuo Chang, "A General Framework for Slow Intelligence Systems", International Journal of Software Engineering and Knowledge Engineering, Volume 20, Number 1, February 2010, pp. 1

1. Introduction

I will introduce the concept of Slow Intelligence and present a general framework for defining and studying Slow Intelligence Systems (SIS). I view Slow Intelligence Systems as general purpose systems characterized by being able to improve performance over time through a process involving enumeration, propagation, adaptation, elimination and concentration. A Slow Intelligence System continuously learns, searches for new solutions and propagates and shares its experience with other peers. A Slow Intelligence System differs from expert systems in that the learning is not always obvious. A Slow Intelligence System seems to be a slow learner because it analyzes the environmental changes and carefully and gradually absorbs that into its knowledge base while maintaining synergy with the environment.

I will start with an example. The rose plant in Figure 1 in many ways behaves like a Slow Intelligence System. In the spring it will grow many branches (enumeration), which interact with the surrounding environment (propagation) to adjust their orientation (adaptation). In late spring some branches will die (elimination), and among the remaining ones only a handful will be provided with most of the resources to grow really tall (concentration). The rose plant not only manages to survive by reacting quickly to environmental changes (quick decision cycle) but also ensures the survival of its species by preserving the strongest branches (slow decision cycle). These characteristics are the trademarks of a Slow Intelligence System.

Figure 1. A rose plant as a Slow Intelligence System.

This paper is organized as follows. In Section Two I will discuss the concept of Slow Intelligence and contrast it to Quick Intelligence. The characteristics of Slow Intelligence

Systems are further examined in Section Three. Section Four describes the structure of Slow Intelligence Systems. Two examples of Artificial Slow Intelligence Systems are discussed in some detail in Section Five. Finally in Section Six I will present a framework of study to explore the applications of Slow Intelligence Systems in software engineering and knowledge engineering.

2. Slow Intelligence vs. Quick Intelligence

Most people have an intuitive notion of what intelligence is, and many words in the English language distinguish between different levels of intellectual skill: bright, dull, smart, stupid, quick, slow, and so on. Yet no universally accepted definition of intelligence exists, and people continue to debate what, exactly, it is. Fundamental questions remain: Is intelligence one general ability, or several independent systems of abilities? Is intelligence a property of the brain, a characteristic of behavior, or a set of knowledge and skills?

The simplest definition proposed is that "*intelligence is whatever intelligence tests measure*" (Encarta Encyclopedia). However such a definition not only is circular but also tends to overlook what can be called Slow Intelligence. In fact, intelligence tests such as IQ tests have time limits and thus favor Quick Intelligence.

What is Slow Intelligence? An interesting story was related by a blogger who calls himself "Master Luke": "I just recently finished reading Robert Pirsig's *Zen and the Art of Motorcycle Maintenance*. There's a point toward the end of the book where Pirsig relates his experiences of a few philosophy graduate seminars at the University of Chicago. At one point, the professor asks him his opinion on, I believe, a Socratic dialogue. Pirsig is unable to respond, not because he's dumb, but because, after the question is asked, he begins to replay in his head possible answers and their logical consequences and contradictions--over and over, until everyone has left the classroom except Pirsig." [1]

Master Luke calls this sort of calculating intelligence "Slow Intelligence", which is *the ability to step back and calculate possibilities without any sort of outside constraints (like time)*: "(Slow intelligence) seems indispensable to our evolutionary history. So why the emphasis on Quick Intelligence in the West? Our educational system generally weeds out the slow and assimilates the quick. Maybe it's just easier? The problem is that the person of Slow Intelligence whose GRE scores (for example) are the only thing holding him back might never get the opportunity to shine." Continues Master Luke: "We are told that Charles Darwin did not have a particularly sharp personality. His virtue was in his ability to sit down, think, and gradually (and meticulously) work out the consequences of whatever the hell he was writing about--evolution or species of butterfly. He probably would have done miserably on the GRE or SAT." [1]

According to Master Luke, Slow Intelligence began to develop more and more as our cities and societies became more and more distant from Nature. In other words Slow Intelligence is distinctively human, although it must be recognized that in our evolution history Slow Intelligence is present everywhere in Natural Slow Intelligence Systems such as plants.

In Eastern philosophy there are also clear recognitions of the value of Slow Intelligence. The Chinese philosopher Confucius says, "Only the wisest man and the man with Slow Intelligence never waver." Slow Intelligence and Quick Intelligence are also contrasted in Buddhist Philosophy. For example in Nyanaponika Thera's teachings, Slow Intelligence and Quick Intelligence each have their advantages in learning and in meditation training. Says Nyanaponika Thera: "For giving up the belief in permanancy, it is easier for the theorizing type of Slow Intelligence to see the inpermanance of consciousness (Citta) in its not-too-diversified classification such as mind and lust." [2]

We can distinguish Natural Intelligence Systems and Artificial Intelligence Systems, in that Natural Intelligence Systems are created and perfected by Nature over time, and Artificial Intelligence Systems are intentionally created by Man, usually within a relatively short span of time. The characteristics of Slow Intelligence manifest themselves in both Natural Slow Intelligence Systems and Artificial Slow Intelligence Systems. Two examples of the latter kind will be described in Section 5.

3. Characteristics of Slow Intelligence Systems

A slow intelligence system is a system that *(i) solves problems by trying different solutions, (ii) is context-aware to adapt to different situations and to propagate knowledge, and (iii) may not perform well in the short run but continuously learns to improve its performance over time*.

The above definition is from the behavioral viewpoint (the black box approach). Our goal is to understand the characteristics and structure of both natural and artificial slow intelligence systems (the white box approach).

Slow Intelligence Systems typically exhibit the following characteristics:

Enumeration: In problem solving different solutions are enumerated, until the appropriate solution or solutions are found.

Propagation: The system is aware of its environment and constantly exchanges information with the environment. Through this constant information exchange, one SIS may propagate information and/or knowledge to other (logically or physically adjacent) SISs.

Adaptation: Solutions are enumerated and adapted to the environment. Sometimes adapted solutions are mutations that transcend enumerated solutions of the past.

Elimination: Unsuitable solutions are eliminated, so that only suitable solutions are further considered.

Concentration: Among the suitable solutions left, resources are further concentrated to only one (or at most a few) of the suitable solutions.

The above five characteristics are well known to the Darwinists. The sixth one, on the other hand, is rather unique for SIS:

Slow decision cycle(s) to complement quick decision cycle(s): SIS possesses at least two decision cycles. The first one, defined as the *quick decision cycle*, provides an instantaneous response to the environment. The second one, defined as the *slow decision cycle,* tries to follow the gradual changes in the environment and analyze the information acquired by experts and past experiences. The two decision cycles enable the SIS to both cope with the environment and meet long-term goals.

Sophisticated SIS may possess multiple slow decision cycles and multiple quick decision cycles. Most importantly, actions of slow decision cycle(s) may override actions of quick decision cycle(s), resulting in poorer performance in the short run but better performance in the long run.

4. Structure of Slow Intelligence Systems

As explained in Section 3, from the structural point of view a Slow Intelligence System is a system with multiple decision cycles such that actions of slow decision cycle(s) may override actions of quick decision cycle(s), resulting in poorer performance in the short run but better performance in the long run.

Now we can consider the structure of SIS by introducing the basic building block and advanced building block. In general a SIS acts according to five main phases, reflecting its five characteristics:

- **Enumeration**: In this phase SIS enumerates all the possible solutions of a task, including new solutions (mutations) that transcend enumerated solutions of the past.
- **Propagation**: During the search of solution for a task, or after the resolution of a task, SIS updates its experience and propagates and shares the new information with other peers.
- **Adaptation**: In this phase SIS acquires information about the environment in which it is situated, to adapt the solutions to the environment.
- **Elimination**: In this phase, SIS, acting according to the information acquired in the previous phases, selects the feasible solutions to solve a task. Information acquired from the environment as well as learned experiences are used.
- **Concentration**: After the selection of the feasible solutions for solving a task, SIS concentrates its resources to pursue only a few of the solutions.

SIS Basic Building Block (BBB): The Basic Building Block (BBB) incorporates the above phases in its problem solving activities. A basic building block is illustrated in Figure 2.

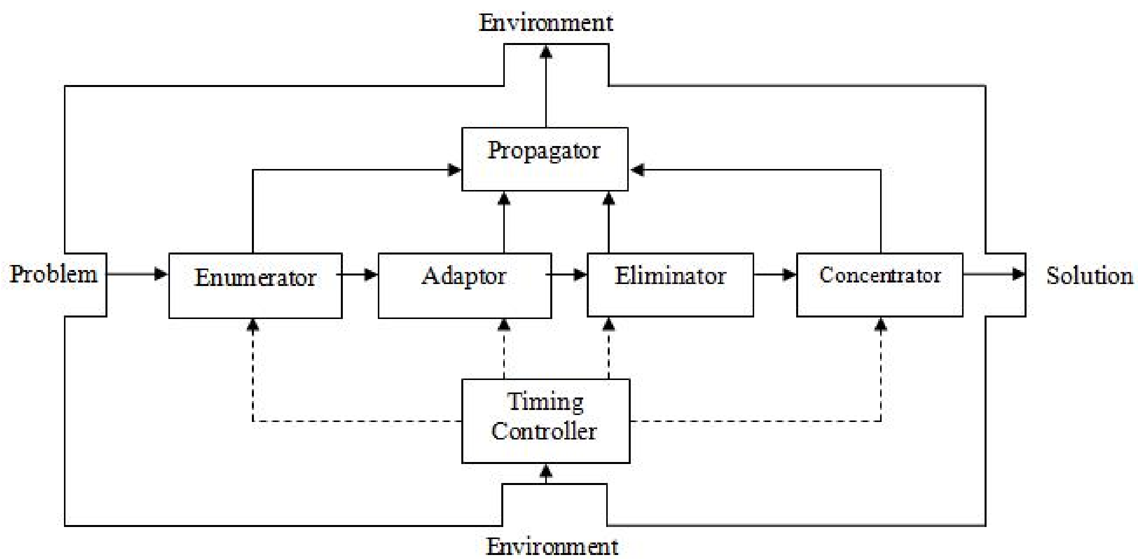

Figure 2. The basic building block BBB.

Problem and solution are both functions of time. Thus we can represent the time function for problem as $x(t)_{problem}$, and the time function for solution as $y(t)_{solution}$. The timing controller is also a time function timing-control(t). For the two-decision-cycle SIS, the basic building block BBB can be expressed as follows:

if timing-control(t) == 'slow'
then /* timing-control(t) is 'slow' */
$y(t)_{solution} = g_{concentrate}(g_{eliminate}(g_{adapt}(g_{enumerate}(x(t)_{problem}))))$
else /* timing-control(t) is not 'slow' */
$y(t)_{solution} = f_{concentrate}(f_{eliminate}(f_{adapt}(f_{enumerate}(x(t)_{problem}))))$

where $g_{enumerate}$, g_{adapt}, $g_{eliminate}$, and $g_{concentrate}$ are the transform functions for enumeration, adaptation, elimination and concentration respectively during slow decision cycles, and $f_{enumerate}$, f_{adapt}, $f_{eliminate}$, and $f_{concentrate}$ are the transform functions for enumeration, adaptation, elimination and concentration respectively during quick decision cycles.

Depending on the **level of abstraction**, the BBB can be viewed differently: At the **problem solving level** the BBB is a *pattern* incorporating the above described phases in its problem solving activities. At the **model level** the BBB is a *network* consisting of interconnected cells (such as the active index [3]). At the **implementation level** the BBB is either a *software system* consisting of interacting software components, or alternatively a *hardware system* consisting of interconnected circuits.

SIS Advanced Building Block (ABB):

An Advanced Building Block can be a stand-alone system as shown in Figure 3. The major difference between an ABB and a BBB is the inclusion of a *knowledge base*, further improving the SIS's problem solving abilities.

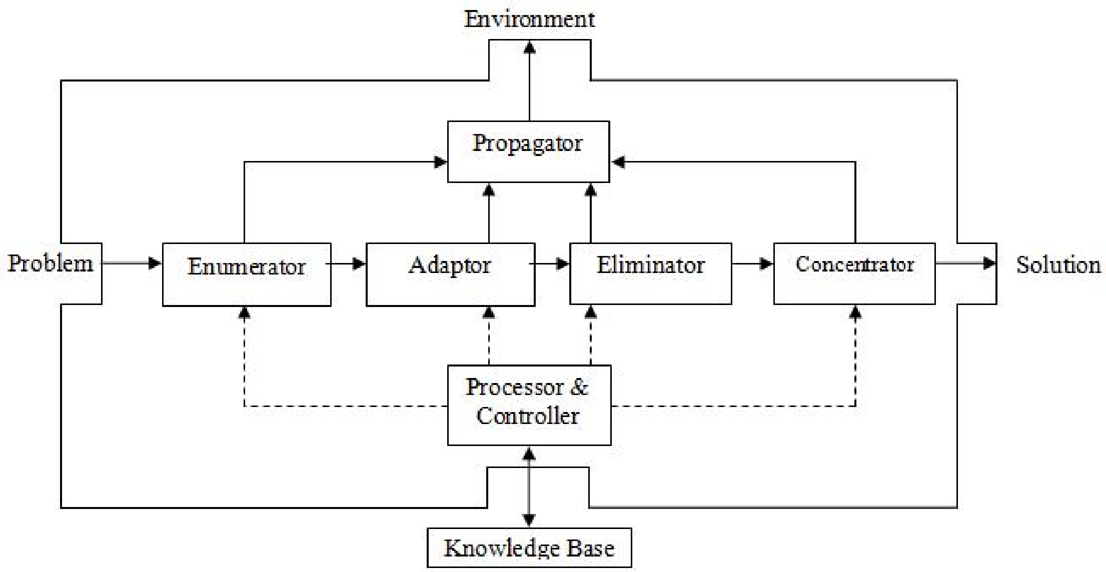

Figure 3. The advanced building block ABB.

Complex SIS constructed from BBBs:

As shown in Figure 4, a complex SIS can be constructed from the BBBs and ABBs governed by slow and quick decision cycles with feedbacks at multiple levels.

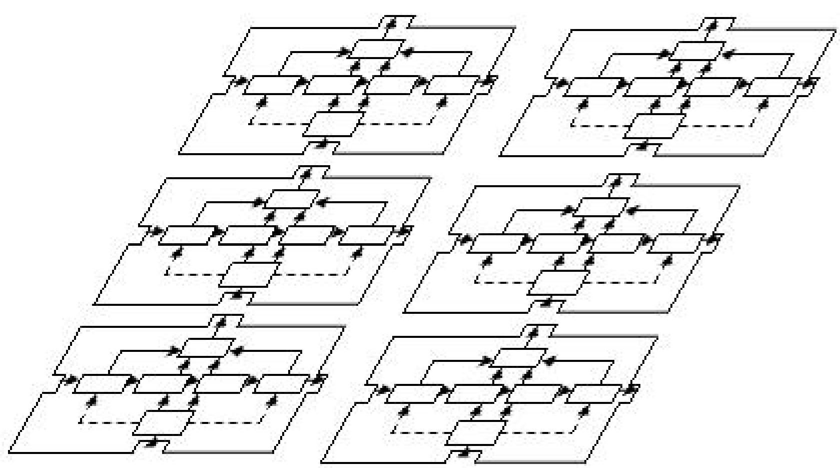

Figure 4. Complex SIS constructed from BBBs and ABBs.

Again, we can view the complex SIS at different levels of abstraction: At the **problem solving level** the complex SIS is a *structured set of patterns*. At the **model level** the complex SIS is a *nested network structure*. At the **implementation level** the complex SIS is either a *complex software system* consisting of encapsulated subsystems of interacting software components, or alternatively a *complex hardware system* consisting of interconnected circuits.

5. Examples of Artificial Slow Intelligence Systems

In this section I will present two examples of Artificial Slow Intelligence Systems. As the first example I will describe evolutionary continuous query (CQ) processing for sensor-based multimedia data streams. As the second example I will illustrate mission initiation and control in emergency/crisis management systems.

5.1. Evolutionary Continuous Query Processing

As discussed above, slow intelligence systems have the following characteristics: Enumeration, Propagation, Adaptation, Elimination, Concentration, and Slow Decision Cycle(s) to complement Quick Decision Cycle(s). Evolutionary query processing system possesses many of the above characteristics.

The basic concept of multimedia data streams query processing is illustrated in Figure 5. Multiple multimedia data streams such as sensor data streams and video streams consist of time sequences of tuples (or frames when dealing with video data). They are either captured and stored, or processed on the fly as the tuples (or frames) arrive. The multimedia continuous queries or m-CQ's are query trees whose nodes are operators on multimedia data.

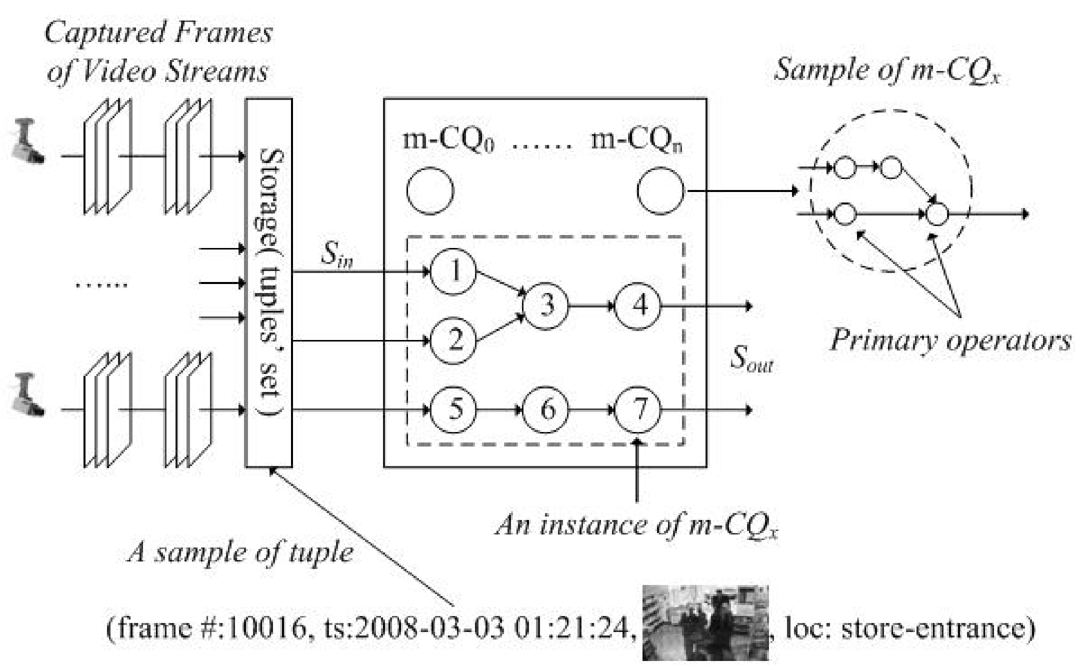

Figure 5. Multimedia data streams query processing.

Continuous Queries (Enumeration):

The Multimedia Data Stream Management System (MMDSMS) is as shown in Figure 6. A user submits a request that is transformed by the user interface into a m-CQ, which is stored and repetitively executed by MMDSMS to produce the query results. Continuous queries are queries that will be repetitively processed over time.

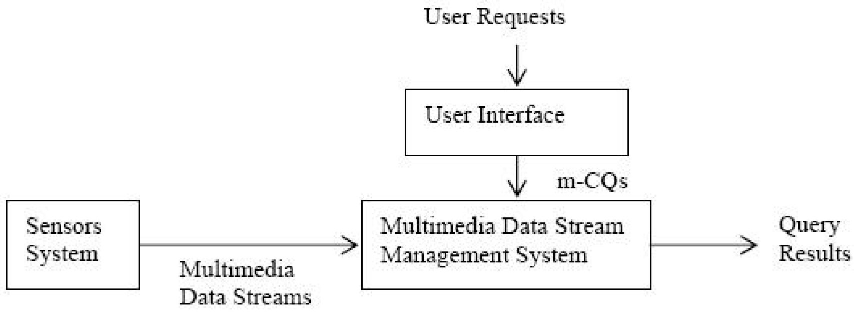

Figure 6. The basic MMDSMS system architecture.

Query Evolution (Adaptation):

In sensor-based information systems, we need to consider evolutionary queries that can change in time [4]. For example in emergency management systems, an emergency worker walks into an area of high risk (fire, radiation, etc.). The spatial/temporal position of the query originator keeps on changing. Evolutionary queries can be generated to identify targets of high risk.

This approach gives better results because the evolutionary query can improve the result from the various sensor data that will also lead to a better result in the fusion process. An evolutionary query may also send a request for new data and thus leads to a feedback process. Normally, refined queries are created to deal with additional constraints to make a query more precise. In this approach, the evolutionary queries can also be created to deal with the lack of information from a certain sensor.

Figure 7 illustrates the three images obtained by an MMDSMS that has three types of sensors: infrared video, laser radar and CCD camera, respectively.

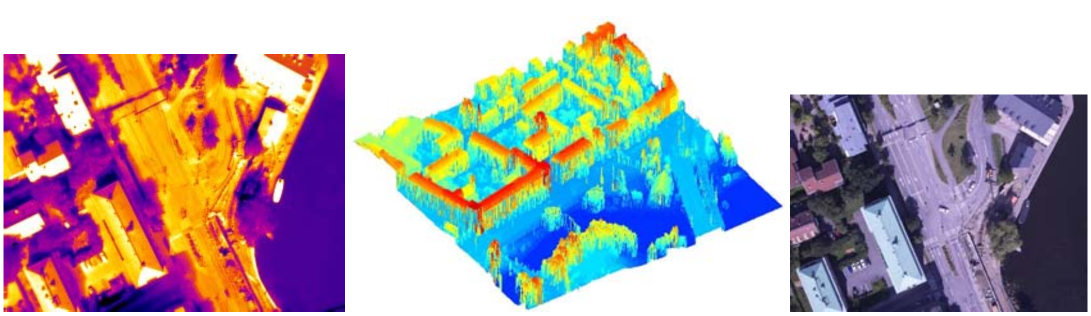

Figure 7. Infrared image (left), laser radar image (center) and CCD camera image (right).

A study was carried out jointly by the University of Pittsburgh and the Swedish Defense Agency FOA [5]. For empirical study FOA collected over 700 G bytes of data from these three types of sensors. An example SigmaQL query is shown in Figure 8.

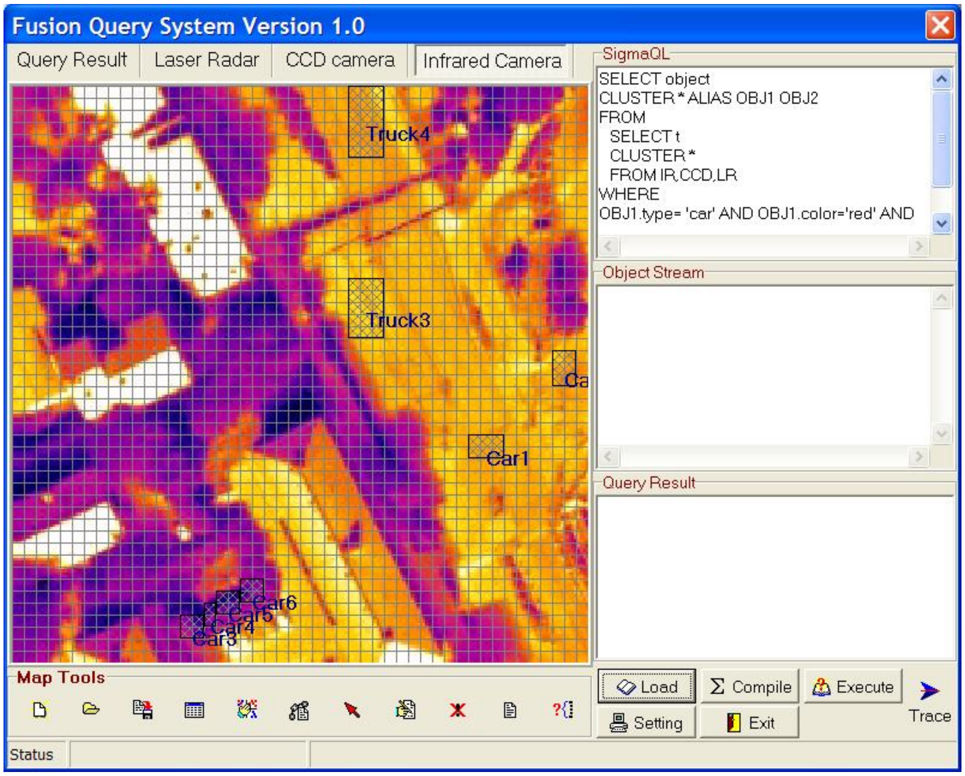

Figure 8. An example SigmaQL query.

The system should be able to detect that some sensors are not providing information, or giving the complete view of the scene, and automatically choose and query the sensors that can help in giving the whole scene. In order to do so the system should have the knowledge about the real world and the sensors. Such a system needs an ontology.

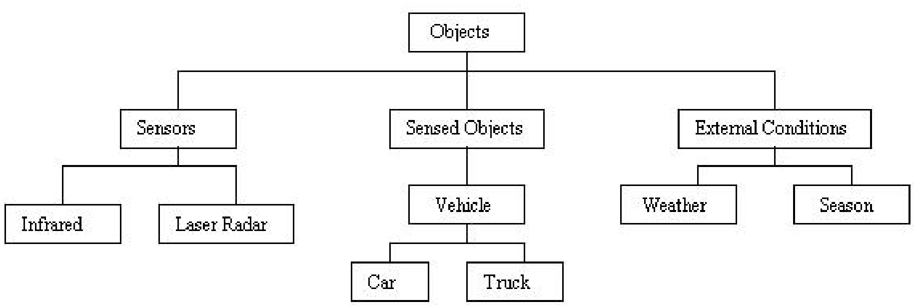

Figure 9. The ontological knowledge base.

As illustrated by Figure 9 this knowledge could be stored in the *ontological knowledge* base consisting of three parts: the sensor part describing the sensors, recognition algorithms and so on, the external conditions part providing a description of external conditions such as weather condition, light condition and so on, and the sensible things part describing objects to be sensed. Multimedia dependencies and constraints for the multiple (quick/slow) decision cycles can also be integrated into the ontological knowledge base.

In our approach we developed a Multimedia Data Streams model (MMDSM) to provide a formal framework to achieve efficient content based retrieval [6]. We also extended the Multimedia Functional Dependency Theory and the Normalization Framework to handle Multimedia Data Streams [7]. Finally, we developed an approach of processing Continuous Multimedia Queries using ontological filters. It is the last part that we will elaborate more in what follows.

Ontological Filtering for Queries (Elimination)

Given the external condition and the object to be sensed, we can determine what sensor(s) and recognition algorithm(s) can be applied through the use of ontological filters. An ontological filter transforms a query into a more efficient query (Figure 10). For example, if we are looking for a car travelling at speed over 100 mph, then the ontology can tell us only to look for certain type of cars (Figure 11). The other type of cars can be eliminated from the query and consequently the solutions as well.

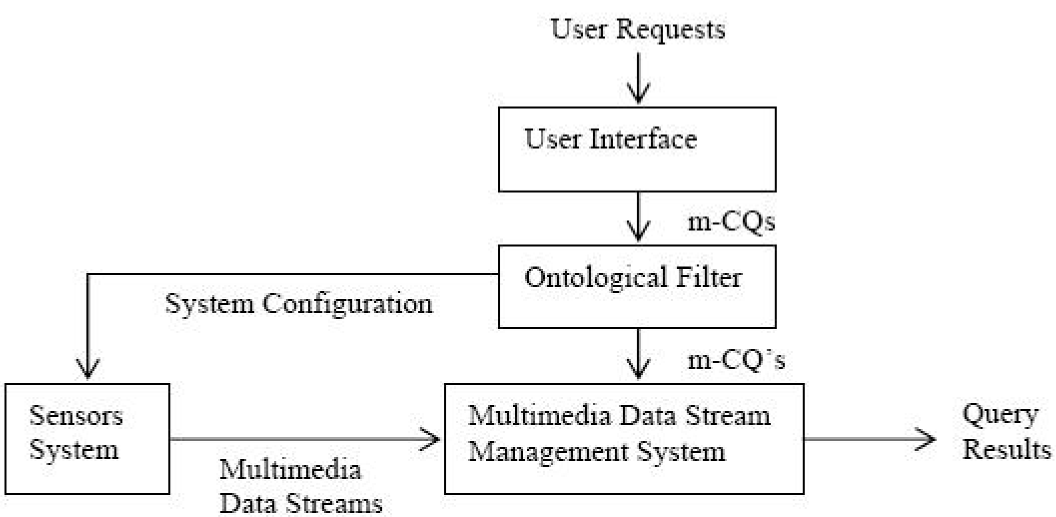

Figure 10. Ontological filtering for queries.

Ontological Filtering for Sensors (Concentration)

An ontological filter can also provide system configuration input to sensors system (Figure 10). For example, the ontology can tell the sensors system to configure its laser radar to detect speedy car. Therefore both the sensors and the query processing will concentrate on such solutions.

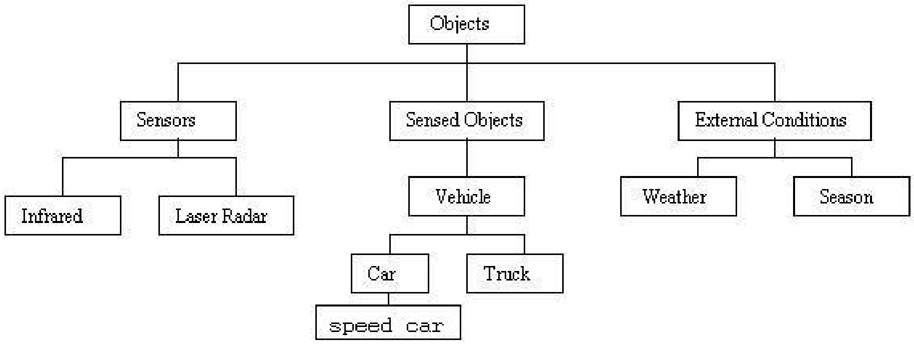

Figure 11. An ontology with a speed car.

The Lightweight Plus Ontology with non-hierarchical relations can be derived from the Lightweight Ontology with hierarchical relations. Multimedia functional dependencies can also be incrementally added to the Lightweight Plus Ontology. The ontology thus also evolves in time. In other words, the knowledge base is slowly changing. Simulation results of processing randomly generated query graphs indicate the advantage of using ontological filters to improve query processing (Figure 12).

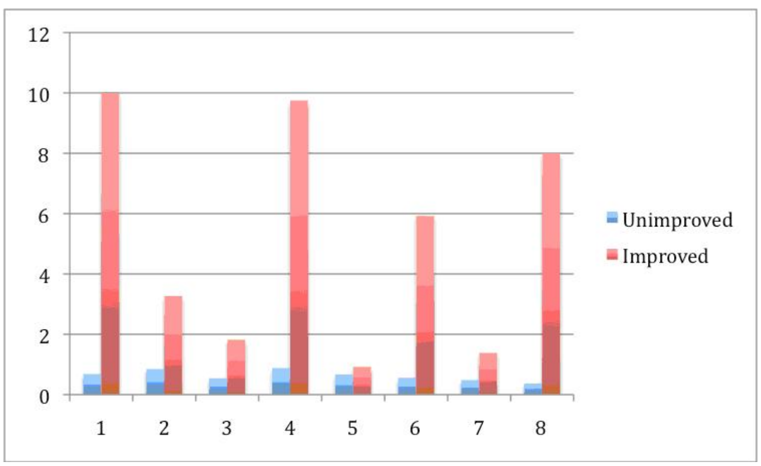

Figure 12. Simulation results.

5.2. Mission Initiation and Control in Emergency/Crisis Management

In crisis/emergency management the involved organizations are continuously working on what to do for all possible events that may occur before and during a crisis/emergency. However it is well known that the first few minutes of an emergency/crisis are the most critical regardless it is medical emergency, fire, flood or other crisis.

Our concern is how agents needed for a mission/operation that will be part of the emergency/crisis management operations can be self-organizing to initialize and control their own missions/operations [8]. The self-organizing process can be seen as part of a management cycle in which the instantiated agent(s) may repeatedly consult some sort of handbook for further information. The handbook is the knowledge base. The instantiated agents can also start a series of **mission control loops**, as shown in Figure 13.

In any mission, with just a few exceptions, the responsible agent must start at least one control loop. Basically, going through a control loop means that a controller (here an agent) develops a plan so that in the end an action of some kind must be carried out. For instance in the case of a forest fire the action can be to water bomb the area from an aircraft.

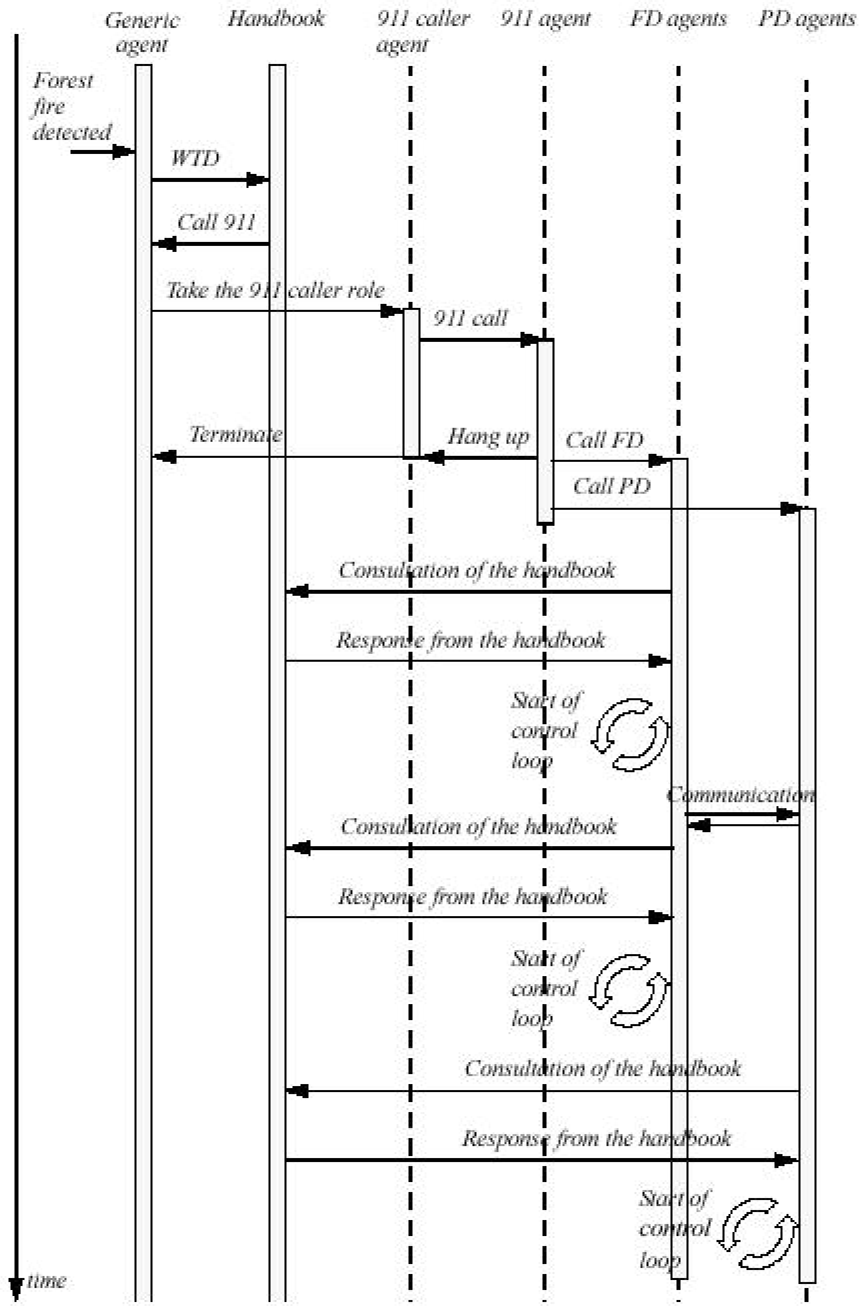

Figure 13. Initial phase of a forest fire.

In the case of forest fire, the event may have started in some unidentified way but it is discovered by some individual who just happens to be present at the place. This individual corresponds to a generic agent and will act as a first responder. After having consulted the Handbook of Protocols, which is available or accessible from his personal digital assistant, the generic agent turns into an agent with the role of a 911-caller. After the call is placed a 911-agent (operator) is activated. The two agents communicate and the caller transmits the event description. Once the event description is transmitted the call is finished and as the caller hangs up the agent goes back to his normal role. The 911-agent, on the other hand, consults the Handbook, if required, and as a consequence engages in its role as an activator of other agents who in this case are the agents at the police and fire departments. The 911-agent now initiates a mission control loop.

Figure 13 is a modified sequence diagram that depicts several such mission control loops. Some of these control loops are short-term (**quick decision cycle**) and others are long term (**slow decision cycle**). The interplay of multiple decision cycles enables the system to handle both initialization and subsequent crisis management operations.

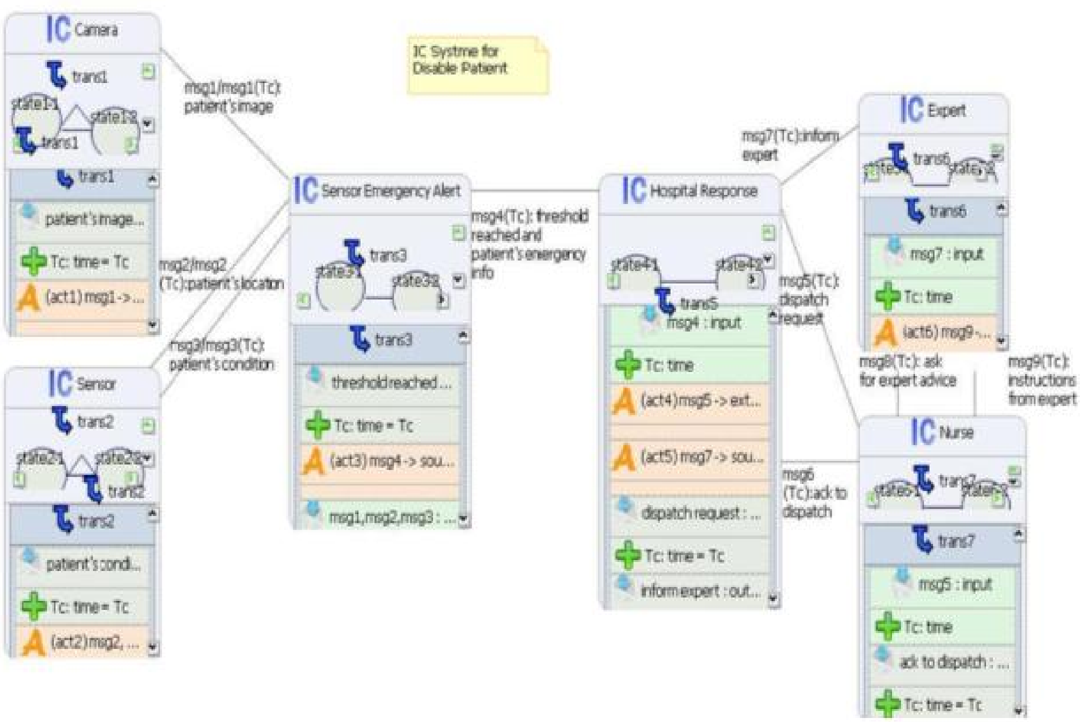

Figure 14. An active index for medical emergency management.

We can also model a Slow Intelligence System as an active index [3]. An example of an active index for medical emergency management is illustrated in Figure 14, where a number of index cells interact during an emergency, and message loops among the index cells can be formed for slow and/or quick decision cycles.

6. A Framework of Study

Two important points need to be emphasized: (1) *Since time is relative, a Slow Intelligence System for some can also be a Quick Intelligence System for others.* (2) *A Slow Intelligence System can evolve into a Quick Intelligence System and vice versa.*

There are a large number of intelligent systems, quasi-intelligent systems and semi-intelligent systems that are "slow". Distributed intelligence systems, multiple agents systems and emergency management systems may exhibit some of the characteristics of Slow Intelligence Systems. Both Natural Slow Intelligence Systems and Artificial Slow Intelligence Systems can be studied and explored.

Study of Natural Slow Intelligence Systems: There are numerous Natural Slow Intelligence Systems to be studied, including ecological systems, social networks and so on. I believe the principled study of social networks as Slow Intelligence Systems may lead to a deeper understanding on how and why such systems work or do not work.

Study of Artificial Slow Intelligence Systems: Topics to be studied may include, but are not limited to, the following: the effects and interplay of multiple decision cycles, levels of abstraction, system architecture, evolutionary ontology, knowledge propagation rules, learning rules, prototype implementation, visual analytics for SIS and visual semantics or SIS and so on.

Researchers and practitioners are invited to explore the applications of Slow Intelligence Systems in software engineering and knowledge engineering, and publish their findings in IJSEKE. A workshop on Slow Intelligence Systems will also be organized concurrent with the DMS2010 conference, and more details can be found at the website: www.ksi.edu/seke/dms10.html. Researchers are also welcome to present their preliminary findings or prototype systems at the DMS2010 Conference. In the future there will be similar workshops on SIS concurrent with SEKE conferences and/or DMS conferences. The contributions from researchers and practitioners are solicited.

References:

[1] http://philosophywithpair.blogspot.com/2009/06/whats-wrong-with-slow-intelligence.html posted by "Master Luke", June 20, 2009.

[2] Nyanaponika Thera, *The heart of Buddhist Meditation : A Handbook of Mental Training Based on the Buddha's Way of Mindfulness,* first American Edition published by Red Wheel/Weiser LLC (www.redwheelweiser.com), York Beach, ME, 1965.

[3] Shi-Kuo Chang, "Towards a Theory of Active Index", *Journal of Visual Languages and Computing*, Vol. 6, No. 1, March 1995, 101-118.

[4] Shi-Kuo Chang, Weiying Dai, Stephen Hughes, Prasad S. Lakkavaram and Xin Li. "Evolutionary Query Processing, Fusion and Visualization", Proc. of Int'l Conf. on Distributed Multimedia Systems, Sept. 20-2, 2003, San Francisco.

[5] Shi-Kuo Chang, Gennaro Costagliola, Erland Jungert and Francesco Orciuoli, "Querying Distributed Multimedia Databases and Data Sources for Sensor Data Fusion", *IEEE Trans. on Multimedia*, Vol. 6, No. 5, October 2004, 687-702.

[6] Shi-Kuo Chang, Lei Zhao, Shenoda Girguis, Rohit Kulkarni, "A Computation-Oriented Multimedia Data Streams Model for Content-Based Information Retrieval", *Journal of Multimedia Tools and Applications*, Volume 46, Issue 2 (2010), pp. 399-423.

[7] Shi-Kuo Chang, V. Deufemia, and G. Polese. A Normalization Framework for Multimedia Databases. *IEEE Transactions on Knowledge and Data Engineering*, v. 19, issue 12, Dec 2007.

[8] Shi-Kuo Chang and Erland Jungert, "A Self-Organizing Approach to Mission Initiation and Control in Emergency/Crisis Management", Proc. of Int'l Conf. on Distributed Multimedia Systems, Sept. 6-8, 2007, San Francisco.