Generic Architecture for Knowledge Base Learning Control System

Aboubekeur Hamdi-Cherif

Abstract— In order to be fully self-controlled, a system needs built-in knowledge obtained by refining information as a further processing of data. All these three components, *i.e.* data, information and knowledge map different hierarchical levels in machine intelligence, with knowledge representing the most complex form. In order to account for diversified situations and issues encountered in control settings embracing both engineering and managerial issues, a generic architecture for knowledge base learning control system (KBLCS) is proposed. Relying on the main technologies available to us, including Web, data mining and cloud computing technologies, we describe the proposed architecture for numeric/symbolic data processing that is capable of addressing issues related not only to knowledge but also to its meaning as used in the characterization of the imprecision and incompleteness of the controlled plant, especially in decision support systems (DSSs) settings. Some tasks trade-offs as part of the control process are also considered.

Keywords— Intelligent control, Intelligent systems design, Knowledge base control system (KBCS), Learning control system, Generic control systems architecture, Hybrid systems, Control engineering processes.

I. Introduction

We consider a knowledge base learning control system (KBLCS) methodology as a generic architecture within the broader context of intelligent control. Unlike conventional control, intelligent control does not level itself to precise formalization. The shift from the first aproach to the second has been done along several lines of sophistication. A swift transition has been done from prescription to description, from model of the system to the model of operation, from relational aspects to rule-based ones, from crisp logic to fuzzy logic, and from rigidity to elasticity of concepts, [27] Additionally, machine learning procedures and concepts have been adopted within intelligent control and incorporated neural networks, genetic algorithms, data mining and other hybrid methods among others [28]. Intelligence-based or AI-based programs and procedures are the basic characteristics of KBLCS. Indeed, for reasons of prohibitive complexity, generated by the "curse of dimensionality", the shear use of brute force procedural heuristics-free algorithms is simply useless. Clearly, because of the nature of most control problems, a more 'intelligent', *i.e.* heuristics-based problem solving approach is required since human expertise codification is necessary. KBLCS principles and tools enhance existing solutions because they incorporate advanced human expertise in the form of easily-modifiable code matching the human expertise improvement. Furthermore, the intelligent control community has adopted general guidelines by which operational constraints continuously adjust the focus of world modeling, the specification of control criteria, and the allocation of resources to address current goals and metagoals. This flexibility is clearly required from systems which are confronted with unfamiliar and uncertain environments, which are subject to critical spatial and temporal constraints and which are expected to perform satisfactorily in a diverse task domain.

As a result to commitment to integrated systems, researchers were challenged to reconsider the assumptions underlying research in modular technologies, [16]. To operate autonomously in a dynamic and non-deterministic environments, such systems must meet the following requirements :

1. Observe the salient features of the world and incrementally refine and extend world models using past and actual results. Here, the world refers to either the physical world where designed systems are to evolve or to the abstract world of design.
2. Construct control procedures which can make use of incomplete and uncertain information and which recover smoothly from the errors that might occur.
3. Determine how to partition available resources dynamically as constraints on timing and solution quality vary.

To contribute to these challenging issues, we propose a generic architecture using the knowledge base learning control system (KBLCS) approach. We further consider Web technologies especially semantic Web (or Web 3.0) for modeling and controlling complex systems especially as a decision support system (DSS) that leaps from knowledge (usually identified with its representation) to meaning [24].

Manuscript received March 20, 2012.

A. Hamdi-Cherif is with Computer College, Qassim University, Saudi Arabia. His permanent address is Faculty of Engineering, Ferhat Abbas University, Setif 19000 (UFAS), Algeria. (e-mail: elhamdi62@gmail.com, shrief@qu.edu.sa) . Tel. +96663800050 ext. 3301.

In the next section we give an overview of the basic ideas and issues raised in intelligent control settings. Section 3 describes the technologies useful for the development of the proposed architecture. In Section 4, an overview of the architecture is described with its main components outlined in Section 5. The interactions with the architecture are described in Section 6. The paper ends with a conclusion summing up the main results and pointing towards some potential future developments.

II. Basic Ideas

A. From conventional to knowledge base control

In control systems settings, a system is said to be controlled if it automatically generates the required control laws, by feedback or otherwise, that guide it towards the achievement of prescribed task or set of tasks in structured or unstructured, deterministic or stochastic, familiar or unfamiliar, known or hostile environments. In order to account for these diversified situations and issues, the so-called knowledge base control system (KBCS) paradigm has been investigated for the last four decades, or so. KBCS represent an artificial intelligence-based paradigm that relies on the use, generation and management of knowledge in control systems. As in any intelligent machine, the codification of knowledge in KBCSs is responsible for the performance of anthropomorphic tasks, autonomously or interactively with a human operator usually in partially-known or unknown environments.

In order to describe the main building blocks of our generic architecture, we make an emphasis on technologies spanning (crude) data, information, refined information including decision support, ultimately leading to the most refined and expensive form of information, *i.e.,* knowledge and its discovery in large and diversified databases over the Web, based on cloud computing solutions. A human-machine interactive knowledge-based learning control system (KBLCS) is our far-reaching goal.

B. Related works

Intelligent control is a term coined in [7] and later developed in [25]. An early, but constantly refined definition of this field describes itself as that area beyond adaptive, learning and self-organizing systems which represents the meeting point between artificial intelligence (AI), automatic control (AC) and operations research (OR). International intelligent control symposia have been held every year since 1985 and numerous contributions appear regularly in the specialized and thoroughly documented literature where novel original definitions of the field are proposed *e.g.* [21]. Extensions of the field are reported in [3], [4], [5] and [9]. Theoretical and applied results go back to [25] who proposed the so-called entropy-based “principle of decreasing precision with increasing intelligence”. Other approaches have also been considered by researchers like the cognition-oriented approach with applications, [16] and soft computing based control [28].

On the managerial intelligent control level, there exist three main integration models which are:

- CIM, or computer integrating manufacturing encompasses various modes of automation.
- MES or manufacturing execution systems which, according to the international association of the same name, “deliver information that enables the optimization production activities from order launch to finished goods”.
- ANSI/ISA-95, or ISA-95 is an International Society for Automation’s (ISA) multi part standard for enterprise/control system integration. It has been developed as an automated interface between enterprise and control systems and destined to global manufacturers in all industries, and in all sorts of processes, such as batch, continuous and repetitive processes. ISA-95 offers models and standard terminology for determining the flow of information between the different departments in the enterprise such as sales, finance, logistics, production, maintenance and quality [29].

C. Intelligent control central issue

One of the fundamental issues that concerns intelligent control is related to the degree to which it is possible to control the dynamic behaviour of a system, including the abstract design process, independently of:

- its complexity,
- our capability of separating it from the environment and localizing it,
- the context in which this system operates,
- the forms of knowledge available and the categories it manipulates,
- the methods of representation.

As stated, this question cannot be handled by either control theory or AI. On the one hand, control theory has a vey localized vision of the problem. This prevents it from looking beyond the localized constraints self-imposed by the designer and hidden within the mechanism of the mathematical representation. On the other hand, the available methods in AI cannot easily handle dynamic systems and have very little consideration for numerical manipulation. Indeed, computations of margins of stability, controllability, observability are indeed alien to this field.

D. Scope of intelligent control

Intelligent control as a discipline provides a generalization of the existing control theories and methods on the basis of the following elements [4]:

- combined analysis of the plant and its control criteria,
- processes of multisensor operation with information (knowledge) integration and recognition in the loop,
- man-machine cooperative activities, including imitation and substitution of the human operator,
- computer structures representing these elements.

III. Technological Components

The knowledge base learning control system (KBLCS) depicted in Fig. 1 below shows the interaction of the main technologies that are useful in the formulation of our control problem. KBLCS and its core technologies are described and used for enhancing actual control systems. It stresses the transition from knowledge ready-made use to its discovery, including Web technologies and cloud computing.

KBCLS incrementally builds on tools like data, information and knowledge, in addition to decision, discovery and control as direct procedures with pervasiveness and learning as supporting or feedback procedures.

A. From data to control

We stress the stratification of KBCLS processes into levels. This stratification motivates for the introduction of incrementally-sophisticated technologies. The proposed solution to the issues addressed here is to be considered under six hierarchical and complementary levels, namely data base management system (DBMS), information system (IS), decision support system (DSS), knowledge-based system (KBS), data mining (DM) and knowledge base control system (KBCS). One of the main aims is to integrate learning methods, encompassed by the knowledge base learning control system (KBLCS) framework.

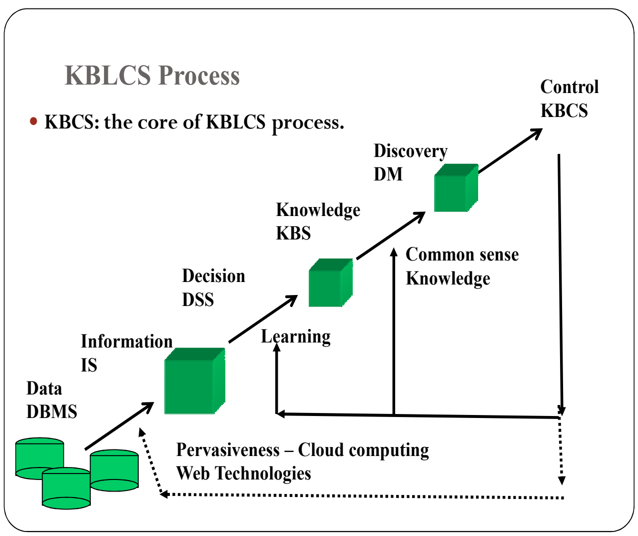

Figure 1 Basic layers of KBLCS

1. Machine learning

In KBLCS, machine learning acts as an inner loop. Machine learning is an adaptive process that makes computers improve from experience, by example, by analogy, or otherwise. Learning capabilities are therefore essential for automatically improving the performance of a computational or control system over time on the basis of previous history. A basic learning model typically consists of the following four components:

- learning element, responsible for improving its performance,
- performance element, or decision support system (DSS) responsible for the choice of actions to be taken, whether these actions are decisions or controls,
- critical element, which tells the learning element whether the criteria are met within some critical boundaries,

and

- problem generator, responsible for suggesting actions that could lead to new or informative experiences [23].

The importance of learning automatically grows as the external environment continues to generate and integrate large quantities of diversified data, generated by the problem generator, or otherwise.

2. Supervised vs. unsupervised learning

As far as relevant machine learning is concerned, there are basically two categories of learning schemes, namely supervised learning and unsupervised learning. Supervised learning learns the data with a known answer at hand. From the control standpoint, this is a typical feedback control system. The parameters are modified according to the difference of the actual output and the desired known output, or expected answer. Grammatical control is one aspect of supervised learning and has been integrated in control systems [10], [11]. Classification falls into this category. On the other hand, unsupervised learning learns without any knowledge of the outcome. Clustering belongs to this category. It finds data with similar attributes and aggregates them in the same cluster [1].

B. Outer loop: control and cloud computing

1. Cloud computing

Another major technology that is susceptible to address stringent control issues is cloud computing. This recent technology provides a way to develop applications in a virtual environment where computing capacity, bandwidth, storage, security, and reliability are not issues because users do not need to install costly software on their own system. In a virtual computing environment, users can develop, deploy, and manage applications, paying only for the time and capacity used while scaling up or down to accommodate changing needs or external requirements.

Cloud computing has the main characteristics of distributed applications, pervasiveness, and service-oriented architecture (SOA),(http://www.cloudcomputingdefined.com/learn/). Ordinary consumers have now migrated to the use of SOA and Web 2.0 applications such as social networking sites, blogs, hosting services, among others. One can choose services from pool of available services and negotiate price through service level agreements (SLAs). Among the popular cloud service providers are: Amazon™ elastic compute cloud (EC2), (http://www.amazon.com/ec2/), Google™ App Engine or GAE, (http://appengine.google.com/) and Microsoft™ Live Mesh, (http://www.mesh.com/), among others.

2. Open Networking Foundation (ONF)

On 21st of March 2011, six companies that own and operate some of the largest networks in the world announced the formation of the Open Networking

Foundation (ONF), a nonprofit organization dedicated to promoting a new cloud-focused initiative approach to networking called Software-Defined Networking (SDN). ONF, through SDN, allows owners and operators of networks to control and manage their networks to best serve their needs. ONF's first priority is to develop and use the so-called OpenFlow protocol [http://www.openflow.org/], [20]. The main functionalities of OpenFlow is to seek to increase network functionality while lowering the cost associated with operating networks through simplified hardware and network management [http://www.opennetworkingfoundation.org/].

3. From Web 1.0 to Web 2.0

There was a natural transition to Web 2.0 after the so-called dot-com crash of Web 1.0 in late 2001. The main Web 2.0 technologies include wikis, blogs, RSS filters, folksonomies, mashups, podcasts, crowdsourcing, social networks, and virtual worlds [2]. A brief comparison is done between Web 1.0 and Web 2.0 is given in Table 1 below [http://oreilly.com/web2/archive/what-is-web-20.html], [18].

TABLE 1 - WEB 1.0 VS. WEB 2.0

Web 1.0	**Web 2.0**
Britannica Online	Wikipedia
personal websites	blogging
domain name speculation	search engine optimization
page views	cost per click
screen scraping	Web services
publishing	participation
content management systems	wikis
directories (taxonomy)	tagging ("folksonomy")
stickiness	syndication

4. Semantic Web as a component of Web 3.0

As a step further, Web 3.0 involves the integration of smart phones and other distributed devices. One of the main components of Web 3.0 is the semantic Web. Under the auspices of the World Wide Web Consortium (W3C), the semantic Web main aim is to provide information in a form that can readily be interpreted by machines. As a result, machines can perform better to reduce the tedious work involved in finding, combining, and acting upon information on the Web. As originally envisioned, the semantic Web is a system that enables machines to give a "meaning" to information and respond to complex human requests based on their objectives, beyond the mere concatenation of keywords. Such an "understanding" requires that the relevant information sources are semantically structured – which represents an issue in its own right. While attributing "meaning" to the symbols it manipulates for ready-made use by humans, the semantic Web represents an important evolution from a Web that consists largely of documents for humans to read through one that includes data and information for computers to manipulate. The Semantic Web derives information/knowledge from data with the help of a semantic theory for interpreting the symbols. The semantic theory provides an account of "meaning" in which the logical connection of terms establishes interoperability between systems [24]. In our KBLCS context, semantic Web is a component that is mostly needed in decision support systems (DSSs) used in human-machine interaction such as in managerial settings, for instance.

5. Need for data integration/data fusion

Data integration might be viewed as a combination of datasets wherein the larger dataset is retained, whereas in data fusion there is a reduction in the resulting dataset with improved confidence. In areas such as business intelligence, for example, data integration is used to describe the combining of data, whereas data fusion is integration followed by reduction or replacement.

(i) Importance of data integration

The need continuously increases for shared semantics and a web of data and information derived from it. Mainly for the purpose of modeling and simulation of dynamic systems behavior, environmental science has been looking for the integration of data from hydrology, climatology, ecology, and oceanography (see http://marinemetadata.org/mmiswinfo/). The need to understand systems across ranges of scale and distribution is evident everywhere in science and presents a pressing requirement for data and information integration [14]. In managerial control systems, various e-government initiatives represent similar efforts, worldwide.

(ii) Universal Data Element Framework (UDEF)

Well-indexed data play a crucial role in the control processes in managerial settings. Indexing data in an enterprise is indeed a major issue. Using the so-called Universal Data Element Framework (UDEF) for indexing enterprise data is similar to using the Dewey decimal system for indexing books in a library. As a result, the UDEF is supposed to enable decision makers to discover data and systems to integrate data across multiple applications and enterprises. Once the enterprise data has been indexed with the UDEF, the data is enabled for simpler interoperability with any other data that has been indexed with the UDEF.

The time and effort to integrate data between any two applications indexed with the UDEF is reduced inducing lesser enterprise costs [26].

6. Data fusion

(i) Data fusion technique

Data fusion is a set of mathematical and computational techniques that combine data from multiple and sometimes heterogeneous sources for the purpose of achieving inferences. The objective is to obtain more efficient and adapted inferences than if they were achieved by means of the separate sources used before fusion. The fused dataset is different from a simple combined superset in that the points in the fused data set contain attributes and metadata which might not have been included for these points in the original dataset. The result of fusion is expected to encompass a qualitatively different knowledge always referred to a context. At the lowest level, data fusion combines several sources of raw data to produce new raw data. The quality of fusion is related to the quality of the original datasets. In any case, the expectation is that fused data is more informative and synthetic than the original datasets. Two levels of integration are usually considered for this purpose. In command/control operations used in military and national defense settings, these levels map the following activities:

- Level 1 data fusion discusses techniques like essential target tracking methods. Level 2 considers fusion methods for applications such as target classification and identification, unit aggregation and ambush detection, threat assessment, and relationships among entities and events, and assessing their suitability and capabilities in each case.
- Level 1/2 interactions emphasizes particle filtering techniques as unifying methods for both filtering under Level 1 fusion and inferencing in models for Level 2 fusion. The various available techniques are temporal modeling techniques including dynamic Bayesian networks and hidden Markov models (HMMs), distributed fusion for emerging network centric warfare environments, and the adaptation of fusion processes *via* machine learning techniques [6].

(ii) Data fusion in distributed data base

In addition to its use in national defense, data fusion is widely used in major industry production, medical treatment, among others. Particularly in the aspect command/control for the purpose of national defense, data fusion played a pivotal role in the Cooperative Engagement Capability (CEC) - a system which promises to transform naval surface warfare to a major extent. It combined and associated the data from a number of sensors, and then estimated the current situation, localized the threat target accurately, evaluated the fusion result. In a complete data fusion system, the background database is vital for the support of each function module. It not only needs to meet security, integrity and consistency requirements of the data like any traditional database, but also needs to meet real-timing of the data in the particular system environment [15].

IV. Overview of the Architecture

A. Knowledge engineering issues

Knowledge engineering is a field concerned with studying and developing technological tools for the processing of knowledge issued from the available information as further refinement of data. Knowledge engineering developed a set of technologies such as expert systems, artificial neural networks (ANNs), case-based reasoning (CBR), genetic algorithms (GAs), and intelligent agents, in addition to other hybrid methods such as neuro-genetic, adaptive neuro-fuzzy inference systems (ANFISs) [13].

B. Raw specifications as a road map

The basic template for using the architecture is designed so as to encompass different classes of approaches such as expert systems (ESs), case-based reasoning (CBR), artificial neural networks (ANNs), evolutionary computation including genetic algorithms (GAs), and intelligent agents (IAs). Each approach is used according to the situation. The basic methodology for raw problem specifications is depicted below.

Methodology 1 - Raw specifications

```
 Choose one of the following accordingly
Case 1
   If IF-THEN rules are available from
   human experts or from other well-
   established sources, then use expert
   systems (ESs)
     1.1 Default use crisp approach.
     1.2 If fuzzy knowledge is available
        then use it.
Case 2
   If similar cases have been
   encountered and available from human
   experts or from other well-
   established sources, then use case-
   based reasoning (CBR) instead of IF-
   THEN rules
Case 3
   If patterns are encountered (such as
   shapes, sounds, videos…) then use
   neural networks (ANNs)
     3.1 If target is known then use
        supervised methods
     3.2 Otherwise use unsupervised
        methods
     3.3 As intermediary use semi-
        supervised methods
Case 4
   If there is an availability of
   infinitely many solutions then use
   evolutionary methods
     4.1 Genetic algorithms (GAs)
     4.2 Evolutionary strategies
     4.3 Genetic programming
Case 5
   If there is an availability of data
   in different form then use hybrid
   methods
     5.1 Neuro-genetic
     5.2 Neuro-fuzzy
     5.3 Adaptive neuro-fuzzy inference
          system (ANFIS)
Case 6
   If in unfamiliar environments, use
   intelligent agents (IAs) among the
   following hierarchy
     6.1 Simple reflex agents
     6.2 Model-based agents
     6.3 Goal-based agents
```

```
    6.4 Utility-based agents
    6.5 Learning agents
```

V. Architecture Components

From Methodology 1 above, we make a mapping onto the six different cases available.

A. Expert systems (ESs)

1. The expert systems approach

Expert systems, or rule-based systems, represent a computational methodology for modeling higher order cognitive functions of the brain such as reasoning inductively and deductively. They are usually used to mimic the decision-making process of human experts and are used in planning, scheduling and diagnostics systems, among others. Although expert systems contain algorithms, many of those algorithms tend to be static, *i.e.* they do not adapt to evolving situations over time. As a result, they are to be separated from learning systems since they do not learn from experience, analogy or through a teacher. Although, expert systems are usually confused with knowledge-based systems (KBSs), they are, in fact, simply one category of KBS [13].

2. Expert systems methodology

Methodology 2
Case 1 Expert systems methodology

```
Step 1 Peripheral input activities
   1.1 Knowledge acquisition
   1.2 Knowledge validation

Step 2 Core activities
  2.1 Knowledge representation
   Rules, semantic nets, frames
  2.2 Inference
   Default use crisp inference
   If vague rules use fuzzy inference
   Chaining: Forward, backward, hybrid

Step 3 Peripheral output activities
Explanation and justification
```

B. Case-based reasoning (CBR) method

1. CBR approach

Cognitively speaking, case-based reasoning (CBR) method works in a similar way as humans when selecting a course of action from previous similar experience. As a result, CBR as a methodology is used for solving problems based on past experiences called source cases to solve new problems known as target cases. CBR systems model the human capability to reason via analogy. Standard applications include legal cases, where the knowledge of the law is not just contained in written documents, but in the practical way this knowledge has been really applied by the courts in real-life situations [22].

2. CBR methodology

The main steps of CBR methodology are described in the box below.

Methodology 3
Case 2 CBR methodology

Step 1 *Elaboration phase*
   Build the specification of the problem.

Step 2 *Retrieval phase*
   Find the previous cases considered most similar to the target.

Step 3 *Reuse phase*
   Build a solution to the problem based on source cases identified in the retrieval phase.

Step 4 *Revision phase*
   Correct solution in case of unsatisfactory results.

Step 5 *Learning phase*
   Use the experience just completed for improvement of subsequent solutions.

C. Artificial neural networks (ANNs)

1. ANNs approach

Artificial neural networks (ANNs), on the other hand, model the brain at a biological level. As the brain easily manages pattern recognition tasks, such as vision and speech, so do neural network systems – in a mimic way. The main standard characteristics of ANNs, as information-processing systems, are complexity, parallel processing and nonlinearity. They have been successfully used to solve problems of classification, prediction, and clustering. They can be trained to learn to read, to recognize patterns from experience and to predict future trends, *e.g.* in the demand for electricity, weather forecasting, financial trends, among others. Their popularity is mainly based on their versatility and abilities to manage both continuous and discrete data, with the production of acceptable results for complex problems [12].

2. ANNs methodology

The main steps of ANNs methodology are described in the box below.

Methodology 4
Case 3 ANNs methodology

Case 3.1 *Supervised learning - MLP*
1. Initialization
2. Activation
3. Weight training
4. Iteration

Case 3.2 *Supervised learning Recurrent network*
1. Storage
2. Testing
3. Retrieval

Case 3.3 *Supervised learning Bidirectional associative memory (BAM)*
Same steps as in 3.2 above

Case 3.4 *Unsupervised learning Self-organizing NNs*
Same steps as in 3.1 above

Case 3.1 *Unsupervised learning Competitive learning*
1. Initialization
2. Activation and similarity matching
3. Weight training
4. Iteration

D. Genetic algorithms (GAs)

1. Genetic algorithms (GAs) approach

Genetic algorithms (GAs) represent a method of evolving solutions to complex problems. They belong to the larger class of evolutionary algorithms (EAs), which generate solutions to optimization problems relying on characteristics inspired by natural evolution, such as inheritance, mutation, selection, and crossover. The term 'genetic' refers to the biologically-metaphoric behavior of algorithms. In this situation, the behavior is similar to biological processes involved in evolution and based on the assumptions that good parents produce good children with a high probability, leading at the end of the process to the improvement of possible solutions over time. However, although genetic algorithms are considered as optimizers they do not guarantee optimality. Usually, the algorithm terminates when either a predetermined maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. For example, such a method is used to find one of many possible acceptable solutions to the problem of scheduling examinations (rooms, students, invigilators and possibly even equipment) from the millions of possible solutions. In many cases, manual adjustments are usually required for improvement [8].

2. Genetic algorithms (GAs) methodology

The main steps of standard genetic algorithms are described in the box below.

Methodology 5
Case 4 Genetic algorithms methodology

Step 1 *Representation*
Represent the problem variable domain
1.1 choose a chromosome of a fixed length,
1.2 choose the size of a population
1.3 choose crossover and mutation probabilities.

Step 2 *Fitness function definition*
Define a fitness function as a performance measure of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.

Step 3 Initial population
Randomly generate an initial population of size N of chromosomes as a sequence $x_1, x_2, \ldots, x_N$

Step 4 *Fitness function calculation*
Calculate the fitness of each chromosome $f(x_1), \ldots, f(x_N)$

Step 5 Selection
Select a pair of chromosomes for mating from current population. Chromosomes are selected with a probability related to their fitness.

Step 6 *New population*
6.1 Create a pair of offspring chromosomes by applying the genetic operators of crossover with chosen probability p_c
6.2 Apply mutation operator on offspring with probability p_m.
6.3 Place the created offspring chromosomes in the new population.
6.4 Repeat *Step 5* until the size of new chromosome population becomes equal to the size of the initial population, *N*.

Step 7 Replace the initial (parent) chromosome population with the new (offspring) population.

Step 8 Go to *Step 4*, and repeat the process until the termination criterion is satisfied.

E. Hybrid methods

1. Hybrid approach

The methodologies above might not be convenient for solving some complex problems. For instance, if the available data is made of signals and patterns with the availability of fuzzy IF-THEN rules, it is highly recommended to use a neuro-fuzzy methodology. If instead of IF-THEN fuzzy rules, infinitely many solutions characterize the problem then use neuro-genetic methodology [17]. Other combinations are possible as described in Methodology 6 below.

2. Hybrid methodology

The main steps for hybrid methods choice are described in the box below.

.

```
                 Methodology 6
          Case 5 Hybrid methodology

1. Data management: use one method of
   data management from below
   1.1 Data with minor preprocessing
   1.2 Use data integration
   1.3 Use data fusion
2. Learning
   Learn from experimental data
   (examples, samples, measurements,
   records, patterns, observations…) by
   ANNs
3. Embedding
   Embed existing structural human
   knowledge such as experience,
   expertise, and heuristics, rules of
   thumb into IF-THEN crisp or fuzzy
   rules.
4. Choose one hybrid method
   4.1 Neural expert systems if crisp
       IF-THEN rules are possible
   4.2 Neuro-fuzzy systems if fuzzy IF-
       THEN rules are possible (Mamdani
       inference method)
```

F. Intelligent agents

1. Intelligent agents approach

An intelligent agent is a software program whose goal or overall task is specified but where the software can autonomously take some decisions. Agents often have the ability to learn and make increasingly complex decisions on behalf of their users. We concentrate on two major types of intelligent agents [23].

- The simple problem solving agent (SPSA) which derives an action from a percept using search techniques in deterministic cases.
- The decision-theoretic agent which derives an action from a percept in uncertain situations *i.e.* when the agent's knowledge of the current state is not fully-known. The basic idea in decision theory is the so-called maximum expected utility (MEU) which states that "an agent is rational if and only if it chooses an action that yields the highest expected utility, averaged over all the possible outcomes of the action" (see [23] page 466).

2. Intelligent agents methodology

The main steps of intelligent methodology are described in the box below.

```
                 Methodology 7
     Case 6 Intelligent agents methodology

Case 6.1 SPSA
Simple problem solving agent (SPSA)
   returns an action from an external
   percept
inputs: a percept
static: sequence an action sequence,
           initially empty
        state: some description of the
           current world state
        goal: a goal, initially null
   problem, a problem formulation
UPDATE_STATE from previous state and
   percept given as input
if sequence is empty then
   FORMULATE_GOAL from state
   FORMULATE_PROBLEM from state and goal
   SEARCH for sequence to solve problem
   RETURN action as the first element in
      sequence

Case 6.2 Decision-theoretic agent (DTA)
inputs: a percept
static: belief_state, probabilistic
   beliefs about current state
   action, the agent's action
UPDATE belief_state based on action and
   percept
CALCULATE outcome probabilities for
   actions, given action descriptions
   and current belief_state
SELECT action with highest expected
   utility, probabilities of outcomes
   and utility information
RETURN action
```

VI. Interactions of Proposed Architecture

We believe that the study and integration of previously-described methodologies will advance our knowledge of control systems in general and of knowledge-based learning control systems, in particular. The processes described are based on the most powerful theoretical and advanced technological tools available to computer control and machine learning scientists, entailing a better understanding of the issue of learning control systems. The impacts on many fields of research are expected to be important, not only on computer science and control systems as such but also on automated managerial processes at large. We expect our architecture will have a strong interaction with the following fields of research and technology.

(i). Data engineering: the interaction for more structured organization of data is needed for efficient response to queries and decision making. For instance, to further integrate ways to choose between preprocessing, data integration and data fusion.

(ii). Human machine integration: to provide distributed control interfaces that will assist the user/decision maker in framing and executing complex queries spanning many levels in the enterprise, from controllers at the production level to administrative procedures at the managerial level.

VII. CONCLUSION

We have shown how to proactively increase the collaboration between different technologies in order to address the issue of knowledge base learning control systems (KBLCS). On top of the multitude of methodologies and tools that are prone to be integrated, it remains highly expected that the technologies reported here will uncover even more useful, but so far hidden, structures in KBLCS processes. In addition to various algorithmic and numerical and symbolic/numeric methods now available, however intelligent these might be, future KBLCS have to include simulation scenarios capable of producing highly anthropomorphic behavior for the resulting intelligent machines, including the discovery and/or generation of common-sense knowledge and hidden patterns. Although the proposed architecture gave a blueprint of solution to an abstract control problem, it remains that further projection and integration of the described methodologies onto specific application-oriented contexts will indeed constitute a challenging task for years to come.

REFERENCES

[1] Alpaydin, E,, *"Introduction to Machine Leaning"*, MIT Press, 2004.
[2] Andriole, J.S., "Business Impact of Web 2.0 Technologies", *CACM*, 58(2):67-79, December 2010
[3] Åström, K.J., "Towards intelligent control", *IEEE Control Systems Magazine*, Vol. 9, no.3, pp. 60-64, ISSN: 0272-1708, 1989
[4] Åström, K.J. & T.J., McAvoy, "Intelligent control", *J. Process Control,* Vol. **2**, no. 3, pp. 115-127, ISSN: 0959-1524, 1992
[5] Cellier, F.E., L.C., Schooley, M.K., Sundareshan, & B.P., Ziegler, "Computer aided design of intelligent controllers: challenges of the nineties", In: M. Jamshidi & C.J Herget (Eds.), "*Recent Advances in Computer-Aided Control Systems Engineering",* Elsevier 1992
[6] Das, S., *"High-Level Data Fusion"*, Artech House, Inc. 2008
[7] Fu, K.S., "Learning control systems and intelligent control systems: an intersection of artificial intelligence and automatic control", *IEEE Trans. Autom. Cont.,* 16:70-72, 1971
[8] Goldberg, D.E., "*The Design of Innovation: Lessons from and for Competent Genetic Algorithms",* Addison-Wesley, 2002
[9] Hamano, F., "Intelligent control", *Expert Systems in Robotics, NATO ASI Series F,* Vol. **71**; pp. 737-740, 1991
[10] Hamdi-Cherif, A., "Towards robotic manipulator grammatical control", Invited Book Chapter In: Suraiya Jabin (Ed.) *"Robot Learning"*, SCIYO Pub., pp. 117-136, October 2010
[11] Hamdi-Cherif, A. & C., Kara-Mohammed (*alias* Hamdi-Cherif), "Grammatical inference methodology for control systems", *WSEAS Transactions on Computers,* 8(4):610-619, 2009
[12] Haykin, S., "*Neural Networks – A Comprehensive Foundation"*, 2nd Edition, Prentice Hall, 1999
[13] Kendal, S.L. & Creen, M., "*An Introduction to Knowledge Engineering"*, Springer, 2007
[14] Lenzerini, M. "Data Integration: A Theoretical Perspective", *PODS*:233-246, 2002
[15] Mansouri, D. & A. Hamdi-Cherif, "Ontology-oriented case-based reasoning (CBR) approach for trainings adaptive delivery", Invited Paper in *Proc. of the 15th WSEAS Int. Conf. on Computers (CSCC'11)*, Corfu Island, Greece, pp. 328-333, July 15-17, 2011
[16] Meystel, A., "Multiscale models and controllers", *Proceedings of the IEEE / IFAC Joint Symposium on Computer-Aided Control Systems Design* (CACSD'94), Tucson, Arizona, pp. 13-26, 7-9 March, 1994
[17] Negnevitsky, M. *"Artificial Intelligence – A Guide to Intelligent Systems, Chapter 8 Hybrid Intelligent Systems"*, 2nd Edition, Addison Wesley, 2005
[18] O'Reilly, T., "What is Web 2.0? - Design patterns and business models for the next generation of software", http://oreilly.com/web2/archive/what-is-web-20.html, 09/30/2005, retrieved March 2012
[19] Park, C., S. Park, "The freeze algorithms for concurrency control in secure real-time database systems", *Data & Knowledge Engineering,* 45(1):101-125, 2003
[20] Pfaff, B. *et al.*, "OpenFlow Switch Specification Version 1.1.0 Implemented (Wire Protocol 0x02)", February 28, 2011
[21] Rao, M., *"Integrated System for Intelligent Control". LNCIS*, Vol. 167, Springer-Verlag, New York, USA, 1992
[22] Richter, M. & A. Agnar, "Case-based reasoning foundations", *The Knowledge Engineering Review*, Cambridge University Press, 20:(3):203–207, 2006
[23] Russel, S., & P. Norvig, *"Artificial Intelligence – A Modern Approach"*, Prentice Hall in Artificial Intelligence, 2nd Edition, pp. 51-54, 2003.
[24] Shadbolt, N., W. Hall, T. Berners-Lee "The semantic Web revisited", *IEEE Intelligent Systems*, pp.96-101, May-June 2006
[25] Saridis, G.N., "Knowledge implementation: structures of intelligent control systems", *IEEE International Symposium on Intelligent Control*, Philadelphia, Pennsylvania, pp. 9-19, 19-20 January 1987
[26] Shuldt, R.L., L.R. Shauger, *"UDEF - Six Steps to Cost Effective Data Integration",* Amazon, 2012
[27] Zadeh, L., "Knowledge evaluation: observation and computation for intelligent control", *IEEE International Symposium on Intelligent Control*, pp. 302-308, 19-20 January 1987
[28] Zilouchian A., M. Jamshidi (Eds.). *"Intelligent Control Systems Using Soft Computing Methodologies"*, CRC Press, 2001
[29] Zuge, C.T.Y., S. L. Pereira & Eduardo Mario Dias **"**Integration of Information Technology and Automation: Facilitators and Barriers", *WSEAS Transactions On Systems and Control*, 5(5):372-381, 2010

Websites accessed as of March 2012.

Examples of KBS
CLIPS: http://clipsrules.sourceforge.net/
Example of DSS: http://www.goaefis.com/about-aefis/faqs/

DBMS/Data Mining Tools/ Machine Learning Software
http://www.microsoft.com/sqlserver/en/us/
http://www.oracle.com/technetwork/database/options/odm/index.html
CART: http://salford-systems.com/cart.php
See5/C5.0: http://www.rulequest.com/see5-info.html
Weka: http://www.cs.waikato.ac.nz/ml/weka/
Tanagra:
http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html

Cloud computing
(http://www.cloudcomputingdefined.com/learn/)
Examples of Cloud Computing Solutions
http://www.ibm.com/developerworks/cloud/ec2.html
http://www.opennetworkingfoundation.org/
http://www.openflow.org/
http://www.microsoft.com/sqlserver/en/us/editions/azure.aspx

Aboubekeur Hamdi-Cherif was born in Setif, Algeria. He received BSc and MSc in Electrical Engineering, both from Salford University, Manchester, England, in 1976 and 1978, respectively, and PhD degree in Computer Science (Artificial Intelligence/Robotics) from Université Pierre et Marie Curie Paris 6, Paris, France, in 1995.

He worked as Control Engineer with Algerian Petroleum Company SONATRACH. He taught at Ecole Supérieure des Transmissions, Algiers, Algeria, at Université de Bretagne Occidentale, Brittany, France, at Ecole Supérieure Libre des Sciences Commerciales Appliquées (ESLSCA), Paris, France, and at Ferhat Abbas University, Setif, (UFAS), Algeria.

In 2001, he joined Computer College, Qassim University, Saudi Arabia, where he is Associate Professor and Deputy Head of Computer Science Department. Concentrating on teaching, he supervised about 95 BSc/Engineer student projects, and 10 master and doctoral students. He is currently interested in machine learning as applied to data mining and to bioinformatics, intelligent control and grammatical inference, and Arabic language processing. In 1998, he was awarded the best technological thesis prize by UFAS. Dr. Aboubekeur received financial support for his research projects from Université Pierre et Marie Curie Paris 6, Paris, France, from UFAS, from Qassim University, and from King Abdulaziz City for Science and Technology (KACST), Riyadh. He can be reached at shrief@qu.edu.sa, elhamdi62@gmail.com

Dr. Aboubekeur is member of IEEE and ACM.