SRI LANKA ASSOCIATION FOR ARTIFICIAL INTELLIGENCE

Proceedings of the 5th Annual Sessions

31st October 2008

Sri Lanka Association for Artificial Intelligence – 2008

Proceedings of the 5th Annual Sessions
31st October 2008
Moratuwa
Sri Lanka

Editors:

Prof. Asoka S. Karunananda
Dr. Nalin Wickramarachchi
Dr. (Mrs) Uditha Ratnayake

Sri Lanka Association for Artificial Intelligence, Department of Mathematics & Computer Science, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka

Contents

Keynote Address

Philosophical Grounding and Computational Formalization for Practice Based Engineering Knowledge

Paper published in Knowledge Based Systems, Vol. 20, Issue 4, May 2007, pp. 382-387

W.P.S. Dias
Department of Civil Engineering
University of Moratuwa, Sri Lanka

Abstract: *This paper describes aspects of Michael Polanyi's epistemology and Martin Heidegger's ontology that provide a strong rationale for the notion of practice based knowledge. Tacit knowing (Polanyi) and pre-theoretical shared practice (Heidegger) are two such philosophical concepts explored. The paper then goes on to classify practice based knowledge as being either historical and structured or horizontal and unstructured. It shows finally that Artificial Intelligence (AI) approaches such as Artificial Neural Networks (ANN), Case Based Reasoning (CBR) and Grounded Theory (with Interval Probability Theory) are able to model philosophical concepts related to practice based knowledge. The AI techniques appropriate for modeling Polanyi's and Heidegger's ideas should be founded more on a connectionist rather than a cognitivist paradigm. Examples from engineering practice are used to demonstrate how the above techniques can capture, structure and make available such knowledge to practitioners.*

1. Background and Objectives

Theoretical knowledge has been prized in academic institutions at least since the scientific revolution. The philosophical underpinnings for this, in the form of privileging the intellectual over the practical, have come from Descartes, but deeper roots lie in Plato himself. In engineering, this focusing on theoretical knowledge has caused a gap between academic training and professional practice, as the latter often calls for practitioner judgement and experience (Dias and Blockley 1995). At the same time, many fields of engineering have craft based origins (Dias 2002), and this has given rise to a rich vein of heuristics or practice based knowledge.

Practice based knowledge has not acquired the same "respectability" as theoretical knowledge in academic institutions. Similarly, theoretically trained engineers who spend their engineering careers doing routine tasks based on heuristic rules could question the value of their training and also have self doubt about their role – e.g. as to how it is different to that of a craftsman. One reason for the above is that there are no formalizing principles for practice based knowledge, such as provided by the scientific method and mathematics for theoretical knowledge. It has been proposed that soft systems methods and artificial intelligence (AI) techniques can provide formalizations for practice based knowledge at the conceptual and technical levels respectively (Dias 2002). In addition to such formalizations however, some philosophical arguments are required to counter the stranglehold that Plato and Descartes wield over our intellectual milieu.

John Dewey's thought is very pertinent in the theory vs. practice debate. He argued both that theory was a kind of practice, and that theory arose from practice (Dewey 1976, Dewey 1981). This importance of a broad philosophy of practice is being actively developed (e.g. Skill 1995), with contributions from philosophers, engineers, craftsmen and actors; parallels have been drawn between actors and engineers. The attempt is to show that knowledge is very often acquired from practice (perhaps under apprenticeship), rather than from theory alone.

This paper however focuses on the philosophers Michael Polanyi and Martin Heidegger. It demonstrates that Polanyi's epistemology and Heidegger's ontology have the potential for placing practice based knowledge on a sound intellectual footing. It then explores two different categorizations of practice based knowledge. Finally, it gives some examples of problems that are not amenable to theoretical knowledge; and demonstrates how AI techniques can be used to capture and process the practice based knowledge related to those problems.

2. Michael Polanyi – Tacit Knowing

One of Polanyi's main contributions to epistemology was the idea of tacit knowing; one of his books is titled "The Tacit Dimension" (Polanyi 1966). A key aspect of tacit knowing was that it attended from *particulars* to a *whole*. Polanyi used the example of recognizing a face to illustrate this - we use our *subsidiary* awareness of the features in order to achieve *focal* awareness of the face (Polanyi 1966). The important thing was that the particulars should not be focused on, but "seen through", like a pair of spectacles. To focus on the spectacles would mean that we can not use them to see anything else (Prosch 1986). Similarly, directing our attention at the isolated features of a face would destroy the act of recognition. This focal recognition of wholes had similarities to Gestalt-type awareness (Polanyi 1958), where the whole "falls into place", when deliberate attention is not paid to the particulars.

Such subsidiary awareness of particulars meant that they could not be fully specified (Polanyi 1966). This is why Polanyi said that "we know more than we can tell" (Polanyi 1966). The "from-to" apprehension of knowledge could not be made explicit either; in other words, the path from particulars to whole is not reversible (Polanyi 1958). This undermines to an extent the "strong Artificial Intelligence" programme of trying to represent the world through cognitive modeling. It is this *unspecifiability* of particulars (Polanyi 1966, p. 18) and *irreversibility* of knowing that constitutes the tacit dimension:

> Scrutinize closely the particulars of a comprehensive entity and their meaning is effaced, our conception of the entity is destroyed. Such cases are well known. Repeat a word several times, attending carefully to the motion of your tongue and lips, and to the sound you make, and soon the word will sound hollow and eventually lose its meaning. By concentrating attention on his fingers, a pianist can temporarily paralyze his movement. We can make ourselves lose sight of a pattern of physiognomy by examining its several parts under sufficient magnification.

Polanyi also argued that knowledge involves skill. It means that knowing is an *active* process, requiring intelligent *effort*, as opposed to passive perception of phenomena (Polanyi 1969). It also means that there is a difference between knowing "what" and knowing "how" and that the former is embedded in the latter (Polanyi 1958). Thus, there is an indefinable component in our knowledge, which cannot be transmitted by propositions alone. This is particularly evident in technological knowledge, and more so in its craft based elements, where apprenticeship within a tradition is essential for the passing on of skills (Polanyi 1958).

Schon (1983) called this "knowing in action", but also called for "reflection in action" or reflective practice, which involved an intimate interaction with one's self, context and with others; his "reflection on action" was to be done after acting, and can be called "learning". He contrasted all of this practice based knowledge with technical rationality, which paid selective inattention to all aspects of problems that could not be theoretically formulated.

3. Martin Heidegger – Pre-theoretical Shared Practices

One of the main thrusts of Heidegger's philosophy is the primacy of practice, or rather practices that we are socialized into, prior to any theoretical understanding (Heidegger 1997). Heidegger approached the question of being from what he called "the human way of being". He did this because humans were the only beings who were concerned about their own being. He used the term *Da-sein* to denote this being. In addition to meaning "the human way of being", this hyphenated German word can also mean "being-there" and "everyday human existence". Heidegger also said that *Da-sein* was not a conscious subject, and that its way of being was "being-in-the-world"; in other words human beings always had the notion of a "world", which meant a "pre-theoretical" shared agreement in practices.

Also, subject-object distinctions were blurred in our everyday lives in the world. Dreyfus, one of Heidegger's best exponents, gives the example of a person turning a doorknob to enter a room. In this very everyday act, he argues, there is no conscious intention on the part of the person directed towards the doorknob, and hence no subject or object as such; rather, there is a seamless web of activity for the fulfillment of a purpose, in which both the person and the doorknob are participants (Dreyfus 1988). Heidegger insisted that the (analytical) isolation of *fundamental* properties of objects by detached subjects was a "way of being" that was derived from a more *primordial* way of being, where a seamless subject-object continuum achieved purpose through practical action.

Heidegger's ontology could serve as an intellectual platform for combating the feelings of inferiority and lack of status that many engineers worldwide experience in a culture (still heavily influenced by Plato and ancient Greece) that values analysis more than synthesis, and theoretical knowledge more than practical intelligence. Patrick Nuttgens, an architecture academic at the University of Edinburgh who became the founding director of Leeds Polytechnic in the U.K. in the early 1970s, argues that children first learn about the world by practice before they acquire a theoretical framework, and that technical education should reflect this (Nuttgens 1980).

Where scientific objectivity was concerned, Heidegger (1997) said that the isolated, so-called fundamental properties of things was a way of being derived from their being "ready-to-hand". In other words, all objective properties had some relation to purpose in everyday life; otherwise they would not have been "shown up". The flip side of this derivative way of being of science is that scientific representation can never capture the totality of nature (Heidegger 1977).

Where cognitive modeling was concerned, Heidegger criticized the symbolic representation of entities because it sought to "free" objective properties of things by stripping away their significance and then to reconstruct a meaningful whole by adding further meaningless elements (Heidegger 1997, p.82):

> The referential context that constitutes worldliness as significance can be formally understood in the sense of a system of relations. But we must realize that such formalizations level down the phenomena to the extent that the true phenomenal content gets lost, especially in the case of such "simple" relations as are contained in significance. These "relations" and "relata" of the in-order-to, for-the-sake-of, the with-what of relevance resist any kind of mathematical functionalization in accordance with their phenomenal content. Nor are they something thought, something first posited in "thinking", but rather relations in which heedful circumspection as such already dwells.

In other words, the holistic, context dependent way in which we encounter the world could not be represented. We would be trying to exchange a *presencing* of the world with a mere *re-presentation*. Our own skills of cognition too could not be captured by a predicate calculus. This would include our embodiment in physical bodies, although Heidegger did not speak much about this aspect, like Mearleau-Ponty does (Dreyfus 1988). At any rate, Heidegger's ontology turns Decartes' epistemological motto of "I think, therefore I am" around completely. To Heidegger, *being* preceded *thinking*; in other words, "I am, therefore I think", and this *sum* ("I am") too should read "I-am-in-the-world" (Heidegger 1997).

4. Categories of Practice Based Knowledge

Both Polanyi and Heidegger are good advocates for the importance, and indeed the primacy of practice based knowledge. How then can this knowledge be formalized and categorized? We have said before that AI can in fact provide a formalization for practice based knowledge. Within AI, Minski (1991) has distinguished between cognitivist and connectionist approaches to knowledge.
The cognitive approach is epitomized by expert systems (Building 1983). Here, the knowledge is made explicit, by eliciting it from an expert, generally in the form of production rules (i.e. if...then... relationships). Uncertainty can also be built in to the system. Once the knowledge base is thus prepared, facts concerning a new problem situation will trigger certain rules, and result in a diagnosis or decision. The triggering or firing of rules is governed by what is called the inference engine of the expert system. One of the important aspects of expert systems is that the rationale for arriving at the end result is also made available to the user.

On the other hand, the connectionist approach does not force the expert to codify his knowledge in the form of rules. All it requires is that he codifies his experience, in the form of case histories. The computer then discovers patterns that even the expert may have been unaware of; this is called the training phase. It can also predict the action that will be taken by the expert if given the parameters that define a new problem situation. In this approach, the knowledge is implicit and no explanations are given to the user (Coyne 1990). An Artificial Neural Network probably best epitomizes this approach.

Polanyi and Heidegger refer to the difficulties of "specifiability" and representation respectively. As such, they would seem to be less in favour of and indeed quite opposed to the cognitivist approach. On the other hand, the connectionist approach of pattern recognition resonates very strongly with Polanyi's tacit knowing, including its feature of being unable to give explanations to users (parallel to Polanyi's irreversibility).

Another categorization of practice based knowledge is that of the historical versus the horizontal. Discipline related information, comprising engineering science theories and codes of practice, can be referred to as *vertical* knowledge, and we shall not refer to it further. In addition, during a given design project, there will be *horizontal* knowledge that is generated by the design team (Konda et al 1992). This will include information regarding the process of design, examples of how design objects are decomposed and knowledge that is specific to the design project, often at the interfaces of disciplines (Reddy et al 1997). Apart from this, various service departments in the design organization will be gathering information from all projects. This could be called *historical* knowledge and is often quite structured in fashion. The distinction between horizontal and historical knowledge can be seen as mapping on to that possessed by generalists and experts respectively (Baird et al 2000).

It has been pointed out (Dias et al 2002) that different kinds of information are generated by product and service departments of organizations. The latter tend to produce generalizations based on historical data that is abstracted from the horizontal information generated by the product departments. There is probably a special need today to document horizontal information, because it is the most unstructured and hence the most difficult to capture. On the other hand, it constitutes information in the most primitive form, and such process information can be invaluable to other product teams if captured and made available.

Horizontal knowledge can also be seen as a collection of stories or narratives. It must be noted that the idea of individual stories (as opposed to "grand" overarching theory or doctrine) is a central tenet in both existentialist and postmodernist philosophy, of which Heidegger is a key figure. The focus in such philosophy is not on overall unifying theory, but rather on the features of particular events. This can also be called a "bottom up" approach to knowledge (as opposed to a "top down" one).

5. Modelling Tacit Knowledge: Construction Bid Decisions

Let us now look at some examples where AI techniques have been used to capture and process practice based knowledge. Consider the modelling of tacit knowledge. There are many areas in engineering that are characterized by such knowledge, none more clearly than bidding for construction projects, decisions for which have been described as being made "on the basis of intuition derived from a mixture of gut feelings, experience and guesses" (Ahmad 1990). The language is very reminiscent of Polanyi. There is also a wide acknowledgement of the poverty of theoretical approaches to this problem.

Hence, a backpropagation neural network called ANNBID was trained to make decisions on percentage mark-up for construction bids, based on the levels (i.e. numbers from 1 to 5) assigned to a set of 6 factors - i.e. nature of job, nature of the client, location of project, risk involved in investment, competition among contractors and current workload (Dias and Weerasinghe 1996); the identification of these key factors itself was based on an industry-wide survey. There were around 31 past cases from a single contractor, who assigned the above levels to all factors in each case; he also indicated the percentage mark-up used. The network was trained on 27 of these cases, each of which had 6 input values (corresponding to the 6 factors) and a single output (corresponding to the mark-up). The training consisted of generating a mapping between the inputs and the outputs for all 27 cases such that the error between the network-generated and declared outputs was below a specified target value for all 27 cases. The remaining 4 cases were used to test the predictions of the trained network, which were quite good. The contractor now had a neural network that "thought like he did" with respect to bid decisions; he could use the network in future bids to guide his guesses. Other researchers have also used such neural networks to model construction bid decisions (Hegazy & Moselhi 1994).

6. Modelling Shared Practice: Layout Design

The objective of this study was to explore the potential for using Artificial Neural Networks (ANNs) and Case Based Reasoning (CBR) for suggesting column spacing and sizing in multistory buildings, based on historical examples (Dias and Padukka 2005). Column spacing and sizing are part of preliminary design, and often based on "engineering judgment", which can be considered an aspect of shared practice. Data was obtained from a total of 45 existing buildings from different design offices; hence the data genuinely constituted shared practice, unlike in the tacit knowledge example, where it was a single contractor's "gut feelings" that were modeled by the neural network.

For the column spacing problem, the inputs were chosen as (i) type of building (residential/office); (ii) building height; (iii) type of foundation (pad/strip/raft/pile); (iv) type of slab (one-way/two-way); and (v) cost per unit area at Year 2000 prices. The output was the (minimum) column spacing. Training was carried out 34 of these cases for the ANN. These same 34 cases were used as the case base for CBR. Testing of the ANN was done using the remaining 11 cases; these same cases were used as the "new" cases in the CBR exercise. For the column sizing problem, the total number of cases was 29 (from among the above 45), with 21 being used for training and 8 for testing. The inputs were chosen as (i) building height; (ii) tributary area; and (iii) concrete grade. The output was the column size, i.e. area, at basement (or ground) level. Two criteria were used to establish the success of decision support tool, namely mean absolute error and the deviation from unity of the average ratio between predicted and desired outputs; these criteria were applied to the testing set.

In the column spacing exercise, the CBR results were better than the ANN ones on both criteria. After carrying out the ANN exercise, a sensitivity analysis was performed on the trained network, by evaluating the change in output when a given input is varied from its lowest value to its highest, all other inputs being held at their average values. This analysis revealed that building height and cost per unit area were the most significant inputs, with slab type being the next, and the others not being so significant. Hence, another CBR exercise (called weighted input CBR) was performed, with the most significant inputs (as per ANN sensitivity analysis) weighted by 3, the next significant by 2 and the others by unity. This resulted in the weighted input CBR results being even better than the original CBR ones. In the column sizing problem too the CBR results were slightly better than the ANN ones. At any rate, both ANNs and CBR were found to be good AI tools for modeling shared practice.

7. Modelling Horizontal Knowledge: Vulnerability of Buildings to Bomb Blast

The examples given above are based essentially on historical knowledge that has been structured into various fields (e.g. factors that affect bid mark-up on the one hand and grid spacing on the other). Such structuring could be construed as imposing a cognitive framework on the practice based knowledge and hence departing somewhat from the connectionist paradigm. The interactions between the fields however are genuinely connectionist, in that there are no cognitive rules that combine evidence in neural network type approaches.

Horizontal knowledge (or knowledge as narrative) is less structured, and we shall now consider an example. The application area is the vulnerability of buildings to bomb blast. To be sure, there are numerical methods of solving blast load problems. However, where overall vulnerability (inclusive of human injury and death) is concerned, the problem is a socio-technical one, and vulnerability depends on such factors as advance warning, level of security and the amount of glass used in buildings (Chandratilake & Dias 2004).

In order to tackle this problem, a hierarchical causal tree was constructed by perusing the case descriptions of 10 blast events, having variations in the type of structures that were targeted, the nature of explosions, the physical and social context and the intentions of terrorists. This was done using a Grounded Theory approach (Glaser & Strauss 1967), and 63 "phenomena" were extracted from the cases. Examples of such phenomena are "long and accurate warnings reduce human casualties"; "reinforced concrete framed construction can withstand considerable blast pressure" etc. By writing each phenomenon on a separate card, and by constant comparison among them, it was possible to cluster them appropriately and generate higher order "concepts" that emerge near the top of the causal tree – e.g. "physical entity", "spatial planning", "context" etc.

Another type of diagram that can be constructed out of event descriptions is what is called the event sequence diagram (Toft and Reynolds 1994). These depict sequential relationships between events that lead, for example to failure. Such diagrams have also been shown to be amenable to connectionist type AI approaches (Stone et al. 1989).

The above approaches are "faithful" to the experience based data, and hence much more holistic and realistic compared to theoretical solutions. The identification of phenomena and connections between them are done by the researchers, and this could introduce cognitive bias. Automation of such procedures, for example through co-word analysis (Monarch 2000) could be explored.

After construction of the fault tree, it can be used in a semi-quantitative way to estimate either a numerical interval between 0 and 1 or a linguistic label (e.g. low, moderate, high etc.) for the top level concept of vulnerability. This estimate is based on the linguistic labels (and associated levels of confidence in those labels) assigned by an assessor to the lowest level phenomena (Sanchez-Silva et al 1995, Dias and Chandratilake 2005). This approach uses interval probability theory (Cui & Blockley 1990), itself based on fuzzy set theory.

8. Discussion

At this stage we discuss an issue each from the philosophical and computational aspects in this paper and seek to further clarify their inter-relatedness. The first issue has to do with a comparison of Polanyi and Heidegger. On the one hand, they are poles apart. Heidegger is a very nihilistic philosopher who advocated a "hermeneutic of suspicion", while Polanyi sought to restore a fiduciary (or faith like) framework for the practice of science. Polanyi's focus is on epistemology, and hence he deals with the way that a human subject apprehends knowledge. Heidegger's focus is on ontology, and the notion of an individual human subject for him is a derivative (and even deficient) way of being, in a world that is characterized by shared practice and a network of relationships (both animate and inanimate).

Both however focus on practice and it is this commonality that has resulted in their being thrown together in this paper. We could say that practitioner involvement is important to Polanyi and context dependence to Heidegger. It is interesting that both these aspects are foundational to an engineering approach, which relies heavily on practice based knowledge (in addition to theoretical knowledge). On the other hand, science tries to strip away practitioner bias and context dependence in framing universal laws.

Another apparent difference between Polanyi and Heidegger can be seen in the directions they move in where wholes and parts are concerned. Where Polanyi is concerned, as stated earlier, knowing is associated in moving from parts to the whole. Knowledge of the whole is an emergent property. On the other hand, where Heidegger is concerned, the whole (or web of relations) is prior to any part thereof. A focus on any part is a derivative (or reductionist) move. For both however networks and connectedness are important. It is for this reason that AI approaches using a connectionist paradigm have been proposed for reflecting their ideas and for modeling practice based knowledge.

This brings us to the second issue for discussion, namely the appropriateness of AI for reflecting the philosophical ideas of Polanyi and Heidegger on the one hand, and for modeling practice based knowledge on the other. It should be noted that the goal of Artificial Intelligence (AI) in general is the solving of practical problems, based very often on practical experience or heuristics. This places AI firmly within a paradigm of practice. The actual AI techniques themselves may indeed be computational and algorithmic, such as an artificial neural network (ANN), which is clearly within the connectionist paradigm in AI. However, the inputs of an ANN are able to accommodate context related information and practitioner judgment (i.e. by converting qualitative information to a number scale). The outputs could be seen as mimicking practitioner involvement in some cases – e.g. the trained backpropagation network described in Section 5 behaves similarly to a human "gut reaction" in arriving at bid mark-up decisions. The outputs could also be seen as springing from the context of practice – e.g. the trained backpropagation network described in Section 6 (and also the case base) encapsulates the way designers in general choose column spacing and size.

It should also be noted that AI techniques within the cognitivist paradigm, such as rule based expert systems, while being useful for practice based knowledge, would not reflect too well the ideas of Polanyi and Heidegger. In fact, as shown earlier in the paper, both of them, either implicitly (Polanyi) or explicitly (Heidegger) rejected the validity of cognitive modeling. Hence, the juxtaposition of these philosophical ideas and computational formalizations has also underlined the conceptual differences between connectionist and cognitivist approaches within AI.

9. Conclusions

1. We have seen that the epistemology of Michael Polanyi and the ontology of Martin Heidegger provide a significant intellectual basis for notion of practice based knowledge.

2. We have demonstrated that Artificial Intelligence (AI) techniques such as Artificial Neural Networks (ANN) and Case Based Reasoning (CBR) can model philosophical concepts such as tacit knowing (Polanyi) and shared practice (Heidegger).
3. The juxtaposition of the above philosophical grounding and computational formalizations for practice based knowledge has highlighted the conceptual differences between connectionist and cognitivist approaches within AI, with the connectionist approach being seen to be more appropriate.
4. We have given examples of how both historical and horizontal practice based knowledge can be captured, structured and made available to practitioners using AI approaches such as ANN and CBR, and also Grounded Theory combined with Interval Probability Theory, itself based on fuzzy set theory.

10. References

Ahmad, I. (1990) Decision support system for modelling bid/no-bid decision problem, ASCE Journal of Construction Engineering and Management, Vol. 116, No. 4, pp. 595-608.

Baird, F., Moore, C.J. and Jagodzinski, A.P. (2000) An ethnographic study of engineering design teams at Rolls Royce Aerospace, Design Studies, Vol 21 No 4, pp. 333-355.

Building Expert Systems (1983) (eds.) Hayes-Roth, F., Waterman, D.A. and Lenat, D.B., Addison-Wesley, London.

Chandratilake, R. and Dias, W.P.S. (2004) Identifying vulnerability of buildings to blast events using Grounded Theory, Proceedings, 10th Annual Symposium on Research for Industry, Engineering Research Unit, University of Moratuwa.

Coyne, R.D. (1990) Design reasoning without explanations, AI Magazine, Vol. 11, No. 4, pp. 72-80.

Cui, W.C. and Blockley, D.I. (1990) Interval probability theory for evidential support, Int. Jnl. of Intelligent Systems, Vol. 5, No. 2, June, pp. 183-192.

Dewey, John (1976) The Middle Works 1899-1924, (ed.) J. A. Boydston, Southern Illinois University Press, Carbondale.

Dewey, John (1981) The Later Works 1925-1953, (ed.) J. A. Boydston, Southern Illinois University Press, Carbondale.

Dias, W.P.S. (2002) Reflective Practice, Artificial Intelligence and Engineering Design: Common trends and inter-relationships, Artificial Intelligence in Engineering Design, Analysis and Manufacture (AIEDAM), Vol. 16, pp. 261-271.

Dias, W.P.S. and Blockley, D.I. (1995) Reflective Practice in Engineering Design, ICE Proceedings on Civil Engineering, Vol. 108, Issue 4, November, pp. 160-168.

Dias, W.P.S. and Chandratilake, S.R. (2005) Assessing vulnerability of buildings to blast using Interval Probability Theory, in preparation.

Dias, W.P.S. and Padukka, U.A. (2005) AI Techniques for Preliminary Design Decisions on Column Spacing and Sizing

Dias, W.P.S. and Weerasinghe, R.L.D. (1996) Artificial neural networks for construction bid decisions. Civil Engineering Systems, Vol. 13, pp. 239-253.

Dias, W.P.S., Subrahmanian, E. and Monarch, I.A. (2002) Dimensions of order in engineering design organizations, Design Studies, Vol. 24, pp. 357-373.

Dreyfus, H.L. (1988) Husserl, Heidegger and Modern Existentialism, in (ed.) B. Magee, Great Philosophers: An Introduction to Western Philosophy, Oxford University Press, Oxford, pp. 252-277.

Glaser, B. and Strauss, A.L. (1967) The Discovery of Grounded Theory: Strategies for Qualitative Research, Weidenfeld and Nicolson, London.

Hegazy, T. and Moselhi, O. (1994) Analogy-based solution to markup estimation problem, ASCE Journal of Computing in Civil Engineering, Vol. 8, No. 1, pp. 72-87.

Heidegger, M. (1977) The Question Concerning Technology and Other Essays, (tr.) W. Lovitt, Harper and Row, New York.

Heidegger, M. (1997) Being and Time, (tr.) J. Stambaugh, SUNY Press, Albany, N.Y.

Konda, S., Monarch, I., Sargent, P. and Subrahmanian, E. (1992) Shared memory in design: a unifying theme for research and practice, Research in Engineering Design, Vol.4, No.1, pp.23-42.

Minsky, M. (1991) Logical vs. analogical or symbolic vs. connectionist or neat vs. scruffy, AI Magazine, Vol. 12, No. 2, pp. 34-51.

Monarch, I.A. (2000) Information science and information systems: converging or diverging? in CAIS 2000, Dimensions of a Global Information Science, Proceedings of the 28th Annual Conference, Canadian Association for Information Science.

Nuttgens, P. (1980) What should we Teach and How should we Teach it?: Aims and Purpose of Higher Learning, Gower Publishing Company, London.

Polanyi, M. (1958) Personal knowledge: Towards a Post-critical Philosophy, University of Chicago Press, Chicago.

Polanyi, M. (1966) The Tacit Dimension, Doubleday & Co., Garden City, N.Y.

Polanyi, M. (1969) Knowing and Being, (ed.) Marjorie Greene, University of Chicago Press, Chicago.

Prosch, H. (1986) Michael Polanyi: A Critical Exposition, SUNY Press, Albany, N.Y.

Reddy, J.C., Finger, S., Konda S.L., Subrahmanian, E. (1997) Designing as building and re-using of artifact theories: understanding and support of design knowledge, Proceedings of the Workshop on Engineering Design Debate, University of Glasgow, Glasgow, Scotland .

Sanchez-Silva, M., Taylor, C.A. and Blockley, D.I. (1995) Evaluation of proneness to earthquake-induced failure of buildings in Buenaventura, Colombia, in (ed.) D. Key, Structures to Withstand Disaster, Institution of Civil Engineers, Thomas Telford, London, pp. 137-152.

Schon, D.A. (1983) The Reflective Practitioner: How Professionals Think in Action. Temple Smith, London.

Skill, Technology and Enlightenment: On Practical Philosophy (1995) (ed.) Bo Goranzon, Springer, London.

Stone, J.R., Blockley, D.I. and Pilsworth, B.W. (1989) Towards machine learning from case histories, Civil Engineering Systems, Vol. 6, pp. 129-135.

Toft, B. and Reynolds, S. (1994) Learning from Disasters, Butterworth-Heinemann, Oxford.

Invited Paper

A New Direction of Research into Artificial Intelligence

George Rzevski
Professor, Design and Complexity Science, The Open University, UK
Chairman, Rzevski Solutions Ltd, London, UK

Abstract: *A new direction of research into Artificial Intelligence is outlined in this paper based on fundamentals of Complexity Science. Intelligence is postulated to be an emergent property of complex networks because it emerges from the interaction of network components and is not traceable to any of these components. Researching artificial intelligence is best conducted by designing artificial complex systems and tuning them to exhibit emergent intelligence.*

Introduction

Research in Artificial Intelligence (AI) has a long tradition. The first paper attributed to the field was published by Warren McCulloch and Walter Pitts in 1943 [1], and the term "artificial intelligence" was proposed and agreed at the famous Dartmouth Workshop held in 1956. The thesis of this paper is that the direction of most of the past AI research was unnecessarily biased towards logic and mathematics. It did produce some interesting results but not intelligent machines, as promised. A new and far more promising research direction is outline below and described in some detail in [2].

What is Intelligence?

The notion of human *Intelligence* is very complex; it comprises the following (and possibly many other) capabilities:

- **Understanding the meaning** of symbols, words, text, data, images, utterances
- **Learning** (acquiring knowledge) from data, text, images as well as from own behaviour and behaviour of others and learning by discovery
- **Analysing** (deconstructing) complicated situations
- **Making choices** (decisions) under conditions of variety and uncertainty and therefore solving incompletely specified problems and achieving goals under conditions of the occurrence of frequent unpredictable events
- **Interacting** (communicating) with other actors in the environment, which include intelligent creatures and machines
- **Autonomously adapting** to changes in the environment
- **Creating** (constructing) new concepts, principles, theories, methods, artefacts, models, literature, art
- **Setting and achieving goals** by competing and/or cooperating with others

An important part of human intelligence is to strive to create **Artificial Intelligence**.

Artificial Intelligence

"Artificial" means man-made rather than natural. Artificial Intelligence is supposed to be man-made intelligence, designed and implemented in computer software and built into artefacts such as robots or intelligent machines [3]. Historically artificial intelligence programs appeared in various disguises such as universal problem solvers [4], expert systems [5], [6] and neural networks [7].

The author proposes that *artificial intelligence should be considered as an emergent property of complex systems* [2]. A network of several billions of neurons in the human brain is such a complex system in which intelligence is not traceable to any individual component – it emerges from the interaction of these components (neurons).

It follows that *the most direct approach to creating AI is to construct artificial complex systems in software and to experimentally seek to obtain aspects of intelligence such as understanding of meaning, learning, autonomous adaptation and decision making under conditions of variety and uncertainty.*

We shall explore this thesis after reviewing the concept of Complexity.

What is Complexity?

The following three paragraphs from Wikipedia [8] are a good introduction to the concept of complexity.

"Complexity has always been a part of our environment, and therefore many scientific fields have dealt with complex systems and phenomena. Indeed, some would say that only what is somehow complex – what displays variation without being random – is worthy of interest.

The use of the term complex is often confused with the term complicated. To understand the differences, it is best to examine the roots of the two words. "Complicated" uses the Latin ending "plic" that means, "to fold" while "complex" uses the "plex" that means, "to weave." Thus, a complicated structure is one that is folded with hidden facets and stuffed into a smaller space. On the other hand, a complex structure uses interwoven components that introduce mutual dependencies and produce more than a sum of the parts… This means that complex is the opposite of independent, while complicated is the opposite of simple.
While this has led some fields to come up with specific definitions of complexity, there is a more recent movement to regroup observations from different fields to study complexity in itself, whether it appears in anthills, human brains, or stock markets."

Following the train of thoughts suggested above, the intuitive interpretation of the term Complex as "difficult to understand" is correct as long as we accept that the reason for the difficulty is *the interdependence of constituent components*.

An example that immediately comes to mind is the Internet-based Global Market, where consumers and suppliers are trading, each pursuing their own goals and targets, and where the overall distribution of resources to demands emerges from individual transactions rather than according to a given plan.

According to Prigogine [9] a system is complex if its global behaviour *emerges* from the interaction of local behaviours of its components (the system creates a new order). Prigogine in his writings emphasises that the behaviour of a complex system cannot be predicted and that, in general, the future is not given [10]; it is being created by actions of all those that participate in the working of the Universe. He discusses examples of complex systems from physics and chemistry, including molecules of air subjected to a heat input, autocatalytic chemical processes and self-reproduction of cells. Emergent behaviour of complex systems is widely covered in literature [11] and applied to many domains, including economics [12].

To locate complex systems on a map of predictability, I proposed [13] the following system classification (see table below), in which complex systems are placed between random and stable systems.

CLASSES/ *Features*	**RANDOM SYSTEMS**	**COMPLEX SYSTEMS**	**STABLE SYSTEMS**	**ALGORITHMIC SYSTEMS**
Predictability	Total uncertainty	Considerable uncertainty	No uncertainty	No uncertainty
Behaviour	Random	Emergent	Planned	Deterministic
Norms of behaviour	Total freedom of behaviour	Some external guidance is essential	Governed by laws and regulations	Follows instructions
Degree of organisation	None	Self-organisation	Organised	Rigidly structured
Degree of control	None	Self-control by self-organisation	Centralised control	No need for control
Irreversible changes	Random changes	Co-evolves with environment	Small temporary deviations possible	None
Operating point	None	Operates far from equilibrium	Operates at an equilibrium	Operates according to the specification

Table 1. A classification of systems

The Key Elements of Complexity

Let us carefully examine the key elements of complexity emphasising those that are essential for the design of artificial complex systems [14].

1. Perhaps the most important feature of complex systems is that *decision-making is distributed* rather than centralised. Complex systems consist of interconnected autonomous decision making elements, often called Agents, capable of communicating with each other. There is no evidence of centralised control.
2. The autonomy of agents is not total. Every complex system has some global and/or local principles, rules, laws, or algorithms for agents to follow. The important point to remember is that agent's behaviour is never completely defined by these rules – they always have alternative possible local behaviours. In other words, complex systems always have a *variety* of possible behaviours and *uncertainty* as to which behaviour will be executed. The degree of freedom that is given to agents (decision makers)

determines the system's ability to self-organise and evolve. When uncertainty is insignificant, the system behaves predictably and lacks capabilities for self-organisation. When uncertainty is equal to 1, the system is chaotic (random). The adaptive complex systems operate "at the edge of chaos" or "far from equilibrium". The occurrence of events that affect their behaviour is so frequent that there is no time for the system to return to its equilibrium.

3. Global behaviour of a complex system *emerges* from the interaction of constituent agents. However, because the decision-making freedom of agents is restricted, complex systems exhibit *patterns* of behaviour. Designers have a choice here. The degree of uncertainty can be adjusted to force the system to follow specified broad patterns. The complete predictability should not be aimed for – it would prevent the system to self-organise and adapt, if and when required.

4. Complex systems are non-linear: the smallest external effects may cause large-scale shifts in system behaviour, the phenomenon known as *butterfly effect* (eg, as in climate systems) or as *self-acceleration* (eg, as in chain reaction in atomic explosions). Also, complex systems exhibit *autocatalytic* behaviour, that is, the ability to create new structures without any external help (eg, creation of organic structures from non-organic materials, under certain thermal conditions).

5. The distribution of decision-making implies interconnectedness of decision-making elements (agents). The links between agents can be strong or weak or nonexistent. The type of link between agents determines the responsiveness of the system when disturbed. Designers can weaken certain links between agents to reduce time required for ripples caused by a chain of changes to settle down.

6. The autonomy implies intelligence. Intelligence implies knowledge and a capability of applying knowledge to resolve uncertainty.

Technology for Constructing Models of Complexity

The most effective technology for constructing artificial complex systems, which exhibits all features described in the previous section, is multi-agent software [15].

In contrast to conventional software such as centralized schedulers, planners and optimizers, which from the start to the end follow algorithms, multi-agent software works primarily by exchanging messages: Intelligent Software Agents negotiate deals with each other, always consulting problem domain knowledge assembled in Ontology. Negotiations are conducted by a concurrent and asynchronous exchange of messages. The system is event-driven: it rapidly self-organises to accommodate events that affect its operation.

Problem domain knowledge is elicited and represented as a semantic network where concepts (classes of objects) are nodes and relations between concepts are links. Each object is characterised by attributes and rules guiding their behaviour. Such a conceptual knowledge repository is called *Ontology*.

A real-life problem situation is represented as a virtual network of instances of objects defined in Ontology and their relations. Such a problem description is called a *Scene*.

The elementary computational element is called *Agent*. An agent is a computer program capable of solving the problem at hand by consulting Ontology and using thus acquired knowledge to negotiate with other agents how to change the current Scene and turn it from the description of the problem into the description of a solution. Agents solve problems in co-operation and/or competition with other agents. As *Events* (new orders, failures, delays) affecting the problem domain occur, agents amend the current scene to accommodate the event, thus achieving *Adaptability*.

An agent is assigned to each object participating in the problem solving process (and represented in the scene) with a task of negotiating for its client (object) the best possible service conditions. For example, Passenger Agents and Seat Agents will negotiate takeoff/landing times and seat prices for requested air taxi flights. Closing a deal between a Passenger Agent and a Seat Agent indicates that a full, or at least partial, matching between *Demand* and an available *Resource* has been achieved. In case of a partial matching (eg, a passenger agrees to accept a later takeoff time but it is not pleased), his Agents may attempt to improve the deal if a new opportunity presents itself at a later stage (eg, if other passengers on the same flight agree an earlier takeoff time). The process continues as long as it is necessary to obtain full matches, or until the occurrence of the next event (say, a new request for a seat), which requires agents to re-consider previously agreed deals.

Agent negotiations are informed by domain knowledge from Ontology, which is far more comprehensive than “rules” found in conventional schedulers and normally includes expertise of practicing operators. Not all of this knowledge is rigid - certain constraints and if-then-else rules may be considered as recommendations and not as instructions and agents may be allowed to evaluate their effectiveness and decide if they should be used. In some cases agents send messages to users asking for approval to ignore ineffective rules or to stress nonessential constraints.

The power of agent-based modelling is particularly evident when the problem at hand contains a very large number of objects with a variety of different attributes; when there is a frequent occurrence of unpredictable events that affect the problem solving process; and when criteria for matching demands to resources are complex (eg, balancing risk, profits and level of services, which may differ for different participants).

As the process is incremental, a change of state of one agent may lead to changes of states of many other agents. As a result, at some unpredictable moment in time, a spontaneous self-accelerated chain reaction of state changes may take place, and after a relatively short transient time the overall structure will switch its state practically and completely. Once the resulting structure has settled, the incremental changes will continue.

It is evident from above discussions that agent-based software exhibits autonomy and emergent intelligence.

Architecture of Multi-Agent Software

A multi-agent software comprises the following key components: (a) Multi-Agent Engine, which provides runtime support for agents; (b) Virtual World, which is an environment where agents cooperate and compete with each other as they construct and modify the current scene;

(c) Ontology, which contains conceptual problem domain knowledge network and (d) Interfaces.

How Multi-Agent Software Works

Software consists of a set of continuously functioning agents that may have contradictory or complimentary interests. Basic roles of agents, based on extended Contract Net protocol, are Demand and Supply roles: each agent is engaged in selling to other agents its services or buying services it needs (Passenger Agents buy seats and Seat Agents sell them).

Current problem solution (current scene) is represented as a set of relations between agents, which describe the current matching of services; for example, a schedule is a network of passengers, seats, aircrafts and flights and relations between them.

The arrival of a new event into the system is triggered by the occurrence of a change in the external world; for example, when a passenger requests a seat on a particular flight, a Seat Request Event is triggered in the system.

The agent representing the object affected by the new event undertakes to find all affected agents and notify them of the event and the consequences (eg, the agent of the failed aircraft undertakes to find Passenger Agents linked to the failed flight and inform them that the flight is not available; the Aircraft Agent breaks the relevant relations and frees the Passenger Agents to look for other available flights.

The process of problem solving can run in parallel and asynchronously and, as a consequence, simultaneously by several active participants; for example, passengers that arrived at the website to book a flight simultaneously can start searching for suitable seats. All aircrafts assigned to flights can start immediately looking for free pilots. This feature is very effective because it eliminates a laborious building of flight schedules only to find out that pilots are not available for all selected flights.

The driving force in decision-making is often the presence of conflicts, which have to be exposed and settled by reconsidering previously agreed matches; for example, if a new flight finds out that the takeoff time slot it needs is already occupied, negotiations on conflict resolution start and, as a result, previously agreed flight-slot matches are adjusted (the takeoff time slot is moved to accommodate both flights) or broken (the time slot is freed). This capability to make local adjustments before introducing big changes is what makes agents-based problem solving so much more powerful in comparison with object-oriented or procedure-based methods.

A multi-agent system is in a perpetual state of processing - either reacting to the arrival of new events or improving the quality of previously agreed matches. The stable solution, when there are no agents that can improve their states and there are no new events, is hardly ever reached (agents are perpetually operating "far from equilibrium").

Solutions developed using multi-agent software fall into the class of open, non-linear and dissipative systems. As the number of relations increases in the system, the level of complexity of the resulting network goes up and, at a certain point, the need may arise to appoint additional agents to represent certain self-contained parts of the network whose nodes are already represented by agents. The increased complexity of solution structures may result in the creation of loops and the system may find itself in a local optimum. To avoid being stuck in a local optimum agents are given power to pro-actively seek alternative solutions from time to time. Attempts to avoid local minima are random (mutations).

Multi-agent systems can learn from experience as follows. Logs of agent negotiations are analysed with a view to discovering patterns linking individual agent decisions and successes/failures of the agent negotiation process. In future negotiations patterns leading to failures are avoided.

The pattern discovery process is itself agent based. An agent is assigned to each data element with a task of searching for similar data elements to form clusters. An agent is assigned to each new cluster with a task to attracting data elements that meet cluster membership criteria. Finally, clusters are represented as "if-then-else rules".

Designing Artificial Complex Systems using Agent Technology

Systems that have been designed under my supervision or with my involvement, using the above principles are described in some detail in [16], [17], [18], [19], [20]. Advantages of adaptability in comparison with rigid systems, such as ERP (Enterprise Resource Planning), are described in a popular format in my paper entitled "ERP: Elephants Rarely Pirouette" [21].

The list is substantial and includes real-time, adaptive multi-agent systems for: managing 10% of world tanker capacity for global crude oil transportation (in use); managing 2000 taxis and other service vehicles in London (in use); managing an extensive road logistic system across the UK (in use); managing social entitlements of citizens in a very large region (in use); managing distribution of rental cars across Europe for a major global car rental organisation (successful trials; in the commissioning stage); managing a car manufacturing system (prototype); simulating virtual enterprises (prototype); managing document flow for a major insurance company (prototype); managing all business processes of a new aviation company (in the design stage); managing a catering supply chain (in the design stage).

To illustrate the power of agent-based adaptive systems let me outline the complexity of the design problem that I am handling at present. The goal is to design an adaptive organisation based on teamwork, and a supporting intelligent multi-agent management system, which will make all operational decisions autonomously (how much to charge a customer for a flight, which pilot, aircraft, ground staff will be assigned to which duty, etc.) and manage domain knowledge required for strategic decisions (on expansion, on market penetration, on increasing business value) for a brand new enterprise. The enterprise network will enable rapid interaction of twelve multi-agent modules, including simulators, several schedulers, a demand forecasting system, a human resource management system, and customer relations management system, and will maintain integrity of all enterprise data and several enterprise ontology. The system is being designed to handle 4,000 travel requests a day, to book 400 taxi seats/flights a day and to schedule or re-schedule a large fleet of small aircraft, flights, crews, ground staff, aircraft maintenance, fuel supply, etc every 7 seconds, which is a task which would be impossible to achieve without agent-based technology.

I have also researched, simulated or prototyped a number of distributed, adaptive *products* following the design principles outlined above, including: a machine tool; an intelligent geometry compressor, an autonomous parcel distribution system and an intelligent family of robots [22], [23].

Perhaps the boldest idea was to design a compressor with moving individual vanes capable of autonomously and dynamically positioning themselves at the optimum angle whenever the operating point of the compressor changes. A software agent is assigned to each individual moving vane equipped with a pressure sensor. As pressure on the vane changes, the Vane Agent negotiates with agents of other vanes how to change vane angles to achieve the optimum pressure distribution along the stator. Before making a decision, agents consult domain knowledge stored in individual agents' minds, which can be updated without interrupting work of agents, to fine tune compressor operation. A very successful simulation [24] showed that the compressor with autonomous vanes, when coupled with an aircraft jet engine, is fully adaptive to sudden changes of loads and is able to prevent stalling of jets caused by lack of air intake.

Replacing a robot by a family of smaller robots illustrates the advantages of designing complexity into artefacts even better. To avoid disasters that ruined both American and British Mars exploration robots (the first died from the accumulation of space dust on its solar cells after a week in space, and the latter fell into a crevice on landing and was immediately lost), I proposed to design a family of five smaller robots capable of cleaning each other, rescuing members of the family from disasters and, more importantly, being able to complete their task successfully even if one or two of the family members were disabled.

A family of robots is an adaptive distributed system, which incorporates all critical complexity features listed earlier in this paper. All decisions are executed after a process of consultation and negotiation among members of the family. There is no "senior" robot ordering others what to do. Each robot is controlled by a set of interacting swarms of agents. Agents consult domain knowledge before making decisions. A copy of domain knowledge is stored in each robot's ontology, making it capable of undertaking any task within domain boundaries. The family represents a "swarm of interacting swarms" of agents and therefore exhibits a considerable emergent intelligence. Robots are trained to help each other, share the workload and self-organise the team if a member is disabled without loosing ability to achieve the goal.

Experimenting with Multi-Agent Systems

Using large-scale complex systems based on agent technology for research purposes is relatively straight forward if one stores all messages exchanged among agents. By shifting through the log of messages one can find connections between agent decisions and particular behaviours of the system and thus can deduce conditions under which desired behaviours emerged. The process is laborious but can be automated using appropriate agent-based software. My interest was in isolating system behaviours that could be described as aspects of emergent intelligence (understanding of meaning, learning, analysis, decision making under conditions of uncertainty, adaptation and autonomous creation of novel structures) as described in [2].

Conclusions

Constructing artificial complex systems using multi-agent technology and conducting experiments aimed at provoking a complex software system to exhibit emergent intelligence is a new approach to studying AI. The pre-requisite skills are skills of designing complexity into software, a notion that appears to be counter-intuitive. The conventional wisdom is to ensure that software is rigidly structured and "correct". The new thinking is to make software autonomous, adaptable and self-organising (and therefore unpredictable), in other words, intelligent.

References

1. McCulloch, W.S. and Pitts,W. "A Logical Calculus of the Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics, Volume 5, 1943, pp 115-137.
2. Rzevski, G., Skobelev, P., "Emergent Intelligence in Large Scale Multi-Agent Systems". International Journal of Education and Information Technology, Issue 2, Volume 1, 2007, pp 64-71.
3. Rzevski, G (ed), "Mechatronics: Designing Intelligent Machines", Butterworth Heinemann, 1995.
4. Newell, A. and Simon, H. A. GPS, a Program that Simulates Human Thoughts. In Billing, H. (ed), Lernende Automaten, 1961, pp 109-124.
5. Feigenbaum, E. A., Buchanan, B. G. and Lederberg, J. "On Generality and Problem Solving: A Case study using the DENDRAL Program". In Meltzer, B. an Michie, D. (eds) Machine Intelligence, Volume 6, 1971, pp 165-190.
6. McDermot, J. "R1: A Rule-Based Configurer of Computer Systems". Artificial Intelligence, Volume 19(1), 1982, pp 39-88.
7. Rumelhart, D. E. and McClleland, J. L. (eds) "Parallel Distributed Processing" 1986, MIT Press, Cambridge.
8. Wikipedia www.wikipedia.com
9. Prigogine, Ilya, "The End of Certainty: Time, Chaos and the new Laws of Nature". Free Press, 1997.
10. Prigogine, Ilya, "Is Future Given?" World Scientific Publishing Co., 2003.
11. Holland, John, "Emergence: from Chaos to Order". Oxford University Press, 1998.
12. Beinhocker, Eric, "The Origin of Wealth: Evolution, Complexity and the Radical Remaking of Economics".
13. Rzevski, G., "Investigating Current Social, Economic and Educational Issues using Framework and Tools of Complexity Science". Journal of the World University Forum, Volume 1, Number 2, 2008.
14. Rzevski, G., Skobelev, P., "Emergent Intelligence in Large Scale Multi-Agent Systems". International Journal of Education and Information Technology, Issue 2, Volume 1, 2007, pp 64-71.
15. Rzevski, G., Skobelev, P. Andreev, V. "MagentaToolkit: A Set of Multi-Agent Tools for Developing Adaptive Real-Time Applications". In Marik, V., Vyatkin, V., Colombo, A. W. (eds.) Holonic and Multi-Agent Systems for Manufacturing. Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2007, Regensburg, Germany, September 2007, pp 303-314. Springer LNAI 4659.

16. Rzevski, G., Himoff, J., Skobelev, P., "Magenta Technology: A Family of Multi-Agent Intelligent Schedulers". Workshop on Software Agents in Information Systems and Industrial Applications (SAISIA). February 2006. Fraunhofer IITB.
17. Andreev, M., Rzevski, G., Skobelev, P., Shveykin, P., Tsarev, A., Tugashev, A. "Adaptive Planning for Supply Chain Networks". In Marik, V., Vyatkin, V., Colombo, A. W. (eds.) Holonic and Multi-Agent Systems for Manufacturing. Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2007, Regensburg, Germany, September 2007, pp 215-225. Springer LNAI 4659.
18. Rzevski, G, Skobelev, P, Batishchev, S, Orlov, A.: "A Framework for Multi-Agent Modelling of Virtual Organisations". In Camarinha-Matos, L M and Afsarmanesh, H (eds), Processes and foundations for Virtual Organisations, Kluwer Academic Publishers, 2003, pp. 253-260.
19. Minakov, I., Rzevski, G., Skobelev, P. and Volman, S., "Creating Contract Templates for Car Insurance Using Multi-Agent Based Text Understanding and Clustering". In Marik, V., Vyatkin, V., Colombo, A. W. (eds.) Holonic and Multi-Agent Systems for Manufacturing. Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2007, Regensburg, Germany, September 2007, pp 361-371. Springer LNAI 4659.
20. Rzevski, G., Skobelev, P., Minakov, I. and Volman, S., "Dynamic Pattern Discovery using Multi-Agent Technology". Proceedings of the 6th WSEAS International Conference on Telecommunications and Informatics (TELE_INFO '07), Dallas, Texas, USA, March 22-24, 2007, pp 75-81. ISBN: 978-960-8457-60-7.
21. Brace, G, Rzevski, G, ERP – "Elephants Rarely Pirouette". Logistics Focus Volume 6, No 9 1998.
22. Rzevski, G.: "On conceptual Design of Intelligent Mechatronic Systems". Mechatronics 13 (2003) pp. 1029 – 1044.
23. Rzevski, G, "Engineering Design for the Next Millennium: The Challenge of Artificial Intelligence". The 86th Thomas Hawksley Memorial Lecture, IMechE, 9 December 1998.
24. Morgan, G, Rzevski, G., Wiese, P.: "Multi-Agent Control of Variable Geometry Axial Turbo Compressors". Journal of Systems and Control Engineering, issue I3 vol. 218 (2004), pp. 157-171.

Using Neural Network for Recognition of Handwritten Mathematical Documents

Nilupa Liyanage , Asoka Karunananda
Faculty of Information Technology, University of Moratuwa, Sri Lanka.
Email : nilupal@uom.lk, asoka@itfac.mrt.ac.lk

***Abstract* - Advancements in modern technologies cannot still override the importance of preparation of handwritten documentations. In particular, handwritten documentations are inevitable in mathematical calculations, mathematical tutorials, preparation of marking schemes and financial reports. This paper presents our approach to the design and implementation of Artificial Neural Network solutions for recognition of handwritten mathematical documents and producing text files. The system consists of three modules for image processing, character recognition and text formation. The Image processing module of the system has been designed to perform thresholding, normalization, segmentation and feature extraction of the handwritten numeric characters. The Image processing module captures the features of handwritten characters and to produce quality inputs for the ANN module. The Artificial Neural Network module for character recognition has been designed with a three layer architecture to use back propagation training algorithm. Image processing has been done through MATLAB while NeuroSolution toolkit has been used for the development of ANN and formation of textual output.**

Keywords- **Image processing, Artificial Neural Networks, Handwritten character recognition**

1. INTRODUCTION

Multifaceted advancements in modern technology, still cannot underestimate the value of handwritten documents. Some handwritten materials can be used as a means of identification of individuals. In addition, hand written documents can be produced at any time without requiring sophisticated technology. The use of handwriting has become even more significant with respect to numeric characters and arithmetic operations. In this sense, policemen, accountants, teachers and examiners are the best examples of people who still use handwritten numeric characters before accessing any device such as a calculator or a computer. As such, computer-based solutions for recognition and transformation of handwritten mathematical documentation have been a research challenge. This area of research goes beyond mere scanning of documents with mathematics and producing editable text documents. This is because, a typical mathematical documents includes more symbols than letters and numbers.

The newer technologies such as personal digital assistants (PDAs) have their impact on handwriting [1]. These inventions have led to the reinterpretation of the role of handwriting, but a pen together with paper is more convenient than a keyboard or a mouse. Obviously, handwritten documents are a preferred way to solve mathematical problems, making schemes, etc. In fact, such documents can be produced without requiring sophisticated technologies, but with a pen and a paper.

Among others, Miguel [2] has developed a numeric character recognition system for mail sorting in the postal department of US. This system has enabled to automate the recognition of postal codes with almost 90% accuracy. Literature mentions that the field of handwritten character recognition is almost thirty years old. There are a number of companies that have been involved in research on handwriting recognition for the past few years [3]. Some handwriting recognition systems go beyond the mere recognition of characters, but work as input preprocessors for some complex computer systems. However, since handwritten characters are specific to individuals, it is rather impractical to develop a handwritten character recognition system for global use.

This paper reports on the design and implementation of recognition of mathematical documents in the Sri Lankan context. The core system has been implemented as an Artificial Neural Networks that has been developed using the Back Propagation training algorithm.

The rest of the paper is organized as follows. Section 2 describes related work in handwritten recognition systems. Section 3 discusses Artificial Neural Network approach to recognition of mathematical documents. Section 4 reports on our design and implementation of Handwritten

Mathematical document Recognition system while section 5 writes on further work.

2. RELATED WORKS IN HANDWRITTERN CHARACTER RECOGNITION

At the outset it is worth mentioning that most character recognitions systems have used techniques in image processing followed by a technology for image recognition. Among other technologies, Artificial Neural Network (ANN) has shown promising results as a technology for image recognition. In fact, ANN has been used not only for image recognition but also for implementing some stages in image processing. For example, Z. Shi and coworkers have used ANN for segmentation and recognition of numeric characters [7]. Further, ANN has been used for segmentation in License Plate Recognition System [18]. However, it should be noted that the scripts which are cursive in nature are difficult to segment. In such events, special algorithms must be used. For example, an area based algorithm has been proposed for the skew detection of characters in Bangla specimen [8]. In this project, before segmentation, the features of images have been extracted by the analysis of specimen through the above algorithm. Finally segmenting points have been recognized through Multilayer Perceptron (MLP) Neural Networks [8].

There are so many applications, where ANN has been used as an approach to recognize characters. For instance, Alexander [17] has used back propagation neural network with one hidden layer to create an adaptive character recognition system. The system was trained and evaluated with printed text, as well as several different forms of handwriting provided by both male and female participants [17]. The handwritten pages were scanned and converting the scanned characters to code readable by MATLAB was achieved with a java application. After that extracted features were fed into neural network.

The off line cursive handwriting recognition system has also used image processing and neural network as technologies [4]. Here image processing is used to capture data from a handwritten document and conventional flat bed scanner has been used. The scanned image must be segmented into separate words and then a series of image processing operations is carried out to normalize the image. After that it has used neural network to estimate data for each frame of data in the representation [4].

Segmentation and Recognition of the hand written numerical chains were mainly based on the evaluation of neural network performances, trained with the gradient backpropagation algorithm [5]. Vertical projection was used to segment the numeric chains at isolated digits and every digit was presented separately. Used parameters to form the input vector of the neural network are extracted on the binary images of the digits by several methods: Distribution sequence, Barr features and centered moments of different projections and profiles [5].

Most of the character recognition systems work as a part of an integrated system. They are also specific to some countries and applications. Therefore, such systems cannot be used globally. ANN based systems cannot be also adapted for a purpose other than for which it has been trained. Although we deal with just 10 numerals, there is a need for developing a different numeric character recognition system that depends on the application and the context. Since our project deals with recognition of mathematical documentation, there is a need for recognition of extra symbols other than ten digits. Theses symbols include brackets, sings of mathematical operations (e.g. +, -, /, x, =, >, <). Next section describes our approach to design and development of an ANN-based system for handling mathematical symbols.

3. PROPOSED APPROACH

Our approach to the identification of handwritten mathematical documents comprises three steps, namely; image processing, image recognition and producing the output as a text document written to a file. We have used the standard techniques of image processing; while the image recognition is handled by an ANN trained in the supervised mode with the aid of backpropagation training algorithm. It is evident that ANN is the best technology for applications such as recognition of handwritten characters, which are rather incomplete and cannot be represented in an algorithmic manner.

Next we briefly describe our approach in terms of input, output, process, users and overall benefits of the system.

Input – The input for the system would be the scanned documents with handwritten numeric characters and mathematical symbols. For example, a typical 8.5 X 11 inch page is scanned at a resolution of 300 dots per inch to create a gray scale image of 8.4 megabytes. The resolution is dependent on the smallest font size that needs reliable recognition.

Output – The output of the system will be a text file that shows numeric characters and mathematical symbols as shown in the original handwritten document. This document is legible and editable if necessary.

User- The users of the system would be persons such as teachers, examiners, students and accountants who produce mathematical documents and wish to transfer the documents as text files that can be edited.

Process – In our approach, firstly the handwritten mathematical document will be scanned and saved as an image. After that the scanned image is normalized because an image can have different sizes and appear at different positions. Then noise will be removed in normalizing the image before thresholding. The pixels of the scanned image will be read as the input for the ANN. The ANN uses backpropagation algorithm for training the input data. Finally the output will be saved as a text file.

Benefits – There are various benefits of the system for handling mathematical documents. Firstly, it can be used for a customized usage by a given person. Since it is convenient to solve mathematical problems manually, our solution will be immensely beneficial to persons like teachers, students, accountants, etc. Secondly, with the use of ANN, even the cursive handwritten documents can be recognized by the system. As such one does not need to worry about his/her handwriting seriously. Thirdly, the system goes beyond the recognition of numeric characters, but processes an editable document that can be used for a secondary purpose. For instance, using this system one can produce handouts and presentation materials upon the preparation of handwritten lecture notes, etc.

4. DESIGN AND IMPLEMENTATION

The top level architecture of the design of our system is shown in Fig. 1. It comprises three modules, namely, Image Processing Module, Artificial Neural Network Module and Output Generator Module. In our design, Artificial Neural network Module is fundamental to the entire system. Next we briefly describe the design and implementation of each module.

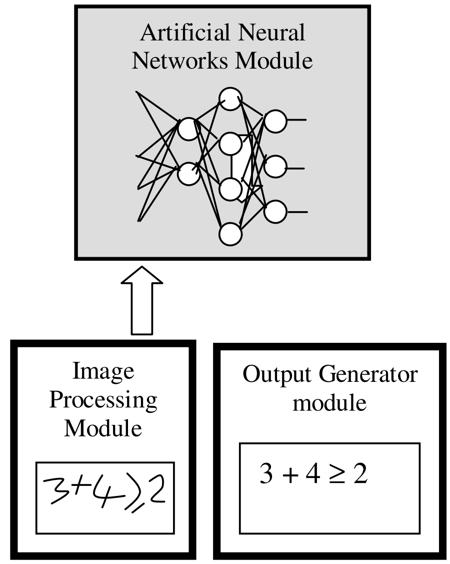

Fig. 1 :Top Level architecture of the system

4.1 Image Processing Module

The image processing module consists of three steps. They are, image normalization, thresholding and feature extraction. This module has been implemented by MATLAB and NeuroSolution. The image processing module works as a preprocessor for the Artificial Neural Networks module. During a training session images of numbers and mathematical symbols can also be used from a drawing tool such as Photoshop. However, in the actual use of the system the image processing module should scan images that are coming to the system.

The image processing module reads a scanned image as the input and normalizes the image in the first place. At this step, this module ensures that features of the image have not been affected due to normalization, which fixes the image into a standard size. The basic steps in image normalization in MATLAB are;

```
r1=imread('image name') // Read  image;
r2=imcrop(r1,[width,Height]) // Crop  image;
r3=imresize(r2,[32,32]) //change the size of image;
```

During the task of thresholding in the image processing module extracts the foreground (ink) from the background (paper) [9]. This process improves the clarity of the image by increasing intensity of some unclear pixels in the original image. It should be noted that thresholding is necessary to be applied in image processing to handle the unclear sectors in original images [14]. Basic steps in MATLAB to threshold an image:

```
level = graythresh(r3)// threshold  normalized
                                    image;
```

Having applied the thresholding on the image, the image processing module next performs the operation of feature extraction. This process digitizes the image. The feature extraction function has been implemented with the use of NeuroSolution [19].

4.2. Artificial Neural Network Module

The Artificial Neural Network Module receives digitized input vectors (32x32) created by the image processing module. The ANN has been implemented as a three layer network to be trained using backpropagation training algorithm. This module has been designed and developed with the use of NeuroSolution to identify 10 digits and other mathematical symbols such as +, -, /, x, =, <, >, ≤, ≠, ≥ (and).

Our study on various environments for ANN development shows that NeroSolution is a better choice due to various reasons. Among other reasons, NeuroSolution is capable of accepting image files directly as the input for the neural networks. Further, NeuroSolution also allows facilities for changing parameters pertaining to the quality of the image. As in most ANN training environments, Neursolution also provides facilities for changing network and training parameters during a training session. However, we have noticed that NeuroSolution version that we have used has a limitation of requiring reentering inputs during a retraining session.

We have been training the images of digits and the above symbols for the multi layer neural network architecture. In this regard, various architectures with a different number of layers and neurons have been experimented. The network accepts input with 32x32 pixels and learns them into 18 different digits and symbols. At present we have tried various architectures and also preliminary testing was done. Currently the training is in progress to accommodate more training data and to increase the accuracy of recognition of images.

4.3 Output Generator Module

The output generator module has been designed to write the output recognized by the trained network into a text file. This file will be as same as the original document that has been converted into a sequence of images. We are currently working in this module, and it requires consideration of some important factors. For instance, this module should be developed to highlight the characters which are identified with some ambiguity. Undoubtedly, a form of post-editing is preferable on the output file to ensure the accuracy of the output generated. However, the process of editing would not be that time consuming if the Output generator module can highlight any ambiguity.

5. CONCLUSIONS AND FURTHER WORK

This paper has discussed the need for transforming handwritten mathematical documents into legible and editable forms on the computer. We are inspired by that fact that many people including lecturers, teachers, students, examiners and accountants are still used to produce handwritten documents before processing those documents on computers. It is evident that ordinary scanning cannot be used as a solution for the requirement except for processing the documents with letters and numbers. However, mathematical documents include various symbols that are unique for mathematics. In view of that we have reported the design and development of Artificial Neural Network-based systems for the recognition of mathematical documents and transforming them into an editable format. We discussed the overall system as per three modules, namely; Image processing module, ANN module and the Output Generator module. We have already completed the development and incremental testing of the first two modules. The development of output generator module is in progress. The system has been developed with the use of MATLAB and NeuroSolution. The system will be able to run on an ordinary PC.

Further improvements to the systems will be done as per training of more inputs to the system to give a high level of generalization and accuracy. The development of the output generator module will also be done in parallel.

6. REFERENCE

[1]. Rejean Plamondon ,Fellow,IEEE, and Sargur N.Srihari, Fellow,IEEE.On- Line and Off-Line Handwriting Recognition:
[2]. Miguel Po-Hsien Wu, “Handwritten Character Recognition”, The school of Information Technology and Electrical Engineering, The University of Queensland.

[3]. A Meyer, “Pen computing: A Technology overview and a Vision,”SIGCHI Bulletin, vol 27, no 3,pp. 46-90,July 1995.

[4]. Andrew w Senior and Anthony J. Robison “An off line cursive handwriting Recognition System”, IEEE Trans. On Pattern analysis and machine intelligence, vol. 20, pp.309 – 321, 1998.

[5]. Salim Ouchtati, Mouldi Bedda and Abderrazak “Segmentation and Recognition of Handwritten Numeric chains”, Jr. on Computer Science 3, vol 4:242 – 248, 2007.

[6]. M. Shridhar and A. Badreldin, “Recognition of isolated and simply connected handwritten numerals “, Pattern Recognition 19(1):1 – 12, 1986.

[7]. Z. Shi, N. Srihari, C. Y. Shin, and A. V. Ramanaprasad , “ A system for segmentation and recognition of totally unconstrained handwritten numeral strings. In Proc. Of 4th International Conference on Document Analysis and Recognition (ICDAR), vol.2,pages 455-458.

[8]. T.K. Bhowmik,A.Roy and U. Roy “Character Segmentation for handwritten Bangla words using Artificial Neural Network”.

[9]. P. Shankar rao and J. Aditya “Handwriting Recognition – “offline” Approach” , ¾ CSE,GITAM.

[10]. Sriganesh Madhavanath ,Member, IEEE and Venu Govindaraju , Senior member , IEEE. “The Role of Holistic Paradigms in Handwrittern Word recognition “

[11]. R.A. Lotufo, A.D. Morgan and A.S. Johnson, “Automatic Number Plate Recognition “, IEEE Colloquium on Image Analysis for Transport Applications, London, Feb. 1990.

[12]. Nawwaf N Krarima and Rabab K. Ward “Character Recognition Systems for the non Expert”, University of British Colombia.

[13]. Haralick,Robert M.,and Linda G.Shapiro. Computer and Robot Vision, Volume I Addison – Wesely ,1992.p.518.

[14]. Floyd,R.W. and L. Steinberg . “An Adaptive Algorithm for Spatial Gray Scale,” International Symposium Digest of Technical Papers. Society for Information Displays, 1975.p.36.

[15]. Lim, Jae S. Two – Dimensional Signal and Image Processing. Englewood Clifts, NJ: Prentice Hall,1990.pp.469 – 476.

[16]. Ardeshir Goshtasby, Piecewise linear mapping functions for image registration , Pattern Recognition, Vol 19, p.459 – 466,1986.

[17]. Alexander J. Faaborg “Using Neural Networks to create an adaptive Character Recognition System”, Cornell University,Ithaca NY.

[18]. Yungang Zhang and Changa Zhang “A /New Algorithem for Character Segmentation of Licence Plate” , Dept. of Automation,Tsinghua University ,The Institute of Information Processing, Beijing 100084,China.

[19] http:///www.neuroSolutions. com

Web-based English to Sinhala Selected Texts Translation system

B. Hettige[1], A. S. Karunananda[2]
[1]Department of Statistics and Computer Science, Faculty of Applied Science,
University of Sri Jayewardenepura, Sri Lanka.
[2]Faculty of Information Technology, University of Moratuwa, Sri Lanka.
budditha@yahoo.com[1] , asoka@itfac.mrt.ac.lk[2]

Abstract – **This paper presents English to Sinhala Machine Translation system that can translate selected English text into Sinhala through the web. This Translation system contains two modules, namely; web-based machine translation system and java based user interface. Core of the translation system runs on a web server and can be accessed by user interface. The core of the translation system contains seven modules, namely, English Morphological analyzer, English Parser, Translator, Sinhala Morphological generator, Sinhala parser, Transliteration module and three Lexicon Databases. Java based user interface provides a mechanism for on-demand translation of selected texts from an English document. This enables users to get translated a selected set of English sentences while reading a document.**

1. Introduction

Sinhala is a member of Indo Aryan family of languages and is the spoken language of the majority of Sri Lankans. Most Sri Lankan people use Sinhala as the spoken and written language. Their ability in Sinhala language usage is at a reasonable level. However their understanding and writing ability of the English language is not comprehensive. This is known as the language barrier that affects both acquisition and dissemination of knowledge.

Machine Translation (MT) is a process that translates one natural language into another. MT is a complex and signal task because it can be used as the solution to the language barrier. Therefore, we have been working on the development of English to Sinhala machine translation system.

In general, a machine translation system contains a source language morphological analyzer, a source language parser, translator, a target language morphological analyzer, a target language parser and several lexical dictionaries. The source language morphological analyzer analyzes a source language word and provides morphological information. The source language parser is a syntax analyzer that analyzes the source language sentences. A translator is used to translate a source language word into the target language. The target language morphological analyzer works as a generator and generates appropriate target language words for given grammatical information. Also the target language parser works as a composer and composes suitable target language sentences. Furthermore, any MT system needs a minimum of three dictionaries such as the source language dictionary, the bilingual dictionary and the target language dictionary. The source language morphological analyzer needs a source language dictionary for morphological analysis. A bilingual dictionary is used by the translator to translate the source language into the target language; and the target language morphological generator uses the target language dictionary to generate target language words.

Many Asian and European countries have already taken steps to develop machine translation systems. In the Asian region, Indians have developed a variety of machine translation systems, including Mantra(Machine assisted translation tool) [6], Matra [8], Anusaaraka [2], AngalaBarathi [4] and Angalahindi [3], Shakit[7] and UNL Based MT system [9]. METIS-II [10] , PLOENG [11], and MANOS [12] being some of the European Machine Translation systems. Among others, EDR [28] by the Japanese is one of the most successful machine translation systems in the world.

These translation systems use various approaches to machine translation, including, Human-Assisted Translation, Rule based Translation, Statistical Translation, Example-based and Knowledge-based Translation etc. However, due to various reasons associated with complexity of languages, for more than fifty five years, Machine Translation (MT) has been identified as one of the least achieved areas in computing. Most of these issues are associated with semantic handling in MT systems.

As for the English to Sinhala machine translation system, we have already developed the Sinhala parser [13], Sinhala morphological analyzer [14], Transliteration module [15] and three lexical databases [16]. The Sinhala parser and morphological analyzer have been tested through various applications such as Sinhala Chatbot [15] and Sinhala Sentence generator [13]. To test and update the English to Sinhala Machine Translation system we have introduced three prototype systems, namely; Human-Assisted machine translation system[18], human-interaction machine translation system for online dictionary update[20] and web-based English to Sinhala MT system[21]. The human-assisted machine translation system uses intermediate-editing approach [19] to semantic handling. This system also introduces an intermediate editor to handle semantics of the sentence. The human-interaction system also uses intermediate-editing approach to semantic handling and it uses human interaction methods to update dictionaries within the translation. Web-based English to Sinhala MT system is developed to translate English text into Sinhala through the web.

This paper reports the development of a mechanism for on-demand translation of selected texts from an English document. This enables users to get translated a selected set of English sentences while reading a document. This system is an extension of our core machine translation system and it enables students and the general public to use our translation system.

The rest of this paper is organized as follows. Section 2 describes the overview of some existing machine translation systems. Section 3 gives design of the developed core machine translation system. Then section 4 briefly describes existing version of the English to Sinhala Machine translation system. Section 5 introduces the selected text translation system. Then section 6 elaborates on how the system works in practice. Finally, Section 7 concludes the paper with a note on further work.

2. Some Existing MT Systems

Machine Translation systems use various approaches for translation; including Human-Assisted translation, Rule based translation, Statistical translation and Example-based translation [24]. Human-Assisted machine translation shares the translation task between man and the machine. The rule based approach translates rule maps from source to the target language representations. The example-based machine translator uses the extended idea of translation memories and reuses existing translation fragments. The statistical machine translation approach is a popular approach that gives alternative possible translations and finds the most probable one in the target language. This method needs a large corpus of the target language. Compared with the existing approach, Human-Assisted machine translation is the most fundamental approach for the machine translation. This approach uses human knowledge to solve translation problems like semantic and multiword-expressions.

However, due to various reasons associated with the complexity of languages, over the past fifty five years, MT has been identified as one of the least developed areas in computing. Most of these issues are associated with semantic handling in machine translation systems. A large number of MT systems have been developed for many languages all over the world. Until recently, there were no initiatives taken to attempt machine translations for Sinhala language. Sinhala is an Indo Aryan language and certain Indian languages like Pali, Sanskrit and Tamil are close to the Sinhala language. Therefore, we need to study some existing MT systems; especially the ones developed for Indian languages.

At present Indians have developed a variety of machine translation systems such as Anusaaraka, Mantra, Angalahindi, Shakti, etc. The Anusaaraka [2] is a popular machine-aided translation system for Indian languages that makes text in one Indian language accessible in another Indian language. Also, this system uses Paninian Grammar (PG) model [1] for its language analysis. The Anusaaraka project has been developed to translate Punjabi, Bengali, Telugu, Kannada and Marathi languages into Hindi. The approach and the lexicon is general, but the system has mainly been applied to children's stories.

MANTRA [5] is one of the web-enabled machine translation systems that translate the English text into Hindi in a specified domain of personal administration, specifically gazette notifications, office orders, office memorandums and circulars. It uses Tree Adjoining Grammar (TAG) [24] for Parsing and Generation and bottom-up parsing algorithm to speed up the parser and online word addition and grammar creation, updating facilities.

Angalahindi [3] is a web-based, English to Hindi Machine-Aided translation system. Its translation methodology has been developed by the author who translates all Indian languages to English.

Shakti[7] and UNL Based MT system [9] are two other web based machine translation systems used for translating English to Indian languages. Among others, Electronic Dictionary Research (EDR)[30], by Japanese, is the most successful machine translation system. This system has taken a knowledge-based approach in which the translation process is supported by several dictionaries and a huge corpus. While using the knowledge based approach, EDR is governed by a process of statistical MT. As compared with other MT systems, EDR is more than a mere translation system but provides lots of related information.

3. Design of the English to Sinhala MT System

Brief description of the core English to Sinhala machine translation system is given below. The core English to Sinhala MT system contains seven modules, namely; English morphological analyzer, English parser, translator, Sinhala morphological analyzer, Sinhala Parser, Transliteration module and three Lexical dictionaries namely the English dictionary, the Sinhala dictionary and the English Sinhala bilingual dictionary. Fig 1 shows the design of the English to Sinhala core machine translation system with these modules. The main input of the system is English sentences and output of the system is translated Sinhala sentences. After reading the input sentence it analyzes the input by using English morphological analyzer and the English parser. Then the system finds the corresponding Sinhala word for the given input word. After that, the system generates suitable Sinhala words by using Sinhala morphological generator. Finally the system generates correspondent Sinhala sentences by using Sinhala parser. This is the basic design of the English to Sinhala machine translation system and it does not handle semantics of the sentence. Each component of the core system describes below.

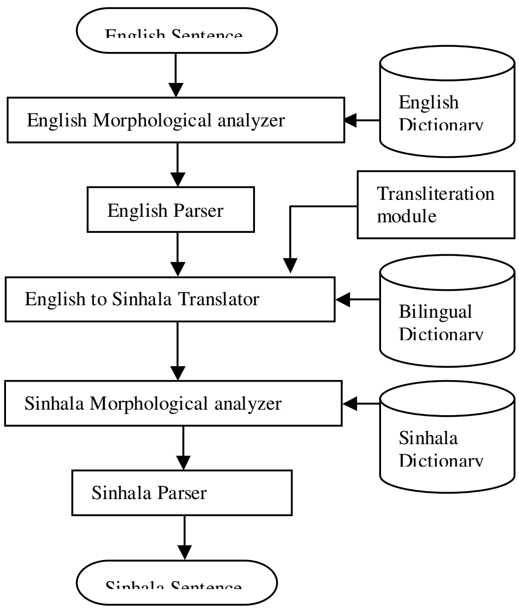

Fig. 1: Design of core English to Sinhala MT system

3.1 English Morphological analyzer

The English Morphological analyzer reads the given English sentence word-by-word and identifies morphological and lexical information on each word such as type of the word (Noun, Verb, Adjective, Adverb etc.), grammatical information for each word types such as (person, number and case for Noun, tense for verb etc.) There are many morphological analyzers available for the English language. Therefore, in this development, we have customized an existing English morphological analyzer. The morphological analyzer in our MT system has linked up with an English dictionary to get grammatical information on the words in the input sentence. SWI-Prolog [25]has been used to implement this morphological analyzer. The present English Morphological analyzer can identify all the Basic English word patterns (inflections). The prolog file named ‘eng_morp_analyzer.pl’ contains prolog based English Morphological analyzer and analyzeWords/1 prolog predicate is used to analyze a given word list. Prolog based English dictionary named ‘eng_word_dic.pl’ is used to analyzed the English words. Output result of the English Morphological analyzer is stored in a file named ‘eng_morp_analyzer_out.pl”. Result of the Morphological analysis is stored in a file and that can be read by others.

3.2 English Parser

The English parser receives source English sentences and the output result of the English morphological analyzer. This parser works as a syntax analyzer. Since there are many English parsers, we have customized an existing parser for our purpose. The current version of the parser used in our MT system is mainly concerned only with simple sentences. At present we have assumed that the input English sentence is grammatically correct therefore we do not need complex parser. The parser has also been implemented using SWI-PROLOG. The prolog file named ‘eng_parser.pl’ contains all in the English parser and ‘analys_eng_sen/2’ prolog predicate is used to analyze a given English sentence. Results of the parsing for a given English sentence is stored in a file named ‘eng_parser_out.pl’. Note that, in the machine translation point of view, English parser identifies the English sentence and categorizes the sentence into sub parts such as subject, object verb and complement. This information is stored in a file named ‘eng_parser_out.pl’.

3.3 Translator

The Translator is used to translate an English base word into a Sinhala base word with the help of the bilingual dictionary. The translator translates subject, object and verb in the English sentence separately. This is a method we have used to reduce the complexity of the translation process. This translator is a simple one and it does not automatically handle the semantics of sentences. Note that, this stage can be supported by human intervention to generate the most appropriate translation for some words in a sentence. As such, handling semantic, pragmatic and Multiword expressions must be addressed with the support of humans, for which we introduce an intermediate-editor. The prolog file named ‘eng_sin_translator.pl’ contains prolog based translator. Translator reads all the output results from the English Morphological analyzer and the English parser. Finally, result of the translation is stored in a file named as ‘eng_sin_translator_out.pl’.

3.4 Sinhala Morphological Analyzer

The Sinhala Morphological analyzer works as a morphological generator. This morphological analyzer reads the words from the Translator word by word, all other information from English Morphological analyzer and English parser. For each word, the Sinhala morphological analyzer generates the appropriate Sinhala word with full grammatical information such as nama (nouns), kriya (verb) and nipatha (preposition) in the Sinhala language[22]. This morphological analyzer works with the help of three dictionaries, namely, Sinhala rule dictionary, Sinhala word dictionary and Sinhala concepts dictionary. All these dictionaries (prolog databases) and the morphological analyzer are implemented using SWI-Prolog.

3.5 Sinhala Parser

The Sinhala parser works as a Sentence composer. It receives tokenized words from the morphological analyzer and composes grammatically correct Sinhala sentences. Generally, a Sinhala sentence contains 5 components, namely, Ukktha visheshana (adjunct of subject), Ukkthya (Subject), karma visheshanaya (attributive adjunct of object), karmaya (object) and akkyanaya (verb) [28][29]. These five components of a Sinhala sentence are the building blocks for the design and implementation of a Sinhala parser. The parser is also one of the key modules of this English to Sinhala machine translation System and is also implemented using SWI-PROLOG.

3.6 Lexical Dictionaries

The translation system uses six dictionaries such as an English word dictionary, English concepts dictionary, English-Sinhala bilingual dictionary, Sinhala word dictionary, Sinhala rule dictionary and Sinhala concept dictionary. The English word dictionary contains English words and lexical information. The English concept dictionary contains synonyms, antonyms and general knowledge about English words. The English to Sinhala bilingual dictionary is used to identify the appropriate Sinhala base word for a given English word and contains the relation between English and Sinhala words. The Sinhala word dictionary stores Sinhala regular base words and lexical information. Similar to the English dictionary, the Sinhala concept dictionary stores semantic information. The Sinhala rule dictionary stores rules required to generate various word forms. These are the inflection rules for the formation of various forms of verbs and nouns from their base words. The rule dictionary also stores vowels, consonants, upasarga (prefixes) and vibakthi (case).

3.7 Transliteration module

The MT system needs to solve Out-of-vocabulary problems and handle technical terms. Machine transliteration can be used as a resalable solution for this. Transliteration is the practice of transcribing a word or text written in one writing system into another writing system [24]. In other words, machine transliteration is a method of automatic conversion of words in one language into phonetically equivalent ones in another language. At present we have developed two types of transliteration models. One of these models transliterates Original English text into Sinhala Transliteration and the other transliterate Sinhala words that are written in English which are transliterated into Sinhala. Finite State Transducers (FST) have been used to develop these two modules.

4. Development Stages of the English to Sinhala Machine Translation

The core of our MT system has seven modules, namely, English morphological analyzer, English parser, Translator, Sinhala morphological analyzer, Sinhala parser, Transliteration module and Lexical dictionaries. Our project has introduced the first ever parser and morphological analyzer for Sinhala language. Fig 2 shows the basic interface of our stand-alone MT system.

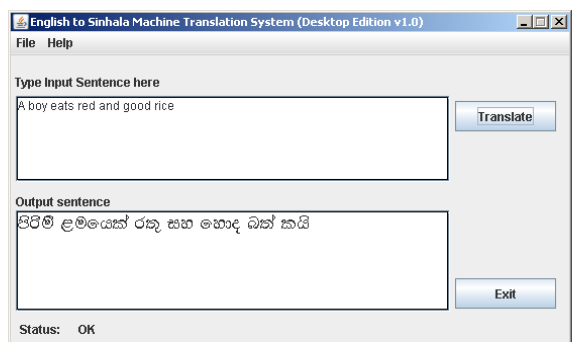

Fig. 2. Basic Interface for English to Sinhala MT system

This project has gone through various development stages. For instance, our English to Sinhala MT system has been improved to enable human-assisted translation [14]. This is a useful approach to improve the lexicon databases [13] and to identify limitations of the system, especially at the early developments. We have also improved our MT system with facilities to apply transliteration to handle pronouns and out-of-vocabulary problems during a translation process. This is essential for handling names of persons and cities, etc, for which there are corresponding Sinhala terms. In addition, we have gone beyond the traditional pre-editing and post-editing concepts in MT systems and introduced a new concept of intermediate-editing for MT system. This concept effectively handles ambiguities in semantic, pragmatic and multiword expressions before proceeding to Sinhala linguistic modules in the MT system.

Fig 3 shows a screen shot of the Intermediate editor. This editor provides facilities such as display of synonym and antonym and related words. The intermediate-editor is linked with both English and Sinhala dictionaries in the MT system. The process of intermediate-editing, before composing a Sinhala sentence, drastically reduces computational costs of running a Sinhala morphological analyzer and parser. In addition, the requirement for post editing [23] can be reduced by the process of intermediate editing. On the other hand, intermediate-editing can be used as a means of continuous capturing of human expertise for machine translation. This knowledge can be reused for subsequent translations. It should be noted that the knowledge used for pre-editing and post-editing cannot be readily captured by a MT system, as these processes are generally done outside an MT system. In contrast, intermediate-editing will be an integral part of the MT system, in which humans directly interact with the system.

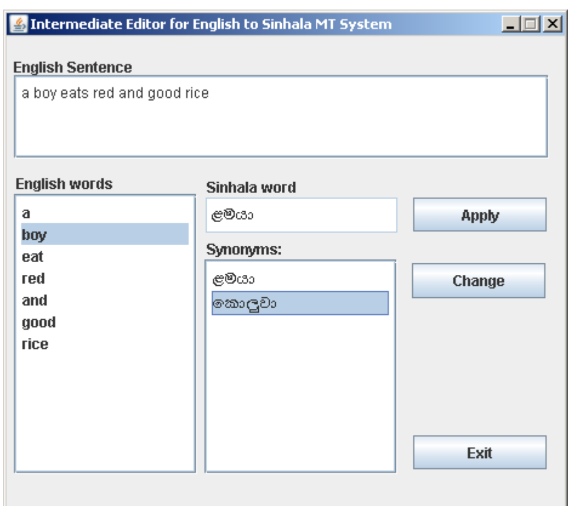

Fig 3: Intermediate editor for the English to Sinhala translation system

We have also improved our MT system with the report on the latest development to introduce web access for our MT system. The extension is primarily based on the use of Prolog sever pages. The architecture of the web-based English to Sinhala machine translation system is shown in Fig. 4.

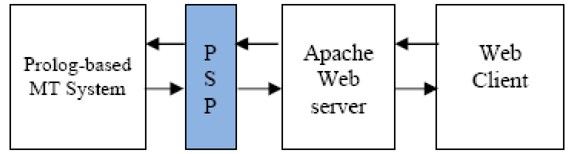

Fig 4. Architecture of the Web-based MT system

The web-based system contains four modules, namely; web client; Apache web server[27], PSP[26] module and Prolog based core translation system. Web browser is the user interface of the system. Apache web server handles all the web-based transactions of the system. PSP provides facilities to run Prolog-based system through the web. Prolog-based system is the core of the machine translation system. Through the PSP scripts, the core system reads input English sentence that comes from the web client. After the translation, the core MT system returns the output Sinhala sentence to the web client.

At This stage we need to test our machine translation system through the prototype working module. The present Parsers and Translator in the MT system have the following features; Handle Simple present tense, Handle determinants, Handle adjectives, Handle adverbs, Handle Compound subject ,Handle Compound adjectives, Handle Transliteration of proper nouns etc.

5. English to Sinhala Selected Text Translation System

English to Sinhala selected text translation system is designed by using web-based English to Sinhala MT System and java-based text selection tool. Web-based English to Sinhala translation system is the core system of the Machine translation system. The Java-based text selection tool is designed to translate English texts in to Sinhala by the help of web based English to Sinhala MT system. This selected text translation tool is an interface of the web based translation system. If the system is running, a translation tool appears in the system tray. To translate the selected text, you need copy it and click the translation icon on the system tray. Then text selection tool reads the clipboards and send the text into web based English to Sinhala MT system. Actually it reduces the access time of the system. Selected text translation tool is developed by using java and it is a simple java application. This java application accesses the English to Sinhala MT system through the internet.

6. How System works

In this section we describe, how the translation system works for a given input sentence. For example, suppose that the system accepts the input sentence: “A boy eats red rice for his lunch”. Then the English Morphological analyzer identifies each word and returns the lexicon information of each word through the following Prolog predicates.

eng_detm([e1000001], id, 'a').
eng_noun([e1000006], td, sg, ma, sb, 'boy').
eng_verb([e1000009], sp, 'eats').
eng_verb([e1000014], pt, 'red').
eng_verb([e1000014], pp, 'red').
eng_adjv([e1000008], p, 'red').
eng_noun([e1000013], td, sg, no, sb, 'rice').
eng_prep([e1000027], v5, 'for').
eng_noun([e1000029], td, uc, ma, ob, 'his').
eng_noun([e1000028], td, uc, no, sb, 'lunch').

Note that eng_detm/3, eng_noun/6, eng_verb/3, eng_adjv/3 and eng_prep/3 prolog predicates are used to store English lexical information such as determination, noun, verb, adjective and preposition respectively. Furthermore, English morphological analyzer provides all the grammatical information for each word. For example, the English morphological analyzer identifies the word 'red' as an adjectival form, and past tense and past participle of the verb 'read'.

After that, the English parser reads the original English sentence together with the output of the Morphological analyzer. After this analysis, the parser returns the following information

eng_sentence_type(simple,sp).
eng_sen_verb([e1000009]).
eng_sen_complement([e1000027, e1000029, e1000028,
e1000008, e1000013]).
eng_sen_subject([e1000001, e1000006]).
eng_sen_ekeys([e1000001,..]).

Rest of the information is subject, verb and complement of the English sentence. Eng_sen_ekeys/1 is the key list that assigns for the each word for the sentence. These keys (Tokenized ID for English words) are used for further operation for the translation system.

Then tokenized IDs of English words are forwarded to the translator. The translator identifies the Sinhala base word for each English word in the sentence, with the help of the bilingual dictionary. The output of the translator is as follows.

estrwords(1001, e1000001, s1000000, dt).
estrwords(1002, e1000006, s1000014, na).
estrwords(1003, e1000009, s1000011, vb).
estrwords(1004, e1000027, s1000023, pp).
estrwords(1005, e1000029, s1000025, na).
estrwords(1006, e1000028, s1000024, na).
estrwords(1007, e1000008, s1000006, aj).
estrwords(1008, e1000013, s1000015, na).

estrwords/4 prolog predicate stores bilingual information for the each word. It stores the tokenized id, English base word key, Sinhala based word key and word type. After this, the Sinhala morphological analyzer reads these words and generates appropriate Sinhala words with all the grammatical information. Sinhala morphological analyzer plays a key role in handling Sinhala language specific features in the MT system.

When executing the Sinhala morphological analyzer, the system uses subject, verb and object based translation to generate the appropriate Sinhala words. This is mainly because of the effect of the determination and proposition in the sentence is different from these two languages and Sinhala sentence has strong subject verb relationship than English (Example: I eat rice 'මම බත් කමි', we eat rice 'අපි බත් කමු') Furthermore, English determinations come with a separate word which is not so in Sinhala.(Sinhala Noun shows person, number, case, sex, live, direct/indirect and case forms for word inflection) Some prepositions work seperately and some affect only the case of a noun (generally, 'to boy' is translate as 'ළමයාට' that affects only for the case). The output of the Sinhala Morphological analyzer is as follows.

snoun([s1000014], td, sg, ma, li, id, v1,'msrsus <ufhla').
sin_sub_info([s1000014]).
sin_sub_word(['msrsus <ufhla', []]).
sin_fverb([s1000011], td, sg, pr,'lhs').
sin_veb_info([s1000011]).
sin_veb_word(['lhs', []]).
snoun([s1000025], td, uc, ma, li, dr, v2,'Tyqf.a').
snoun([s1000024], td, uc, no, nl, dr, v5,'osjd wdydrh ioyd').
sin_adjv([s1000006],'r;=').
snoun([s1000015], td, sg, no, nl, dr, v1,'n;a').
sin_cmp_info([s1000025, s1000024, s1000006, s1000015]).
sin_cmp_word([Tyqf.a', osjd wdydrh ioyd, r;=, n;a, []]).

Finally, the Sinhala parser composes the corresponding Sinhala sentence 'පිරිමි ළමයෙක් ඔහුගේ දිවා ආහාරය සඳහා රතු බත් කයි'. Figure 5 shows output of the web-based English to Sinhala machine translation system.

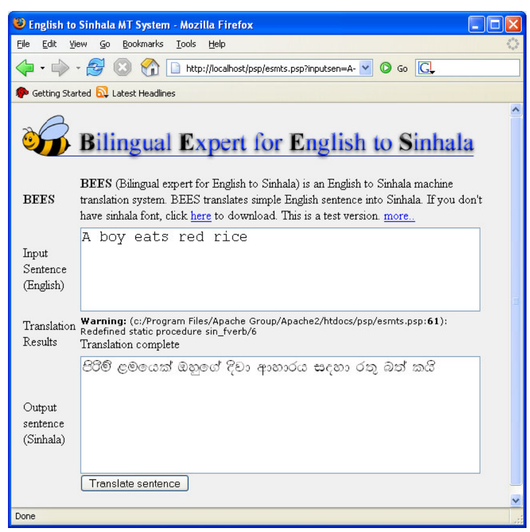

Fig. 5: web-based English to Sinhala machine translation system.

5. Conclusion and Further Works

This paper has reported on the ongoing project on English to Sinhala machine translation system and its current extension to provide selected text translation capabilities through web-based English to Sinhala Machine translation system. We have briefly described the developments of the previous work of this project by giving an emphasis on the modules of the core of our MT system. In this sense, we have reported on the first ever Sinhala Parser, Morphological analyzer, Intermediate editor and Human Assisted translator for English to Sinhala machine translation. More importantly we reported on the expansion to the core system thereby enabling the access to our MT system to a wider audience. This system will be useful for the general public, and students and teachers in particular. Further, work of this project has many directions. A system implemented with a powerful English parser and the WordNet lexicon database will be yet another further work of this project. In addition, the system will be improved to handle more complex English sentences.

References

[1] Askhar B, Chaitanya V, Sangal R, “Natural Language Processing:
A Paninian Perspective”, Prentice Hall of India, New Delhi, India, 1995.

[2] Bharathi A, Chaitanya V, Kulkarni A. P, Sangal R., “Anusaaraka: Overcoming language barrier in India”, to appear in "Anuvad: Approaches to Translation", Rukmini Bhaya Nair, (editor), Sage, New Delhi, 2001.

[3] Sinha R.M.K, Jain A., “AnglaHindi: an English to Hindi machine-aided translation system”, MT Summit IX, New Orleans, USA, 23-27 September 2003; pp.494-497.

[4] Sinha R.M.K, “Integrating CAT and MT in AnglaBhart-II architecture”, 10th EAMT conference, May. 2005, pp. 235-244.

[5] Durgesh R., “Machine Translation in India: A Brief Survey”, National Centre for Software Technology, Mumbai, India. http://www.elda.org/en/proj/scalla/scalla2001/scalla2001Rao.pdf

[6] Mantra: web based Machine assisted translation tool for English to Hindi, http://www.cdac.in/html/aai/mantra.asp

[7] Shakti: A web based Translation system, http://shakti.iiit.net

[8] MaTra : English to Hindi MT system, http://202.141.152.9/ matra/index.jsp

[9] UNL Based MT System http://www.cfilt.iitb.ac.in /machinetranslation/eng-hindi-mt

[10] P. Dirix, I. Schuurman, V. Vandeghinste, “METIS-II: example based machine translation using monolingual corpora - system description”, MT Summit X, Phuket, Thailand, September, Proceedings of Second Workshop on Example-Based Machine Translation; 2005, pp.43-50.

[11] J. Krzysztof, F. Gralinski, G. Krynicki: “POLENG - adjusting a rulebased Polish-English machine translation system by means of text corpus analysis”, 5th EAMT Workshop "Harvesting existing resources", Ljubljana, Slovenia, pp. 47-50. 2000.

[12] B. Xu, Z.B. Chen, W. Wei, W.Pan, Z.D.Yang, “Phrase-based statistical machine translation for MANOS system”. MT Summit X, Phuket, Thailand, September 13-15, Conference Proceedings: the tenth Machine Translation Summit: invited paper; 2005, pp.23-26.

[13] Hettige B., Karunananda A. S., “A Parser for Sinhala Language – First Step Towards English to Sihala Machine Translation”, International Conference on Industrial and Information Systems (ICIIS2006), IEEE, Peradeniya, Sri Lanka, 8-11 Aug 2006, pp 583-587.

[14] Hettige B., Karunananda A. S., “A Morphological analyzer to enable English to Sinhala Machine Translation”, Proceedings of the 2nd International Conference on Information and Automation (ICIA2006), Colombo, Sri Lanka, 2006 pp. 21-26.

[15] Hettige B., Karunananda A. S., "First Sinhala chatbot in action", Proceedings of the 3rd Annual Sessions of Sri Lanka Association for Artificial Intelligence(SLAAI), University of Moratuwa, 2006.

[16] Hettige B., Karunananda A. S., “Transliteration System for English to Sinhala Machine Translation”, Industrial and Information Systems 2007, (ICIIS 2007)International Conference on Volume, Issue, 9-11 Aug. 2007, pp 209 – 214.

[17]Hettige B., Karunananda A. S., “Developing Lexicon Databases for English to Sinhala Machine Translation”, Industrial and Information Systems 2007, (ICIIS 2007)International Conference on Volume, Issue, 9-11 Aug. 2007, pp 215 – 220.

[18] B. Hettige, A. S. Karunananda, “Using Human Assisted Machine Translation to overcome language barrier in Sri Lanka”, proceedings of 4th Annual session of Sri Lanka Association for Artificial Intelligence, University of Moratuwa, 2007.

[19] Hettige B., Karunananda A. S., “Intermediate-Editing Approach for English to Sinhala Machine Translation”, Proceedings of the 3rd International Conference on Information and Automation (ICIAFS07), Melborn, Australia, 2007

[20]Hettige B., Karunananda A. S., "Human Interactive NLP System for English to Sinhala Machine Translation", paper submitted to Industrial and Information Systems 2008, (ICIIS 2008) International Conference, India, 2008..

[21] Hettige B., Karunananda A. S., "Web-based English-Sinhala translator in action", paper submitted to 4th International Conference on Information and Automation (ICIAFS07), Colombo , Sri Lanka 2008

[22] A. M. Gunasekara, A Comprehensive Grammar of the Sinhalese Language", Asian Educational Services, New Delhi, Madras, India., 1999.

[23] Jeff L., Christopher H, "Toward the development of a post-editing module for Machine Translation raw output", Presented at the Third International Controlled Language Applications Workshop (CLAW2000), Washington, 2000.

[24]wikipedia, the free encyclopedia, http://en.wikipedia.org.

[25] SWI-PROLOG home page; http://www.swi-prolog.org

[26] Prolog server pages: http://193.226.6.174/roedunet2003/ site/conference/papers/SUCIU_A-Prolog_Server_Pages.pdf

[27] Apache web server: Apache software foundation,http://www.apache.org/

[28] S. Karunarathna, Sinahala Viharanaya, Washana prakasakayo, Dankotuwa, Sri Lanka., 2004.

[29] W. S. Karunathilaka, Sinahala Basha Viyakaranaya, M.D. Gunaseena & Company, Clolombo 11, Sri Lanka.,2003.

[30] Toshio Y, "The EDR electronic dictionary", Communications of the ACM, Volume 38, Issue 11 (Nov. 1995), pp. 42 – 44.

Advising Farmers on Crop Selecting Using Expert Systems

M. M. Anuradha Premachandra and Uditha Ratnayake
Department of Electrical and Computer Engineering,
The Open University of Sri Lanka
E-mail: anu.tinko@gmail.com

Abstract- Agriculture and plantation is an important and interesting research area everywhere in the world and Sri Lanka is no exception. Nowadays available land area for a plantation is becoming scarce. This scarce resource is frequently wasted through our bad practices and improper management. Cultivation is a more economical but complex process. Selecting and maintaining suitable crops for the maximum profit involves a sequence of tasks. These tasks and the whole process need a lot of expert knowledge and experience. But unfortunately, people having this type of knowledge are very limited. Their assistance is not available when the person who is going to cultivate needs it.

We propose a knowledge-based approach to land evaluation for the selection of suitable agricultural crops - Crop Advisor."Crop Advisor" is a Knowledge-based Decision Support System (KBSS) for crop selection. The expert system is powered primarily by human knowledge collected from crop experts. It also considers economic feasibility of raising a crop by taking market price, cost of production, access to market and yield levels. The "Crop Advisor" expert system then suggests with consultation with farmer (through a graphical user interface) a suitable agricultural crop that can be grown in a land unit with reasoning.

Key Words: *Expert system, Inference Engine, Knowledge Base*

1. Introduction

Agriculture plays a major role in our country's economy. Approximately 21% of the gross domestic product and 23% of the total export earnings are derived from agriculture. The crop subsector has a large number of small farmers on 1.8 million small holdings. Nearly half of the holdings are less than 0.5 ha.

But unfortunately, present farm family income is comparatively low. Hence farming is becoming a non attractive sector in the economy. The yields of most crops have remained stagnant for over a decade. Some lands are uncultivated. Food imports are increasing. Crops cultivated in marginal lands give very low yields which should be avoided. This project indicates new approaches in order to transform the present low productivity farming into more productive farming systems with the goal of significantly increasing the farm family incomes much above the poverty line.

This system contains information about distribution of a number of crops, considering soil, biological, social and economic considerations.

Currently, when the farmer wants to cultivate his land he wants to get an appointment for the necessary information. Without this correct information, sometimes farmers will be lost at the end. This can happen in many ways such as,

- Agricultural instructors contain little knowledge,
- Instructions are not available at the correct time.

In order to alleviate this problem, our project identified a powerful tool with extensive potential in agriculture.

The goal of this project is to increase the yield quality of agricultural production through the introduction of expert systems to enhance crop management and farm support.

The rest of the paper is organized as follows. Section 2 describes about the background information related to the project. Section 3 is on the design aspects considered in the development, and section 4 reports on the implementation of the proposed system incorporating the expert system. The evaluation process carried out on this research, the evaluation results and their explanations are described in Section 5.

2. Background Information

2. 1. Climate and Rainfall

Most parts of Sri Lanka are hot and humid. Despite the relatively small size of the country, there is a considerable variation in climate over time and space.

The rainfall pattern in Sri Lanka is bimodal with two periods of monsoonal precipitation resulting in two distinct cultivation seasons. The major cultivation season, called Maha, is from October to February. The precipitation during this season comes from the northeast monsoon of October-December. The harvesting period of crops cultivated in Maha is at the end of January, usually a dry period. The second crop season is called Yala and it extends from May to July. The rains in this season come from the southwest monsoon during mid-April to June. The remaining months of the year are dry, and almost no cultivation occurs during this period.

The annual average rainfall varies from below 1000 mm (39") over a small region in the semi-arid parts of the north-west and south-east of the island to over 5000 mm (197") at a few places on the south-western slopes of the central hills.

There are four rainfall seasons during the year. These are:

1. The south-west monsoon period (May to September)
2. The inter-monsoon period following the south-west monsoon (October to November)
3. The north-east monsoon period (December to February)
4. The inter-monsoon period following the north-east monsoon (March to April)

2.2 Temperature

There is little seasonal variation of temperature in Sri Lanka. It depends on the elevation ranges. The temperature variation through the year is low with the mean ranging between 21.1 and 31.7°C.

2.3. Zones and Regions

With in the two main rainy seasons, rainfall distribution across the country is determined by topography. The whole island benefits from the northeast monsoon. The southwest monsoon is intercepted by the central mountains resulting in 2000-5000 ml of rain per year in the highlands and southwest part of the island. This is the "wet" zone of the country covering 1.53 million ha. The "intermediate" zone covering 4.17 million ha receives 2000-2250 ml of rainfall per year, and is hotter than the "wet" zone. The dry zone receives only 900-1000 ml of rain, with the highest temperatures ranging between 28 and 30°C. In terms of "Wet Zone" in the southwestern region, it includes central hill country, and "Dry Zone" covering predominantly, northern and eastern part of the country, being separated by an "Intermediate zone," skirting the central hills except in the south and the west

2.4.Cultivation Time

Most vegetables are grown about equally in both Maha and Yala, except for some up-country vegetables, such as onion, which are mainly grown in the Yala season, and tropical vegetables, such as chili, which are mainly grown in the Maha. Total production of vegetables in Maha is slightly higher than that in Yala. No significant difference in the yield of most vegetables grown in Maha and Yala was observed.

2.5. Soils of Sri Lanka

The soils of Sri Lanka have been classified at Great Group level for the whole country and Series level for some parts.

The chemical fertility of Wet Zone soils is poor because these soils have been extensively leached due to high rainfall. The base saturation of the Dry Zone soils remains at a higher range.

Apart from chemical soil categories, basically soil can be categorized in to three , according to the soil texture.

- Sand
- Silt
- Clay

When considering their mixtures there are 11 sub categories.

- Clay
- Silty clay
- Silty clay loam
- Silty loam
- Silt
- Loam
- Sandy loam
- Sand
- Sandy clay loam
- Clay loam

2. 6. Introduction about Expert System

An expert system is a computer program designed to simulate the problem-solving behavior of a human who is an expert in a narrow domain or discipline.

An expert system is normally composed of a knowledge base (information, heuristics, etc.), inference engine (analyzes the knowledge base), and the end user interface (accepting inputs, generating outputs).

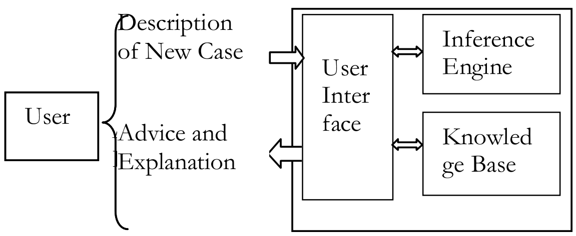

Figure 2.1: Major parts of an expert system

3. System Overview

The Expert System for Agriculture uses a rule base which uses the experience and knowledge of a human expert captured in the form of IF-THEN rules and facts which are used to solve problems by answering questions. Questions can be typed at a keyboard attached to a computer.

Yield of the particular crop depends on several factors; such as rainfall pattern, land elevation, soil texture, soil pH, soil drainage, temperature. Experts related to the Department of Agriculture have identified some categories of above mentioned factors.

According to the rainfall pattern, Sri Lanka has basically three zones. They are,

- Wet Zone
- Dry Zone
- Intermediate Zone

When considering the elevation and temperature, all these zones can be categorized in to another three groups.

- Up Country
- Mid Country
- Low Country

According to these factors there are special regions called Agro ecological zones. Sri Lanka has 22 Agro ecological zones.

- WU_1, WU_2, WU_3, WM_1, WM_2, WM_3, WL_1, WL_2, WL_3,
- IU_1, IU_2, IU_3, IM_1, IM_2, IM_3, IL_1, IL_2, IL_3,
- DL_1, DL_2, $DL_{3\&4}$, DL_5

WU_1 – Wet zone up country
IM_1 – Intermediate zone mid country
DL_5 – Dry zone low country

Soil is very important to the crop cultivation. It depends on the land elevation and also soil texture. I have explained the majority of soil textures in Sri Lanka in chapter two.

Land elevation can be considered as high land or low land. These two factors make the soil as,

- Well Drain
- Poor Drain
- Imperfect

After studying the factors which affect the healthy growth and quantity of yield, we have identified some input information for the expert that the user wants to deal with the system. Selection of the crop depends on these parameters.

Mainly “Crop advisor” consists of three modules. These modules are designed for the user known information and for the user unknown information. These modules are

- Crop Selection
- Fertilizer Plan Selection
- Soil Selection

First two modules are for the user known information and the last module for the user unknown information.

3. 1. Input Parameters

After studying of the background information and most of the other information, we have identified that there are five input parameters to give the data to the expert. Those are

➢ Zone
- Wet Zone (WZ)
- Dry Zone (DZ)
- Intermediate Zone (IZ)

- Sub Zone
  - WU_1, WU_2, WU_3, WM_1, WM_2, WM_3, WL_1, WL_2, WL_3, IU_1, IU_2, IU_3, IM_1, IM_2, IM_3, IL_1, IL_2, IL_3, DL_1, DL_2, $DL_{3\&4}$, DL_5

- Soil & Drainage
  - Clay (C)
  - Silt clay (StC)
  - Silt clay loam (StCL)
  - Silt loam (StL)
  - Silt (St)
  - Loam (L)
  - Sandy loam (SL)
  - Sand (S)
  - Sandy clay loam (SCL)
  - Clay loam (CL)
  -

- Land Orientation

  - High Lands
  - Low Lands

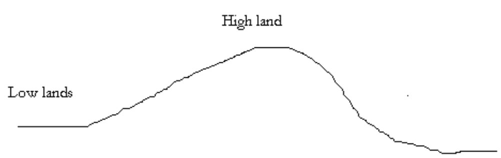

Figure 3. 1 – Land Orientation

- Crop Period

  - January – March (JM)
  - April – June (AJ)
  - July – September (JS)
  - October - December (OD)

If one doesn't know what his/her sub zone is, he/she can give the **district** and also **agrarian services area** as the input. Then the system will find the correct subzone accordingly, through the SQL data base.
If the soil type is not known, there is a module to select soil texture for user unknown information.

3.2. Design of the Knowledge Base

When designing a knowledge base we have studied many theories related to the knowledge representation.

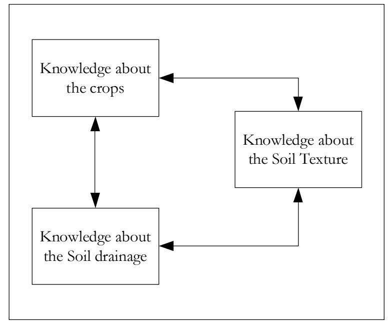

Figure 3. 2 – Knowledge Base

Then we have designed a rule based system to store the above knowledge, whose basic components are:(Shown in figure - 3.3, figure – 3.4, and figure – 3.5.)

Attributes :$X_1, X_2, \ldots, X_{n1}$
Conditions :$C_1, C_2, \ldots, C_{n2}$
Rules :$R_1, R_2, \ldots, R_{n3}$
Actions :$A_1, A_2, \ldots, A_{n4}$

We only need to execute an action when a rule containing it is fired. A rule is fired only when all of its conditions are satisfied. If there is a missing parameter, several rules have to be fired. To detect this we assigned a value to each condition and use it to keep track of exactly how many of the conditions in the rule are currently satisfied. Thus we only check to see if a rule is ready to fire when one of its conditions has become true. From our inference algorithm we can select the most corrective actions from the total evaluation points.

Here are mainly two types of rules.

- Main Rules
- Sub Rules

At the beginning, the system considers only main type rules. It contains five conditions. It involves making decisions about suitable crops according to the given input factors. Following figure shows the format of main rule.

R	A	C1	C2	C3	C4	C5
		X1	X2	X3	X4	X5
		3 Point	5 Points	7 Points	11 Points	19 Points

Figure 3. 3 – Format of Main Rule

we have assigned values for each condition. It is based on the priority.

C_1 = Zone
C_2 = Sub Zone
C_4 = Soil
C_5 = Soil Drainage
C_6 = Crop Period
A = Most suitable Crop/s
R = Rule Number

Expert systems have to handle incomplete inputs in decision making. CROP ADVISOR also has this common problem. Therefore, the knowledge base must contain knowledge to handle the incomplete inputs. Sub rules contain this type of knowledge. Sub rules involve decision making, only some parameters missing at the inputs.

R	A	C1	C2
		X1	X2

Figure 3. 4 – Format of Sub Rule

This type of rule can be used to two purposes.

It can be used to select the soil drainage, because it is not an input parameter. The system finds it using other parameters. Given below is the condition and action of this rule.

C_1 = Soil

C_2 = Land Orientation

A = Soil Drainage

the other purpose is to find the soil of the particular land, when it is missing in the input parameters. Using information about the sub zone and orientation of the land we can find the soil as an action.

C_1 = Sub Zone
C_2 = Land Orientation
A = Majority Soil

But some input parameters can not be found from other inputs such as sub zone, land elevation, and crop period. When a user misses this type of input, system will show all crops related to the other given inputs.

When rules are examined by the inference engine, actions are executed if the information supplied by the user satisfies the conditions in the rules.

Conditions are expressions involving attributes and the logical connective . For example,

Zone = wet and
Subzone = WM1 and
Soil = CL and
Drainage = Well
Crop period = JM

We do not consider the cultivation method under Rain fed, because under sub zone and crop period, system can identify the amount of rain for the particular sub zone.

Thus, a full example of a rule would be as the following.

Main Rule:

Most suitable Crop (xxxx) :- Zone(wet), Sub Zone(WM1), Soil(St), Soil Drainage(well), Crop Period (JM)

Sub Rule:

Soil Drainage (Well Drained) :- Soil(St), Land Orientation(High)

Majority Soil(C) :- Sub Zone (WM1), Land Orientation (High)

After the selection of the suitable crops, CROP ADVISOR may advise on fertilizers. When selecting a fertilizer plan for the selected crop, there is another type of rules needed to fire.

R	A	C1	C2	C3	C4
		X1	X2	X3	X4

Figure 3. 5 – Format of secondary knowledge base Rule

C_4 = Period
C_3 = Soil
C_2 = Sub Zone
C_1 = Crop
A = Most suitable fertilizer Plan

3.3. Working memory

- The contents of the working memory are constantly compared to the production rules.
- When the contents match the condition of a rule, whenever a condition is matched, it is added to certain points, and the rule contains the maximum points copy to the another place to fire and its action is executed.

– The system then fires the rules sequentially, within the working memory.

– More than one production rule may match the working memory.

3. 4. The Inference engine

Two methods of inferences are used often;forward chaining and backward chaining. In our project we present an inference engine which operates by the method of forward chaining.

In order to execute a rule based expert system using the method of forward chaining, it is merely needed to fire actions whenever they appear on the action list of a rule whose conditions are true. This involves assigning parameters to attributes, evaluating conditions, and checking to see if all of the conditions in the rule are satisfied. A general algorithm for this might be,

While parameters for attributes remains to be input
Read parameters an assign to attribute
Evaluate conditions
Fire rules whose conditions are satisfied

Several points about this require consideration. First, decide which rules are fired first. In here it is to fire the selected rules sequential.

There are several activities to do within an inference engine.

- Check input parameters against the conditions contain rules.
- Calculate the total points.
- Ask for missing parameters.
- Guess missing parameters using sub rules.
- Explanation against the inputs.
- Select most suitable crops.
- Give priority with compare of demand forecasting

4. Implementation

4. 1. User interfaces

The system has several interfaces to communicate with the user

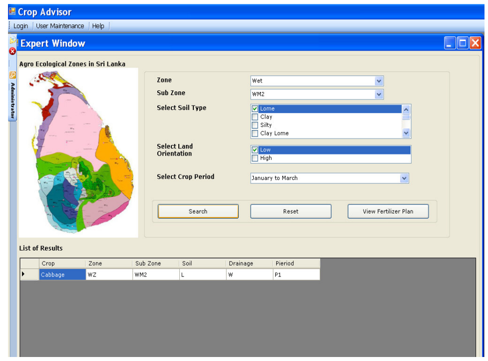

Figure4.1- Main interface of the system

This interface indicates all parameters. If a user misses some parameters,the CROP ADVISOR can identify those inputs from the given inputs.

Figure 4.2 – Interface to find the sub zone

4. 2. Implementation Tools

- Visual Studio 2005 C#.net - inference engine design and interfaces.
- Microsoft Access – Develop knowledge base.

5. Evaluation and conclusion

The main goal is to help the crop selecting task and this has been achieved. Experts at the Ganoruwa research center and the Department of Agriculture helped to analyze the system and to build a 90% accurate knowledge base.

Another goal is to develop a fertilizer plan for the particular crop which could not be achieved.

The system has been tested with about 100 people and farmers. Outputs were 80% accurate.

Since it is difficult to apply anything learnt from books directly in the real world we faced some operational problems.

Another lesson was about planning. In industry, plans can be made with 100% accuracy but going according to that plan is very much difficult. Practically a lot of problems come and they have to be changed accordingly.

5.1. Future Work

A system like this is some what new to the department of agriculture, and also to the farmers who are used to traditional methods. Therefore it has to be people; especially farmers who should be educated on this regard. This system can be developed further including some new features and improving the existing features.

The future work of this system can be described as followed. The responsible operator/ user should be able to use the system.

The targeted main users are farmers. Then they will be able to use this system. Therefore, first and foremost, the system must be user friendly with the option of selecting a familiar language to work.

Another extension is to develop web enable system with this knowledge base. It will give more facilities to the farmers.

Other than the input factors, there are various factors affecting the crop production,such as soil pH and temperature conditions. But in this stage we have not considered those factors and we are hoping to improve the knowledge base and also the whole system using these most of factors.

BIBLIOGRAPHY

[1] A guide to Expert System by Donald A. Waterman. ISBN 0-201-08313-2

[2] Artificial intelligence by Elaine Rich and Kevin Knight. ISBN 0-07-052263-4

[3] Field Drops of Ceylon by S. T. Senewiratne and R. R. Appadurai.

[4] The cultivation of Ceylon by W. R. C. Paul

[5] Administrative Report of the Director of Agriculture for 2004

[6] "Govi Sevana Niyamana Puhunuwa- Agro Techniques" Part Two-1999

[7] Agro Techniques for 21st century by Department of agriculture, Uva province.

[8] De Alwis KA and CR Panabokke (1972) Handbook of the soils of Sri Lanka. J. Soil Sci. Soc. Ceylon 2, 1-98

Using Agent technology to access the Semantic Web

Sanjaya Ratnayake, Ruvindee Rupasinghe, Anuruddha Ranatunga, Shalinda Adikari,
Sajayahan de Zoysa, Kamala Tennakoon, and Asoka Karunananda
Faculty of Information Technology, University of Moratuwa, Katubedda, Sri Lanka
Email:sanjurat@yahoo.com

Abstract- **Semantic web consists of heterogeneous sources of knowledge including texts, graphics, blogs, animations, audios, and videos. However, there have been limited researches conducted to present semantic web information in the form suitable to individuals. This paper reports on the design and implementation of 'Divon', a swarm of agents that emulate a user profile driven approach to present semantic web information in the forms suitable to individuals. Divon has been structured with four agents, namely, Message Agent, Query Handler Agent, Presentation Agent, and User Profiler Agent. The Message Agent plays a key role in guiding the search process and displaying the information in a suitable form for the user whereas User Profiler Agent creates individual user profiles according to individual preferences. Divon has been developed on JADE environment and can run on any computer in connection with an arbitrary search engine.**

1. Introduction

Internet has been founded by the American authorities in the early 90's as an infrastructure for the defense and scientific work [3]. However, after identifying its immense potential as a facilitator for a myriad of human activities it has grown on a massive scale. An essential result of this unprecedented growth is that the amount of information residing in the web has also grown to a huge amount. Nowadays, Internet has grown into a multifaceted source of knowledge and postulated what is known as the Semantic Web [13]. Within this complex web environment, it is not easy for the web users to locate a particular piece of information they need. The traditional solution for this problem was that search engines came into operation. However, the results returned by these search engines were too general and did not match the exact requirement of the user. Hence, the need for personalized web searching mechanisms became apparent. Many researchers have proposed Agent based solutions for information retrieval [9].

User modeling techniques together with machine learning have been generally used to build Intelligent Agents [5]. Starting from a small knowledge base, Agent enhances its knowledge base through the machine learning techniques. Furthermore, there are also some agents specially designed for the purpose of information retrieval [9]. In addition, swarm of agents or multi agent systems have been experimented with to facilitate information retrieval on the semantic web [2].

However, all these approaches have a serious limitation when it comes to the user modeling. That is the agent's knowledge about the user is mostly static. The reason for this is that the knowledge parameters which are considered by the agent remain static over time. As a result, the user profile available to the agent does not reflect the actual user requirement at a particular time. More importantly, none of these approaches have handled the need for presenting the retrieved information in a format suitable to respective individuals.

In responding to these issues, we have developed a swarm of agents, Divon, that can dynamically modify the user requirements and guide the searching and presentation of information accordingly. Divon is designed to be run totally at the client end and hence would not add any burden to the network traffic. Basically it would maintain a dynamic user profile. User's current context would be identified by the attributes like location, time, and searching habits etc. Divon has been developed on the Java Agent Development Environment (JADE), and can be run on any client computer.

The rest of the paper has been organized as follows. Section 2 briefly describes the current approaches to information retrieval on the Internet. The swarm intelligence based approach is described in section 3. Section 4 describes the design of the Divon. Implementation details of the Divon are described in section 5. Work flow order of the Divon is described in section 6. Finally, section 7 contains the conclusion.

2. Current Approaches To Information Retrieval

Over the last many decades, researchers have introduced various approaches to information retrieval on the Internet and the semantic web. Undoubtedly, the Search Engines are the most known and the very first technology for information retrieval on the Internet. Subsequently, Meta search engines [1], Distributed information retrieval systems [11], Agent-based information retrieval [20] have also been introduced by various researchers.

2.1 Search engines

Search engine is a web site that collects and organizes content from all over the Internet [17]-[19]. Those who wish to locate something would enter a query about what they would like to find and the engine provides links to content that matches what they want [15]. Among these, Web search engines are used to retrieve information from World Wide Web (WWW). Using a set of algorithms, search engines built different methods to find information that are required by users [14]. For instance, S. Asadi and H.R. Jamali (2004) have analyzed the shift in search engine development and identified eight aspects, namely, general aspects of web search, query formulation, search process, reference feedback, ranking, and retrieval of models, information filtering, clustering, and selection process [18]. According to this study, during the period 2003-2004, the general aspects of web search have been reduced, whereas information filtering and search process have been increased. This shows that users are more interested in information retrieval in a more personalized manner. In line with these requirements, in today's context, even the general purpose search engines such as Yahoo and Google have improved with basic natural language processing abilities.

Meta search engine operate based on the virtual databases. However, it does not compile a physical database of the web. As an alternative first they take a user request and pass it to several other various databases and then compile the results in a standardized manner based on a specific algorithm [7]. All Search engines have been primarily implemented as traditional software systems.

Distributed information retrieval techniques have been used to address the problem of long duration of the update intervals. The conventional architectures are centralized, nevertheless these techniques are capable of shortening the update intervals [22]. Distributed information retrieval systems can be created with the contribution of multiple local Meta search engines that corporate with each other [16].

2.2 Semantic Web approach

Another way of distributed information retrieval is the semantic web method. Semantic web provides the means to use metadata that help determine which documents are relevant. Simply semantic web means data with meaning. Semantic web is a technology which describes things in a way that computer's applications can understand [6]. Unlike normal web search engines semantic web does not consider links between pages; it is based on the relationship between things and properties of things. Semantic web approach considers not only the total amount of data but also the differences among the local metadata vocabularies [21].

The Semantic web is governed by a three step framework, where the first step is *resource selection*, in which for any given query the full Web has to be queried. The second step is *query reformulation and ontology alignment*, which deals with the differences in the vocabularies used by the user and the selected information resources. *Aggregation and data* fusion is the third step, which integrates the ranked results from the individual resources into a single list. The three step framework of semantic web has generally been implemented as a piece of traditional software that does not use any intelligence for information retrieval.

2.3 Agent-based information retrieval

There are verities of research projects that have adapted the agent technology for information retrieval [12]–[4]–[20]. It is quite natural to use agent technology for information retrieval, since agents are autonomous and can work to a large extent without user intervention all the time. These agent solutions work as a front end for standard search engines. Some of these agents also enable personalized searching on the Internet [10]. However, pre-structured agents with static user profile may restrict the effective, personalized searching on the Internet. In particular, effective agent must be sensitive not only to user profile, but also to the capability and the choice of search engines, resources on the semantic web, presentation format of resources, etc. Obviously, all these aspects are too much to handle by a predefined single agent.

3. Approach to Using swarm of agents

As we discussed, the existing approaches to information retrieval have been primarily based on technologies that improve performance of search engine. Further, agent-based approach to information retrieval has also faced with serious limitations as per personalizing the search process through the dimensions of user profile, resources, location, nature of the Internet connectivity, time of the day, etc. whereas the agent technology would be one of the most effective approaches to information retrieval; it is too ambitious for a single agent to assign all the tasks related to search process.

Therefore, we propose an approach to effective searching on the semantic web, which composes a group of agents (swarm of agents). Using this approach, the communication among agents produces the best result and allows the evolution of the performance of the swarm of agents. More importantly, this approach ensures effective information retrieval through the collaboration of four agents, namely, Message Agent, Query Handler Agent, User Profiler Agent, and Presentation Agent. The overall approach comes out as intelligent software, called Divon, which can run on a client machine.

Key features offered by Divon are autonomous, personalized, reactive, adoptable, proactive, and collaborative and facilitate effective information retrieval on the Semantic web. Divon is necessarily autonomous as it runs all the time with minimum user intervention. In particular, agents in the swarm communicate without bothering the user, yet come up with solutions through their interaction. Altogether the system would carry the following main benefits for the end user.

1. Highly personalized content
2. Sensitiveness to the user's current context
3. Adaptability to the changes in information requirements of the user
4. Proactive search on behalf of the user
5. Reduced search time
6. Enhanced quality of the search results

High personalized content is supported by the User Profile Agent. Divon identifies user by the windows authentication and develops the profile based on criteria such as location, time, search query, viewed sites, time spend on particular sites, and repeat views. Based on the profile, user is categorized in to the relevant domain by the User Profiler Agent. By default Divon consists of ten domains, technology, entertainment, business, news, information, online transaction, education, research, children, and general. Divon is capable of adding new domains while killing isolated domains.

Divon is reactive to be sensitive to the user's current context. As such, Divon is capable of representing semantic web in the form of blogs, news, videos, books, web, local sources, images, and general. As Divon is continuously updating user profile, the system is reactive to the changes in information requirements of the user. Due to its reactive nature, although the Divon is autonomous, the user can access the Divon at any time.

Adaptable nature of Divon enables us to install Divon for any client machine that uses an arbitrary search engine. Divon is also adaptable of incorporating new requirements, the changes of previous interests, etc., of a user into the system.

Proactive search on behalf of the user takes place when the user logs in to the system. The user can view the previously visited, preferred sites before he starts a search. Divon reduces the searching time at the user level by proactive search, query optimization, result filtering and ranking. At the process level, search is improved by caching and accessing local data sources. Divon enhances the quality of search results by initially accessing local data sources. If the search is done outside local sources, it still filters results based on the user profile. When swarm is not busy it analyzes the user details and patterns. So with the same time the user will be classified into several domains and data representation styles.

Divon produces results through the collaboration of four Agents. The collaboration has addressed the problem of not being able to handle the entire workload of effective searching by a single Agent. This is the key contribution in our research work.

4. Design of Divon

Fundamentally, Divon can be seen as a swarm of agents that deal with messaging, request handling, and resource allocation. Fig. 1 shows the functional overview among these processors. In order to implement the concept, we have defined Query Handler Agent that has access to semantic web and the repository of URLs which are collected during execution. The Message Agent is responsible for overall communication among the user and the agents in the system. The User Profiler Agent plays a central role by communicating with all other agents whereas Presentation Agent is responsible for dynamic data representation.

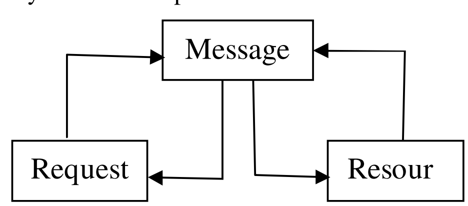

Fig. 1. Functional overview-Divon deals with accepting requests displaying messages and resource/agent allocation to perform the request

The top level architecture of the Divon is shown in Fig. 2. It comprises four agents, Local repository, and the connection to the Semantic Web. Next we shall briefly discuss the role of four agents and Local Repository in Divon.

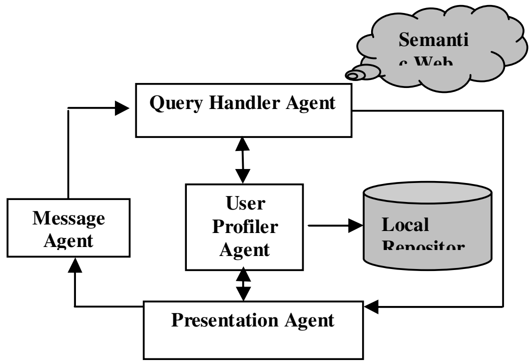

Fig. 2. High level modular architecture of Divon – Four Agents with Local repository.

4.1 Message Agent

Message Agent is responsible for handling the messages over the system. This works as the communicator between all four agents of the system and the user. Blocking the messages, prioritizing them, storing the messages, and invoking the right agent at the right time are major functions of the agent. At the search request, the Message Agent checks the status of the system agents. If the agents are in process Message Agent will store the data in its local data sources and later utilize swarm of agents when they are free. Message Agent is also capable of providing advanced warnings. For example if the user tries to do a bulky download, first the Message Agent analyzes the network traffic and if the traffic is high it informs the user about the alternatives available.

4.2 User Profiler Agent

User profiler Agent handles the user specific data for each individual users of Divon. The complex architecture of the agent monitors the complex behaviors of each user and identifies their key features. Based on those features users are categorized into different domains by the User Profiler Agent. Each domain consists of domain specific keywords and those are matched with the search query. In keyword identification the agent works with a complex algorithm which maps the parameters such as, the web sites each user visits, the number of clicks performed and the time spent on them. The keywords identified by the agent are more personalized to the user and are able to map the user behavior in an appropriate manner. More importantly the changes of the user behavior can be identified by the User Profiler Agent and the user's profile can be customized accordingly. When the user is not using the machine over a period more than six months , the User Profiler deletes the profile of the specific user. Once the agent identifies the behaviors of the user, the agent itself searches through the web to retrieve the related URLs for the user automatically. Every improvement done to the users are stored in the repository.

TABLE I
KEY WORDS WITH SIMILAR MEANING

Key Word	Meaning	Domain
SOAP	Protocol	Technology
	Cleaning Utility	General
MOUSE	Animal	Animal
	Computer peripheral	Computer
OFFICE	Working Environment	Job
	MS Office Software	Computer
BOW	To bend forward at the waist in respect	Culture
	Front of the ship	Traveling
	Weapon which shoots arrows	History

4.3 Query Handler Agent

Once the Message Agent sends the query to the Query Handler Agent, it tries to personalize the query according to the feedback given by the User Profiler Agent. The user's domain has a huge importance here as different users can use the same word to interpret different meanings. Table I depicts such words and the domains that they can be categorized.

Thus the keywords given by the User Profiler are prioritized by the Query Handler and mapped with the given query to create a meaningful implementation to it. More importantly with the experiences gathered by the Query Handler Agent, it can improve its ability of creating the suitable query. Query optimization is also done within the Query Handler Agent to make it more specific to search the given query. The optimized query is sent to the search engine by the Query Handler Agent in order to retrieve the search results. The Query Handler Agent communicates with the Presentation Agent for further processing of the query.

4.4 Presentation Agent

Presentation Agent is responsible for representing data in an understandable manner to the user. More importantly the Presentation Agent can analyze, filter, sort, and rank the search results according to the user preferences. In case of a scenario where the given results do not match the user profile, the Presentation Agent communicates with the Query Handler Agent through the Message Agent and requests Query handler to process again. So this goes in a cyclic manner until the best search results are retrieved. The direct communication between the User Profiler Agent makes the Presentation Agent aware of the user preferences, so that the Presentation Agent can present the results in a user preferred format, for example in pictures, audios, videos, blogs, images, news, books etc.

4.5 Local Repository

Local repository stores all the information transmitted through the system. The repository is updated with the feedback of the User Profiler Agent. So in case of a failure the local repository can be used to recover the system from the failed state.

5. Divon Implementation

Divon has been implemented using JADE, which provides the critical features of agent oriented systems such as a distributed, fault tolerant, highly secured, and semantic framework [8]. The Google search engine has been used as the search engine for development purposes. The middleware technologies of the agents such as agent management, agent communication, and agent software interaction have been implemented according to the specifications of The Foundation for Intelligent, Physical Agents (FIPA) [8]. The basic functionalities of the agents such as birth, registration, location, communication, migration, and operation of agents were implemented according to the FIPA specifications in agent management.

The main function of Divon, which is the agent communication, was done using Agent Communication Language (ACL). The communication among agents is an asynchronous message communication and being autonomous and proactive are the key features of the agent. It has its own thread of execution and has the knowledge to be invoked at the right time and to perform the correct action.

The actual tasks that the agents should perform are defined within the behaviors of the agents. Therefore each agent who has been described above have their own behaviors. The behaviors can be invoked concurrently. Table II describes the dedicated behaviors of each and every agent.

TABLE II
THE BEHAVIORS OF AGENTS

Agent	Behaviors
User Profiler	createDomain , deleteDomain, addKeywords, matchUser, createUser, getDomain, getKeyword, createUser, dropUser
Query Handler	getQuery, optimize, connectSemanticWeb
Presentation	getPreferences, presentation, searchResults, sortResults, rankResults, checkValidity, askQueryHandler
Message	display, checkBuzyUsers

The Message Agent implements cyclic behaviors in order to achieve its main feature, which is the continuous checking for resources. Here the agent communication is done in an asynchronous manner in which all the messages passed over the system are stored in a queue and for selecting the messages from the message queue, the Message Agent has to check the status of the other agents periodically. Fig. 3. depicts the state and transmission diagram of Message agent and its communication among other agents.

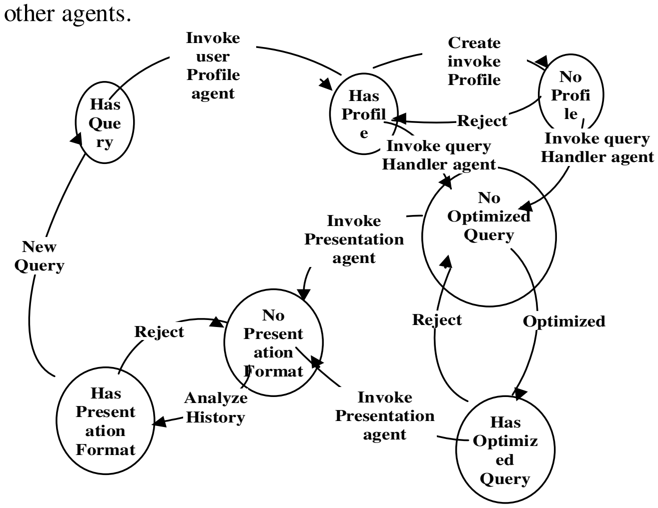

Fig. 3. state and transition diagram of the Message Agent

The dynamic data representation is handled by the Presentation Agent using its presentation behavior. Presentation Agent will communicate with the User Profiler Agent and negotiate about the user preferred data format. Once the User Profiler Agent responds, the Presentation Agent starts its process accordingly. Meanwhile checkValidity behavior checks the relevance of the search results to the user profile and if there is any mismatch Presentation Agent's askQueryHandler behavior communicates with the Query Handler Agent through Message Agent.

User Profiler agent is able to categorize users based on his location. Here it considers the user's location specific requirements for a more accurate searching strategy. createDomain behavior in User Profiler Agent is able to generate domains for specific users. The keywords relevant to those domains are categorized and listed by addKeywords behavior. Here createUser behavior creates new users whereas dropUser deletes frequently not logged in users.

Query Handler Agent is able to expand and optimize the search query with specific keywords using its expand behavior. The data retrieval from the semantic web is performed by the connectSemanticWeb behaviors. The query optimization is based on the combination of different searching strategies, namely; click history based personalization, ontology based personalization, personalization based on long term search history and Group Based Personalization.

6. Divon Functionalities

When a user logs in, Divon presents the user with the result of the most recent search session. This information and preferred sites are filtered by the Message Agent based on user's domain and recent searches. For example, a researcher who searched on "Agent Theory" recently will be getting some agent theory related links at the time of logging. The links consist of sites that a researcher has not visited previously but related to the "Agent Theory".

At the time of browsing the user may also experience the effect of Divon. For example, lets assume a house wife and a researcher have explored the search queries in Table III. These are based on past searches and now stored in the Local Repository.

TABLE III
QUERIES BY DIFFERENT USERS

House Wife	Researcher
Washing machine powder (images)	Protocols(tutorials)
Sun Light (images)	Computer (tutorials)
How to clean clothes(video)	XML (tutorial)
Laundries in Texas(images)	Remote Procedure (thesis)

Now according to the information available, Divon will classify the researcher who prefers to search computer related tutorials into "Computer" category and house wife, who is interested in images into "General" category. Let us assume that both make a search on "SOAP". At this point, Divon generates different results for the house wife and researcher, because they belong to different categories. Furthermore the presentation criterion also changes according to the user interest, by providing the researcher more tutorials and the house wife more images. Table IV shows the output generated by Divon for the above queries. The researcher will be getting the results from tutorials and thesis and house wife will be getting the results from general web or as images.

TABLE IV
ANSWERS BY DIVON

House Wife	Researcher
Washing Soap(images)	Simple Object Access Protocol (SOAP) 1.1 (tutorial)
Effect of washing hands with soap(images)	SOAP - Wikipedia, the free encyclopedia (tutorial)
SOAP POWER(images)	SOAP Tutorial
Getting Clothes Clean(video)	Simple Object Access Protocol – SOAP (thesis)

Let's consider another situation where the user retrieves different search queries according to the user domains they belong. Assume two users of the Divon who belong to computer and animal domains respectively are searching for the term "mouse". If they use a normal search strategy the first three search results they would obtained are listed in table V.

TABLE V
QUERIES BY DIFFERENT USERS IN A GENERAL SEARCH

Index	Normal serach engine
1	Mouse (computing) - Wikipedia, the free encyclopedia en.wikipedia.org
2	Apple – Mighty Mouse www.apple.com/mightymouse
3	What is mouse? - a definition from Whatis.com Searchexchange.techtarget.com

Nevertheless when they use the personalized strategy they can directly go for their desired solution quickly. As depicted in table VI, through Divon, users interested in computers get computer related information while users interested in animals can view animal related data. Therefore the personalized search could provide users relevant search results with a high precedence.

TABLE VI
QUERIES RETRIEVED BY DIFFERENT DOMAIN USERS

Index	For a computer domain user	For an animal domain user
1	Mouse (computing) - Wikipedia, the free encyclopedia en.wikipedia.org/wiki/Computer_mouse	Pet Mouse, Fancy Mice, Pet Mice, Mus musculus animal-world.com/encyclo
2	What's Your Favorite Computer Mouse? www.youtube.com	Jungle Mouse www.junglemouse.ne
3	What is mouse? - A Word Definition From the Webopedia Computer www.webopedia.com/TERM	Biomethodology of the Mouse - Animal Research - The University of Iowa Research.uiowa.edu/animal

7. Conclusion

Since the number of internet users is growing rapidly, the interests of internet users vary rapidly, making it hard to find relevant information according to individual user interests. Therefore personalized search engines, agent based search engines, and semantic web approaches have come into existence. Nevertheless those approaches have a serious limitation in user modeling and data presentation due to static user modeling methodologies.

Agent based systems are the newest tools which replace other web searching technologies. In this paper we have presented Divon, a swarm of agents who learn the user behaviors by observing the users searching habits and activities, and to present the search results in a way appropriate to individuals. Our solution is implemented one step ahead of the existing agent based designs as it has the ability to search results from the semantic web depending on the users' profile. Divon handles requests, responses, and resources in a more meaningful manner to achieve maximum resource utilization. The four agents, namely; Message Agent, User Profiler Agent, Query Handler Agent, and Presentation Agent collaboratively work together to provide highly accurate information to the user from the semantic web. User Profiler Agent who creates separate user profiles for every user communicates with Presentation Agent and Query Handler Agent to provide required user information and keywords to the personalized search. Query Handler Agent creates meaningful queries using the user entered queries and user specific keywords and retrieve search results from the semantic web. Message Agent communicates with the user, get their requirement and present him the results with his preferred format.

In the future we hope to expand the number of agents in our system to take more complex behaviors into consideration. In addition to the currently available user profiling criterions, we can consider users email information and messenger information. Furthermore we hope to integrate the system with the web browser so that we can monitor the users scrolling behavior also. We expect to generate more sophisticated learning, filtering, and ranking algorithms to further increase our system performances.

Acknowledgements

All the faculty members and our colleagues deserve to be appreciated for contributing to the success of this project in various ways. Also the authors of the references we have used throughout this project are highly appreciated.

References

[1] A. E. Howe, D. Dreilinger, "SAVVYSEARCH: A Metasearch Engine That Learns Which Search Engines to Query", *Jour:* AI Magazine, 1997.

[2] A. M Fard, M. Kahani, R. Ghaemi and H. Tabatabaee, "Multi-agent data fusion architecture for intelligent web information retrieval", *Proceedings of World Academy of Science*, Engineering and Technology, vol. 22, 2007, ISSN 1307-68.

[3] B. M. Leiner, V. G. Cerf, D. D. Clark , R. E. K. Leonard, K. Daniel, C. L. Jonathan, B. P. Lawrence, G. Roberts and S. Wolf,(1999), "A Brief History of the Internet", The Computing Research Repository.

[4] D. Bogdan, C. Ruth, M. Malcolm, C. Taylor , and M. Rusinkiewicz, "Distributed Processing of Queries for XML Documents in an Agent Based Information Retrieval System, "Kyoto International Conference on Digital Libraries, 2000.

[5] D. Riecken, Agents that reduce work and information overload, Communications of the ACM, Vol.37,1994, No 7

[6] Dr. B. Matthews, Semantic Web Technologies, CCLRC Rutherford Appleton Laboratory.

[7] E. J. Glover, S. Lawrence, W. P. Birmingham, and C. L. Giles, "Architecture of a Metasearch Engine that Supports User Information Needs," *Conference on Information and Knowledge Management*, Kansas City, Missouri, United States, 1999.

[8] F. Bellifemine, G. Caire, and D. Greenwood, "*developing multy agent systems with JADE*", Wiley series in Agent Technology, 2007, pp. 20-45.

[9] G. Graart, Agent based information retrieval supported by information markets, Master's thesis, University of Maastricht, 2003.

[10] Giovanni, S. Marco, D. Pasquale, L. U. Thiel , and M. L'Abbate, "A Personalized Information Search Process Based on Dialoguing Agents and User Profiling", *European Colloquium on IR Research – ECIR,* 2003.

[11] I. A. Macleod, P. Martin, B. Nordin, J. R.Phillips, "Strategies for building distributed information retrieval systems", Information Processing and Management, 1987.

[12] J. B. Odubiyi, D. J. Kocur, S. M. Weinstein, N. Wakim, S. Srivastava, C. Gokey, and J. Graham, "SAIRE - A Scalable Agent-Based Information Retrieval Engine", Autonomous

Agents & Multiagent Systems/International Conference on Autonomous Agents, 1997.

[13] J. Hendler, "Agents and the Semantic Web", IEEE Expert / IEEE Intelligent Systems, vol.16, March 2001.

[14] Massimo," Hyper Search Engines",journal: Computer Networks [and ISDN Systems].Marchiori," The Quest for Correct Information on the Web, 1997.

[15] M. Peshave, How search engines work and a web crawler application, Department of computer science, university of Illinois at Springfield Springfield, Midwest instructions and computing symposium, April 2005.

[16] N. E. Craswell, Methods for Distributed Information Retrieval, The Australian National University, May 2000.

[17] P. Thomas, An introduction to search engines, Search & Delivery; CSIRO ICT Centre, May. 2008

[18] S. Asadi and H. R. Jamali,(2004), "Shifts in Search Engine Development: A Review of Past, Present and Future Trends in Research on Search Engines, " *Webology*, vol. 1, December 2004.

[19] S. Brin and L. Page, The anatomy of a large scale hyper textual web search engine, Doctorial thesis, University of Stanford, 2001.

[20] S. Ugurlu , N. Erdogan, "An Agent-Based Information Retrieval System", Advances in Information Systems, 2000.

[21] T. Schreiter; An Introduction into Semantic Web Services; Hasso-Plattner-Institute for IT Systems Engineering, University of Potsdam, Germany.

[22] U. Straccia, R. Troncy, towards distributed information retrieval in the Semantic Web: Query reformulation using the oMAP Framework, *3rd European Semantic Web Conference*, June 2006.

Organizing Committee

- Dr. (Mrs) Uditha Ratnayake
- Dr. Nalin Wickramarachchi
- Prof. Asoka. S. Karunananda
- Ms. Menaka Ranasinghe
- Mr. Budditha Hettige
- Mr. Kalana Mendis
- Ms. Nilupa Liyanage

Panel of Judgers

- Dr. Sisil Kumarawadu
- Dr. Kapila Ponnamperuma
- Ms. Thashmee Karunarathne

Review Panel

- Dr. Nalin Wickramarachchi
- Prof. Asoka S Karunananda
- Dr. (Ms) Uditha Rathnayake
- Dr. Janaka Balasuriya

Special Supports

- **Language editing** : Ms. Umesha Weerakkody
- **Compiling** : Ms. Uditha Jayasinghe