Canadian Artificial Intelligence

Intelligence Artificielle au Canada

Canada's National AI Publication — **La Publication Nationale en IA au Canada**

Spring 2000 **No. 46** **printemps 2000**

Contents / Contenu

Canadian Artificial Intelligence is published by the Canadian Society for Computational Studies of Intelligence (CSCSI). ***Intelligence Artificielle an Canada*** est publiée par la Société canadienne pour l'étude de l'intelligence par ordinateur (SCEIO). Canadian Publications Mail Product Sales Agreement No. 1497200.

ISSN 0823-9339

Web Version Web

Please **don't throw away** the **envelope** in which CAI/IAC arrived. It contains information about your userID and password, which are needed to access the members-only area of the CSCSI/SCEIO website:

http://cscsi.sfu.ca/cai.html

Sample issues and articles are accessible to non-members. The members-only area contains this issue (#46) and some past issues of *CAI/IAC*. To access the area, type your userID and password at the login window.

Your **userID** is the first letter of your first name plus up to seven letters of your last name. For example, the userID for Anne Murray is amurray.

Your **password** is based on your CSCSI/SCEIO membership number which is **printed on the envelope** in which this issue arrived. Take that number and prepend to it the first letter of your first and last name. (e.g. if Anne Murray's membership number was 876543, then her password would be am876543.)

COMMUNICATIONS

President's Message

Robert Mercer

In my first President's message, my theme was "the challenges facing the society and the opportunities that we have to meet these challenges". I would like to return to that theme in this, my last President's message.

Our greatest challenge seems to be communicating with our members. In discussions with the executive and other members, the common theme has been that this challenge also presents an interesting opportunity. The people in a society like CSCSI/SCEIO are its greatest resources. What is needed is a way to coordinate and utilize the talents that exist to address the challenges.

Some of the issues that we have discussed are: the use of regional editors and special edition editors, more involvement from industry members in the form of special editors, and more items of interest to the student members.

We hope to discuss these and other issues at the next Annual General Meeting which will take place at AI2000 in Montreal on May 15 at 17:45. I hope you all attend the conference and make your voice heard at the AGM.

CSCSI SCEIO

AI 2000

The Thirteenth Canadian Conference on Artificial Intelligence

14-17 May, 2000

Palais Des Congrès, Montréal, Québec, Canada

http://www.cs.uregina.ca/~hamilton/ai2000.html

Secretary's Report

Guy Mineau

Minutes of the 1999 CSCSI/SCEIO AGM

Held on December 22nd 1999 over the phone (conference call), the meeting started at 16h00 and ended at 17h00. Only the executive of CSCSI participated in the meeting:

Fred Popowich, past president
Bob Mercer, president
Russ Greiner, vice-president
Guy Mineau, secretary
Howard Hamilton, treasurer

The agenda was:

1. Minutes of the 1998 meeting in Vancouver
2. Financial Report
3. Magazine
4. Other Business:
   - Update on AI 2000
   - Update on bid for AAAI 2002
   - Future plans: membership, etc. The agenda was proposed by Bob Mercer, supported by Fred Popowich, and was adopted unanimously.

1. Minutes of the 1998 Meeting

Guy Mineau presented the minutes of the 1998 meeting. Fred Popowich moved to adopt them, seconded by Russ Greiner, the motion to adopt them was carried unanimously.

2. Financial Report

Howard Hamilton presented the annual financial report (available on the web). Howard moved that we adopt it, seconded by Russ Greiner. It was adopted unanimously. Howard bought a GIC for 30K$ for a term of one year. This policy was approved by the executive and is something to repeat each year, for a one-year term. In brief, the society makes money the year it holds the conference, and loses money the second year. As a whole a dim profit would be made if the membership would remain stable. However, the membership is dropping and subsequent financial losses could occur. For the time being, a cash equity is still available to the society, of the order of 45K$. Ways to promote the membership using this money were discussed and will be brought up to the next AGM in Montreal in May of 2000 (during the conference). For instance it is proposed to use this money as financial support: to organize workshops and tutorials at the conference, to support students who have papers accepted to major conferences in the field of AI, to help pay for travelling expenses of members to give seminars throughout the country, etc.

3. Magazine

Ann Grbavec has now completed her first issue. It is obvious that the quantity of work required is much more than expected, mainly to collect the material for the different issues. It is moved by Guy Mineau and seconded by Russ Greiner that the amount of time paid for by the society to the CAI editor be doubled, going from a half-day a week to a full day a week. It is also suggested to seek regional coeditors who would bring 1 or 2 articles each year from their region; and it is suggested to use press release from Canadian industries to add information to the magazine (industrial involvement must also be sought). It is proposed to ask workshop organizers sponsored by CSCSI to become coeditor of the magazine for an issue. All these proposals will also be discussed at the next AGM.

4. Other Business

4.1 Update on AI 2000 Howard Hamilton reported that 60 papers were submitted and are in the process of being reviewed. One workshop was confirmed, the other one is pending, depending on the availability of rooms at the location where the conference will be held. Bob Mercer must check this out with the organizers of IRIS/Precarn, with whom we collocate this year.

4.2 Update on Bid for AAAI 2002

Russ Greiner informed the executive that the bid was done and according to him, that things are looking good with regard to our chances of being awarded the organization of AAAI 2002 in Edmonton. It is proposed that if this is the case, that the conference would not be held that year. Russ is confident that we should have news of this by March though it could be as late as December.

During the early months of next year we should be planning AI 2001. Bob Mercer will contact potential organizers for that year. The details of the exact format of the conference will have to be discussed at the next AGM however.

4.3 Future Plans: Membership

Guy Mineau will send an email to every past member of CSCSI just to remind them of the advantages of the society. This email message will promote the society by outlining all the advantages associated with the membership (current and the ones being considered as discussed above), will remind the past members of our low rates, offering the last three issues of the CAI magazine with the renewal of the membership, and will advertise AI 2000. Again, the drop in the membership must be discussed at the next AGM.

Minutes by Guy Mineau, Secretary of CSCSI
Associate Professor, Dept. of Computer Science
Université Laval, Quebec City, Quebec, Canada, G1K 7P4
Tel: (418) 656-5189 Fax: (418) 656-2324
Email: mineau@ift.ulaval.ca

Papers to be Presented at AI 2000

May 14-17, 2000,

Montréal, Quebec, Canada

CSCSI
SCEIO

Twenty-five papers have been accepted for oral presentation and twelve papers have been accepted for poster presentation. The conference proceedings will be published by Springer in the Lecture Notes in Artificial Intelligence series.

Papers Accepted for Oral Presentation

Barker, K., and Cornacchia, N., University of Ottawa, Canada;
Using Noun Phrase Heads to Extract Document Keyphrases.

Barriere, C., University of Ottawa and Popowich, F., Simon Fraser University, Canada;
Expanding the Type Hierarchy with Nonlexical Concepts.

Ghorbani, A. A., and Bayat, L., University of New Brunswick, Canada;
Accelerated Backpropagation Learning: Extended Dynamic Parallel Tangent Optimization Algorithm.

Han, J., An, A., and Cercone, N.; University of Waterloo, Canada;
CViz: An Interactive Visualization System for Rule Induction.

Hannon, C., and Cook, D., University of Texas at Arlington, U.S.A.;
A Parallel Approach to Unified Cognitive Modeling of Language Processing within a Visual Context.

Japkowicz, N., and Eavis, T., Daltech/Dalhousie University, Canada;
A Recognition-based Alternative to Discrimination-Based Multi-Layer Perceptrons.

Kubon, P. P., Popowich, F., Tisher, G., Technical University of British Columbia, Canada;
An Extendable Natural Language Interface to a Consumer Service Database.

Lallement, Y., and Fox, M.S.; University of Toronto, Canada;
Interact: A Staged Approach to Customer Service Automation.

Liscano, R., Mitel Corporation, Baker, K., and Meech, J., National Research Council, Canada;
The Use of Ontologies and Meta-Knowledge to Facilitate the Sharing of Knowledge in a Multi-Agent Personal Communication System.

McDonald, S., University of Edinburgh, Scotland, Turcato, D., McFetridge, P., Popowich, F., and Toole, J., Simon Fraser University, Canada;
Collocation Discovery for Optimal Bilingual Lexicon Development.

Moulin, B., Laval University, Kattani, D., Defense Research Establishment Valcartier and Laval University, Gauthier, B. and Chaker, W., Laval University, Canada;
Using Object Influence Areas to Quantitatively Deal with Neighborhood and Perception in Route Descriptions.

Nagarajan, S., Goodwin, S., University of Regina, Canada and Satar, A., Griffith University, Australia;
A Constraint Directed Model for Partial Constraint Satisfaction Problems.

Najarian, K., Dumont, G. A., Davies, M. S., and Heckman, N. E., University of British Columbia, Canada;
Neural ARX Models and PAC Learning.

Scarlett, E., and Szpakowicz, S., University of Ottawa, Canada;
The Power of the TSNLP: Lessons from a Diagnostic Evaluation of a Broad-Coverage Parser.

Schaeffer, J., and Plaat, A., University of Alberta, Canada;
Unifying Single-Agent and Two-Player Search.

Schulz, S., and Hahn, U., Freiburg University, Germany;
A Methodology for Knowledge Engineering in the Large.

Sebban, M., and Nock, R., Universite des Antilles et de la Guyane, France;
Prototype Selection Based on Information Theory.
Stenz, G., and Wolf, A., Institut für Informatik der Technischen Universität München, Germany;
Scheduling Methods for Parallel Automated Theorem Proving.
Upal, M. A., Dalhousie University and Renee Elio, University of Alberta, Canada;
Learning Rewrite Rules versus Search Control Rules to Improve Plan Quality.
van Rijswijck, J.; University of Alberta, Canada;
Are Bees Better than Fruitflies? Experiments with a Hex Playing Program.
Vitoria, A., and Mamede, M.; Universidade Nova de Lisboa, Portugal;
On the Integration of Recursive ALN-Theories.
Wiese, K., University of British Columbia, Nagarajan, S., and Goodwin, S., University of Regina, Canada;
ASERC--A Genetic Sequencing Operator for Asymmetric Permutation Problems.
Xiang, Y., University of Massachusetts, U.S.A., Hu, J., Fulcrum Technologies Inc., Cercone, N., University of Waterloo, and Hamilton, H.J., University of Regina, Canada;
Analysis and Experiments on Learning Pseudo-Independent Models.
Yan, J., Tokuda, N., and Miyamichi, J., Utsunomiya University, Japan;
Simulating Competing Alife Organisms by Constructive Compound Neural Networks.
Yang, Q., Simon Fraser University and Wu, J., University of Waterloo, Canada;
Keep it Simple: A Case-base Maintenance Policy Based on Clustering and Information Theory.

Papers Accepted for Poster Presentation

Baillie, J.-C., and Ganascia, J.-G., LIP6 - Universite Pierre et Marie Curie, France;
Qualitative Descriptors and Action Perception.
Bowes, J., Neufeld, E., Greer, J. E., and Cooke, J., University of Saskatchewan, Canada;
A Comparison of Association Rule Discovery and Bayesian Network Causal Inference Algorithms to Discover Relationships in Discrete Data.
Chatterjee, N., Indian Institute of Technology, India;
Analysis of Plans for Knowledge Acquisition.
Garzone, M., and Mercer, R. E., University of Western Ontario, Canada;
Towards an Automated Citation Classifier.
Han, J., and Cercone, N., University of Waterloo, Canada;
Typical Example Selection for Learning Classifiers.
Hussin, M. F., Arab Academy for Science, Technology and Maritime Transport and Abouelnasr, B. M. and Shoukry A. A., Alexandria University, Egypt;
Comparative Study of Neural Network Controllers for Nonlinear Dynamic Systems.
Karimi, K., University of Regina, Canada;
The Iterative Multi-Agent Method for Solving Complex Search Problems.
Morin, J., and Matwin, S., University of Ottawa, Canada;
Relational Learning with Transfer of Knowledge Between Domains.
Scheutz, M., University of Notre Dame, USA;
Surviving in a Hostile Multi-Agent Environment: How Simple Affective States Can Aid in the Competition for Resources.
Situ, Q., and Stroulia, E., University of Alberta, Canada;
Task-Structure Based Mediation: The Travel-Planning Assistant Example.
Suderman, M., and Delgrande, J., Simon Fraser University, Canada;
Considerations on Compositional Update Operators.
Tawfik, A. Y., University of Prince Edward Island and Barrie, T., University of Toronto, Canada;
The Degeneration of Relevance in Uncertain Temporal Domains: An Empirical Study.
Zhang, H., Ling, C. X., and Zhao, Z., University of Western Ontario, Canada;
The Learnability of Naive Bayes.

Ask Jeeves Natural Language Patent Suit

Ask Jeeves, the Internet search engine with a natural language front end, has been sued yet again over patents for natural language processing.

Two MIT scientists, Patrick Winston (author of a classic Lisp/AI book) and Boris Katz, are suing Ask Jeeves over two patents issued in to them 1994 and 1995.

The researchers are seeking an injunction prohibiting Ask Jeeves from "making, using or selling Internet search products." Additionally, the suit seeks to prevent Ask Jeeves from licensing its search engine to other companies and to collect damages and royalties.

A spokeswoman for Ask Jeeves said that the suit is without merit and that the company will aggressively defend itself.

Previously, Ask Jeeves was sued for infringing a natural language processing patent owned by IPLearn. Ask Jeeves has denied those allegations as well.

Distributed Simulation as a Tool for Artificial Intelligence Research

John Anderson

Résumé

En dépit de beaucoup d'avances récentes dans le milieu de l'Intelligence Artificielle, grace a l'utilisation d'approches complètement fondées, la simulation persiste à être un outil extrèmement important pour les chercheurs dans de nombreux domaines de l'IA. Les raisons sont nombreuses et incluent des problemes de controle, de largeur de sujets, de point focal de recherches, ainsi que des problemes de dépenses et de fiabilité pratiques. Cependant, l'application de la simulation au développement des systèmes intelligents, place d'énormes demandes sur un outil de simulation: incluant le support d'agents multiples et leurs interactions, et le contrôle détaillé des épreuves dans un environnement, ainsi que la perception précise dans les limites de calcul. Présentement, notre travail de développement d'agents intelligents utilise la simulation comme processus de développement et d'évaluation. Récemment nous avons investigué les techniques de simulation distribuée comme solution aux problèmes pratiques et philosophiques difficile a traiter dans le cadre de la simulation de système simple. Cet article trace les grandes lignes de l'approche utilisée dans notre création d'une simulation distribuée pour le développement d'agents intelligents et certains avantages contribués par la simulation distribués specifiques a cet application.

Abstract

Despite many recent advances in AI through completely-grounded approaches, simulation continues to be an extremely important tool to research in numerous areas of AI. The reasons for this are many and include control issues, issues of breadth and research focus, and issues of practical expense and reliability. The application of simulation to the development of intelligent systems, however, places enormous demands on a simulation tool: everything from supporting multiple agents and their interactions, to providing detailed control over trials in an environment, to accurate perception within computational bounds. Our current work in the development of intelligent agents employs simulation as part of the development and evaluation process, and we have recently been turning to distributed simulation as a solution to practical and philosophical problems that are difficult to deal with in single-system simulation. This paper outlines the approach we are taking to the creation of a distributed simulation tool for intelligent agent development and some of the benefits that distributed simulation brings to this application.

1 Introduction

From its outset as a field, Artificial Intelligence has relied on simulation to provide environments in which intelligent behaviour is demonstrated. From the earliest toy problems within which heuristic search techniques were developed, to today's robotic simulators, simulation continues to be an important part of intelligent systems research and development. This reliance on simulation has at times been to the detriment of the field. Any introductory AI text worth its name chronicles the overgeneralizations made in AI's early days, by researchers eager to show the broad applicability of results obtained from simulated domains that were grossly oversimplified from the real world. Instances of such overgeneralization became one of the most oftcited arguments against the utility and future of intelligent systems (e.g. [Dreyfus, 1981; McDermott, 1981]). Howe [1993] also points out that the very ease of use that simulators bring to the testing process can allow AI researchers to construct experiments too quickly, without the clearly stated hypotheses and careful methodology that physical domains encourage. While such subtle problems are insidious however, it the fact that simulation allows researchers to model only those aspects of an environment that they choose to be significant, ignoring other aspects that may have a much greater impact on results than is initially assumed, that led to the trend in AI to eschew simulation in favour of completely grounded physical models (e.g. [Agre, 1988; Maes and Brooks, 1990]). The effects of this trend are also seen more recently in the emphasis on multirobot systems and standardized problem-solving environments such as robotic soccer within the area of multiagent systems.

This emphasis on physical grounding has been beneficial to AI in several ways, producing results that not only show how far simple architectures can be used to deal with complex domains under the right conditions, but furthering work in placing perception and low-level action into more significant positions in intelligent agent architectures. However, this emphasis also limits applicability to many types of problems simply because of the level at which implementation must proceed and the detail in which the world must be dealt with – both elements easily controlled using simulation. The value of simulation itself derives in part from the fact that the simplification of the physical world is often necessary to isolate a given phenomena in a complex system for experimentation purposes, and is as critical in AI as in any other scientific endeavour [Hanks et al., 1993]. This simplification is also necessary specifically in AI research for much more than experimental isolation. AI is an immature science, and as such it does not yet possess a wealth of broadly accepted theories to form a concrete foundation for ongoing research. The result of this is that researchers in learning, for example, cannot make use of a generally accepted theory of planning to qualify or simplify their research. Similarly, researchers in vision cannot make use of a generally accepted theory of learning. In both of these cases, any overlap from one subfield to another entails making assumptions about unsolved problems in the other subfield (qualifying and possibly compromising one's own research) or embarking on a number of highly tangential research projects in order to provide a more solid foundation for the original research.

The use of a simulator in intelligent agent research is thus largely an issue of practicality: the large collection of interrelated problems encountered when implementing an intelligent agent necessitates the use of a simulator to aid in accounting for those pieces of the theory that are not yet complete. Simulators are also often required because of a lack of physical resources. Few research facilities can afford the staggering cost of supplying (and keeping technologically current) enough robotic technology to meet the needs of all the intelligent systems research projects they support [Etzioni and Segal, 1992] nor the manpower to maintain this equipment.

Beyond the nature of physical environments, simulation provides means of control not available in the physical world. A simulator can present a common environment across many trials, providing exactly the same initial state and planned course of events for each, and can be used to isolate the testing environment from interference from aspects not controllable in the real world [Cohen et al., 1989]. Because of these control abilities, domains can also be saved and shared, allowing simulation environments to serve as a broad metric for comparing intelligent agents [Howe, 1993; Hanks et al., 1993].

The trend of very purposely avoiding simulation is abating to some degree today, and is approaching a happy medium wherein emphasis is placed on thorough analysis of assumptions made in a domain, and physical verification where appropriate and possible, but with strong recognition of the value that simulation as a research tool brings to AI. This can be seen in such systems as Balch's TeamBots (formerly JavaBots [Balch and Ram, 1998]), where a simulator provides a physical world that accepts the same instructions and perception bounds as physical robotic equipment, so that the same code can be used in both a simulated environment and physical robots. This brings the advantages of both approaches and limits their detriments.

The difficulty with employing simulation in the development of intelligent agents lies in the wide range of application domains and the wide range of intelligent agent internal components, whose nature changes as development in relatively unexplored domains proceeds. In order to perform ongoing research in areas where agent designs and the environments in which they are examined change as development pursues, we require a tool that will easily support such changes. Similarly, there are many applications outside of the development of intelligent agents themselves in which complex environments populated by such agents are useful: natural resource management [Deadman and Gimblett, 1994] and economics [Deadman, 1999], for example. Ideally, a simulator generic enough to support diverse agents and environments should also be applicable to these areas and others.

With this in mind, we developed *Gensim*, a generic time-shared simulator for multi-agent systems [Anderson and Evans, 1995]. We have employed this system ourselves in intelligent agent design and verification [Anderson, 1995; Anderson and Evans, 1996] and have shown its potential in areas outside of this environment [Anderson, 1997; Anderson and Evans, 1994]. This system is generic in that agents and environments can be easily defined and interchanged in a modular fashion. The system also provides pragmatic support for agent sensing, control over agent timing, and facilities for constructing domains and agents. However, the most problematic aspects of a realistic environment for a generic simulator to provide are precisely those that are present due to the nature of simulation itself. For example, an agent perceiving objects around it must specify objects or areas of interest (in the way that we might focus on one area in spite of having a much larger visual field), and must somehow be given a list of perceived objects fitting with both the environmental restrictions on perception and the agent's own abilities for perception. Creating such a perception selection for a significant number of agents is extremely time-consuming for a simulator. This also simply does not occur in the real world, as such communication is not necessary: the world's physics manages itself, and agents perceive from a low level image of the world. It is a compromise necessary for adequate simulation while not implementing low-level agent vision (one of the main reasons for employing a simulator in intelligent agent design). Similarly, the agent's commitments to action must be recorded and manifested by the environment as the result of an asynchronous decision-making process on the part of an agent. Internal timesharing of such agent processes is awkward in that this does not naturally model this asynchronous process. It also creates timing problems, in that agents must have similar operating cycle lengths, as well as problems in the results of this timing, such as the predictability of the outcome of competing actions given the order in which the agents are time-shared. It is also inordinately slow, making the simulator impractical for large numbers of agents (a particular interest in our own multi-agent systems work, as well as one of the most primary demands of a tool for agent-based work in other fields).

We are currently attempting to deal with these problems and others through the development of *DGensim* [Anderson, 2000], which employs distributed simulation techniques to provide a more natural underlying approach to simulation for the purposes of supporting intelligent agents. While in most cases, the purpose of turning to distributed simulation technology is the increased computing power this approach makes available [Hamilton et al., 1997; Pham et al., 1998], this work involves using distribution to increase the fidelity of simulations and ultimately the ability to more adequately experiment with intelligent agents in complex domains.

The remainder of this paper describes the methodology employed in creating DGensim, and details the solutions that distribution brings to the problems mentioned above.

2 Gensim and DGensim

As mentioned above, Gensim [Anderson and Evans, 1995] is a generic simulation system for intelligent agent designs. In Gensim, a LISP-based simulation process manages an object-oriented view of the environment, including the physical embodiment of the associated agents. Collections of agent processes (also LISP-based) making up the decision-making components of agents are time-shared, with equal timeslices usually given to each agent. Agent processes are run, and during their timeslices may or may not commit to particular actions, which are communicated to the simulator. The simulator manifests these changes in a time-based manner (i.e. manifesting the changes made during the agent timeslices just completed), using an event queue to manage future change. Agents also make requests for perceptual information within their current context, and after the simulator manifests the changes initiated during the current cycle, this perceptual information is provided based on limitations of bandwidth and recorded agent perceptual abilities. Details of these and other significant aspects of the system may be found in [Anderson and Evans, 1995].

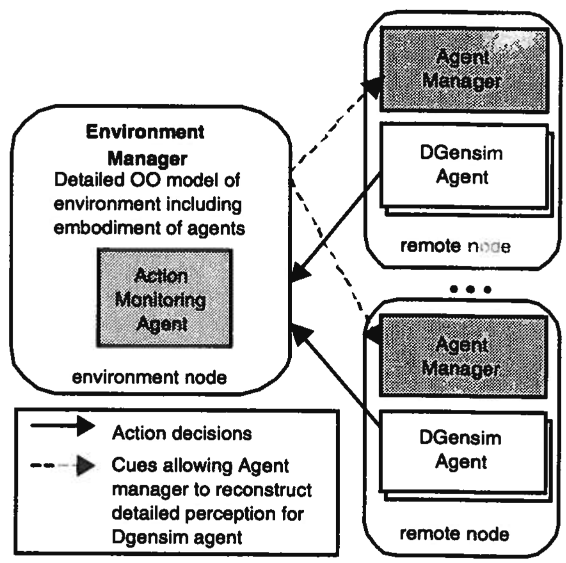

Figure 1. Overview of DGensim Organization.

In moving to a distributed simulation to deal with the problems mentioned in Section 1, we take the approach of moving agent decision-making processes to separate hardware systems. The basic model employed in DGensim is illustrated in Figure 1. DGensim revolves around a set of n node machines, n-1 of which are dedicated to executing agent internals. The remaining node runs the *environment manager*, which manifests the changes made by agents over time as well as the ongoing physics of the rest of the environment. In order to deal with the management of multiple agents per agent node, an *agent manager* is employed. This process is initially run on each machine that is to participate in the simulation by supporting agent internals, and connects to the environment manager via a prespecified Linux port. When the environment manager is started on the environment node, it makes contact with remote nodes and transfers agent code to them. This transfer is intended in future to allow the environment node to choose the most appropriate machine for a particular agent, and to allow agents themselves to be mobile. The agent manager is also sent limited environmental information, which allows it to play another important role that will be described shortly.

2.1 Actions and Timing

The most immediate result of the distribution of agent internals is a much more natural, realistic flow of agent decision-making over time due to a more realistic execution of the underlying agent processes. This is due in part to improvements in implementation platform (Allegro Common Lisp under Linux), which among other things allows OS-level threads as opposed to the application-level timesharing of the original Gensim system. However, there are also significant improvements in timing accuracy brought about by distribution *per se*. Rather than allowing an agent to take as many fixed time steps as required to come to a decision upon action, and then processing that decision along with those of other agents during that same cycle, agents in DGensim send their timestamped decisions asynchronously to an action monitoring agent (see Figure 1). This relatively simple agent (running on the same system maintaining the object-oriented environment) organizes incoming decisions and assists in correcting for limited network delays using the timestamps on incoming actions to order the queue of pending events.

While agent processes in DGensim make asynchronous decisions for action, the environment around those agent processes flows at a constant rate through time – thus defining the flow of time for the agents involved. This timing model is both more accurate and far simpler to employ within DGensim agents than the original. As it is however, it is susceptible to problems with long network delays: in a wide-area network scenario, it is entirely possible to have an action commitment from an agent delayed enough that others that might have interfered with it would have been processed, and even related sensory information already delivered to other agents. It is a fairly simple matter to deal with small delays, based on the fact that the environment is updated in discrete time steps. Every event intended to occur during a particular unit time is processed in sequence, and as long as an action is received by the action monitoring agent within this time boundary, the action monitoring agent will rearrange the event queue such that everything is updated as if the actions arrived at the appropriate point in time. That is, there is a window of safety around with an event can be delayed, and a simulation may be designed to increase that window.

In the event that an action arrives after the environment manager has processed its time unit, compromises must be made. In this case, other agents may have already been given perceptions that might have been different had the action been received in a timely fashion (and if the delay is long, even made further action commitments on that basis). Given that the simulator is supposed to be an ongoing interactive real world for agents to inhabit, rollback is generally not an option. Rollback would also be difficult for large numbers of agents, and more importantly the mechanics of being able to roll back agent state change the nature of the agents themselves. The latter would lead to different results in experimentation than would be seen otherwise. Other options are made available however, in order to keep the simulator as general as possible. First, an action may be simply invalidated, as if it simply had not occurred, and the agent(s) involved informed of this. Another available alternative is to process the event as if it had happened during the time unit in which it arrived as opposed to the one in which it was generated. Envision one agent throwing a ball, and a second (the late agent) attempting to catch it. In this situation, the end result would be as if the late agent moved to catch the ball too late to receive it. This is somewhat more suitable, but still inadequate for most agent evaluation experiments because the actions of the late agent itself were optimal – it was the nature of the simulation that caused the delay. Finally we allow a preventative measure that slows the simulation down significantly but deals with these problems as completely as can be expected. In DGensim, it is also possible to force each agent to transmit its actions regularly, including a *no-op* action if the agent is not performing any useful activity. This allows the environment manager to process one timeslice after a complete set of these actions is received. This is overwhelming in the case of a large number of agents however, or a particularly small time unit setting for the environment manager.

Note that none of these approaches to dealing with delay (except the case above where an agent can effectively be made to act in synch with a slower environment) is particularly appealing in the case of experimentally evaluating agents: network delays significantly affect results. Even in the situation where the environment manager can expect a completely regular response from each agent, it may not be worth continuing experimentally if it does not get one (i.e. an agent is severed from the simulation temporarily). However, this is a particularly exacting application of distributed simulation, and these approaches are available to be used where that may be more suitable. In our own work, because of the nature of the problems we are investigating, we naturally use small dedicated networks, and would consider an experiment invalid if delays in action reception such as those considered above occurred. On this scale however, such delays are rare and thus an insignificant downside in comparison to the timing benefits alone.

2.2 Perception

Perception and its relationship to the agent and environment is another significant problem in Gensim that was dealt with there through compromise, but which can be significantly improved through a distributed implementation. In Gensim (and DGensim as well), perception is implemented pragmatically at the object level [Anderson and Evans, 1995], in order to remove the burden of pixel-level perception for agents – one of the major reasons simulators are employed in intelligent agent development [Hanks et al., 1993; Anderson, 1995]. In both Gensim and DGensim, agents specify their interest using concepts familiar to both the agent and the simulator (e.g. scan for blue objects; where is the other agent) or may simply gaze in a direction of interest. In the original Gensim system, the simulator responds to this request with object-attribute-value specifications for a limited range (based on a model of the agent's perceptual abilities) of what can be perceived (given the agents' perceptual and focus biases). However, this is an artificial organization of perception, in that the simulator is doing more than just removing the overly-detailed burden of low-level vision – it is actually doing all the agent's perception for it aside from the highest level of integrating those perceptions into the state of the agent's decision-making components. The environment simulation process is deciding which objects the agent can see, and which it is limited from seeing. While some perceptual limitation is due directly to the environment and does seem to fall within these bounds (e.g. objects block one another), others are due to the agent (what type of objects it is biased toward focusing perceptual attention upon, for example). Placing the perceptual component completely within either the agent or environment is philosophically inaccurate, and technically problematic in that this sensory preparation is computationally intensive and slows simulation significantly.

The solution to this problem in DGensim is to move the management of perception to a point *between* the agent decision-making components and the simulated environment. In DGensim, the agent managers running on each remote machine manage the bulk of perception. In response to an agent's perceptual request, the agent manager receives a description of objects that could be perceived based on the physical aspects of the environment – it is up to the agent manager to filter these according to the particular biases of the agent itself. This places perception in a much more natural position in a simulation organization, and removes a significant amount of unnecessary work from the environment manager. However, it also results in the potential for an inordinate amount of information transfer, and potential network overload for a large number of agents.

In order to restrict the amount of information that must be physically sent across the network by the environment manager, agent managers contain simplified environmental information: stereotyped views of objects in the environment that are exported to agent managers by the environment manager each time a new environment is used. When the agent manager accepts a particular agent, a registration is created indicating the mapping between agent and manager, a part of which states the frequency with which sensory information should be sent to the particular DGensim agent (via its manager). This effectively states the speed at which the agent can perceive objects. The environment manager maintains the agent's current orientation and maximum sphere of attention (physical attributes of the agent's body, which it is managing as part of the environment), and relays very basic object information to the appropriate agent manager. This information is essentially which object stereotypes to invoke and the particular changes beyond those stereotypes. The agent manager then reconstructs detailed attribute perceptions (the information originally provided by the simulator in Gensim) based on the information received and its local knowledge. This approach pragmatically balances reasonable perception with network bandwidth, and also allows us to deal with the perception-related problems described earlier.

This rebalancing of the function of perceptual information service is also in part a consequence of distribution: given that agents are now separated from the environment processes by machine boundaries, it makes sense to move those elements of perception that do not belong in the environment there as well. This also assists significantly in improving efficiency: In Gensim, the bulk of the simulation time was spent preparing perceptual information and translating it into the object references or descriptions an individual agent would comprehend for each unique agent (DGensim employs the same *deictic* object description facilities provided for Gensim; the interested reader is referred to [Anderson and Evans, 1995] for more details on this). Moving a significant portion of the work previously done by the environment simulator to the distributed components allows the simulation to run more efficiently and keep up with distributed agent internals. This in turn helps to minimize the problems associated with getting perception information back to agents in a timely fashion as discussed above.

3 Discussion and Future Work

As mentioned earlier in this paper, DGensim is in ongoing development using LISP on a network of Linux-based machines. We have currently made what we believe are reasonable assumptions to ultimately provide a useful tool supporting a broad range of agents and environments efficiently and effectively. While we have designed the system in particular for experimenting with intelligent agents in spatially explicit multiagent settings from the point of view of the design and performance of those agents, we believe this system to be applicable to other areas where multiagent simulation is of use as well.

As stated earlier, one of the major motivations of this project is to create a simulation tool that is as generic as possible. One of the difficulties with maintaining this generic nature in a distributed simulation is the much larger range of variability that is introduced once a distributed implementation must be considered. In distributed simulations, primary motivation ranges from providing significantly detailed graphical environments suited to human participants, to focusing on the interactions of computational agents. Environments range from the corners of the Internet to local area networks, and reliability, security, control, and level of fidelity vary considerably, as do the number of expected agents. While such a huge range cannot be expected to be dealt with in a single tool without weakening that tool to the point of uselessness, we are currently working with local networks and attempting to provide as broad a range as possible within this scheme. Issues of generality in such a simulator are beyond the space limitations of this forum, and the interested reader is directed to [Anderson, 2000] for further discussion on these topic.

DGensim is also in ongoing development in a larger sense. What is described here is the first phase of a larger distributed simulation system which will hopefully be just as powerful while dealing with some of the issues affecting the implementation of significantly large numbers of agents along with a wider range of distribution. The next phase will involve the distribution of the environment across multiple machines as well as agent internals. As stated earlier, the tightly interactive nature of environmental components makes synchronization difficult. However there are strong elements of locality of reference in agent interaction in virtually any physical environment, pointing toward geographical division as a logical choice. This would be especially useful in domains where fairly strong geographic boundaries could be defined (i.e. where the likelihood of agent sensory range overlapping a boundary and thus having to be serviced from two different machines would be minimal). We also intend to add the ability for agents to migrate from machine to machine to rebalance loads and (in the case of a distributed environment as well) make the overall simulation more efficient and dynamic.

In this paper, we have endeavoured to illustrate that despite the common view of distributed simulation as a necessary evil for large-scale simulations of difficult problems, with its overhead paid back in increased performance, the distributed approach in fact brings about many practical advantages in this situation. In particular, simpler timing (and at a lower level, the implementation of timing), and the provision of a pragmatic agent perception component benefit greatly specifically from a distributed setting. While this is by no means a simulator that can be used to implement every possible distributed simulation, we believe it will be applicable not only to the initial problem to which it was designed, but a wide range of applications outside of this. Future work will extend this range while keeping the pragmatic choices outlined here in mind.

References

[1] Agre, P. E., *The Dynamic Structure of Everyday Life*, Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1988. 282 pp.

[2] Anderson, J., "A Generic Distributed Simulation System for Intelligent Agent Design and Evaluation", to appear in Proceedings of the International Conference on AI, Simulation, and Planning, Tucson, AZ., March, 2000.

[3] Anderson, J., "Supporting Intelligent Agents in Individual-Based Ecosystem Models", *Proceedings of the Eleventh Annual Conference on Geographic Information Systems*, Vancouver, BC, February, 1997, pp. 3-6.

[4] Anderson, J., *Constraint-Directed Improvisation for Everyday Activities*, Ph.D. dissertation, Department of Computer Science, University of Manitoba, March, 1995. 397pp.

[5] Anderson, J., and M. Evans, "Real-Time Satisficing Agents for Complex Domains", *Proceedings of the Ninth Florida AI Symposium*, Key West, FL, May, 1996, pp. 96-100.

[6] Anderson, J., and M. Evans, "A Generic Simulation System for Intelligent Agent Designs", *Applied Artificial Intelligence*, Volume 9, Number 5, October, 1995, pp. 527-562.

[7] Anderson, J., and M. Evans, "Intelligent Agent Modelling for Natural Resource Management", *International Journal of Mathematical and Computer Modelling*, Volume 20, Number 8, October, 1994, pp. 109-119.

[8] Balch, T., and A. Ram, "Integrating Robotic Technologies with JavaBots", *Working notes for the 1998 AAAI Spring Symposium on Integrating Robotic Research*, AAAI, 1998.

[9] Deadman, P., "Modelling Individual Behaviour and Group Performance in an Intelligent Agent-Based Simulation of the Tragedy of the Commons", *Journal of Environmental Management* 56, 1999, pp. 159-172.

[10] Deadman, P., and R. H. Gimblett, "A Role for Goal-Oriented Autonomous Agents in Modeling People-Environment Interactions in Forest Recreation", *Mathematical and Computer Modelling* 20:8, 1994, pp. 121-131.

[11] Dreyfus, H. L., "From Micro-Worlds to Knowledge Representation: AI at an Impasse", in Haugeland, J. (Ed.), *Mind Design* (Cambridge, MA: MIT Press), 1981, pp. 161-204.

[12] Etzioni, O., and R. Segal, "Softbots as Testbeds for Machine Learning", *Proceedings of the Machine Learning Workshop at AI/GI/VI '92*, University of British Columbia, May, 1992, pp. v1-v8.

[13] Hamilton, J. A., D. A. Nash, and U. W. Pooch, *Distributed Simulation* (Boca Raton, Fl.: CRC Press), 1997. 390 pp.

[14] Hanks, S., M. E. Pollack, and P. R. Cohen, "Benchmarks, Test Beds, Controlled Experimentation, and the Design of Agent Architectures", *AI Magazine* 14(4), Winter, 1993, pp. 17-42.

[15] Howe, A., "Evaluating Planning through Simulation: An Example Using Phoenix", *Proceedings of the workshop on the Foundations of Automatic Planning: The Classical Approach and Beyond*, AAAI Technical Report SS-93-03, 1993, pp. 53-57.

[16] Maes, P., and R. A. Brooks, "Learning to Coordinate Behaviours", in *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, 1990, pp. 796-802.

[17] McDermott, D. V., "Artificial Intelligence Meets Natural Stupidity", in Haugeland, J. (Ed.), *Mind Design* (Cambridge, MA: MIT Press), 1981, pp. 143-160.

[18] Pham, C., H. Brunst, and S. Fdida, "How Can We Study Large and Complex Systems", *Proceedings of the 13th Annual Simulation Symposium*, Boston, April 1998, pp. 126-134.

About the Author

John Anderson (andersj@cs.umanitoba.ca) is an Assistant Professor in the Department Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada, R3T 2N2. He received his Ph.D. in Computer Science from the same institution in 1995. His current research involves autonomous agents and multiagent systems, including intelligent agent collaboration, simulation environments for developing and experimenting with agents, and mobile robotics.

Representing Time and Repetition: Survey and Future Directions

Diana Cukierman

Résumé

Ceci est une presentation d'étude de solutions de rechange pour modeler les objets temporels dans les communautés de bases de données d'intelligence artificielle et les bases de données temporelles. Nous nous concentrons principalement sur l'ontologie et les aspects de modelization formelle. Une attention particulière est pretée à l'algèbre d'intervalle de Allen et aux travuax reliée. L'étude se tourne vers la recherche sur les répetitions temporelles et la formalization de calendrier, y compris sur notre recherche effectuée presentement sur les structures d'objets et de boucles temporelles.

Abstract

A survey of alternatives to model temporal objects in the Artificial Intelligence and Temporal Databases communities is presented. We mainly focus on the ontology and formal modeling aspects, paying special attention to Allen's Interval Algebra and work based on it. The survey then focuses on work about temporal repetition and calendar formalizations, including our current research on *structured temporal objects* and *time loops*.

1 Introduction

The notion of time is inherent in many activities which involve intelligence. Change, causality and action are intrinsically related to the passage of time; the information associated to any dynamic environment is time-dependent. Hence, it is not surprising that researchers in a wide variety of areas related to Artificial Intelligence and Databases need to deal with temporally-based information. In particular, much work is devoted to deal with time per se, where the intent is to formalize what temporal objects are and how they relate, abstracting from what occurs during them. Different options are proposed to best model temporal objects according to the availability of information and to accommodate the different requirements in applications.

Application areas where time is a fundamental component include medical diagnosis and therapy monitoring, scheduling of classes in a university, ecological modeling; the list is endless.

Questions that we mainly address in this survey and arise when formalizing time include the following: Which should be the primitive temporal objects: time points or time intervals? Should time be formalized in a discrete or continuous manner? Is time linear, parallel, branching? Should temporal objects and their relations be expressed in a qualitative manner or also including quantitative information? Should one single system allow different levels of granularities or time units? How can temporal objects best represent repetition? In the next sections we address these questions as we survey different proposals in the area; we only extract from the surveyed literature what concerns to the ontology of the temporal objects.

Section 2 describes interval based work, with special emphasis on Allen's Interval Algebra, a qualitative approach where intervals are primitive temporal objects. Section 3 briefly describes point based research, as well as research where both points and intervals are considered within the same formalism. Qualitative and quantitative formalisms are included. Section 4 describes temporal objects representing temporal repetition; work about calendar formalizations is also included. Finally we briefly describe research dealing with structured temporal objects and qualitative relations. In particular time loops represent repetition. Some final comments conclude this paper.

2 Time Intervals

Allen [1] proposes that *intervals* should be considered as primitive temporal objects. An objection posed about interpreting assertions over *time points* is that this leads to paradoxical situations. For example, imagine a time interval during which a light bulb is on, meeting another interval during which the same light bulb is off. Where these two intervals meet, there is a transition of the truth value of the property *on(light)*. The question that arises is, what is the truth value of this property in the meeting point? Is the light on, off, both or neither? Allen proposes that by having assertions interpreted only over intervals consistency is maintained.

Another argument in favor of a non-point based but rather "quantum oriented" view of time is explained in [5], within the context of databases and conceptual modeling of time. The basis of the argument is the limitation in measurement given by the Heisenberg uncertainty principle. From a cognitive perspective, this principle corresponds to the fact that events have to have a certain extent in time (and in space) to be perceivable. Hence some physicists have proposed a *time quantum* or a minimal time division of about 10^{-22} seconds. In [5] the time quantum concept is generalized to be the minimal time division conceivable subject to the application or context, which may be (much!) longer than 10^{-22} seconds. The glossary in [18] defines this concept with the term *chronon*, as "the shortest duration of time supported by a temporal database". [4] axiomatize *time moments* as indivisible time intervals. Intuitive and linguistic evidence is proposed to support the existence of such concept. When intervals are defined as primitive temporal objects points are defined in terms of the former; for example [4] defines time points by the sets of intervals which meet (at the point being defined).

2.1 Allen's Interval Algebra

A fundamental and inspiring work about temporal intervals is that of Allen [1, 2]. Allen's interval algebra is a *relational algebra*. Very briefly, an algebra is composed of a non-empty set S of elements and a sequence of operations, such that the set S is closed under these operations, and other properties hold. The elements in a *relational algebra* are binary relations and the operations are the set-theoretical operations union, intersection, converse and composition. (Hence, this is not to be confused with the algebraic formulation of operations on relational databases)

Relation		Graphical Representation
before after	$.b.$ $.b^{-1}.$	i j
meets met by	$.m.$ $.m^{-1}.$	i j
starts started by	$.s.$ $.s^{-1}.$	i j
during contains	$.d.$ $.d^{-1}.$	i j
overlaps overlapped by	$.o.$ $.o^{-1}.$	i j
finishes finished by	$.f.$ $.f^{-1}.$	i j
equals	$.eq.$	i j

Table 1: Basic 13 binary interval relations. Direct relations are graphically represented between intervals i and j: $i\ .r.\ j$, therefore, the inverse holds between j and i: $j\ .r^{-1}.\ i$.

[1] defines a set of mutually exclusive relations between convex time intervals, where intervals are considered as primitive temporal objects. Convex intervals are those that do not have any gaps within them. This is to be contrasted with *non-convex intervals*, a term introduced by [22] and dealt with later in this article, representing temporal repetition. The elements of the Interval Algebra are the *relations* that may exist between two convex intervals of time. There are 13 basic relations between convex intervals; see Table 1. These are the combinatorial possibilities of how two intervals can relate, such as *before* and *during*. Temporal relations can be indefinite, i.e. a disjunction of basic relations may be all what is known to hold between two intervals, therefore there are 2^{13} possible relations between convex intervals. A transitivity table, expressing the results of composing any two basic relations appears in [1]. For example, *before* composed with *meets* results in *before*. This corresponds intuitively to when one interval is *before* a second interval, and the second interval *meets* a third interval, from which it follows that the first one is *before* the third one.

In [1] the interval algebra is used as a basis for a constraint propagation algorithm to solve the closure problem; intuitively this problem can be described as finding all possible consistent scenarios for a set of intervals inter-related with relations in the algebra. The algorithm runs in polynomial time, but it was found to be non complete; the problem of determining consistency of assertions in the full interval algebra was proven to be NP-complete [31]. Starting from these seminal papers, a very extensive line of research evolved. Work has been done to obtain more efficient reasoners to solve the closure and other equivalent problems. Some authors consider point-based algebras [31]. Different interval sublagebras (with less expressive power but complete closure polynomial algorithms) have been studied as well [28, 14]. Other research includes Allen's algorithm as a module to propagate temporal constraints within other systems; for example [19, 26] combine Allen's system with metrics.

We next turn to analyze work dealing with time points and with both intervals and points.

3 Time Points: Both Points and Intervals

In [31, 32], the *Time Point Algebra* is developed, based on the notion of *time point* in place of interval. Three qualitative basic relations are possible; {<, >, =}. Indeterminacy is represented with disjunction of the basic relations. Operations of composition, intersection and inverse are defined between such relations. Furthermore, Vilain and Kautz also propose an *Interval-Point Algebra* including relations between *both* points and intervals. Thus point-to-point, interval-to-interval and point-to-interval qualitative relations are defined. For example a point can be *before*, *starting* or *during* an interval and so on.

In [12] time points are the primitive temporal object but quantitative (rather than qualitative) information is dealt with. [12] defines *metric networks* to deal with the *TCSP*: *temporal constraint satisfaction problem*. Metric networks are based on points, where *unary constraints* are point-to-date constraints and *binary constraints* are point-to-point metric constraints. An example tackled by such system is: "John goes to work either by car and it takes him less than 20 minutes. He can also go by bus, which takes at least 60 minutes. Fred goes by car (15-20 minutes) or by car-pool (40-50 minutes). Today John left home between 7:00 and 7:05 and Fred arrived between 7:50 and 7:55. John arrived between 10 to 20 minutes after Fred left home". Queries that can be asked are: Is the information consistent? Is it possible that John took the bus and Fred used the car-pool? What are the possible times that Fred left home?" A *Simple temporal problem* (or *STP*) is a special case of the TCSP where all the constraints are of a simple case and is solvable in polynomial time.

[26] defines the generalized TCSP and deals with both time points and time intervals in the same formalism. Furthermore, this formalism also incorporates qualitative and quantitative information. This combines and extends *Metric networks* [12] and Interval and Interval-Point Algebras-based reasoners [1, 32].

"Graph based" temporal reasoners provide another example of point-based formalisms. These works exploit the inherent temporal structure in certain domains; not all relations are explicitly included and/or only some types of graphs are considered. Hence important improvements are obtained in practical performance. [16] proposes a *timegraph* to represent qualitative, point-based temporal information. Timegraphs were originally inspired in narrative or plans, where events occur mostly as a chain of sequential temporal events and there is not much interrelation between chains.

[13] proposes variations of this approach and define *series-parallel graphs* to represent (qualitative, point-based) temporal information which adapt to a certain type of domain. Highly efficient algorithms are obtained to deal with some class of temporal problems.

A point-based formalism which is widespread in research about change, causality and action is the *Situational Calculus* [25]. The Situational Calculus is an extension of first order language with temporal arguments; a *situation* reflects the state of the world at a particular time or instant. One situation is transformed to another situation through some *action*. A truth-valued *fluent* is a property or relation that may or may not hold in a given situation. Situation Calculus originally did not deal with a metric of time nor concurrency; however there have been proposals of extensions to the original formulation which include such. Kowalski and Sergot's Event Calculus [21] defines intervals in terms of instantaneous events. Intervals are defined by events happening at the beginning or end of an interval, with the possibility of leaving undefined one of the extremes, or later adding the information about the other extreme to the knowledge base.

[17] addresses the issue that intelligent reasoning involves being able to view the world in *different granularities* and move from one scale to the other. Essentially, depending on the level of granularity the same event can be conceived as happening during a time interval or a time point. An adequate perspective not only provides a better formalization, but it can turn an intractable reasoner into a useful tool in a coarser and more appropriate scale of the problem. A similar argument is proposed by Freksa [15] suggesting that coarse reasoning is more akin to the way cognitive agents perceive reality, and can also simplify reasoning tasks (computationally). In Freksa's work, coarseness refers to the temporal relation between two intervals. Relations between two intervals are defined based on the intervals extreme points, but it does not necessarily involve the four extreme points. This allows one to represent coarser information, which in some applications may be sufficient and can be simpler.

4 Repetition and Periodicity

The issue of which element to choose as a primitive temporal object extends to the representation of temporal repetitive objects. Likewise, some proposals are exclusively qualitative, some include metrics. The issue of granularity is especially dealt with within temporal repetition when defining calendars and time units.

The term *non-convex intervals* (also referred to as *unions of convex intervals*) was coined by Ladkin: They "consist intuitively of some (maximal) convex subintervals with convex gaps in between them", [22], page 360. Qualitative binary relations extending those in the Interval Algebra are suggested. The number of such relations is at least exponential in the number of sub-components [22]. Some relations are defined and exemplified in the referred paper. One subset of such relations is generated by functors *mostly, always, partially, sometimes* on convex interval relations. For example, *I mostly R J* iff "for every j in J there exists an i in I such that i R j", where the *j in J* stands for a subinterval *j* in the non-convex interval *J*.

Non-convex intervals constitute a repetitive temporal object which has been used in many other works. For example [27] defines *N-intervals* as a subclass of Ladkin's non-convex intervals; they constitute finite sequences of pairs of points, each pair representing a subinterval. An algebra of n-interval relations is defined with operations of intersection, composition and inverse. These operations allow the application of a path consistency constraint propagation algorithm.

[24] generalizes further the non-convex interval concept to *generalized intervals*, where intervals are defined by either an even or odd number of points. Relations among these intervals are expressed as conjunctions of conditions on the interval defining points. These relations are referred as *(p,q)-relations*, relating one interval defined by *p* points with another defined by *q* points. For example, Allen's (convex intervals) relations are (2,2)-relations.

4.1 Quantitative Information and Repetition

[23] is one of the inspirational papers about multiple time unit, calendar-based expressions and repetition. This proposal relies on sequences of consecutive intervals combined into "collections". The collection representation makes use of "primitive collections" (essentially circular lists of integers), and two basic operators, *slicing* and *dicing*, which subdivide an interval and select a subinterval from another collection respectively. New collections are built with these operators, and thus calendar-based repetitive intervals are represented. An example of a primitive collection is "days of 1999". A collection resulting from applying selection to the previous collection could be "*Mondays* of 1999". Many other calendar formalizations have been recently proposed; these also include repetitive expressions subsuming the collection expressions, see for example [29, 10, 6]. Work where collection expressions are used within formal systems and/or temporal reasoners include [20, 30].

In [10] we define *classes of intervals* (in the sense of sets). Particularly a core concept developed is *time unit classes*, which are classes with non-intersecting intervals. Time unit classes are the building blocks to define time unit hierarchies or *calendars*. Based on time unit hierarchies, one can represent specific intervals or time points in the time line with *calendar expressions*, as well as sets of intervals with a certain repetition pattern, corresponding to (calendar based) non-convex intervals. These expressions subsume the language of [23] adding set operators, a displacement operator and universal and existential quantification. For example it is possible to express "Two days per week", "All weekdays from 7:00 to 8:00 am except day number 5 of any month", "The first Monday in February 2000 and the day occurring 10 days later", etc.

[7] propose calendric formalizations and also constraint propagation networks with metric information expressed in different time units. The idea proposed is to have a separate network for each time unit, and within each network the constraint propagation problem is a Simple Temporal Problem [12].

Some forms of repetition have been formally characterized in temporal databases related research. *Linear repeating points, (lrps)* is a mathematical characterization of repetitive sequences, developed by [8] in the context of generalized databases. These are extensions of the classical relational databases which allow a finite representation of infinite temporal information. A generalized tuple includes linear repeating points as temporal components and constraints on them. A *lrps* is a set of integers represented by a linear equation $c + kn$, where c and k are integer constants and n takes values from $-\infty$ to $+\infty$ in the integer numbers. Linear repeating intervals extend lrps to (fixed length) intervals [29].

4.2 Structured Temporal Objects and Repetition

We conclude this article by briefly describing part of our current research about qualitative repetitive temporal objects. In [11] we propose temporal objects where an essential characteristic is that they include structure within them. This approach is based on convex interval relations. The simplest structured temporal object is a *related pair*, consisting of two temporal objects where an interval relation holds between them. As well, we have sets of related pairs and *time loops*. Our primary object of interest is the time loop, used to represent *temporal repetition*. We abstract (possibly infinite) lists of temporal terms which follow certain repetitive structure with a *time loop*. Three parameters provide the necessary information to define a loop: a single (representative) *cycle*, an order relation between consecutive instances of the cycle or *repeats* and a *dimension*. A loop cycle can be an interval, but it can also be recursively made up of other (related) intervals and/or other loops. Examples such as *sometimes lunches meet seminars, sometimes lunches are before seminars* are covered within one loop. To express the above example with non-convex intervals instead of loops, there would exist two different non-convex intervals, "seminars" and "lunches", one corresponding to the set of seminars and the other to the lunches. This example would then be expressed as a relation between matched pairs using non-convex intervals: each single seminar *meets* or *is after* each single "closest" lunch. In a loop the linking relation between successive seminars and lunches is made explicit, as being part of a cycle given in the loop construct.

Hence non-convex intervals are subsumed by the temporal objects we propose; the language can represent sets of temporal objects (which may recursively include structure), with any qualitative relation holding between them and arbitrary gaps.

5 Concluding Remarks

This brief survey in temporal representation has described works where different modeling alternatives are proposed for temporal objects. We have mainly addressed ontological issues not deepening into computational complexity results. In particular we have focused on temporal repetition. For other overviews in the area see for example [3, 9].

As far as the issues dealt in the present survey, one conclusion that emerges is that the intended application may determine the granularity required, and this in turn determines how a temporal event is best represented, whether point or interval-based. Likewise, the application and the available information determine whether qualitative, quantitative or both kinds of information can or should be considered to solve a given problem. As is to be expected, there is a trade-off between the expressive power and preciseness on one hand, and the algorithms computational complexity on the other. It has become clear that important computational improvements can be gained when restricting the temporal representation and the reasoning to specific characteristics of the domain. Hence work dealing with specially structured domains such as temporally repetitive situations provide promising results.

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832- 843, 1983.

[2] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123-154, 1984.

[3] J. F. Allen. Time and time again: The many ways to represent time. *International Journal of Intelligent Systems*, 6:341-355, 1991.

[4] J. F. Allen and P. J. Hayes. Moments and points in an interval-based temporal logic. *Computational Intelligence*, 5:225-238, 1989.

[5] T. L. Anderson. *Modeling Time at the Conceptual Level*, pages 273-297. Academic Press, Inc., 1982.

[6] C. Bettini and R. DeSibi. Symbolic representation of user-defined time granularities. In C. Dixon and M. Fisher, editors, *International Workshop on Temporal Representation and Reasoning, TIME'99*, pages 17-28, Orlando, Florida, USA, 1999. IEEE Computer Society.

[7] C. Bettini, X. S. Wang, and S. Jajodia. A general framework and reasoning models for time granularity. In *International Workshop on Temporal Representation and Reasoning, TIME'96*, pages 104-111, Key West, FL, USA, 1996.

[8] J. Chomicki and T. Imielinski. Temporal deductive databases and infinite objects. In *PODS'88, Proceedings of the 7th ACM Symposium on Principles of Database Systems*, pages 61-73, Austin,Texas, 1988.

[9] D. Cukierman. Temporal reasoning in artificial intelligence: A survey, with focus on repetition and calendar formalization. Technical report SFU CMPT TR 1998-06, Simon Fraser University, School of Computing Science, 1998.

[10] D. Cukierman and J. Delgrande. Expressing time intervals and repetition within a formalization of calendars. *Computational Intelligence*, 14(4):563-597, November 1998.

[11] D. Cukierman and J. Delgrande. A formalization of structured temporal objects and repetition. July 2000. To appear in *TIME 2000*.

[12] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61-95, 1991.

[13] J. P. Delgrande, A. Gupta, and T. vanAllen. Point-based approaches to qualitative temporal reasoning. In *Proc. of AAAI'99*, Orlando, FL, July 1999.

[14] T. Drakengren and P. Jonsson. Towards a complete classification of tractability in allen's algebra. In *Proc. of IJCAI'97*, pages 1466-1471, Nagoya, Japan, 1997.

[15] C. Freksa. Temporal reasoning based on semiintervals. *Artificial Intelligence,* 54:199-227, 1992.

[16] A. Gerevini and L. Schubert. Efficient algorithms for qualitative reasoning about time. *Artificial Intelligence,* 74:207-248, 1995.

[17] J. Hobbs. Granularity. In *Proc. of the Ninth IJCAI-85,* pages 1-2, 1985.

[18] C. Jensen, J. Clifford, S. Gadia, A. Segev, and R. Snodgrass, editors. *A Glossary of Temporal Database Concepts,* chapter 21, pages 621-633. Benjamin/Cummings Publishing Company, Inc., 1993.

[19] H. A. Kautz and P. B. Ladkin. Integrating metric and qualitative temporal reasoning. In *Proc. of AAAI-91,* pages 241-246, 1991.

[20] J. A. G. M. Koomen. Reasoning about recurrence. *International Journal of Intelligent Systems,* 6:461-496, 1991.

[21] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing,* 4:67-95, 1986.

[22] P. B. Ladkin. Time representation: a taxonomy of interval relations. In *Proc. of the AAAI-86,* pages 360- 366, 1986.

[23] B. Leban, D. D. McDonald, and D. R. Forster. A representation for collections of temporal intervals. In *Proc. of the AAAI-86,* pages 367-371, 1986.

[24] G. Ligozat. On generalized interval calculi. In *Proc. of the AAAI-91,* pages 234-240, 1991.

[25] J. M. McCarthy and P. J. Hayes. *Some philosophical problems from the standpoint of artificial intelligence.* Tioga Publishing Co., 1981.

[26] I. Meiri. Combining qualitative and quantitative constraints in temporal reasoning. In *Proc. of the AAAI91,* pages 260-267, 1992.

[27] R. A. Morris, W. D. Shoaff, and L. Khatib. Path consistency in a network of non-convex intervals. In *Proc. of the IJCAI-93,* pages 655-660, 1993.

[28] B. Nebel and H. J. Burckert. Reasoning about temporal relations: A maximal tractable subclass of allen's interval algebra. In *Proc. of the AAAI-94,* pages 356- 361, 1994.

[29] M. Ni'ezette and J. St'evenne. An efficient symbolic representation of periodic time. In *Proceedings of the 1st. International Conference on Information and Knowledge Management,* Baltimore, Maryland, 1992.

[30] P. Terenziani. Qualitative and quantitative constraints about numerically quantified periodic events. In *International Workshop on Temporal Representation and Reasoning, TIME'97,* pages 94-101, FL, USA, 1997.

[31] M. Vilain and H. Kautz. Constraint propagation algorithms for temporal reasoning. In *Proc. of the AAAI86,* pages 377-382, 1986.

[32] M. Vilain, H. Kautz, and P. VanBeek. Constraint propagation algorithms for temporal reasoning: a revised report. In Weld and de Kleer, editors, *Readings in Qualitative reasoning about physical systems,* pages 373-381. Morgan Kaufmann, 1989.

About the Author

Diana Cukierman (diana@cs.sfu.ca) is a Ph.D. student in the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, V5A 1S6. Her research interests include artificial Intelligence, knowledge representation, logic, logic programming, and temporal reasoning.

Editor's Note:

Related work by this author [11] has been selected as one of two Best Papers at TIME 2000: The Seventh International Workshop on Temporal Representation and Reasoning.

The Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems

(IEA/AIE-2001)

Budapest, Hungary, June 4-7, 2001

Sponsored by the International Society of Applied Intelligence and cooperated with major international organizations, including ACM/SIGART, AAAI, INNS, IEE, CSCSI, JSAI, SWT, ERCIM, and SZTAKI.

Submission Deadline: November 17, 2000

Submit the required materials by November 17, 2000, to Dr. Laszlo Monostori.

Authors can obtain details of the call for papers announcement either from the conference web page,
http://www.sztaki.hu/conferences/ieaaie2001/

or from

Dr. Laszlo Monostori,
Program Chair IEA/AIE-2001,
Computer and Automation Research Inst.,
Hungarian Academy of Sciences,
Kende u. 13-17, H-1111
Budapest, Hungary.
FAX (+36 1) 466 7503;
E-mail ieaaie2001@sztaki.hu,
kindl@sztaki.hu

Dr. Moonis Ali, General Chair,
IEA/AIE-2001,
Southwest Texas State University,
Department of Computer Science,
601 University Drive,
San Marcos, TX 78666-4616, USA.
Telephone (512) 245-3409;
FAX (512) 245-8750;
E-mail ma04@swt.edu;

ISR 2000

International Symposium on Robotics

May 14-17, 2000

Palais des Congrès, Montreal, Quebec (Canada)

http://www.precarn.ca/isr2000/

CALL FOR PAPERS

PRICAI 2000

Sixth Pacific Rim International Conference on Artificial Intelligence

August 28 - September 1, 2000

Melbourne, Victoria, Australia

http://www3.cm.deakin.edu.au/pricai/

First Call for Papers

ISI 2001

International Congress on Information Science Innovations

March 17-21, 2001

to be held at the American University in Dubai, U.A.E.

Submissions must be received by June 30, 2000.

http://www.icsc.ab.ca/isi2001.htm

Tenth Annual PRECARN • IRIS Conference

May 14-17, 2000

Palais des Congrès, Montréal, QC

We're very excited about this year's conference. Not only is it the Tenth PRECARN • IRIS Annual Conference but the Conference is also being held for the first-time in conjunction with the 31st International Symposium on Robotics (ISR 2000), and the Artificial Intelligence (AI) 2000, Graphics Interface (GI) 2000 and Vision Interface (VI) 2000 conferences. Five Conferences for the price of one!! We are hoping to attract over 800 people from around the world.

You won't want to miss Canadian astronaut, Julie Payette, who will launch this first-time event at the Opening Plenary session. An accomplished computer science researcher, Ms. Payette will be sharing with us reports of her voyage into space as part of the first shuttle mission to dock with the International Space Station (ISS).

As a favorite of the Conference in the past years the Demo and Poster exhibitions return. These exhibits will provide a unique opportunity for conference participants to see, touch and learn about PRECARN and IRIS ongoing research projects.

For conference information, contact:

Conference Coordinator Office
Golden Planners Inc.
126 York Street, Suite 301
Ottawa, Ontario K1N 5T5
TEL (613) 241-9333 / FAX (613) 565-2173
EMAIL ISR2000@goldenplanners.com

For conference information updates:

PRECARN•IRIS www.precarn.ca/Events

ISR 2000 www.precarn.ca/isr2000

PRECARN Associates Inc.

announces a

Request for Proposals

for a

Technology Development Program in Intelligent Systems

Deadline for submissions:

Proposal Concept Document:
June 9, 2000

Full Proposal:
July 10, 2000

This RFP is for projects that will become part of the ongoing PRECARN program of industry-led collaborative research projects in the broad area of Intelligent Systems (a list of current PRECARN projects can be found on our web site http://www.precarn.ca or, may be requested by facsimile or mail).

This RFP is open to all PRECARN Members and to new partner companies that commit to become PRECARN Members if their projects are approved.

For more information, please contact:

PRECARN Associates Inc.
300-30 Colonnade Rd.
Nepean Ontario, Canada K2E-7J6
Phone: (613) 727-9576
Fax (613) 727-5672
E-Mail: info@PRECARN.ca

http://www.precarn.ca

CSCSI/SCEIO Membership Application

I wish to join CSCSI/SCEIO and receive *Canadian Artificial Intelligence /Intelligence Artificielle au Canada*

Yes ___ / No ___ Web Access Only ($30.00* Cdn./yr.)

Yes ___ / No ___ Printed Copy and Web Access ($40.00* Cdn./yr.)

I am a student

Yes ___ / No ___ Web Access Only ($15.00* Cdn./yr.)

Yes ___ / No ___ Printed Copy and Web Access ($15.00* Cdn./yr.)

I am a member of CIPS

Yes ___ / No ___ Web Access Only ($25.00* Cdn./yr.)

Yes ___ / No ___ Printed Copy and Web Access ($30.00* Cdn./yr.)

*Includes applicable G.S.T.

Name ....

Job Title ....

Company ....

Address ....

....

....

....

E-mail ....

Phone ....

Fax ....

Visa or Amex Number ....

Expiry Date ....

Signature ....

Please send your membership application to:

Mail: CIPS National Office, One Yonge St., Suite 2401, Toronto, Ontario M5E 1E5

Phone: (416) 861-2477

Fax: (416) 368-9972

E-mail: aff@cips.ca

For more information, contact CIPS or a member of the CSCSI/SCEIO executive.

Canadian Artificial Intelligence
Intelligence Artificielle au Canada

The editor welcomes contributions from members and non-members. Possible formats include, but are not limited to:

- commentary, in the form of letters or short articles, on any AI issue of concern to you (e.g. research priorities, funding, educational trends, cultural, political, or ethical aspects of AI)
- interdisciplinary work
- survey papers
- reports on new work by students
- abstracts
- web site reviews
- news
- humour
- employment opportunities
- conference announcements.

Submission information • Guide de soumission

Canadian Artificial Intelligence / Intelligence Artificielle au Canada welcomes submissions on any matter related to artificial intelligence. Please send your contribution, preferably in electronic form, with an abstract and biography to:

Ann Grbavec
Editor, *Canadian AI / IA au Canada*
School of Computing Science
Simon Fraser University
Burnaby, British Columbia, Canada
V5A 1S6 or — agrbavec@cs.sfu.ca
Phone (604) 291-3208
Fax (604) 291-3045

Please contact the editor as early as possible to obtain complete submission guidelines. **Submission deadline:** July 21, 2000.

Sample issues and articles can be viewed at: http://cscsi.sfu.ca/cai.html

Advertising rates are available upon request from the editor.

Recycled / Recyclable