Special Issue:
Advances in
Artificial Intelligence

Research in Computing Science

Series Editorial Board
Comité Editorial de la Serie

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 26**, Noviembre, 2006. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Se usó la imagen obtenida de la siguiente dirección, para el diseño de la portada: http://dosmanos.com/Merchant2/graphics/00000001/RACAC.jpg Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela,* RFC *SOAJ560723*

Volume 26
Volumen 26

Special Issue: Advances in Artificial Intelligence

Volume Editors:
Editores del Volumen

Alexander Gelbukh
Carlos Alberto Reyes-García

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2006

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.ipn.mx
http://www.cic.ipn.mx

Printing: 500
Impresiones: 500

Printed in Mexico
Impreso en México

Preface

Artificial Intelligence is a branch of computer science that develops heuristic methods for solving super-complex problems that cannot be solved by direct calculations or mathematics methods. A large class of such problems is modeling human-like behavior such as ability to think, learn by example, doubt, act, see, and speak. Another class are highly combinatorial optimization problems. Many of the methods suggested for solving such problems have been influenced and inspired by nature, since many processes in nature involve huge number of participating entities—particles, living cells, animals, etc.—interacting in a very complex manner.

In this year, we celebrate 50 years since foundation of Artificial Intelligence as a ranch of sciences, as well as 20 years since foundation of the Mexican Society of Artificial Intelligence.

This volume presents 16 original research papers structured into eight thematic fields representative of the main current areas of interest within the AI community: Knowledge Representation and Reasoning, Evolutionary Algorithms, Machine Learning and Classification, Optimization Algorithms, Computer Vision, Agents, Intelligent Tutoring Systems and Dynamic Systems.

Total of 36 papers by 107 authors from 16 different countries where submitted for evaluation; see Tables 1 and 2. Each submission was reviewed by three independent members of the Editorial Board of the volume. This volume contains revised versions of 16 papers, by 50 authors, selected for publication after thorough evaluation. Thus the acceptance rate was 44.4%. In Table 1, the number of papers by country was calculated by the shares of all authors of the paper: e.g., if a paper has three authors: two from Mexico and one from USA, then we incremented the counter for Mexico by 0.66 (two authors of three) and the counter for USA by 0.33. Table 2 presents the statistics of papers by topics according to the topics indicated by the authors; note that a paper can be assigned more than one topic.

The academic and editorial effort resulting in this volume was carried out in collaboration with, and was supported by, the Mexican Society for Artificial Intelligence (SMIA). We cordially thank all people involved in its preparation. In the first place these are the authors of the papers constituting it: it is the excellence of their

Table 1. Statistics of submissions and accepted papers by country

Country	Authors		Papers		Country	Authors		Papers	
	Subm	Accp	Subm	Accp		Subm	Accp	Subm	Accp
Australia	2	–	2	–	Japan	1	–	1	–
Brazil	3	3	1.2	1.2	Macedonia	1	–	0.33	–
Chile	3	3	0.99	0.99	Malaysia	3	–	0.99	–
China	23	5	8.66	1.99	Mexico	33	19	9.66	5.99
Cuba	10	4	2	1	Portugal	8	8	1.8	1.8
Germany	2	–	0.66	–	Spain	6	4	1.66	1
India	4	3	1.99	0.99	Turkey	4	–	1	–
Israel	3	–	0.99	–	UK	1	1	1	1
					Total:	107	50	36	16

Table 2. Statistics of submitted and accepted papers by topic

Topic	Submitted	Accepted
Assembly	1	–
Belief Revision	1	–
Bioinformatics	1	–
Case–Based Reasoning	1	1
Common Sense Reasoning	2	1
Computer Vision	3	2
Data Mining	3	2
Expert Systems / KBS	1	–
Fuzzy logic	2	–
Genetic Algorithms	6	3
Hybrid Intelligent Systems	3	1
Intelligent Interfaces: Multimedia; Virtual Reality	1	1
Intelligent Organizations	3	1
Intelligent Tutoring Systems	4	2
Knowledge Acquisition	2	1
Knowledge Management	3	1
Knowledge Representation	5	3
Knowledge Verification; Sharing; Reuse	1	–
Logic Programming	4	1
Machine Learning	4	3
Model-Based Reasoning	3	1
Multiagent systems and Distributed AI	4	2
Natural Language Processing / Understanding	1	–
Navigation	1	–
Neural Networks	6	4
Nonmonotonic Reasoning	2	–
Ontologies	2	–
Philosophical and Methodological Issues of AI	1	–
Planning and Scheduling	5	3
Qualitative Reasoning	1	1
Robotics	3	2
Spatial and Temporal Reasoning	2	1
Uncertainty / Probabilistic Reasoning	2	1
Other	4	2

research work that gives sense to the work of all other people involved. We thank the members of the Editorial Board of the volume and additional referees. We express our gratitude to Ángel Kuri, Héctor I. Ortiz Ortiz, Ernesto Daza Ramírez, Leoncio González Hernández, Roberto Acoltzi Nava, Nicolás Alonzo Gutiérrez, Orion Fausto Reyes Galaviz, and Sulema Torres for their significant contribution at various stages of preparation of the volume. The submission, reviewing, and selection process was supported for free by the EasyChair system, www.EasyChair.org.

Alexander Gelbukh November 2006
Carlos Alberto Reyes García

Table of Contents

Índice

Page/Pág.

Computer Vision

Agents

Intelligent Tutoring Systems

Dynamic Systems

Knowledge Representation and Reasoning

Modeling a Decision Making Process in a Risk Scenario: LOCA in a Nucleoelectric Plant Using Fuzzy Cognitive Maps

Ana Lilia Lauren-Cruces[1], Javier Ramírez-Rodríguez[2], Martha Mora-Torres[3] and Gilberto Espinosa-Paredes[4]

Universidad Autónoma Metropolitana
[1,2] Azcapotzalco, Departamento de Sistemas
San Pablo 180, 02200 México D. F., México
<clc@correo.azc., jararo@correo.azc.>uam.mx
[4]Iztapalapa, Departamento de Ingenieria de Procesos e Hidraulica
Av. San Rafael Atlixco 186, Col. Vicentina, 09340 Mexico D.F., Mexico
gepe@xanum.uam.mx

[3] Posgrado de Ciencia e Ingeniería de la Computación del IIMAS-UNAM, Circuito Escolar
04510 Ciudad Universitaria.D.F., México
kabhun@yahoo.com.mx

Abstract. Decision making in a nucleoelectric plant is a complex process, due to many elements involved in its operation, and the permanent attention demanded by its maintenance. Presently, the decision making process in the plant is analyzed and developed by a human operator, using diagrams whose main characteristic is a linear representation of events within a scenario. That is a slow process, and can lead to the generation of new failures. We propose the development of an expert system that will help in the decision making, In this paper our main objective will be the design of the of knowledge representation and the design of reasoning through the latter. The dominion is located in the failure events in a nuclear plant. To automate the decision making process, a representation of knowledge is developed using Fuzzy Cognitive Maps (FCM), which allow us to model an expert's behavior, in decision making with uncertainty.

1 Introduction

An important part of fuzzy logic and possibility theory is centered in aspects such as: representation of knowledge, approximate reasoning and reasoning with uncertainty, main topics of Artificial Intelligence [3].

Fuzzy control logic is very close to the experts' spirit, and is known as artificial intelligence–based control. A byproduct of this theory are fuzzy cognitive maps [4, 5].

Fuzzy cognitive maps are digraphs used to represent causal reasoning. The fuzzy part allows us to have degrees of causality, represented as links between nodes of these digraphs, also known as concepts. This structure allows the forward and backward propagation of causality, allowing the knowledge base to increase when

© A. Gelbukh, C.A. Reyes-García. (Eds.) *Received 31/05/06*
Advances in Artificial Intelligence. *Accepted 03/10/06*
Research in Computing Science 26, 2006, pp. 3-13 *Final version 10/10/06*

increasing nodes and links between them. Causality is represented as a fuzzy relation between nodes. For this type of representation a causal algebra that allows the implementing of this type of knowledge representation has been developed[7, 9, 12].

In a nuclear plant whose objective is to generate electric power, an important feature could be some critical situations due to failures in control systems; in this case the effects of mechanical and/or electrical failures of the different variables that keep the reactor in operation must be attenuated, and and if equilibrium needs to be restored through preventive decision making [10, 11, 13]. In this paper we use FCMs to model the behavior of an expert in decision making [2].

In AI, there is a variety of techniques used for representing knowledge: production rules, semantic networks, frameworks, scripts, statements logic and fuzzy cognitive maps, among others [3]. The choice of a particular technique depends on two factors: the nature of the application and the user's choice. In this case, fuzzy cognitive maps (FCMs) were chosen to represent behavior. The behavior to be modeled is centered in the decision making process, whose reasoning implies to reach a predefined goal, coming from one or more initial states; hence, the less number of transitions to reach the final goal, the most efficient the reasoning system will be. Thus, increasing efficiency implies to minimize intermediate states, and that is represented in the organization of the knowledge base. The former is accomplished by a fast and efficient implementation of the inference engine; in other words, the circuit between perception and action is closed in a faster way [3]. Cognitive maps allow us to represent many rules efficiently through links.

As a first step we must detect by means of a behavior analysis, the elements generating that behavior and its process. Elements will be represented by nodes and behavior by the links between this nodes [6, 8].

2 Fuzzy Cognitive Maps

Fuzzy cognitive maps (FCMs) comprise a new approach to the model of behavior and operation of complex systems. They were introduced by Bart Kosko in 1986 to describe the behavior of a system in terms of concepts and causal relations between those concepts [4, 5, 7, 12].

FCMs are represented by a diagram, in which nodes are concepts describing the main characteristics of the process, and the edges between the nodes establish causal relations (positive or negative) between the concepts. This graphical representation illustrates the influence that each one of the concepts has on the rest [7].

Concepts in a FCM are events, whose values change in time and originate in the system. Concepts take values in the interval [0,1], and the interconnection weights in the interval [-1,1] (see [2, 7]).

Positive causality implies a directly proportional relation[1] between one concept and the other, whereas the negative one implies an inversely proportional relation[2] [2, 7].

1 **A directly proportional relation b**etween two elements means that if one **increases** its possibility of being present the other is **incremented** proportionally. And if this possibility **decreases** in one of them, then the possibility of the other **decreases** proportionally.

This qualitative approach allows observing the general behavior of the system. However, quantification must be taking into account the causal relation in the map. How much does a node cause another? This is where fuzzy logic proves its worth.

The state of a given node is derived from all the other nodes causing it. These states are multiplied by the weight of the arc between the two nodes, and the sum is used as the input of a threshold function, transmitting a non bound input in a bound signal, which allows the comparison of nodes. Different threshold functions can be used; in this paper the following logistic signal function is used [5]:

$$S(x) = \frac{1}{\left(1 + e^{-cx}\right)} \tag{1}$$

3 Application case: Accident by Loss of Coolant (LOCA)

A small LOCA (*Loss Of Coolant Accident*) is a rupture in liquid flow, small enough so that low capacity systems are sufficient to compensate the loss of coolant. The size of this LOCA category can be approximated to a relief/security valve stuck in the open position. Within this category are considered liquid leaks less than 0.004 sq. ft. and vapor leaks less than 0.005 sq. ft. Arriving to this LOCA scenario implies an initial failure of loss of coolant. Although LOCA's have not happened in nuclear plants according to operational experience, this kind of initiators scenarios are examined because they represent a menace to the core integrity and to the primary contention [1].

4 Elements of the Model of the HPCS System Within a Small LOCA

The behavior model is conformed by 6 elements, and this model is inspired in the physical system of Fig. 1: a) A1 (Reactor valve failure); b) A2 (Tank valve failure); c) M (Pool manual valve failure); d) B (Pump failure); e) HPCS operational; f) Vessel in good condition.

The reactor valve allows feeding water from the deposits (tank and pool) to keep a stable temperature of the reactor by means of cooling it when its temperature surpasses acceptable levels. Besides, this valve, given the relation existing between the variables temperature and pressure, also influences the keeping of pressure inside the reactor vessel. Because of the former, a *reactor valve failure (A1)* becomes a first order failure[3], and thus leaves the *High Pressure Core Spray (HPCS)* cooling system inoperative, which consequently leaves the vessel in a bad condition.

---

2 **An inversely proportional relation b**etween two elements means that if the possibility of one of them being present **increases**, then the possibility of the second element **decreases** proportionally; and again, if the possibility of one **decreases** then the possibility of the second element **increases** proportionally.

3 **First order failure**: one whose effect causes a total failure of the system, leaving it non operational.

The tank valve (Tx Valve) allows feeding water from this deposit (Tx) to the reactor, and so it is part of the cooling system; in other words, if a *tank valve failure (A2)* is present, this affects the operation of the HPCS system. However, this relation is not determining, because the tank is not the only deposit available to supply the water necessary to keep the HPCS system operational.

The pool valve (Ax Valve) allows feeding water from the second deposit (Pool) available to the cooling system towards the reactor. Hence, a *pool valve failure (M)* affects the operation of the HPCS system, and, like the former relation, it does not so in a determining way, since the pool is not the only deposit available to supply the water necessary to keep the HPCS system operational.

The pump allows the suction of water from any of the two deposits available for the HPCS system to provide the necessary cooling, and then a *pump failure (B)* first order one, it affects the reactor directly.

The *operation of the HPCS system* is the main element of the model, it has a positive effect on the objective element, that is, to keep the vessel in good condition. The rest of the elements of this system: pump and valves, contribute to keep the HPCS system in good working order, and hence a failure in such elements affects the overall operation of the system.

The *good condition of the vessel* implies a reactor operating properly, and this state is directly dependent on the operation of the HPCS system, which means a correct operation of the pump and the valves contained in the system.

Fig. 1 shows in a general way the operation of a nucleoelectric plant: generating electric power by means of the vapor obtained from the heating of water by the reactor nucleus.

Keeping the reactor operation in a failure scenario implies the activation of the systems attenuating such a failure scenario [1, 10, 11].

In our model, the attenuating system is the HPCS and the failure scenario is the loss of coolant (known as small LOCA) [1].

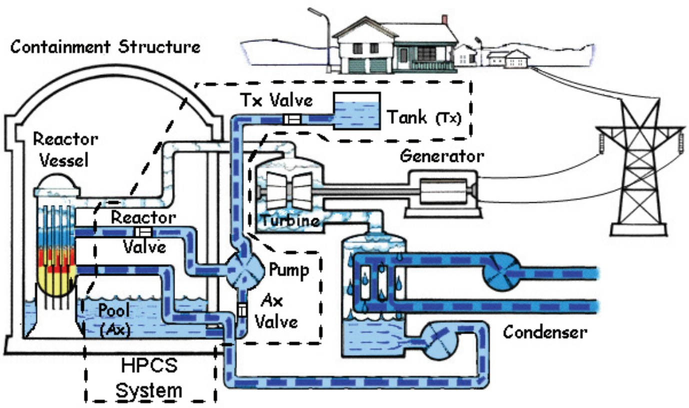


Figure 1. Operation of a nucleoelectric plant.

Our proposal associates to each possible event a possibility allowing us to make a decision based on the state of the parameters that are part of the different starting events, that is, the possible paths generated within the Small LOCA scenario. For that purpose, a representative set is implemented and analyzed (Fig. 1). The fuzzy cognitive map and the relations established between the different elements can be examined in Fig.2 .

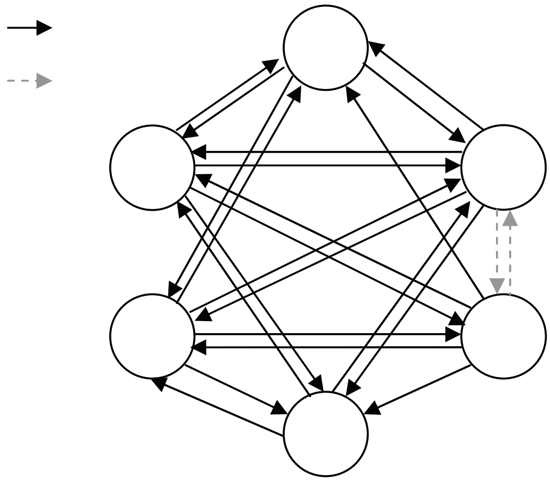


Figure 2. Fuzzy Cognitive Map of a representative set for the Small LOCA scenario.

4.1 A1 (Reactor Valve Failure)

- Once the event A1 is present, all that happens with A2 or M does not matter, because it is a first order failure, which implies affecting the system globally. In this case, the relation with those failures (A2 or M) is an excluding one (-1), *i.e.*, once the system fails, the effect of either A2 or M does not contribute to the system failure, and they are excluded.
- With regard to event B, this is independent (0). In this case, the fact that A1 is present does not affect B
- As for the operation of the HPCS system, this is in an inversely proportional relation (-1) with event A1.
- The relation between A1 and the event of the vessel being in good condition, is inversely proportional (-1).

4.2 A2 (Tank Valve Failure)

- The fact that event A2 is present, has no positive or negative effect on M (0).
- The fact that A2 is detected implies that there is no first order failure, for otherwise the effect of event A2 would have no importance, being excluded. Hence, the relation with events A1 and B is an excluding one (-1).
- The relation of event A2 with the fact that the HPCS system is in operation, has a possibility of (-0.5), although it is not determinant, for even, if event A2 were

present, the HPCS system could continue its operation, as long as events M, A1 and B are not present.
- With regard to the vessel condition, the effect can be negative if A2 is combined with event M, but it could be positive if event M is not present, and thus the resultant effect would be indifferent (0).

4.3 M (Manual Pool Valve Failure)

- The fact that event M is present has no positive or negative effect on A2 (0).
- The fact that event M is detected implies that there is no first order failure, because if there were, the effect of M would have no importance, being excluded. Thus the relation with A1 and B is an excluding one -1).
- There is a relation between event M and the operation of the HPCS system, although it is not determinant (-0.5), because, even if event M were present, the HPCS system could continue in operation if event A2 were not present, which implies that neither A1 nor B are present.
- The effect of failure M on the good condition of the vessel can be negative if it is combined with event A2, and positive if event M is not present, and thus the total effect is indifferent (0).

4.4 B (Pump Failure)

- Event B is independent of event A1 (0), which implies that whether event A1 is present or not, it has no influence, since the HPCS system would have already failed upon the presence of event B.
- The relation between event B and failures A2 and M is excluding (-1); *i.e.*, once the event B is present, events A2 and M do not contribute further to the overall effect on the system, since the HPCS system would be automatically inoperative.
- The relation of event B with respect to the operation of the HPCS system is inversely proportional (-1).
- The relation of event B with regard to the fact that the vessel is in good condition, is inversely proportional (-1).

4.5 When the HPCS is Operational

- The relation of this event respective to any of the failure events is inversely proportional (-1).
- With regard to the vessel being in good condition, this is directly proportional (1), because for a good operation of the HPCS system, the vessel must be in good condition.

4.6 When the Vessel is in Good Condition

- The relation of this event respective to any of the failure events is inversely proportional (-1).
- With regard to the operation of the HPCS system, the relation is directly proportional (1), because if the vessel is in good condition, then the HPCS system is operational.

5 Tests and their Interpretation

In the next section, we give a detailed account of the tests made with the fuzzy cognitive map presented in Fig. 2, which are represented in the following matrix:

Table 1. Matrix for the HPCS System in the LOCA scenario

	A1	A2	M	B	HPCS	Vessel
A1	0.000000	-1.000000	-1.000000	0.000000	-1.000000	-1.000000
A2	-1.000000	0.000000	0.000000	-1.000000	-0.500000	0.000000
M	-1.000000	0.000000	0.000000	-1.000000	-0.500000	0.000000
B	0.000000	-1.000000	-1.000000	0.000000	-1.000000	-1.000000
HPCS	-1.000000	-1.000000	-1.000000	-1.000000	0.000000	1.000000
Vessel	-1.000000	-1.000000	-1.000000	-1.000000	1.000000	0.000000

Table 2. First scenario, with event A1 present (reactor valve failure).The result is a scenario with failing HPCS system and vessel malfunction present.

	A1	A2	M	B	HPCS	Vessel
V_i	1	0	0	0	0	0
V_1	0.500000	0.006693	0.006693	0.500000	0.006693	0.006693
V_2	0.466586	0.006262	0.006262	0.466586	0.006693	0.006919
V_3	0.467376	0.008715	0.008715	0.467376	0.009354	0.009638
V_4	0.454598	0.008420	0.008420	0.454598	0.009293	0.009689
V_5	0.455342	0.009557	0.009557	0.455342	0.010565	0.010992
V_6	0.449336	0.009367	0.009367	0.449336	0.010496	0.010980
V_7	0.449907	0.009944	0.009944	0.449907	0.011148	0.011648
V_8	0.446845	0.009824	0.009824	0.446845	0.011090	0.011620
V_9	0.447249	0.010131	0.010131	0.447249	0.011436	0.011974
V_{10}	0.445627	0.010055	0.010055	0.445627	0.011393	0.011946
V_{11}	0.445900	0.010222	0.010222	0.445900	0.011580	0.012137
V_{12}	0.445023	0.010175	0.010175	0.445023	0.011550	0.012115
V_{13}	0.445201	0.010266	0.010266	0.445201	0.011652	0.012219
V_f	0.444722	0.010238	0.010238	0.444722	0.011633	0.012204

Table 3. Second scenario, with event A2 (tank valve failure) present. The result is a correct operation of the HPCS system and the vessel.

	A1	A2	M	B	HPCS	Vessel
V_i	0	1	0	0	0	0
V_1	0.006693	0.500000	0.500000	0.006693	0.075858	0.500000
V_2	0.000378	0.049915	0.049915	0.000378	0.483274	0.577462
V_3	0.003010	0.004930	0.004930	0.003010	0.933015	0.917782
V_4	0.000091	0.000093	0.000093	0.000091	0.989378	0.990387
V_5	0.000050	0.000050	0.000050	0.000050	0.992970	0.992938
V_6	0.000049	0.000049	0.000049	0.000049	0.993063	0.993066
V_7	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_8	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_9	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_{10}	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_{11}	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_{12}	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_{13}	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_f	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069

Table 4. Third scenario, with event M (pool valve failure) present. The result is a correct operation of the HPCS system and the vessel..

	A1	A2	M	B	HPCS	Vessel
V_i	0	0	1	0	0	0
V_1	0.006693	0.500000	0.500000	0.006693	0.075858	0.500000
V_2	0.000378	0.049915	0.049915	0.000378	0.483274	0.577462
V_3	0.003010	0.004930	0.004930	0.003010	0.933015	0.917782
V_4	0.000091	0.000093	0.000093	0.000091	0.989378	0.990387
V_5	0.000050	0.000050	0.000050	0.000050	0.992970	0.992938
V_6	0.000049	0.000049	0.000049	0.000049	0.993063	0.993066
V_7	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_8	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_9	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_{10}	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_{11}	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_{12}	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_{13}	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069
V_f	0.000049	0.000049	0.000049	0.000049	0.993068	0.993069

Table 5. Fourth scenario, with event B (pump failure) present. The result is the failure of the HPCS system and a vessel malfunction.

	A1	A2	M	B	HPCS	Vessel
V_i	0	0	0	1	0	0
V_1	0.500000	0.006693	0.500000	0.006693	0.006693	0.006693
V_2	0.466586	0.006693	0.006693	0.466586	0.006693	0.006919
V_3	0.467376	0.006262	0.006262	0.467376	0.009354	0.009638
V_4	0.454598	0.008715	0.008715	0.454598	0.009293	0.009689
V_5	0.455342	0.008420	0.008420	0.455342	0.010565	0.010992
V_6	0.449336	0.009557	0.009557	0.449336	0.010496	0.010980
V_7	0.449907	0.009367	0.009367	0.449907	0.011148	0.011648
V_8	0.446845	0.009944	0.009944	0.446845	0.011090	0.011620
V_9	0.447249	0.009824	0.009824	0.447249	0.011436	0.011974
V_{10}	0.445627	0.010131	0.010131	0.445627	0.011393	0.011946
V_{11}	0.445900	0.010055	0.010055	0.445900	0.011580	0.012137
V_{12}	0.445023	0.010222	0.010222	0.445023	0.011550	0.012115
V_{13}	0.445201	0.010175	0.010175	0.445201	0.011652	0.012219
V_f	0.444722	0.010266	0.010266	0.444722	0.011633	0.012204

Table 6. Fifth scenario, with events A2 (valve tank failure) and M (pool valve failure) present. The result is the failure of the HPCS system and a vessel malfunction..

	A1	A2	M	B	HPCS	Vessel
V_i	0	1	1	0	0	0
V_1	0.000045	0.500000	0.500000	0.000045	0.000045	0.000045
V_2	0.006690	0.499830	0.499830	0.006690	0.006691	0.006691
V_3	0.006273	0.474932	0.474932	0.006273	0.006485	0.006485
V_4	0.008049	0.476229	0.476229	0.008049	0.008329	0.008329
V_5	0.007802	0.469505	0.469505	0.007802	0.008153	0.008153
V_6	0.008354	0.470339	0.470339	0.008354	0.008729	0.008729
V_7	0.008238	0.468245	0.468245	0.008238	0.008635	0.008635
V_8	0.008419	0.468652	0.468652	0.008419	0.008822	0.008822
V_f	0.008370	0.467969	0.467969	0.008370	0.008779	0.008779

The interpretation of the results considers the following rounding off:

0.011633, 0.012204 and 0.008779 are taken as 0.

0.993068 and 0.993069 are taken as 1.

Where:

0 means the absence of an element, or either its opposite effect.

1 means the element is present.

These results were compared with an expert's analysis; all of them were congruent. Other tests were performed, like considering: “HPCS operational” or “vessel in good condition", or "HPCS and Vessel operational", the results were, as should be expected, "HPCS and vessel functioning correctly" in all of them.

6 Conclusions

Fuzzy cognitive maps allow handling the complexities of failure analysis in the system operation, involving all the elements of the system considered as a whole, especially if we compare it with the traditional failure trees method used by the expert, where each element of the system is considered individually in the reasoning process.

The interpretation of the results obtained helps, in an automatic fashion, to the supervisor in charge of overseeing the plant performance to make decisions about critical situations. This point is of paramount importance when the available information is so extensive as to make difficult making an adequate decision. With the interpretation of the results obtained in real time according to the present state of the plant, they will allow to make a faster and more reliable decision.

The challenge will be to model most of the systems intervening in the plant operation, to have all the time assistance to make decisions automatically.

References

1. Huerta, A., Aguilar, O., Núñez, A., López, R.: Análisis de Eventos Internos para la Unidad 1 de la Central Nucleoeléctrica de Laguna Verde –Eventos Iniciadores y Secuencias de Accidentes. Comisión Nacional de Seguridad Nuclear y Salvaguardias (1993)
2. Khan, M.S., Chong, A., Quaddus, M.: Fuzzy Cognitive Maps and Intelligent Decision Support –A Review. School of Information Technology, Murdoch University, Graduate School of Business, Curtin University of Technology, GPO Box U (1987)
3. Konar, A.: Artificial Intelligence and Soft Computing –Behavioral and Cognitive Modeling of the Human Brain. CRC Press (2001)
4. Kosko, B.: Fuzzy Cognitive Maps. International Journal of Man-Machine Studies 24 (1986) 65-75
5. Kosko, B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, New Jersey (1992)
6. Laureano-Cruces, A., De Arriaga-Gómez, F., García-Alegre, M.: Cognitive task analysis: a proposal to model reactive behaviors. Journal of Experimental & Theoretical Artificial Intelligence 13 (2001) 227-239
7. Laureano-Cruces, A., Ramírez-Rodríguez, J., Terán-Gilmore, A.: Evaluation of the Teaching-Learning Process with Fuzzy Cognitive Maps. IBERAMIA-04 Advances in Artificial Intelligence. Lecture Notes in Artificial Intelligence Series, Vol. 3315. Springer-Verlag (2004) 922-931
8. Laureano-Cruces, A., Espinosa-Paredes, G.: Behavioral design to model a reactive decision of an expert in geothermal wells. International Journal of Approximate Reasoning 39 (2005) 1-28

9. Laureano-Cruces, A., Ramírez-Rodríguez, J., Mora-Torres, M., Terán-Gilmore, A.: Inference Engine to Elicit the Cognitive and Emotive Interventions of a Pedagogical Agent. International Journal of Intelligent Systems. Wiley (Submitted: March 2006)
10. Peláez, C. E., Bowles, J. B.: Applying Fuzzy Cognitive-Maps Knowledge-Representation to Failure Modes Effects Analysis. Proceedings Annual Reliability and Maintainability Symposium X. IEEE (1995) 149-144
11. Peláez, C. E., Bowles, J. B.: Using Fuzzy Cognitive Maps as a System Model for Failure Modes and Effects Analysis. Information Sciences 88. Elsevier Science Inc. (1996) 177-199
12. Stylios, C. D., Groumpos, P. P.: Fuzzy Cognitive Maps: a model for intelligent supervisory control systems. Computers in Industry 39. Elsevier Science B.V. (1999) 229-238
13. Uhrig, R. E., Tsoukalas, L. H.: Multi-Agent Based Anticipatory Control for Enhancing the Safety and Performance of Generation-IV Nuclear Power Plants During Long-Term Semi-Autonomous Operation. Progress in Nuclear Energy, Vol. 43. Elsevier Science Ltd (2003) 113-120

A Versatile Characterization of Causal Relationships

Jixin Ma, Brian Knight and Miltos Petridis

School of Computing and Mathematical Sciences, the University of Greenwich, U.K.
j.ma@gre.ac.uk

Abstract. This paper introduces a framework for representing versatile temporal relationships between events and their effects. The framework is based on a simple time model which characterizes each time element as a subset of the set of real numbers and allows expression of both absolute time values and relative temporal relations. The formalism presented here formally specifies the so-called most general temporal constraint (GTC), which guarantees the common-sense assertion that "the beginning of the effect cannot precede the beginning of the cause". It is shown that there are in fact 8 possible causal relationships which satisfy GTC, including cases where, on the one hand, effects start simultaneously with, during, immediately after, or some time after their causes, and on the other hand, events end before, simultaneously with, or after their causes. The causal relationships characterized in this paper are versatile enough to subsume those representatives in the literature.

1 Introduction

Representing and reasoning about events and their effects is essential in modeling the dynamic aspects of the world. Over the past 40 decades, a multitude of alternative formalisms have been proposed in this area, including McCarthy and Hayes' framework of the situation calculus [17, 18], McDermott's temporal logic [19], Allen's interval based theory [1, 2], Kowalski and Sergot's event calculus [11], Shoham's point-based reified logic and theory [26, 27], and Terenziani and Torasso's theory of causation [28]. In particularly, noticing that temporal reasoning plays an important role in reasoning about actions/events and change, a series of revised formalisms have been introduced to characterize richer temporal features in the situation calculus or the event calculus, such that of Lifschitz [12], of Sandewall [23], of Schubert [24], of Gelfond et al. [10], of Lin and Shoham [13], of Pinto and Reither [21, 22], of Miller and Shanahan [20, 25], and of Baral et al. [5, 6, 7].

In most existing formalisms for representing causal relationships between events and their effects, such as the situation calculus and the event calculus, the result of an event is represented by the effect takes place immediately after the occurrence of the event. However, as noted by Allen and Ferguson [3], temporal relationships between events and their effects can in fact be quite complicated. In some cases, the effects of an event take place immediately after the end of the event and remain true until some further events occur. E.g., in the block-world, as soon as the action "moving a block from the top of another block onto the table" is completed, the block being moved

© A. Gelbukh, C.A. Reyes-García. (Eds.) *Received 31/05/06*
Advances in Artificial Intelligence. *Accepted 03/10/06*
Research in Computing Science 26, 2006, pp. 15-24 *Final version 05/10/06*

should be on the table (immediately). However, sometimes there may be a time delay between an event and its effect(s). E.g., 30 seconds after you press the button at the crosswalk, the pedestrian light turns to green [10]. Also, in some other cases, the effects of an event might start to hold while the event is in progress, and stop holding before or after the end of the event. Examples can be found later in the paper.

The objective of this paper is to propose a framework, which allows expression of versatile temporal causal relationships between events and their effects. As the temporal basis for the formalism, a simple point-based time model is presented in section 2, allowing expression of both absolute time values and relative temporal relations. In section 3, fluents and states are associated with times in the manner of temporal reification [16, 26]. Section 4 deals with representation of event/action and change, as well as temporal constraints on the causal relationships between events and their effects. Finally, section 5 concludes the paper.

2 A Simple Time Model

In what follows in this paper, we shall use **R** to denote the set of real numbers, and **T**, the set of time elements. Each time element t is defined as a subset of **R** and must be in one of the following four forms:

$$(p_1, p_2) = \{p \mid p \in \mathbf{R} \wedge p_1 < p < p_2\}$$
$$[p_1, p_2) = \{p \mid p \in \mathbf{R} \wedge p_1 \le p < p_2\}$$
$$(p_1, p_2] = \{p \mid p \in \mathbf{R} \wedge p_1 < p \le p_2\}$$
$$[p_1, p_2] = \{p \mid p \in \mathbf{R} \wedge p_1 \le p \le p_2\}$$

In the above, p_1 and p_2 are real numbers, and we shall call them the left-bound and right-bound of time element t, respectively. The absolute values as for the left and/or right bounds of some time elements might be unknown. In this case, real number variables are used for expressing relative relations to other time elements.

In this paper, if the left-bound and right-bound of time element t are the same, we shall call t a time point, otherwise t is called a time interval. Without confusion, we shall take time element [p, p] as identical to p. Also, if a time element is not specified as open or closed at its left (right) bound, we shall use “<” instead of “(” and “[” as for its left bracket; similarly, we shall use “>” instead of “)” and “]” as for its right bracket. In addition, we define the duration of a time element t, Dur(t), as the distance between its left bound and right bound. In other words:

(2.1) $t = <p_1, p_2> \Rightarrow Dur(t) = p_2 - p_1$

Following Allen’s terminology [1], we shall use Meets to denote the immediate predecessor order relation over time elements:

(2.2) $Meets(t_1, t_2) \Leftrightarrow \exists p_1,p,p_2 \in \mathbf{R}(t_1 = <p_1, p) \wedge t_2 = [p, p_2> \vee t_1 = <p_1, p] \wedge t_2 = (p, p_2>)$

It is easy to see that the intuitive meaning of $Meets(t_1, t_2)$ is that, on the one hand, time elements t_1 and t_2 don’t overlap each other (i.e., they don’t have any part in common, not even a point); on the other hand, there is not any other time element standing between them.

N.B. For any two time elements t_1 and t_2 such that Meets(t_1, t_2), t_1 and t_2 define a unique time element as the "ordered-union" of t_1 and t_2, denoted as $t_1 \oplus t_2$.

Analogous to the 13 exclusive relations introduced by Allen for intervals [1, 2], in this paper, we shall use **TR** to denote the set of exclusive temporal order relations over time elements including both time points and time intervals:

TR = {Equal, Before, After, Meets, Overlaps, Overlapped-by, Met-by, Starts, Started-by, During, Contains, Finishes, Finished-by}

It is important to note that, for a given pair of time elements, some of the relations in **TR** may be non-applicable. In fact, when the pair of time elements, t_1 and t_2, are specified as a point and a point, a point and an interval, an interval and a point, and an interval and an interval, respectively, all the exclusive temporal order relations between t_1 and t_2 can be classified into the following four groups, which we shall call the Comprehensive Temporal Order Relations (CTOR):

- 3 relations relating a point to a point:
  {Equal, Before, After}
- 7 relations relating a point to an interval:
  {Before, After, Meets, Met-by, Starts, During. Finishes}
- 7 relations relating an interval to a point:
  {Before, After, Meets, Met-by, Started-y, Contains, Finished-by}
- 13 relations relating an interval to an interval:
  {Equal, Before, After, Meets, Met-by, Overlaps, Overlapped-by, Starts, Started-by, During, Contains, Finishes, Finished-by}

The definition of the derived temporal order relations in terms of the single relation Meets is straightforward [4]. In fact:

$\text{Equal}(t_1, t_2) \Leftrightarrow \exists t', t'' \in \mathbf{T}(\text{Meets}(t', t_1) \wedge \text{Meets}(t', t_2) \wedge \text{Meets}(t_1, t'') \wedge \text{Meets}(t_2, t''))$
$\text{Before}(t_1, t_2) \Leftrightarrow \exists t \in \mathbf{T}(\text{Meets}(t_1, t) \wedge \text{Meets}(t, t_2))$
$\text{Overlaps}(t_1, t_2) \Leftrightarrow \exists t, t_3, t_4 \in \mathbf{T}(t_1 = t_3 \oplus t \wedge t_2 = t \oplus t_4)$
$\text{Starts}(t_1, t_2) \Leftrightarrow \exists t \in \mathbf{T}(t_2 = t_1 \oplus t)$
$\text{During}(t_1, t_2) \Leftrightarrow \exists t_3, t_4 \in \mathbf{T}(t_2 = t_3 \oplus t_1 \oplus t_4)$
$\text{Finishes}(t_1, t_2) \Leftrightarrow \exists t \in \mathbf{T}(t_2 = t \oplus t_1)$
$\text{After}(t_1, t_2) \Leftrightarrow \text{Before}(t_2, t_1)$
$\text{Overlapped-by}(t_1,t_2) \Leftrightarrow \text{Overlaps}(t_2, t_1)$
$\text{Started-by}(t_1, t_2) \Leftrightarrow \text{Starts}(t_2, t_1)$
$\text{Contains}(t_1, t_2) \Leftrightarrow \text{During}(t_2, t_1)$
$\text{Finished-by}(t_1,t_2) \Leftrightarrow \text{Finishes}(t_2,t_1)$,
$\text{Met-by}(t_1,t_2) \Leftrightarrow \text{Meets}(t_2,t_1)$

For the convenience of expression, we define two non-exclusive temporal relations as below:

$\text{In}(t_1, t_2) \Leftrightarrow \text{Starts}(t_1, t_2) \vee \text{During}(t_1, t_2) \vee \text{Finishes}(t_1, t_2)$
$\text{Sub}(t_1, t_2) \Leftrightarrow \text{Equal}(t_1, t_2) \vee \text{In}(t_1, t_2)$

Another important fact needs to be pointed out is that the distinction between the assertion that "point p Meets interval t" and the assertion that "point p Starts interval t" is critical: while Starts(p, t) states that point p is the starting part of interval t, Meets(p, t) states that point p is one of the immediate predecessors of interval t but p is not a part of t at all. In other words, Starts(p, t) implies interval t is left-closed at point p, and Meets(p, t) implies interval t is left-open at point p. Similarly, this applies to the distinction between the assertion that "interval t is Finished-by point p" and the assertion that "interval t is Met-by point p", i.e., Finished-by(t, p) implies interval t is right-closed at point p, and Met-by(t, p) implies interval t is right-open at point p.

As mentioned earlier, the simple point-based time model introduced here allows the openness (or closeness) of some interval at their left and/or right bounds to be unspecified. Such an approach provides a satisfactory representation of possibly incomplete relative temporal knowledge, and hence retains the appealing characteristics of interval-based [1], and point&interval-based [15] temporal systems. Specially, it can successfully bypass puzzles like the so-called Dividing Instant Problem [1, 8, 9, 14, 29].

3 Fluents and States

Representing the dynamic aspects of the world usually involves reasoning about various states of the world under consideration. In this paper, we shall define a state (denoted by, possibly scripted, s) of the world in the discourse as a collection of fluents (denoted by, possibly scripted, f), where a fluent is simply a Boolean valued proposition whose truth-value is dependent on the time.

The set of fluents, **F**, is defined as the minimal set closed under the following two rules:

(3.1) $f_1, f_2 \in \mathbf{F} \Rightarrow f_1 \vee f_2 \in \mathbf{F}$

(3.2) $f \in \mathbf{F} \Rightarrow \text{not}(f) \in \mathbf{F}$

In order to associate a fluent with a time element, we shall use Holds(f, t) to denote that fluent f holds true over time t.

As pointed out by Allen and Ferguson [3], as well as by Shoham [26], there are two ways we might interpret the negative sentence. In what follows, the sentence-negation will be symbolized "¬", e.g., ¬Holds(t, f), distinguished from the negation of fluents, e.g., not(f) [9]. In the weak interpretation, ¬Holds(t, f) is true if and only if it is not the case that f is true throughout t, and hence ¬Holds(t, f) is true if f changes truth-values over time t. In the strong interpretation of negation, ¬Holds(t, f) is true if and only if f holds false throughout t, so neither Holds(t, f) nor ¬Holds(t, f) would be true in the case that fluent f is true over some sub-interval of t and also false over some other sub-interval of t.

In this paper, we take the weak interpretation of negation as the basic construct:

(3.3) $\text{Holds}(f, t) \Rightarrow \forall t'(\text{Sub}(t', t) \Rightarrow \text{Holds}(f, t'))$

That is, if fluent f holds true over a time element t, then f holds true over any part of t.

(3.4) $\text{Holds}(f_1 \vee f_2, t) \Leftrightarrow \text{Holds}(f_1, t) \vee \text{Holds}(f_2, t)$

That is, if fluent f_1 or fluent f_2 holds true over time t, then at least one of them holds true over time t.

(3.5) $\text{Holds}(f, t_1) \wedge \text{Holds}(f, t_2) \wedge \text{Meets}(t_1, t_2) \Rightarrow \text{Holds}(f, t_1 \oplus t_2)$

That is, if fluent f holds true over two time elements t_1 and t_2 that meets each other, then f holds over the ordered-union of t_1 and t_2.

Following the approach proposed in [25], we use Belongs(f, s) to denote that fluent f belongs to the collection of fluents representing state s:

(3.6) $s_1 = s_2 \Leftrightarrow \forall f(\text{Belongs}(f, s_1) \Leftrightarrow \text{Belongs}(f, s_2))$

That is, two states are equal if and only if they contain the same fluents.

(3.7) $\exists s \forall f(\neg \text{Belongs}(f, s))$

That is, there exists a state that is an empty set.

(3.8) $\forall s_1 f_1 \exists s_2(\forall f_2(\text{Belongs}(f_2, s_2) \Leftrightarrow \text{Belongs}(f_2, s_1) \vee f_1 = f_2))$

That is, any fluent can be added to an existing state to form a new state.

Without confusion, we also use Holds(s, t) to denote that state s holds true over time t, provided:

(3.9) $\text{Holds}(s, t) \Leftrightarrow \forall f(\text{Belongs}(f, s) \Rightarrow \text{Holds}(f, t))$

4 Events, Effects and Causal Relationships

The concepts of change and time are deeply related since changes are caused by events occurring over the time. In order to express the occurrence of events (denoted by e, possibly scripted), following Allen's approach [2], we use Occur(e, t) to denote that event e occurs over time t, and impose the following axiom:

(4.1) $\text{Occur}(e, t) \Rightarrow \forall t'(\text{In}(t', t) \Rightarrow \neg \text{Occur}(e, t'))$

We shall use formula Changes(t_1, t, t_2, s_1, e, s_2) to denote a causal law, which intuitively states that, under the precondition that state s_1 hold over time t_1, the occurrence of event e over time t will change the world from state s_1 into state s_2, which holds over time t_2. Formally, we impose the following axiom about causality to ensure that if the precondition of a causal law holds and the event happens, then the effect expected to be caused must appear:

(4.2) $\text{Changes}(t_1, t, t_2, s_1, e, s_2) \wedge \text{Holds}(s_1, t_1) \wedge \text{Occur}(e, t) \Rightarrow \text{Holds}(s_2, t_2)$

In order to characterize temporal relationships between events and their effects, we impose the following temporal constraints:

(4.3) $\text{Changes}(t_1, t, t_2, s_1, e, s_2) \Rightarrow \text{Meets}(t_1, t) \wedge (\text{Meets}(t_1, t_2) \vee \text{Before}(t_1, t_2))$

It is important to note that axiom (4.3) presented above actually specifies the so-called (most) general temporal constraint (GTC) (see [2, 19, 27, 28]). Such a GTC guarantees the common-sense assertion that "the beginning of the effect cannot precede the beginning of the cause".

There are in fact 8 possible temporal order relations between times t_1, t and t_2 which satisfy (4.3). These are illustrated in Figure 1 as below:

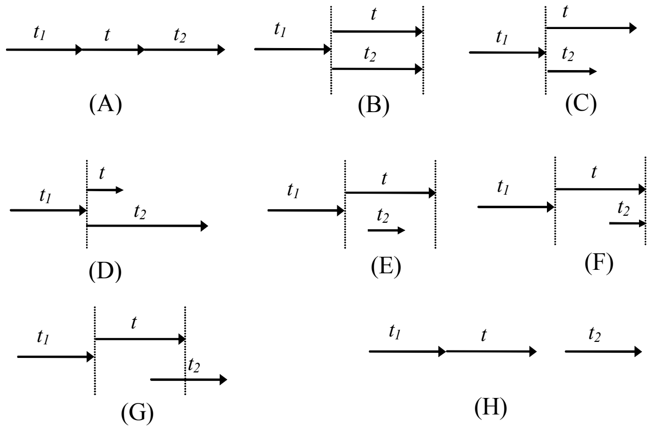


Figure 1. Temporal order relations between times t_1, t and t_2 which satisfy (4.3).

- Case (A) where the effect becomes true immediately after the end of the event and remains true for some time after the event. E.g., the event of putting a book on the table has the effect that the book is on the table immediately after the event is completed.
- Case (B) where the effect holds only over the same time over which the event is in progress. E.g., as the effect of the event of pressing the horn of a car, the horn makes sounds only when the horn is being pressed.
- Case (C) where the beginning of the effect coincides with the beginning of the event, and the effect ends before the event completes. E.g., consider the case where a landmine is set on the first half of a bridge. If someone is walking through the bridge, he will be in danger just over the first half of the bridge crossing.
- Case (D) where the beginning of the effect coincides with the beginning of the event, and the effect remains true for some time after the event. E.g., as the effect of the event of pressing the button of the bell on the door (say, for one second), the bell sounds a tune for fifteen seconds.
- Case (E) where the effect only holds over some time during the progress of the event. E.g., he goes through a wall of tiredness over the fiftieth minute of the event of running for four hours.
- Case (F) where the effect becomes true during the progress of the event and remains true until the event completes. E.g., consider the event of discharging some water from a basket by means of lifting one side of the basket. In the case where the basket is not full, the effect that the water flows out takes place only after it has been lifted to the edge of the basket, and will keep flowing out until the event ends.
- Case (G) where the effect becomes true during the progress of the event and remains true for some time after the event. E.g., he becomes tired for days after the thirtieth minute of the event of running along the track for three hours.

- Case (H) where there is a time delay between the event and its effect. E.G., 25 seconds after the button at the crosswalk is pressed, the pedestrian light turns to yellow; and after another 5 seconds, it turns to green.

As mentioned in the introduction, various theories have been proposed for representing and reasoning about action/event and change. However, the temporal causal relationships between events and their effects as specified in most of the existing formalism are quite limited. An exception is the relatively general theory of Time, Action-Types, and Causation, introduced by Terenziani and Torasso's in the middle of last 90s [28]. Due to the limit to the length of the paper, in what follows, we shall briefly demonstrate that the causal relationships characterized in this paper are more general than that of Terenziani and Torasso, and therefore versatile enough to subsume those representatives in the literature. In fact:

If t and t_2 are specified as a point and a point, a point and an interval, an interval and a point, and an interval and an interval, respectively, by applying the CTOR as classified in section 2, we can reach the following four theorems straightforwardly:

(Th1) $Changes(t_1, t, t_2, s_1, e, s_2) \wedge Dur(t) = 0 \wedge Dur(t_2) = 0$
$\Rightarrow$ $Equal(t, t_2) \vee Before(t, t_2)$

That is, if the event and the effect are both punctual, then either the event precedes (strictly) the effect, or they coincide with each other (i.e., they happens simultaneously at the same time point).

(Th2) $Changes(t_1, t, t_2, s_1, e, s_2) \wedge Dur(t) = 0 \wedge Dur(t_2) > 0$
$\Rightarrow$ $Starts(t, t_2) \vee Meets(t, t_2) \vee Before(t, t_2)$

That is, if the event is punctual and the effect is durative, then either the event precedes (immediately or strictly) the effect, or the event coincides with the beginning of the effect.

(Th3) $Changes(t_1, t, t_2, s_1, e, s_2) \wedge Dur(t) > 0 \wedge Dur(t_2) = 0$
$\Rightarrow$ $Started\text{-}by(t, t_2) \vee Contains(t, t_2)$
$\vee\ Finished\text{-}by(t, t_2) \vee Meets(t, t_2) \vee Before(t, t_2)$

That is, if the event is durative and the effect is punctual, then either the event precedes (immediately or strictly) the effect, or the effect coincides with either the beginning or the end of the event, or the effect happens during the event's occurrence.

(Th4) $Changes(t_1, t, t_2, s_1, e, s_2) \wedge Dur(t) > 0 \wedge Dur(t_2) > 0$
$\Rightarrow$ $Started\text{-}by(t, t_2) \vee Contains(t, t_2) \vee Finished\text{-}by(t, t_2) \vee Equal(t, t_2)$
$\vee\ Starts(t, t_2) \vee Overlaps(t, t_2) \vee Meets(t, t_2) \vee Before(t, t_2)$

That is, if both the event and the effect are durative, then the beginning of the event either precedes (immediately or strictly) or coincides with the beginning of the effect, where the end of the event can either precedes (immediately or strictly), coincides with, or succeeds (immediately or strictly) the end of the effect.

It is easy to see that (Th1) and (Th4) are equivalent to Terenziani and Torasso's Theorem 4 (p.541, [28]) and Theorem 1 (p.540, [28]), respectively, while (Th2) and (Th3) can be seen as the extension to Terenziani and Torasso's Theorem 3 and

Theorem 2 (p.541, [28]), respectively. This is due to the fact that, while the "Meets" relation between a punctual event and a durative effect, and between a durative event and a punctual effect, is accommodated in (Th2) and (Th3), respectively, Terenziani and Torasso's Theorem 3 and Theorem 2 do not allow such relations.

In fact, follow Terenziani and Torasso's Theorem 3, either there must be a gap between the punctual cause and its durative effect, or the punctual cause must coincide with the beginning part of its durative effect. In other words, the interval over which the effect happens must be either "After" or "closed at" the point at which the cause happens. Therefore, the case where a punctual cause "Meets" its durative effect (that is, the interval over which the effect happens is "open" at the point at which the cause happens) is not allowable. However, consider the following example:

Immediately after the power was switched on, a robot that had been stationary started moving.

If we use $s_{Stationary}$ to represent the state that "the robot was stationary, $e_{SwitchOn}$ to represent the event that "the power was switched on", and s_{Moving} to represent the corresponding effect that "the robot was moving", then

$$Changes(t_{Stationary}, t_{SwitchOn}, t_{Moving}, s_{Stationary}, e_{SwitchOn}, s_{Moving})$$

should be consistent with:

$$Meets(t_{SwitchOn}, t_{Moving})$$

That is, the "Switching" point $t_{SwitchOn}$ is immediately followed by the "Moving" interval, but not included in the "Moving" interval itself. In other word, the robot was moving immediately after the "Switching" point $t_{SwitchOn}$, but at the time point when the power was switching on, the robot was not moving. Obviously, such a scenario cannot be expressed in Terenziani and Torasso's Theorem 3.

Similarly, in Terenziani and Torasso's Theorem 2, the case where a durative event "Meets" its punctual effect (that is, the interval over which the cause happens is "open" at the point at which the effect happens) is not allowable. Then again, consider the following example:

Immediately after the ball being falling down from the air, it touched the ground.

If we use s_{InAir} to represent the precondition that "the ball was at a certain position in the air", $e_{FallingDown}$ to represent the event that "the ball was falling down", and $s_{TouchGround}$ to represent effect that "the ball touched the ground", respectively, then

$$Changes(t_{InAir}, t_{FallingDown}, t_{TouchGround}, s_{InAir}, e_{FallingDown}, s_{TouchGround})$$

should be consistent with:

$$Meets(t_{FallingDown}, t_{TouchGround})$$

That is, the interval over which the ball was falling down is immediately followed by the point when the ball touched the ground, but does not include the point itself. In other word, the ball was falling down immediately before the instant when it touched the group, but at the time point when the ball touched the ground, the ball was no longer falling down. Again, such a scenario is not allowed in Terenziani and Torasso's Theorem 2.

5 Conclusions

Based on a simple point-based time model which allows expression of both absolute time values and relative temporal relations, we have presented in this paper a framework for representing flexible temporal causal relationships. The formalism presented here formally specifies the most general temporal constraint (GTC), ensuring the common-sense assertion that "the beginning of the effect cannot precede the beginning of the cause". It is shown that the causal relationships characterized here are versatile enough to subsume those representatives in the literature. Ideally, any useful theory about action/event and change has to be able to handle the frame problem adequately. However, due to the length of this paper, we didn't tackle such a problem. An interesting topic for further research is to extend this framework to include representing and reasoning about concurrent actions/events and their effects.

References

1. Allen J.: Maintaining Knowledge about Temporal Intervals, Communication of ACM. Vol. (26), (1983) 832-843.
2. Allen J.: Towards a General Theory of Action and Time, Artificial Intelligence, Vol. 23, (1984) 123-154.
3. Allen J. and Ferguson G.: Actions and Events in Interval Temporal Logic, the Journal of Logic and Computation, Vol. (4:5), (1994) 531-579.
4. Allen J. and Hayes P.: Moments and Points in an Interval-based Temporal-based Logic, Computational Intelligence, Vol. (5), (1989) 225-238.
5. Baral C.: Reasoning about actions: non-deterministic effects, constraints, and qualification. In Proceedings of IJCAI'95, (1995) 2017-2023.
6. Baral C. and Gelfond M.: Reasoning about Effects of Concurrent Actions. Journal of Logic Programming, vol. (31:1-3), (1997) 85-117.
7. Baral C, Son T. and Tuan L.: A transition function based characterization of actions with delayed and continuous effects. In Proceedings of KR'02, (2002) 291-302.
8. van Benthem J.: The Logic of Time, Kluwer Academic, Dordrech, (1983).
9. Galton A.: A Critical Examination of Allen's Theory of Action and Time, Artificial Intelligence, Vol. (42), (1990) 159-188.
10. Gelfond M., Lifschitz V. and Rabinov A.: What are the Limitations of the Situation Calculus? In Working Notes of AAAI Spring Symposium Series. Symposium: Logical Formalization of Commonsense Reasoning, (1991) 59-69.
11. Kowalski R. and Sergot M.: A Logic-based Calculus of Events, New Generation Computing, Vol. (4), (1986) 67-95.
12. Lifschitz V.: Formal theories of action, In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, (1987) 966-972.
13. Lin F. and Shoham Y.: Concurrent Actions in the Situation Calculus, in Proceedings of AAAI-92, (1992) 590-595.
14. Ma, J. and Hayes, P.: Primitive Intervals Vs Point-Based Intervals: Rivals Or Allies?", the Computer Journal, Vol.49(1). (2006) 32-41
15. Ma J. and Knight B.: A General Temporal Theory, The Computer Journal, Vol. (37:2), (1994) 114-123.
16. Ma, J. and Knight B.: A Reified Temporal Logic, The Computer Journal, Vol. (39:9), (1996) 800-807.

17. McCarthy J.: Situations, actions and causal laws, Stanford Artificial Intelligence Project: Memo 2, (1963).
18. McCarthy J. and Hayes P. J.: Some philosophical problems from the standpoint of artificial intelligence, in Machine Intelligence, Vol. (4), Eds. Meltzer B. and Michie D., Edinburgh University Press, (1969) 463-502.
19. McDermott D.: A Temporal Logic for Reasoning about Processes and Plans, Cognitive Science, Vol. (6), (1982) 101-155.
20. Miller R. and Shanahan M.: Narratives in the Situation Calculus, the Journal of Logic and Computation, Vol. (4:5), (1994) 513-530.
21. Pinto J. and Reiter R.: Temporal Reasoning in Logic Programming: A Case for the Situation Calculus, In Proceedings of 10th Int. Conf. on Logic Programming, Budapest, Hungary, (1993) 21-24.
22. Pinto J. and Reiter R.: Reasoning about Time in the Situation Calculus, Annals of Mathematics and Artificial Intelligence, Vol. (14:2-4), (1995) 251-268.
23. Sandewall E.: Filter preferential entailment for the logic of action in almost continuous worlds. In Proceedings of the 12th International Joint Conference on Artificial Intelligence, (1989) 894-899.
24. Schubert L.: Monotonic Solution of the Frame Problem in the Situation Calculus: an Efficient Method for Worlds with Fully Specified Actions, in: H.E. Kyburg, R. Loui and G. Carlson, eds., Knowledge Representation and Defeasible Reasoning pp.23-67, Kluwer Academic Press, (1990).
25. Shanahan M.: A Circumscriptive Calculus of Events, Artificial Intelligence, Vol. (77), (1995) 29-384.
26. Shoham Y.: Temporal logics in AI: Semantical and Ontological Considerations, Artificial Intelligence, Vol (33), (1987) 89-104.
27. Shoham Y.: Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence, MIT Press, (1988).
28. Terenziani P. and Torasso P.: Time, Action-Types, and Causation: an Integrated Analysis, Computational Intelligence, Vol. (11:3), (1995) 529-552.
29. Vila, L.: A survey on temporal Reasoning in Artificial Intelligence. AI Communications, Vol. (7), (1994) 4-28.

Evolutionary Algorithms

Multi-Objective Evolutionary Algorithms with Immunity for SLAM

Meiyi Li

College of Information Engineering, Xiangtan University.
Xiangtan China, 4110005
meiy_li@yahoo.com.cn

Abstract. The simultaneous localization and mapping problem with evolutionary computations is translated to a multi-objective optimization problem since it possesses of characters of multi-objective at a certain extent, and in order to efficiently solve the simultaneous localization and mapping problem, a local searcher with immunity is constructed. The local searcher employs domain knowledge, which is named as key point grid which is developed in the paper. The experiment results of a real mobile robot indicate that the computational expensiveness of designed algorithms is less than evolutionary algorithms of single-objection for simultaneous localization and map-ping and accuracy of obtained maps are better.

1 Introduction

SLAM (Simultaneous Localization And Mapping or Concurrent Mapping and Localization) is to acquire a map of an unknown environment with a moving robot, while simultaneously localizing the robot relative to this map[1-2]. The SLAM problem addresses situations where the robot lacks a global positioning sensor, and instead has to rely on sensors of incremental ego-motion for robot position estimation (e.g., odometry, inertial navigation. etc.). Such sensors accumulate error over time, making the problem of acquiring an accurate map into a challenging one. This is a hard problem because noisy sensor data must be simultaneously used for both mapping and localization. Within mobile robotics, the SLAM problem is often referred to as one of the most challenging ones[3-4].

Traditional algorithms are based on extended Kalman filters (EKFs) [5-6]. However, several problems arise when applying the Extended Kalman Filter approach. Especially, this method is not able to deal with uncertainty as follows: the combinatorial labeling problem of data association (e.g., landmark identification, feature recognition and place recognition, etc.) in which a correspondence must be found between sensor measurements and the features already represented in the map.

In order to overcome the difficulty about the data association problem, Reference [7] and [8] employed evolutionary computations to solve SLAM problem. Their investigations indicate that evolutionary computations is a hopeful approach for NP-hard SLAM problem.

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 27-36

Received 18/06/06
Accepted 03/10/06
Final version 09/10/06

In SLAM occupancy grid present by Moravec and Elfes[9] often is employed, where a map is consisted of grids or cells. A occupancy grid is described as grid[i][j], and every grid[i][j] has a probability or belief occ[i][j] which is occupied and a probability or belief emp[i][j] which is free. The calculations of occ[i][j] and emp[i][j] depended on data from range-finder. Usually, since the reliable degree of data from range-finder is relevant to the distance. The reliable degree of data from range-finder is projected to occ[i][j] and emp[i][j] by means of a sensor fusion approach like theories of evidence. In this paper Dempster-Shafer theory of evidence is employed and works as a sensor fusion.

SLAM possesses of characters of multi-objective at a certain extent since several formulations are combined with weights and the fitness is defined. From Reference [8], we can get that the target of evolutionary computations for SLAM is to solve multi-objective problem(MOP) as follow.

The reliable degree of data from range-finder is projected to occ[i][j] and emp[i][j] by means of Dempster-Shafer theory of evidence. So, the overall consistency of the sensory information contained in the grid-map, which is one target to be optimal, is described as follows:

$$f_1 = \sum \min(1\text{-}occ[i][j],\ 1\text{-}emp[i][j]). \tag{1}$$

The other objectives are to reward the algorithm for producing smaller, more compact maps:

$$f_2 = \sum \delta_1(i,j), \quad f_3 = \sum \delta_2(i,j), \tag{2}$$

where

$$\delta_1(i,j) = \begin{cases} 1, & if\ \ occ[i][j] > 0.5 \\ 0, & other \end{cases}, \quad \delta_2(i,j) = \begin{cases} 1, & if\ \ emp[i][j] > 0.5 \\ 0, & other \end{cases} \tag{3}$$

Objectives or of combining them together as a weighted, linear sum, is the method used in Reference [7] and [8]. This method, as is so often done, will lead bad solutions to since choice of weights is crucial and is difficult to be determined. Even a large change in the weights of a weighted sum scalarization would result in finding a bad solution.

The weighted sum is only one possible method in this family of scalarizing methods and has some serious drawbacks. One of others is to consider alternately one objective function then another; and there are various ways this could be organized. Another approach is to use some form of relative ranking of solutions in terms of Pareto dominance. The latter is the most favoured approach in the EA community because it naturally suits population-based algorithms and avoids the necessity of specifying weights, normalizing objectives, and setting reference points.

Mikkel and Jensen[10] used the flowtimes of the individual jobs Fi to build helper-objectives, and optimize the primary objective. The helper-objective simultaneously will be equivalent to simply optimizing the primary objective. Knowles and Corne[11] defined additional objectives with arbitrary sub-tours for travelling salesman problem, and local optima in single-objective optimization problems can be removed. These results enlighten that multi-objective methods can guide the search and deal with sources of difficulty in single-objective optimization. Their investigations also show that the performance of algorithm depends on whether dominant knowledge in problems is used efficiently.

2 Key Point Grid and Its Detection

Sensors like sonar and laser scan, grabbing range-finder data, provide environment information around the robot. From the information the local map can be constructed and a local path for robot can be planned. If there is some structure information like lines in environments, the structure information is used to building a global map more efficiently and exactly. Considering a situation showing as Fig. 1 in which range-finder data for a laser scanner are simulated, it is can seen that a large gap of range-finder data occurred at some point and on left side of which long scan radial line, and on the right side short scan radial line occurred. Furthermore, all long scan radial line forms a continuous sector. We call the occupancy grid that is located by gap point as key-point grid.

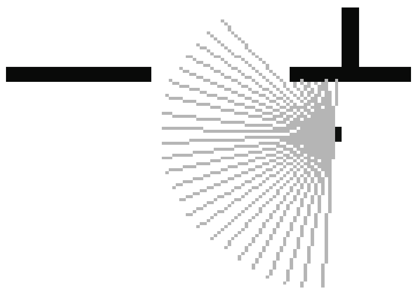


Figure1. Simulations of range- finder data at a key point grid

Fig.2 shows a key point grid in a running for a real robot in our experiment room. Local occupancy/empty grids were calculated by grid range-finder data, so the map in Fig 1 is local where the cells with black are the edges of obstacles calculated from range-finder data gotten from the robot. The ones with white are free of obstacles and gray cells are unknown. It is remarked that these colors in other figures are the same meaning in this paper. On the right side in Fig 1, a large gap in range-finder data occurred, since the obstacle goes to the end. Although range-finder data are not continuous, there is no key point grid since varying scopes of range data are not enough.

In the opinion of the geometry, the concept of key point grid comes form the convex point in polygon obstacle. When the robot samples near a convex point in polygon obstacle, largely discontinuous gap in range-finder data will happen. Because of noise, largely discontinuous gap that is founded haphazard in range-finder data doses not always mean that a key point grid occurs. Therefore, a method for eliminates false key point grid is needed. In this paper if a grid is a candidate key point grid, the grid will be scanning for many times and a key point grid will be determined by means of Dempster-Shafer theory of evidence.

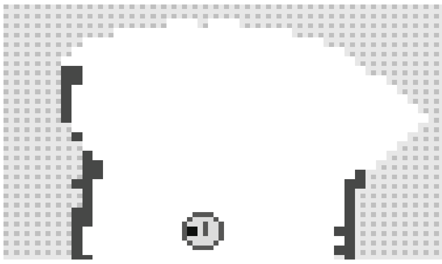


Figure 2. A possible key point grid is found by a real robot

— A robot, ■ — Obstacles, — Unknown area, Space —Obstacle-free area

In the implementation of determining key point grid, when a large gap in range-finder data occurs, the grid which the gap belongs to is consider as a candidate key point grid and the robot samples and takes range-finder data for 3-4 times while it move in a curved path. In Fig. 3 it is showed that a real robot sampled at some key point grid in moving along a curved path. If there is really a key point grid, the grid is marked as K_i=[x_i^b , y_i^b], where x_i^b , y_i^b are coordinates of the grid assigned by the map under global coordinate system. Since K_i depends on the robot's own measurements of its trajectory, these measurements will be corrupted by noise. Hence, real coordinates of Ki can be described as follows:

$$x_i = x_i^b + \Delta x_i \, , \; y_i = y_i^b + \Delta y_i \, , \tag{4}$$

where Δx_i ,Δy_i =0, ±1.

The following algorithm provides a method of finding a key point from range-finder data.

(1) Let range-finder data is a set of $\{(r_i,\varphi_i)\}$ in the local coordinate system. (r_i,φ_i) is coordinates under the polar coordinates. In general the polar coordinates (r_i,φ_i) can be translated to the Cartesian coordinates $p_i(x_i,\ y_i)$ by a simple coordinate transformation:

$$\begin{cases} x_i = r_r \cos\varphi_i \\ y_i = r_r \sin\varphi_i \end{cases} , \tag{5}$$

(2) Let a set A1 ={ p_1 }. The set A_1 is called as a successive section. Let i=2, j=1;

(3) If the point pi in range-finder data satisfies the condition as follows:

$$\frac{\| p_i - p_{i-1} \|}{\min((\| p_i \|, \| p_{i-1} \|)} < \delta_{\text{Key}} \tag{6}$$

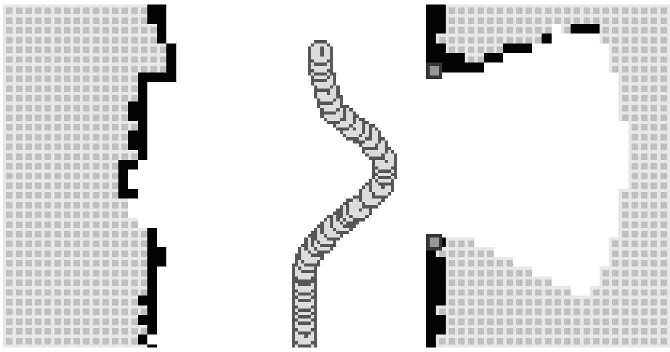


Figure 3. Trajectories and two key point grids.
— Trajectories, — Key point grids, — Obstacles,
— Unknown area, Space — Obstacle-free area

p_i is inserted into successive section A_j. Else, a new successive section A_{j+1}, $j=j+1$, is created, and let $i=i+1$;

(4) The step (3) is repeated until range-finder data is empty.

(5) If the size of successive section A_j is less than 10, this successive section A_j will be deleted.

(6) Let all successive sections can be described as follows:

$$Q_i = \{(x_{i,1}, y_{i,1}), \Lambda\ , (x_{i,N}, y_{i,N})\}., i=1,2,...,N_q. \tag{7}$$

(i) If N_q =1 there is not a c. The algorithm goes to end;

(ii) If N_q =2, the last point $p_{1,N}(x_{1,N}, y_{1,N})$ in Q_1 is a key point. The algorithm goes to end;

(iii) If N_q >2, successive section pairs $\{Q_i, Q_{i+2}\}$($i=1,2,...,N_q-2$) are constructed, a distance of $\{Q_i, Q_{i+2}\}$ is defined as follows:

$$d(Q_i, Q_{i+2})=\sqrt{(x_{i,N}-x_{i+2,1})^2+(y_{i,N}-y_{i+2,1})^2}\ . \tag{8}$$

This distance is called as successive section pair distance.

(7) If $d(Q_i, Q_{i+2}) > d_{\mathrm{Key}}$, $i=1,2,...,N_q-2$, points $p_{i,N}(x_{i,N}, y_{i,N})$ and $(x_{i+2,N}, y_{i+2,N})$ are considered as key points.

In the implementation of the algorithm in this paper key point grids were stored in a database, so computations of extracting key point grids from range-finder data were carried out only once.

3 Multi-Objective Evolutionary Algorithms with Immunity for SLAM

3.1 Multi-Objective Algorithms (MOA) with Immunity

Multi-objective algorithms with immunity for SLAM proposed in this paper are described as follows.

(1) Let G and P' empty
(2) Generate an initial population P
(3) Calculate the fitness of each individual
(4) **repeat**
  (5) Let G and P' empty
  (6) **for** j =1 to #recombinations **do**
    (7) Select a set X_{par} in P∪G to be parents with ranking-based selection[12]
    (8) Perform recombination on X_{par} with probability p_R to generate x_{off1}
    (9) Perform vaccination on x_{off1} with probability p_I
    (10) Perform immune selection to generate x_{off} from x_{off1}
    (11) Replace P' with P'∪{x_{off} }
  (12) **endfor**
  (13) **for** j =1 to #mutations **do**
    (14) Select x□P for mutation
    (15) Perform Mutate with probability p_M to generate x'
    (16) Perform vaccination on x' with probability p_I
    (17) Perform immune selection to generate x"
    (18) Replace P' with P'∪{x"}
  (19) **endfor**
  (20) Replace P with Select from(P∪P') using ranking-based selection[12]
  (21) **if** P has converged **then** replace P with an initial population P
(22) **until** termination condition is TRUE

In the implementation of the algorithm, both termination conditions are used.

3.2 Chromosome Encoding, Recombination and Mutation

SLAM is treated as a continuous global optimization problem where the search is carried out in the space of possible robot trajectories. A trajectory can be defined as a vector $[T_1, T_2, ..., T_N]$, where T_j=[d_j, θ_j], d_j and a_j are the relative distance and rotation that are traveled by the robot in one small step j, and there are N steps in total.

The robot's own measurements of its trajectory are used to generate candidate solutions by applying different correction factors, which are described as follows, to the measured values of d_j and a_j.

$$d'_j = d_j + \Delta d_j,\ \theta'_j = \theta_j + \Delta\theta_j. \tag{9}$$

Each chromosome is encoded as a string of floating point numbers [X_1‹ X_2‹▷▷▷‹ X_N, K_1‹ K_2‹ ▷▷▷‹ K_{Nk}] corresponding to the correction factors and the key point grids, where X_i=[Δd_j, $\Delta\theta_j$], and □d_{max}ℜΔd_kℜd_{max} , ↖θ_{max} ℜ$\Delta\theta_k$ ℜ+θ_{max}, k=1,2,▷▷▷,N. d_{max} and θ_{max} are real positive integers.

Pairs of selected strings are then combined by recombination. Recombination is carried out with probability p_c. Mutation is carried out by picking single value within the strings with very low probability p_m and replacing those values with randomly generated values, as upon initialization.

3.3 Vaccination Operator

A local exploration process, named as a vaccination operator, is constructed by means of the feature of key point grid. A key point grid K_i is selected uniformly, and a trajectory T_j is found such that at T_j the robot can detect the key point grid K_i by means of laser scanner. If the point where a large gap occurs in range-finder data at T_j does not belong to K_i, the correction factors Δd_j applied to the distance will be adjusted. This proccess will be performed with probability p_I.

3.4 Immune Selection Operator

If the correction factors $\Delta\theta_j$ and $\Delta\theta_{j+1}$ applied to the angle measurements at T_j and T_{j+1} are adjusted in the vaccination operator, an immune selection operator, which will be described as below, will be performed.

The change of the correction factors $\Delta\theta_j$ and $\Delta\theta_{j+1}$ will lead to improvements of the consistency and compactness of the map. Measure of improvements will be used to evaluate the performance of a vaccination operator. In detail, either of both trajectories T_j and T_{j+1} is evaluated by constructing a local occupancy map using the recorded range-finder data of the robot along the path, and a value f_1 is calculated according to formation (1) within the local occupancy map. If T_j and T_{j+1} are replaced with T'_j and T'_{j+1}, as the same as above calculations, a value f'_1 is calculated. If $f'_1 < f_1$, T_j and T_{j+1} are replaced with T'_j and T'_{j+1} in the individual in which the vaccination operator has been performed, else, the operator of replacement will be done[13] with probability $\exp(-(f'_1 - f_1)/\ln(k+1))$, where k is the evolutionary generation in evolutionary process.

4 Implementation of the Algorithm and Experiment Results

The algorithm was tested using data recorded by an AmigoBOT mobile robot produced by ActivMedia Robotics, LLC with the addition of a SICK laser scanner showed at the Intelligence Control Lab of the Central South University in China. The odometer trace was divided into segments of about from 1 to 2 meters in length and was 0.2 meters at every key point grid. For the environment of Fig. 4, there were 26 segments corresponding to the about 30 meters traveled by the robot. Because movable scope for robot is not large enough, all range-finder data are truncated such that the lengths of range-finder data are less than 3 meters, which means that if r_i>3 in range-finder data (r_i,φ_i), r_i=3.

In our experiments $D_{\max}$=0.32, δ_{Key} =1, $d_{\max}$=20cm, $\theta_{\max}$=10, β_0=1meters, integer $d_k \in$[-20, 20], integer $\theta_k \in$[-10, 10]. In implementation of the algorithm the population size is 50, p_{I}=0.3, p_{C} =0.9 and p_{M}=0.08.

The algorithms proposed in this paper and in Reference [7] and [8] run 10 times for the same test case (environment, trajectories, and range-finder data) in order to prove that the algorithm proposed in this paper outperforms other approaches. Running results of algorithms are list in Table 1, and a grid-map gained through the algorithm in this paper is shown in Fig. 4, where the terminate condition is that running generation is 300. From Table 1 it can be seen that the convergence rate is higher than the algorithms in Reference [7]. Similar experiments were conducted for the traditional multi-objective algorithm without vaccination operator and immune selection operator, the results are shown in table 2, where the terminate condition is if the best fitness values in the population are not improved in N_ς =20 generations, the algorithm will go to the end, which is often the convergence critical. The table holds one column for every algorithm. The first column reports the average total FS of the traditional algorithm, while the remaining columns report the performance for the multi-objective algorithm. The experiments reveal that the multi-objective algorithm with immunity performs better than the traditional algorithm without immunity.

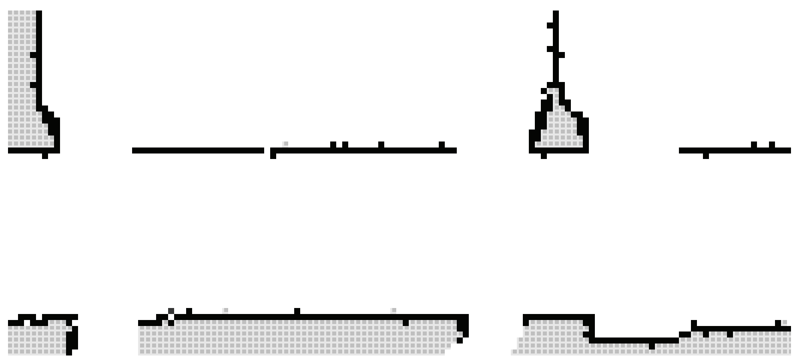


Figure 4. A grid-map was gotten with proposed algorithm when the robot ran in our experiment Lab.

In order to compare the accuracy of maps achieved by several algorithms, manual build a map using stored data captured by the mobile robot.

Table 1. Comparisons of running results for two algorithms

Algorithms	Mean value of best fitness values	Standard deviation of best fitness values
MOA with immunity	510.4	22.7
MOA without immunity	697.3	30.1
Algorithm in Ref.[7]	631.8	37.5
Algorithm in Ref.[8]	582.0	29.8

Table 2. Comparisons of running results for two algorithms

Algorithms	Mean number of fitness function evaluations
MOA with immunity	2074.8
MOA without immunity	3893.6

In the implementation of the algorithm in this paper, computations of extracting line segments and key point grids from range-finder data are carried out only once, hence in a vaccination operator and an immune selection, major computation time is spent on the computation of values f_1 in a local grid-map that was constructed by only both of trajectories. The average ratio of the computation time of values f_1 in a local grid-map to the one in a global grid-map is about $4/N$ (N is the total of trajectories). So, its computation time is more less than computation of the fitness value f in formulation (1). To sum up, the algorithm proposed in this paper can increase the convergence rate of SLAM based on evolutionary algorithms, and the larger the scope is for robot to travel the higher the convergence rate of our algorithm is, since the total of trajectories will increase.

5 Conclusions

(1) Multi-objective algorithms with immunity for SLAM have bee proposed, which are combined with feature of key point grids in range-finder data in order to increase the convergence rate of SLAM based on evolutionary algorithms.

(2) The feature of large gap in range-finder data at a convex vertex in polygonal obstacle is employed, and the feature of key point grids is extracted and used to construct a local search operator of key point grid with immunity.

(3) Experiments results showed that multi-objective evolutionary algorithms with immunity could improve optimization for SLAM in some cases.

Acknowledgements. This work was supported in part by the Crossover Item of XTU (No. 05IND04). Thanks to Prof. Zixing Cai & other fellows in Central South University for help in experiments.

References

1. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M Csorba. A solution to the simultaneous localization and map building (SLAM) problem. Transactions of Robotics and Automation, 2001, 17(3), pp. 229-241.
2. J. J. Leonard and H. F. Durrant-Whyte. Directed Sonar Sensing for Mobile Robot Navigation. Kluwer, 1992.
3. Thrun S. Robotic mapping: A survey. In: Lakemeyer G, Nebel B eds, Exploring Artificial Intelligence in the New Millenium. Kaufmann M, 2002, pp. 1-35
4. Wang L and Cai Z X. Progress of CML for mobile robots in unknown environments. Robot. 2004,26(4), pp. 380-384(in Chineses)
5. Smith RC and Cheeseman P. On the representation and estimation of spatial uncertainty. International Journal of Robotics Research, 1986, 5(4): 56-68
6. Smith R, Self M and Cheeseman P. Estimating uncertain spatial relationships in robotics. In: Wilfong G T and Cox I J, eds, Autonomous Robot Vehicles, Springer-Verlag, 1990:167-193
7. Duckett T. A genetic algorithm for simultaneous localization and mapping. In: IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003: 434-439
8. Meiyi Li, Zixing Cai. Immune Evolutionary Computations for Concurrent Mapping And Location. Fuzzy Sys-tems and Knowledge Discovery, Springer, 2005.8. Volume(3), pp.1308-1311,
9. Moravec H, Elfes A. High resolution maps from wide angle sonar. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'85), 1985, pp. 116-121
10. EE, Mikkel T. Jensen: Guiding Single-Objective Optimization Using Multi-objective Methods. EvoWork-shops 2003, pp. 268-279.
11. J. Knowles and D. Corne. Reducing local optima in single-objective problems by multi-objectivization. In First International Conference on Evolutionary Multi-criterion Optimization (EMO'01), Springer-Verlag, 2001, pp. 269–283.
12. David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
13. Jiao L C, Wang L. A novel genetic algorithm based on immunity. IEEE Trans. Syst., Man, Cybern. A. 2000, 30(5), pp. 552-561.

An Evolvable Hardware Approach

T. S. P. C. Duque[1] and A. C. B. Delbem[1]

Instituto de Ciencias Matematicas e de Computacao – Universidade de Sao Paulo (USP)
Av. Trabalhador Sao-carlence, 400 – 13.566-590 – Sao Carlos – SP – Brasil
thyago@grad.icmc.usp.br, acbd@icmc.usp.br

Abstract. Hardware based systems shows in general better performance for embedded applications than those based on software. Software based systems use a generic purpose processor, which is not adequate for several problems. Hardware systems, on the other hand, are in general very inflexible. Many hardware solutions cannot be updated. Even when this process is possible, it requires specific knowledge and tools.
The proposed system aims at investigating a solution for these drawbacks involving hardware systems. The proposal is based on an Evolutionary Algorithm to reconfigure Field Programmable Gate Arrays.

1 Introduction

This work proposes the use of Evolutionary Algorithms and Field Programmable Gate Arrays to produce an intelligent hardware system with adaptive capabilities.

This system could be used for real embedded applications, in fields where few systems can achieve the required performance. Several problems cannot be efficiently solved using generic purpose processors as, for example, embedded systems. For these problems, dedicated hardware solutions are in general employed. On the other hand, these hardware solutions are too inflexible and may become obsolete with small changes on problem requirements. If these requirements are expected to often change, reconfigurable systems have been the usual solution.

However, reconfigurable systems are in general difficult and expansive to be built. In order to overcome this drawback, we propose a self-reconfigurable approach, based on Evolutionary Algorithms. This approach can reconfigure a FPGA until it becomes a dedicated system for the target problem. Moreover, this process never finishes since the system continues to evolve and adapt to the problem when its requirements change.

2 Evolutionary Algorithms

Evolutionary Algorithms (EAs) have been largely employed for complex design problems [5], [7], combinatorial optimization [8], [10] and multi-objective problems [3], [4], [8]. Moreover EAs are plausible solution strategies for problems

Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 37-47
Received 03/06/06
Accepted 03/10/06
Final version 10/10/06

requiring adaptive capabilities. In this way, we choose an evolutionary approach to guide the FPGA reconfiguration.

An EA has typically the following characteristics. A set (named population) of solutions (named individuals) is generated at random. At each iteration, or generation, evolutionary operators are applied to the individuals modifying them and generating new individuals. In order to compose the new population a selection strategy and a criteium of adequacy for the target problem are employed. This criterium is named fitness. The selected individuals compose a new population. After several generations, the EA produce a population with individuals that should correspond to adequate solutions. Several relevant references about EAs are available in the literature [3], [4], [8], [9], [10], [11].

3 Field Programmable Gate Arrays

For problems whose requirements are expected to change often, reconfigurable systems have been the usual solution. The EA can be considered the intelligence of the proposed system, the FPGA can be viewed as its physical component. It holds the IO interfaces, stores the system logic and executes the processing.

A FPGA is a hardware system designed to hold another hardware system. It is composed by a set of logical blocks or elements and a set of connections among them. In an ideal architecture, there would be one connection between each pair of elements and one connection between each element and each IO port. This architecture however is not necessary for most real problems and is too expansive and complex to implement. Real FPGA uses only a subset of this complete set of connections.

The FPGA works as follow. The input ports have the function of reading external signals from the environment. These signals are propagated through the connections and logical elements until they reach an output port. During this route, the logical elements modify the signals according to a FPGA configuration. Once the signals are stabilized in the output ports, they produce the output of the system. Introductory references about FPGA architectures can be found in [1], [2], [12].

4 FPGA and EA Integration

The architecture of the FPGA is very important since it defines the complexity of the problems the FPGA can be used to solve. The architecture has also large influence on the EA efficiency to reconfigure the FPGA.

If the architecture is too complex, the FPGA can be employed for a wide range of problems. On the other hand, complex architectures can reduce the EA performance. It is important to define architectures that both give the system enough power to solve relatively complex problems and also allow an efficient EA for FPGA reconfiguration.

In this way, the first step to build the proposed system should be the definition of the FPGA architecture. The second step is the definition of the data structure (named chromosome in EAs) to computationally represent the architecture. From the chromosome, evolutionary operators are elaborated. Afterward, a Fitness function and a selection method are determined according to the target problem characteristics.

4.1 Architecture of the FPGA

The FPGA architecture plays an important role for the proposed system, both determining the FPGA capability of solving problems and the EA performance. This section describes the employed architecture.

The proposed architecture organizes the FPGA as a logical element matrix. Each logical element receives n input signals and produces one output signal. In FPGAs, a logical element is built as an array of 2^n binary memory units. Each memory can store one single bit. One of these bits is selected using the input signals and multiplexers. Figure 1 shows an architecture example, where S1, S2 and S3 are the received signals and f is the produced output.

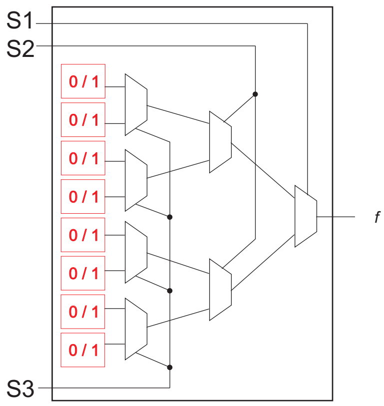


Fig. 1. Architecture of a Logical Element

Each element in the matrix has two indexes indicating its line and column in the architecture. Given a block in line l, each input terminal of this block can be connected to the output terminal of any block in line l-1. Figure 2 shows the proposed architecture.

The elements in the first line have their input terminals connect to the FPGA input ports. They transmit signals from environment through the FPGA. The output terminals of the elements in the last line are also the output ports of the FPGA. Then, a FPGA of c columns will also have c output terminals.

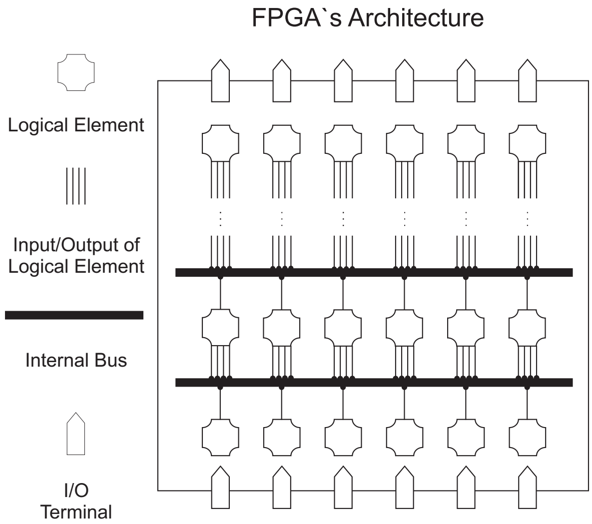


Fig. 2. Architecture of the FPGA

4.2 Individual Representation and Evolutionary Operators

Based on the architecture described in Section 4.1, individual's computational representation (chromosome) was defined as follow. Each logic element is represented as a string with 2^n bits, where n is the number of input terminals in the element. Based on this string of bits, we define the evolutionary operators: crossover and mutation.

The crossover operator is applied to two individuals named *father1* and *father2*, generating two new individuals named *child1* and *child2*. This operator generates a random mask of 2^n bits used to alter the bit string producing a new one. For each position of the bit string, if the bit in the mask is 0, *child1* keeps the corresponding bit value from *father1* and *child2* retains the corresponding bit value from *father2*. If the bit in the mask is 1, *child1* keeps the bit value from *father2* and *child2* retains the bit value from *father1*.

The mutation operation is applied to one individual, and generates a new individual. It consists on the negation of a random bit from a bit string of the individual.

The connections among elements are represented as an array of n integers. Each element in a line of the architecture has an array of integers indicating the connections of this element with the elements of previous line. The mutation operator alters element connections by randomly changing the value of one position in the array.

A chromosome is structured as a pair composed by a logical element matrix and a connection matrix. The former can be changed by a crossover operator that consists of the recombination of each logical element from *father1* with the logical element in the corresponding position from *father2*. This process generates two children. The mutation operator for the logical element matrix consists on randomly choosing a matrix position and altering the corresponding element.

The connection matrix mutation process is similar to the mutation of logical element matrix. For the connection matrix no crossover operator is employed.

It is important to notice that small changes in the logical element matrix produce small changes on how the FPGA works. However, small changes on the connection matrix may drastically affects the way the FPGA works. Moreover, a good logical element set for a given connection matrix may be inadequate for another connection matrix. These characteristics demand an efficient evolution strategy, which is discussed in Section 4.3.

4.3 Strategies for the Evolution Process

Given a connection matrix, the evolution strategy employed first evolves the logical element matrix. Only if this matrix is not adequate, a new connection matrix is produced.

This strategy requires two stopping criteria. The first criterion determines when a satisfactory solution is found. The second criterion decides when the evolution of a logical element matrix for a given connection matrix should stop. In this case, the evolution of a logical element matrix is restarted for a new connection matrix.

As consequence, if a given connection matrix propagates all input signals to all output signals, it will not be necessary to change it. In this case, it is possible to reach the expected operation of the FPGA by reconfiguring properly the logical elements matrix.

If it is possible to create a connection matrix that propagates all input signals to all output signals, then we should first find this matrix and then run the EA only for the logical element matrix. This improves the performance of the proposed system, avoiding unnecessary connection matrix changes.

However, it is not always possible to create connection that propagates all input signals to all output signals. For example, if the number of columns is much larger than the number of lines for a FPGA, it is not possible to create such matrix. In this case, it may be impossible to solve the target problem using this FPGA, requiring a larger one.

5 Target Problem and Fitness Evaluation

The focus of this work was to produce an efficient EA to configure a FPGA. The implementation of the proposal on hardware will be performed in a future work. In order to validate the proposed approach, we use a simulator. The simulation is a simple, fast and easy procedure to debug and evaluate the developing system.

The simulator was written in Java and the computational efficiency of the simulator was considered secondary criterion for its evaluation. The main objective was the investigation of how the EA work with the problem of reconfiguration of FPGAs. Main criterion to evaluate the system is the number of individual evaluations.

In order to obtain a target problem which was relatively easy to simulate and evaluate solutions, we use the one-max problem formulation **??**. First, we generate a random individual named target individual. For all possible inputs (2^n for n input terminals), all possible outputs are generated for the target individual. The fitness of an individual is calculated comparing its outputs with the corresponding outputs of the target individual. The individual score is increased by one for each bit that matches with the target output. The target individual score is the maximum score and can be calculated as:

$$2^{nInputs} * nOutput \tag{1}$$

where *nInput* is the number of input ports and *nOutput* is the number of output ports.

The simulator works with a population of n individuals. At each generation a recombination process generates n new individuals. The recombination consists on the crossover of the logical function matrix of two individuals (say *father1* and *father2*) of the population. These individuals (*father1* and *father2*) are selected using the roulette-wheel selection method.

From the resulting population of $2n$ individuals, n individuals are selected using a *steady state* strategy, i.e., the better individual among parent and child are selected for the next generation. Other selecting strategies are also possible; however, the reported results were based on this strategy.

6 Statistic Measures of the EA Performance

This section presents statistics showing the performance of the EA. The system was tested using different combination of the following parameters: number of individuals on the population (n), size of the FPGA (number of columns c and lines l), mutation rate (for logical elements only), and convergence criterion.

For the first test, we used a population of 10 individuals ($n = 10$). The mutation rate was set to 0. The EA executes until all individuals reach the maximum score. Each individual is a FPGA of eight lines and eight columns. Although this FPGA may look small, it is complex enough, for example, to be applied to a robot navigation problem where robots navigate through the environment using binary distance sensor [6].

At each run of the EA, the number of generations and running time for convergence were measured. The data of 2000 executions were then divided into classes to drawn an approximated graphic of the probability distribution of the number of generation and running time for convergence of the EA. The interval between the maximum and the minimum value of the measured data was divided into subintervals (or classes) of equal length and the number of values on each class was counted. The relative frequency of each class was calculated as the number of elements on this class divided by the number of total executions (2000). This process was applied to both number of generation and running time.

Figure 3 shows the approximated probability density for the number of generations required to all individuals in the population converge. Figure 4 shows the approximated probability density for the time until the convergence of all individuals of the population.

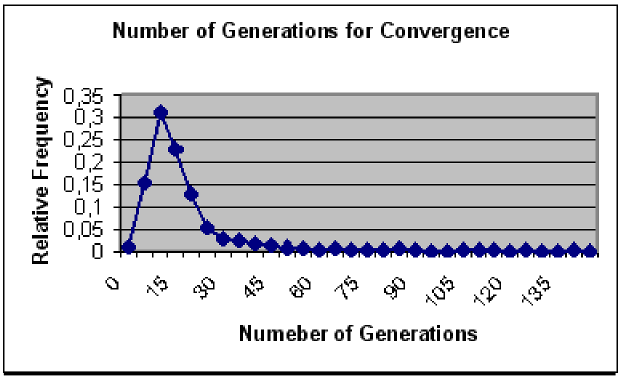


Fig. 3. Graphic of the probability distribution of the number of generations for convergence

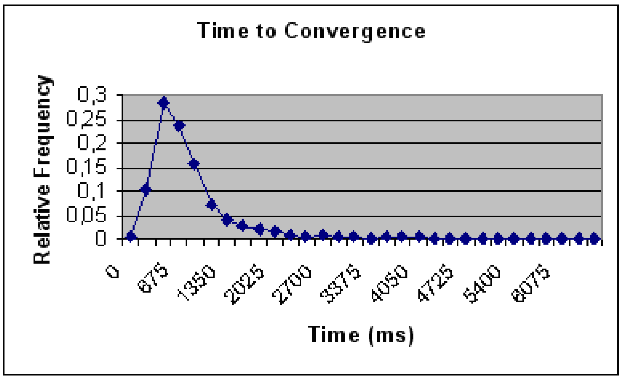


Fig. 4. Graphic of the probability distribution of the time for convergence

As expected, the graphic for probability distribution for the number of generations is very similar to the graphic probability distribution for running time. This occurs since the running time the EA requires to converge is proportional to the number of generations. There are also other factors that influence the convergence process; however, they are less important than the number of generations.

Table 1 presents the main descriptive statistics involving running time and number of generations for convergence.

	Numeber of Generations	Time (ms)
Mean	18.572393	949
Mean Deviation	8.58767	421
Minimum	3	186
First Quartile	11	564
Median	15	767
Third Quartile	21	1036
Maximum	141	6960

Table 1. Descriptive statistics

Figure 5 is a box plot of number of generations for convergence without outliers. Figure 6 shows a box plot including outliers. Box Plots of running time for convergence are similar to those of number of generations and thus were omitted.

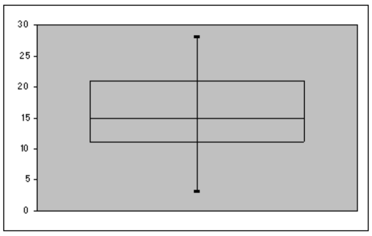


Fig. 5. Box Plot without Outliers

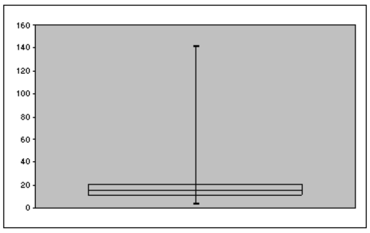


Fig. 6. Box Plot with Outliers

Notice that although the maximum value for the number of generations is high, the third quartile is low. This means that, for the majority of cases, the number of generations for convergence is low. Figure 3 confirms this behavior. Cases with high number of generations were rare and spread. This means that the system has a good performance and convergence.

The running time required to the EA converge is also low. The mean running time is lower than 1 second. This shows that we could evolve the system right on the environment without requiring high performance computers, since this process should require few minutes in a FPGA.

6.1 Relationships between Parameters and Performance

This section discusses how the parameters affect the system performance and how the system behaves for more complex problems. First, we investigated the relationship between the number of individuals in the population and the performance of the system. We used again a FPGA of eight lines and eight columns and the convergence of all individuals as stopping criterion. The population size was set to 2, 5, 7, 8, 10, 15, 20, 50 and 100. For each size 200 runs were performed.

Figure 6.1 shows that the number of generations for convergence depends on population size. For small population, the diversity of the population is low, and lots of generations are required for convergence. For medium size populations, there is already enough diversity, so, few generations are required. For large populations it is difficult to converge all individuals, and the number of generations increases

Figure 6.1 shows that, although the number of generations for small populations is high, the running time required is not so high. Moreover, for large populations, that have low number of generations for convergence, the corresponding running is very high. Figure 9 helps to explain this behavior.

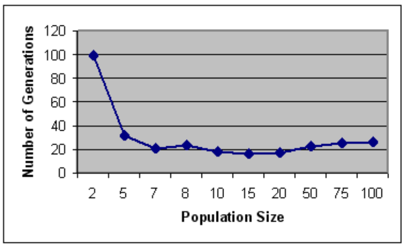


Fig. 7. Relationship between population size and number of generations for convergence

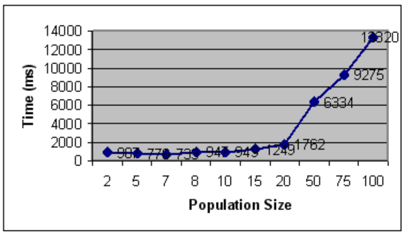


Fig. 8. Relationship between population size and time for convergence

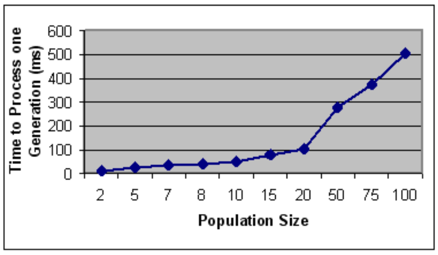


Fig. 9. Relationship between time to process one generation and population size

For Figures 6.1 and 6.1, the stopping criterion was the convergence of one individual. As in Figure 6.1 for small populations, the number of generations is large. As the population size increases, the number of generations decreases. In contrast with Figure 6.1, the number of generations for convergence does not increase. This occurs since the convergence of all individuals is not required.

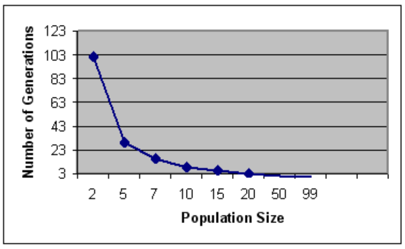


Fig. 10. Relationship between the population size and the number of generations for convergence of one individual

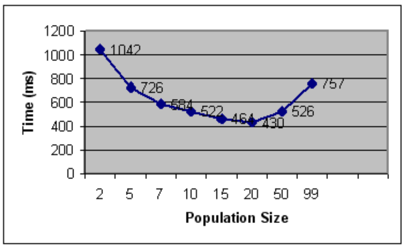


Fig. 11. Relationship between the population size and the time for convergence of one individual

Figure 12 shows the influence of mutation operator over the EA performance. The relationship between mutation rate and EA performance in the proposed system is not clear.

More accurate statistic investigation of such effects should be carried out.

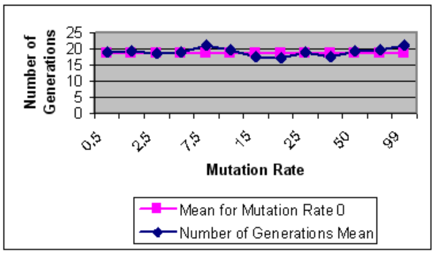


Fig. 12. Effect of mutation rate

7 Final Considerations

The development of an intelligent system capable of solving real problems without needing general-purpose processors and with self-adapting capabilities for problem requirement changes is a computational difficult problem. This work presents a method to create a system with these characteristics using EA and FPGA.

The proposed system showed satisfactory results for small FPGAs. For larger systems, the complexity of configuring a FPGA increases and more efficient approaches should be investigated.

In order to use more efficient data structures and operators, larger processing capabilities are required. The available processors for embedded system are in general not adequate to work with complex structures. Hardware modules to implement complex operators or store large data structures require a large amount of logical elements, increasing cost and size of the final system.

Definition of the most adequate FPGA size for a problem is also complex. Large FPGAs are capable of solving hard problems. However, their configuration is more difficult to be achieved by the EA. On the other hand, the configuration of small FPGAs is easier, although these FPGAs can only be applied to relatively simple problems.

The obtained results show that the proposed approach can produce a system that reaches the requirements of the test problem. Additional aspects of the proposal could be investigated in future works in order to obtain an improved system.

8 Acknowledgments

This research was financed by FAPESP (Process N. 04/10394-7), a foundation of the state of Sao Paulo, Brazil

References

1. Altera. *Altera Data Book.* 2003.
2. S. Brown and Z. Vranesic. *Fundamentals of Digital Logic with VHDL Design.* McGraw Hill, 2000.
3. Lamont G. B. Coello C. A. C., Van Veldhuizen D. A. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic Publishers, 2002.
4. K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley and Sons, 2001.
5. A.C.B. Delbem, A. C. P. L. F. Carvalho, and Policastro C. A. Node-depth encoding for evolutionary algorithms applied to network design. In *Lecture Notes in Computer Science*, volume 3102, pages 678–687, 2004.
6. D. Floreano and F. Mondada. Evolution of homing navigation in a real mobile robot.
7. M. Gen, R. Cheng, and S. S. Oren. Network design techniques using adapted genetic algorithms. *Adv. Eng. Softw.*, 32(9):731–744, 2001.
8. D. E Goldberg. *Genetic Algorithms for search, Optimization, and Machine Learning.* Addison-Wesley, 1989.
9. J. H. Holland. *Adaptation in Natural and Artificial Systems.* MIT Press, 1975.
10. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer, 1998.
11. D. M. Tate and A. E. Smith. Expected allele coverage and the role of mutation in genetic algorithms. In Stephanie Forrest, editor, *Proc. of the Fifth Int. Conf. on Genetic Algorithms*, pages 31–37, San Mateo, CA, 1993. Morgan Kaufmann.
12. Xilinx. *Xilinx Data Book.* 2003.

Machine Learning and Classification

A tool for Multi-Strategy Learning

Francisco Reinaldo[1,2], Marcus Siqueira[1,2], Rui Camacho[1], and Luís Paulo Reis[1]

[1] LIACC, Faculty of Engineering, University of Porto, Portugal
[2] GIC, Department of Computer Science, UnilesteMG, Brazil
reifeup,rcamacho,lpreis{@fe.up.pt}, marvinsiq@gmail.com

Abstract. This paper presents the AFRANCI tool for the development of Multi-Strategy learning systems. AFRANCI allows users to build, in an interactive and easy way, complex systems. Systems are built using a two step methodology: design of the structure of the system; and fill in the modules. The structure of the target system is a collection of interconnected modules. The user may then choose among a variety of learning algorithms to construct each module. The tool has several built-in Machine Learning algorithms and interfaces that enable it to use external learning tools like WEKA or CN2. AFRANCI uses the interdependency of the modules to determine the sequence of their training. To improve usability, the tool uses a wrapper that hides from the user the parameter tuning procedure for each algorithm. In a final step of the design sequence AFRANCI generates a compact and legible ready-to-use ANSI C++ open-source code for the final system.

To illustrate the concept we have empirically evaluated the tool in the context of the RoboCup Rescue domain. We have developed a small system that uses both neural networks and rules in the same system. The experiment have shown that a very significant speed up is attained in the development of systems when using this tool.

1 Introduction

Over the past few years, there has been a significant increase in the interest of working with heterogeneous learning algorithms in order to achieve fast and complex behaviours in autonomous agents. An Autonomous Agent (AA) can be seen as a collection of components of heterogeneous learning modules with well-defined interfaces and fine tuned behaviours. It has been recognised that different behaviours may require different learning strategies.

A popular AA architecture considers several learning algorithms/modules arranged in horizontal and/or vertical levels to compose behaviours with different levels of abstraction. This layered learning is specially adequate for domains that are too complex for a direct mapping from the input to the output representation to work [1]. This approach brings some new challenges on the arrangement of these modules to achieve inner reasoning, prediction and abstraction methods instead of classical planning research. Additionally, the use of robust tools are necessary to support this kind of approach. Although not in the context of AA

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 51-59

Received 02/06/06
Accepted 03/10/06
Final version 11/10/06

construction, but also using a hierarchical organisation of modules, Alan Shapiro [2] proposes the use of "structured induction" to address the problem of inducing complex concepts.

However, when using a conventional programming language, like C or C++ to encode such modules, an expert programmer is often required. Each new user starts a project from scratch, and occasionally it results in a bad program code structure revealing problems whenever the code needs to be extended or updated. To overcome such difficulties some tools propose uniform and easily modelling facilities for the learning modules. Such tools include Matlab© [3] and SNNS (Stuttgart Neural Network Simulator)© [4] and [5]. These tools are limited as far as the user interface aspects are concerned since these are often neglected. In addition, the tools do not support the design of several modular, hierarchic and complex structures of heterogeneous learning modules in the same environment as AFRANCI does.

The AFRANCI tool [6] offers solutions for assembling and linking together graphically on the screen the modules required to create large scale behaviour-based systems. In the development of AA applications there is the need to aggregate different kinds of behaviours working together to increase the complexity of behaviours. The user's choice is not restricted to low levels but is free to choose the suitable abstraction level. Another advantage of AFRANCI is the possibility of using external learning tools/algorithms such as the WEKA [7] library or CN2 induction algorithm [8] to compose heterogeneous learning modules. The tool makes easily adjustable and extensible connection with WEKA learning modules and CN2 without the user's perception. These aspects will be presented in Section 2.

According to the proposed objectives, this work has enabled the development of a tool that can: a) support graphic designs of large and extensible behaviour-based architectures composed of heterogeneous clusters with cognitive processes; b) make available a wide set of varied functionalities of control modules; c) offer facilities and resources to interconnect several structures of external control processes so that they could communicate in a multi-environment; d) arrange learning modules in horizontal or vertical levels to insert new or substitute the previous agent characteristics by an adopted technique from software engineering [6] and [9]; e) offer parallelised network training; f) automatically generate a ready-to-use ANSI C++ open-source code from the screen.

The contribution of this paper goes beyond a object-oriented implementation of ideas. It addresses abilities, specialisations and policies to accomplish specific tasks in an Autonomous Agent setting by using a multi-strategy learning in order to achieve complex behaviours. We offer a scalable modelling and parallel environments with the purpose of speeding up behaviour agent simulations. Furthermore, the tool is a valuable contribution to users without knowledge about deep programming language and learning modules because it offers high level of abstraction and automatic creation of source-code, as observed in Section 5.

The rest of the paper is organised as follows. Section 2 presents an overview of AFRANCI architectures and how it handles the generated system's modules.

Section 3 describes how to design behaviour structures in the interconnected architecture in order to achieve the best agent performance. The use of wrappers is described in Section 4. Experiments, results and discussion are presented in Section 5. Related work is presented in Section 6. Finally, some conclusions are drawn in the last section.

2 AFRANCI Tool

The AFRANCI Tool [6] is a tool built over some classes of open-source PyramidNet Framework [10] platform, which extends the use to external libraries of learning algorithms like the WEKA© library and CN2 induction algorithm. PyramidNet was restricted to a subset of Artificial Neural Network (ANN) algorithms whereas with the new ability to use WEKA© library and CN2 the user has access to a useful repository of machine learning algorithms for data pre-processing, classification, regression, clustering, association rules and visualisation.

The AFRANCI tool is composed of three main parts, which are the Graphic User Interface (GUI), the Machine Learning Modules (MLM) and the Automatic Open-Source Code Generator (ACG). First, GUI is a set of main classes for handling and modelling learning modules, using graphic elements that will interact with the user. MLM implements the construction of the modules using the chosen learning algorithms. Finally, ACG receives a description of pictorial representation of the system structure and produces the short automatic open-source code.

The tool allows users to design and implement behaviour-based architectures through the interconnection of elementary heterogeneous control modules in the form of a circuit diagram. New structures can be constructed by linking standard structures or modules together. This form is very familiar to engineers in digital systems design and in model analysis systems tools (e.g. SIMULINK©). It makes some items available, such as desktop constructor, sensor, actuator, line links, skins of learning modules, menus, dialogue box. Each graphical element can be dragged and dropped to a re-sizable screen and its features (colour, label, format and size) be adjusted. Heterogeneous control modules are standard boxes with input and output of data. For instance, if the user is working with ANN, he/she has access to a number of hidden neurons, neuron transfer functions, training parameters etc.

With the purpose of speeding up the implementation of learning modules, the tool offers three other important features: the multi parallel environment, the automatic network design of modules and the automatic development of open-source code. First, each new environment includes a powerful parallel resource to training the proposed designed architecture. The user can link and put together several sub-projects and run them, at the same time because it is free of manual scheduler. In the design process, once adjustments have been made, the module is unlighted and the training and simulation can be run again quickly. The second feature is the on screen automatic/wizard design of networks of learning modules by importing of CSV (Comma Separated Values) files. When using ANN/WEKA/CN2, the user links standard learning modules together; each learning module can be simulated at any level of abstraction for fine tuning of the assembled system. The user needs little knowledge about it. As a last step the tool produces clean and ready-to-use ANSI C++ open-source for information fusion, planning and coordination with a few mouse clicks. By using a high-performance interpretation algorithm, this functional but compact ANSI C++ executable core is created from the drawn project. This ANSI C++ code can be edited on screen in order to be modified and compiled easily in different operating systems.

3 Designing a Structure

In this section we briefly analyse the simple steps to design, train and obtain a finished system composed of different learning modules.

First, it is necessary to design the agent's structure. The user should plan how many modules, how they are interconnected and what algorithms to use in each module. In the next step the user draws a project in the tool desktop (screen). The project must have input, control modules, outputs and links. After designing the project, every module receives the database file that stores data to feed it in the training phase.

The interconnections are established between inputs, control modules and outputs to make a complete wired network. The user can connect two environments: from the output of modules in the first environment to the input of other assembled of modules in a second environment. This tool allows the user to interconnect everything without loss of performance because it uses a parallel environment. The user can tune each module by changing the default module parameters. Careful is needed in this step because a badly planned set up may lead to the emergence of wrong reasoning processes. This step is the only one that requires a little more knowledge about arranging of modules.

The training process is simple and consists of training and fine tuning of the behaviour-based architecture to achieve the agent's goal. The automatic training process is responsible for almost everything and based on the data flux sequence to trigger modules to training. Independently of the horizontal or vertical architecture level, the user can follow the iteration training process of each module by graphics, data windows and others.

Finally, the user can generate a ready-to-use ANSI C++ open-source to plug it in the agent. The source code is a codification of the graphic modelling. Inside of this code, user will find input and output connections, weights, an activation functions, in the case of ANN, a compact main executable core, and some input and output matrixes. Summing up, these quick steps help user to substitute large programs, time consuming projects and confusing lines of code.

4 Wrappers

Almost all Machine Learning systems have parameters that must be tuned to achieve a good quality of the constructed model. An experienced practitioner knows that changes in the parameter's values may lead to quite different results. To tune a system's parameter requires knowledge of the system. This is most often a severe obstacle to the wide spread use of such algorithms.

As proposed by John [11] one possible approach to overcome such a situation is by the use of a *wrapper*. A *wrapper* produces several models using different combinations of the learning algorithm and returns the "best" model. In our tool the wrapper optimises the test set error rate estimation. This automatic tuning of parameter completely hides the details of using the learning algorithms from the user. It is therefore a way to make the tool usable by a wider range of users.

When the total number of combinations of parameter's values is small[3] the tool tries all the combinations and chooses the best one. When the number of combinations is larger the parameters are tuned using a Genetic Algorithm (GA) [12] with mutation and cross-over GA operators. For the ANN we also choose automatically their structure. A GA is used to choose the best number of hidden layers and number of neurons in each layer for each ANN. The type of ANN is also included in the automatic choice.

As future work we intend to extend the use of *wrappers* to do feature (subset)-selection as in [13] and [14]. This facility would require however a tighter relationship between the modules synthesis and their inter-connection. If the wrapper decided that some feature is not relevant for the classifier then that input would have to be removed in the modules inter-connection design stage.

5 Experiments

To illustrate the features of AFRANCI and facilitate the reader's understanding we generated a simple artificial problem and dataset in the RoboCup Rescue setting. The problem we devised is to decide if a ambulance or fireman should rescue or not a civilian to a nearest refuge. The civilian is somewhere in a burning building. The decision is commonly made based on the localisation and agent's and civilian's life conditions. The main independent variables include: the position (X, Y) of the ambulance, fireman, building on fire, fire brigade, the nearest refuge (rescue building), and of the civilian; the life condition measure of the fireman and civilian [4], the building volatile information is composed of earlier burnt, state and structure; the state of the ambulance (busy/free) to receive the civilian and of the fireman (busy/free) to extinguish the fire or to rescue the civilian; a measure that assesses the difficulty of the civilian rescue situation.

The system devised is composed by several modules that encode the decision of the fireman and the civilian and particularly a main module to combine fireman and ambulance decisions. Figure 1 shows the modular structure of the

[3] In the current implementation *small* is less than 20.

[4] A measure between 0 and 100 of the energy the fireman can use.

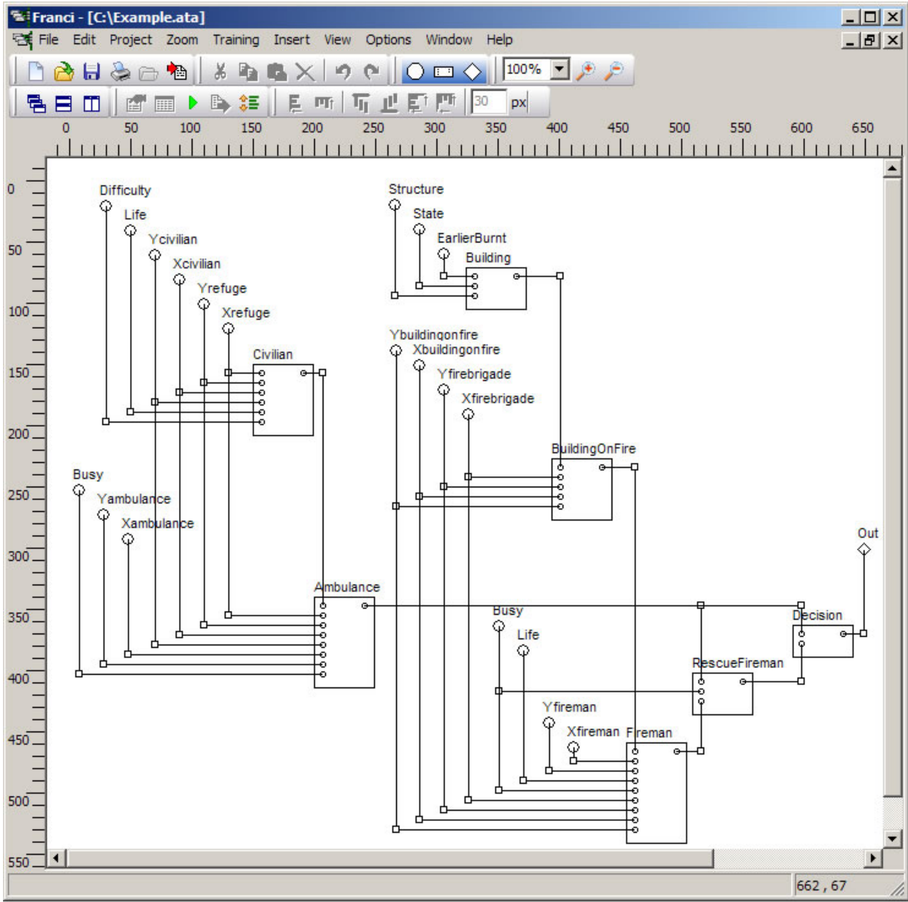


Fig. 1. Architectural Design.

system constructed interactively by the user. Not perceived in the figure is the heterogeneity of the modules. Different modules use different representations sand where induced using different learning algorithms. Referring to the module's labels in Figure 1 the following algorithms were used. The Civilian module was constructed using AFRANCI built-in feedforward ANN. The Ambulance, BuildingOnFire, Fireman and RescueFireman modules were constructed using CN2 rule learner. The other modules, Building and Decision, were constructed using WEKA's J48 Decision Tree algorithm. The user had only to select the number and place in the window of the modules, to connect them and choose input and output variable names. All this was done using drag-and-drop operations. He then provided the dataset and the tool trained the modules in the correct sequence and generated a C++ program that encodes the system.

a)	b)

<pre>J48 pruned tree
------------------ 	
ambulance_apt = TRUE: RescueAmbulance
 	
ambulance_apt = FALSE: RescueFireman</pre>
	<pre>IF occupied = occupied
	THEN class = n [52 0]
	ELSE
	IF civilian = notapt
	THEN class = n [20 0]
	ELSE
	IF Yamb > 8164.50
	  AND Xciv < 9204.50
	THEN class = y [0 7]
	ELSE
	IF Yamb > 340.00
	  AND 386.00 < Yciv < 7731.50
	  AND Xrefuge > 3800.00
	THEN class = n [9 0]
	ELSE
	IF Yamb < 5721.00
	  AND Xrefuge > 866.00
	THEN class = y [0 8]
	ELSE
	(DEFAULT) class = n [4 0]</pre>

Fig. 2. a) Decision Tree generated by WEKA's J48 learner. b) Rule set generated by CN2.

The action of deciding which agent will be responsible for rescuing the injured civilian is taken in the Decision module (see Figure 1). The fireman agent will perform a rescue action only if the ambulance agent will not be able to do so. In the case of both, the fireman and ambulance, are capable of rescuing the civilian, the module decides in favour of the ambulance agent because a fireman agent has to extinguish fires in burning buildings with the aim of preserving the city. In order to check if the ambulance is entirely apt to rescue a civilian (see Figure 2) the rules induced by CN2 establish that: (rule 1) if the ambulance is occupied then it is useless to attempt the rescue; (rule 2) if the civilian has not enough "vitality" then it is also not rescued; (other rules) the civilian will be rescued if it has enough "energy" and the ambulance is between the civilian and the rescue place otherwise it will not be rescued.

6 Related Work

In order to develop a robust tool, we analysed two relevant ones. The MatLab © and SIMULINK © tools are mainly concerned with the offer of a detailed design of a control process [15] and to automatically generate a ready-to-use code for it. However, that tool has two main limitations that affect the design of the whole project. First, the user cannot design a complex structure composed of several heterogeneous modules nor a interconnected architecture because the tool does not offer facilities nor a specific multi-environment to work with. These limitations inhibit the user's ability to handle different levels of abstraction, several processes in the same environment. Second, the ready-to-use code was generated from a simple learning algorithm is large to be worked, complex to be understood and unpractical to be used. Other studied software simulator was the SNNS © (Stuttgart Neural Network Simulator) [4] [5] that it exclusively works with ANN in order to create their applications. Although the tool offers a good graphic environment, its repertory is limited to only a behaviour level a at time, without offering a parallel environment to train the ANNs. Moreover, it is limited concerning expandability or new ways to graphically preview the whole structure that is being worked with.

In Section 2, we presented the tool which offers a user friendly parallel graphic environment that facilitates the development and training of Modular and Hierarchic Systems by including several modules with different learning algorithms. The system architecture produced with the AFRANCI tool promoted the Multi-Strategy learning as long as various learning modules are used to compose the structure.

7 Conclusions

In this paper we described a tool for the development of Multi-Strategy learning systems. Using a friendly graphical interface the user may define the modular structure of the system and choose the learning algorithms to construct each module. He then provides the dataset and lets the tool train, in the correct sequence, each module produces a complete self containing program encoded in the C++ language.

The tool provides several learning algorithms and is able to call external learning algorithms to construct the modules. The deployment of the tool confirmed the assumption that it was easy and fast to develop Multi-Strategy system with AFRANCI.

As future work we intend to develop *wrappers* (John, 1994b) to automatically tune the modules parameters, optimising the test set error rate estimation.

References

1. Peter Stone and Manuela M. Veloso. Layered learning. In *Machine Learning: ECML 2000, 11th European Conference on Machine Learning, Barcelona, Catalonia, Spain, May 31 - June 2, 2000, Proceedings*, volume 1810, pages 369–381. Springer, Berlin, 2000.
2. A.D. Shapiro. *Structured Induction in Expert Systems.* Adison-Wesley, Wokingham, 1987.
3. Mathworks. *Matlab.* Mathworks, Inc, Natick, MA, 1999.

4. A. Zell, N. Mache, T. Sommer, and T. Korb. Design of the snns neural network simulator. In H. Kaindl, editor, *7. Österreichische Artificial-Intelligence-Tagung*, pages 93–102. Springer, Berlin, Heidelberg, 1991.
5. Andreas Zell, Niels Mache, Ralf Huebner, Michael Schmalzl, Tilman Sommer, and Thomas Korb. SNNS: Stuttgart neural network simulator. Technical report, Stuttgart, 1992.
6. Francisco Reinaldo, Rui Camacho, and Luís Paulo Reis. Afranci: An architecture for learning agents. Phd report, FEUP, Porto, Portugal, August 2005.
7. Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd. edition, 2005.
8. P. Clark and T. Niblett. The CN2 algorithm. *Machine Learning*, (3):261–283, 1989.
9. Robert Wray, Ron Chong, Joseph Phillips, Seth Rogers, and Bill Walsh. *A Survey of Cognitive and Agent Architectures*. University of Michigan, 1994.
10. F. A. F. Reinaldo. Projecting a framework and programming a system for development of modular and heterogeneous artificial neural networks. Dept. of computer science, Federal Univ. of Santa Catarina, Florianopolis, Brazil, Feb 2003.
11. H. George John. Cross-validated c4.5: Using error estimation for automatic parameter selection. Technical note stan-cs-tn-94-12, Computer Science Department, Stanford University, California, October 1994.
12. J.H. Holland. *Adaption in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan, 1975.
13. Ron Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Stanford University, 1995.
14. Ron Kohavi and Dan Summerfield. Features subset selection using the wrapper method:overfitting and dynamic search space topology. In *First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, 1995.
15. Richard C. Dorf and Robert H. Bishop. *Modern control systems*. Tenth, 2004.

Determination of the Optimum Error and Reject Validity Regions as Functions of threshold in the Case of Three Classes

Jorge Isidro Aranda[1], Arturo Baltazar[2]

[1] UMSNH, Facultad de Ciencias Físico-Matemáticas,
Ciudad Universitaria, Morelia, Michoacán,
chilo@fismat.umich.mx
[2] Cinvestav-Saltillo, División de Robótica y Manufactura Avanzada,
Carretera-Saltillo-Monterrey, Km. 13.5, Ramos Arizpe, Coah., 25900, México
arturo.baltazar@cinvestav.edu.mx

Abstract. The performance of a pattern recognition system is characterized by its error rate and the reject rate tradeoff. The error rate can be directly evaluated from the reject function assuming a threshold t, which is a parameter that limits the reject and acceptance regions. In this paper, optimum rejection rule of a recognition system for three classes is used to calculate the reject function and the error rate evaluated directly from the reject rate. A Bayes decision rule as a function of threshold t is used to determine the minimum risk. A simple parametrization that considers the distance between means of the normal distribution of classes is presented. Finally, the rejection rate, the error rate and conditional risk were estimated in terms of the threshold t to illustrate the effect of the proposed parametrization and Bayes decision rule for minimum risk.

Keywords: Bayes error; minimum risk; reject rate.

1 Introduction

It is known that the Bayes error provides the lowest error rate for a given pattern classification problem. An optimum rejection rule and a general relation between the error and reject probabilities are presented by Chow (1970). There are several classical approaches used to estimate or to find bounds of the Bayes error including those proposed by Thumar et al. (1966) and Doermann (2004) which considered the second order dependency between the class and decision, and found that a combined-based method renders better estimates than the classical methods of dependency-based product approximation (DBPA). Pierson (1998) used boundary methods for estimating class separability since it does not require knowledge of the posterior distributions. In this work, optimum rejection rule of a recognition system to calculate the reject function and the error rate evaluated directly from reject rate and the results are illustrated for a case involving three classes, commonly found in actual classification problems but hardly described in the literature. The Bayes decision rule for minimum error and reject option for n classes considering the minimum risk,

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 61-71

Received 03/06/06
Accepted 03/10/06
Final version 11/10/06

assuming a threshold t with a simple parametrization for a three class case is described.

2 Background

2.1 Decision Rule for Minimum Error and Reject Option

In actual classification problems where classes are not fully separable, it is unrealistic to expect absolute classification performance of the pattern recognition system. The object of a statistical classification problem is to reach the best possible performance. The question that arises is how to determine the optimum classification rate which can be answered by the determination of the Bayes error since the Bayes decision rule provides the lowest error rates (Tumer, 1996).
In general, to assign a pattern x to n classes wi (where i =1,…,n), a model for classification with a decision rule to partition the measurement space into n regions Ωi, i =1,…, n is needed. The boundaries between the regions Ωi are known as the decision boundaries or decision surfaces; usually it is near to these boundaries that the highest probability of misclassifications can occur. In such situations, the decision on the pattern may be withheld or rejected until further information is available, this option is known as the reject option (Webb, 2002).

According to the Bayes decision rule where assigning with minimum error a pattern x to a class wi, we have that:

$$p(w_i) > p(w_k) \quad k, i = 1,...,n;\ k \neq i \,, \tag{1}$$

where $p(w_1)$,…, $p(w_n)$,are known prior probabilities.

The optimum decision rule is to reject a pattern x if the maximum of the posterior probabilities does not exceed some predefined threshold t, which can take values between 0 and 1 (0 ≤ t ≤1). (Chow, 1970; Pierson, 1998). More explicitly, the optimum recognition rule is to accept the pattern x and to classify it as belonging to the kth class whenever the following is true:

$$p(w_k)p(x|w_k) \geq p(w_i)p(x|w_i) \quad , \tag{2}$$

and

$$p(w_k)p(x|w_k) \geq (1-t)\sum_{i=1}^{n} p(w_i)p(x|w_i)\,, \tag{3}$$

and to reject the pattern whenever:

$$\max_i \left[p(w_i)p(x|w_i) \right] < (1-t)\sum_{i=1}^{n} p(w_i)p(x|w_i) \ . \tag{4}$$

The term (1-t) (see Figure 2) indicates the maximum values that can be assigned to the posterior probability p(*w*i| x) to do a correct classification of a measurement pattern.Through Bayes's theorem, this posterior probability function is related to the class conditional density by:

$$p(w_i|x) = \frac{p(x|w_i)p(w_i)}{p(x)}, \tag{5}$$

where

$$p(x) = \sum_{i=1}^{n} p(w_i)p(x|w_i) \tag{6}$$

is the probability of pattern x. The class probability distributions $p(w_i)$ can be estimated using an iterative method with the patterns belonging to each class (Baram, 1999).

For any fixed value of t, the decision rule (this is used for the correct classification of samples) partitions the pattern space into two disjointed sets (or regions) A(*t*) and R(*t*) given by:

$$A(t) = \left\{ x \middle| \max_i p(w_i)p(x|w_i) \geq (1-t)p(x) \right\}, \tag{7}$$

$$R(t) = \left\{ x \middle| \max_i p(w_i)p(x|w_i) < (1-t)p(x) \right\}, \tag{8}$$

where A(*t*) is the acceptance region which implies that once the maximum posterior probability exceeds the threshold (1-t), a classification decision can be made; R(*t*) is the rejection region where the equations (3) and (4) hold. The integral of regions A(*t*) and R(*t*) defines the reject rate r(t) and correct classification c(*t*) expressed as:

$$r(t) = \int_{R(t)} p(x)\,dx\,, \tag{9}$$

which describes the unconditional probability of rejecting a measurement x and

$$c(t) = \int_{A(t)} \max_i \left[p(w_i)p(x|w_i) \right] dx\,, \tag{10}$$

is the probability of correct recognition of the patterns of the measurements. The probability $e(t)$ of accepting a pattern for classification and incorrectly classifying it is known as error rate given by:

$$e(t) = 1 - c(t) - r(t). \quad \textbf{(11)}$$

A correct recognition can be done if given an error rate (error probability) the reject rate (reject probability) is minimized. In this work, a parametrization to illustrate its effect on error and reject rate as well as the minimum risk for three classes is introduced following the work by Chow (1970) for the case of two classes.

Assuming two classes and a pattern x with equal prior probability of occurrence, $p(w_1) = p(w_2) = 1/2$, the condition for rejection (Eq. (4)) can never be satisfied when $t > ½$; this can be explained by the fact that the minimum value which $\max_i [p(w_i|x)]$ can attain is $1/n$ since $1 = \sum_{i=1}^{n} p(w_i|x) \le n \max_i p(w_i|x)$; using Eq. (4) the threshold rule $t \le 1 - 1/n$ can be obtained (Chow (1970); Webb (2002)) to activate the rejection option. The reject rate is always zero if t exceeds 1/2, therefore t only can have values in the range $0 \le t \le 1/2$. To estimate $r(t)$ and $e(t)$ two normal distributions $p(x|w_i) = (1/\sigma\sqrt{2\pi})\ \exp(-(x-\mu_i)^2 / 2\,\sigma^2)$ are assumed with means μ_1 and μ_2 ($\mu_1 > \mu_2$) and equal covariance σ^2. Chow (1970) used the following parametrization:

$$s_2 = \frac{\mu_1 - \mu_2}{\sigma}, \quad \textbf{(12)}$$

which describes the separation between the means of the distributions and is the only parameter of the distributions that $r(t)$ and $e(t)$ depend upon. The error and reject rates can be expressed using the standard cumulative distribution function $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp(-x^2/2)\,dx$ as: $e(t) = \Phi(a)$ and $r(t) = \Phi(b) - \Phi(a)$, where $a = -s_2/2 - \ln(1/t-1)/s_2$ and $b = -s_2/2 + \ln(1/t-1)/s_2$ (Chow, 1970) . Figure 1, shows results of error and reject rate in terms of the threshold t and $s_2 = 1, 2, 3, 4$. All curves tend to zero when t=1/2 and to 1 when t=0 proving consistency with the threshold rule when n=2. The results for two classes are now compared with results in the case of three classes.

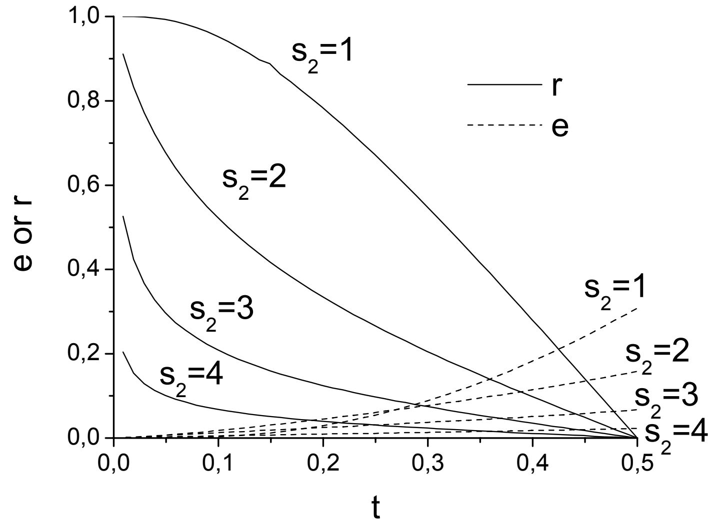


Fig. 1. Reject and error rate in terms of threshold t for two classes considering several values of the parameter s_2 are shown.

3. Estimation of Error and Reject Rate and Risk for Three Classes

3.1 Error and Reject rate

Accordingly with previous section, to calculate the error rate, Eq. (11), it is necessary to have the reject rate and the probability of correct recognition; to obtain such results the prior class-conditional probability density functions $p(x|w_i)$ for each class is needed. In the case of three classes, assuming a Gaussian distribution, these can be found as follows:

$$p(x|w_i) = \frac{1}{\sqrt{2\pi}\,\sigma}\left(e^{-(x-\mu_i)^2/2\sigma^2}\right). \quad \textbf{(13)}$$

In this case a parametrization is introduced after a change of variable in terms of the means and standard deviation as follows: $y = (x - \mu_1)/\sigma$, s_2 (given by Eq. 12) and $s_3 = (\mu_3 - \mu_1)/\sigma$. After some algebra and arranging terms in Eq. (13) (for i=1,2,3), the corresponding prior class-conditional probability density functions for three classes are:

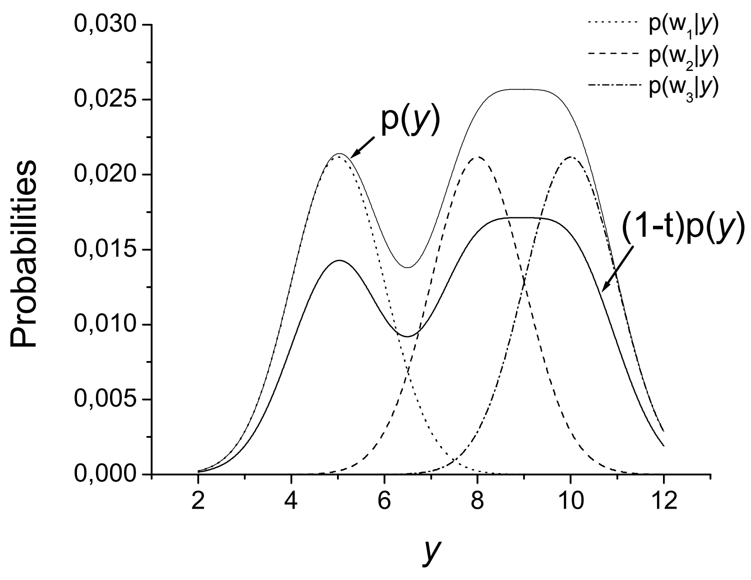
$$p(y|w_1) = \frac{1}{\sqrt{2\pi}\,\sigma}\left(e^{-y^2/2}\right),$$

$$p(y|w_2) = \frac{1}{\sqrt{2\pi}\,\sigma}\left(e^{-(y-s_2)^2/2}\right), \qquad (14)$$

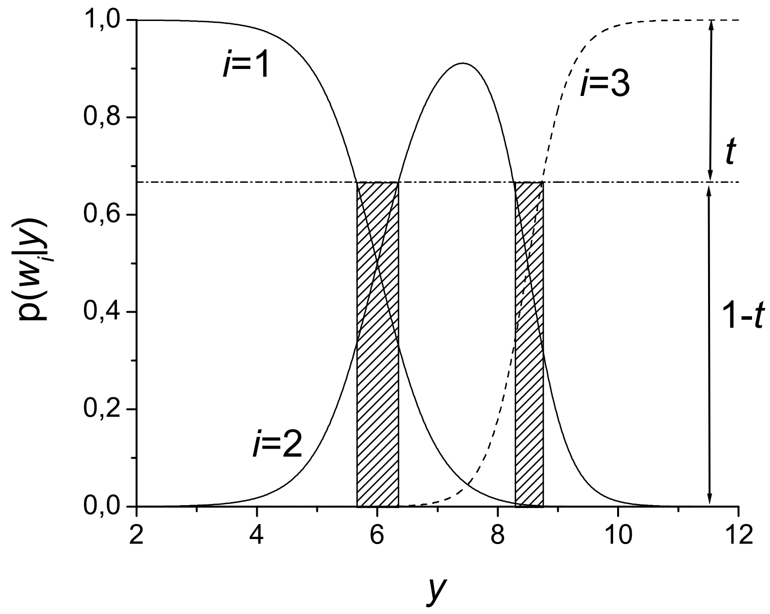
$$p(y|w_3) = \frac{1}{\sqrt{2\pi}\,\sigma}\left(e^{-(y-s)^2/2}\right),$$

The posterior probability densities for each class $p(w_1|\,y\,)$, $p(w_2|\,y\,)$ y $p(w_3|\,y\,)$ are now given using eq. (14) in Eq.(5).

It can be seen that Eq. (5) allows easy visualization of the intersection points for different values of s_3 and s_2 facilitating to calculate the area under the reject rate curve, as shown in Figure 2a. Two reject regions can be found, the first between the intersection points of $p(w_1|\,y\,)$ and $p(w_2|\,y\,)$ with (1-t), and the second between the intersection of $p(w_2|\,y\,)$ and $p(w_3|\,y\,)$ with (1-t) (Figure 2b).



(a)



(b)

Fig. 2. (a) Probability densities considering threshold *t* calculated from Eq. (3) assuming equal class probabilities; (b) a posterior probability densities estimated using Eq. (5); where the horizontal dashed line represents (1-*t*) the horizontal threshold; *i* = 1, 2, 3 indicates the three class-condition probabilities used in the calculations. The threshold *t* varies between the interval ($0 \leq t \leq 2/3$) in the case of three classes.

The error and reject rate for two cases $s_3 = 1$, $s_2 = 0.5$ and $s_3 = 2, s_2 = 1$ are shown in Figure 3; here it can be seen that the limits of error rate and reject rate corresponds to the maximum value of threshold *t* =2/3 for three classes and with a minimum value in *t*=0. Hence, this proves the consistency with the threshold rule $t \leq 1 - 1/n$ when n=3. It can be observed that the curves for *e* and *r* in the case of three classes shown in Figure 3 are different of those for two classes described in Figure 1.

3.2 Decision Rule for the Minimum Risk.

In the previous section, the decision rule was such that the selected class has the maximum posterior probability $p(w_i| y)$ minimizing the probability of making an error. Therefore, a new decision rule that minimizes the expected *loss* or risk is

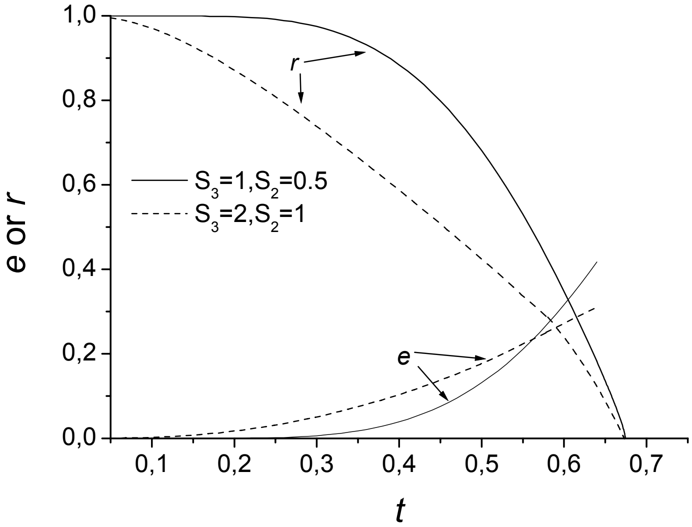


Fig. 3. Estimation of error (*e*) and reject (*r*) rate in terms of threshold *t* for three classes assuming $s_3 = 1$, $s_2 = 0.5$ (solid line) and $s_3 = 2$, $s_2 = 1$ (dash line).

described in Webb (2002); this is very important since in many applications the costs associated with misclassification depend upon the true class of the patterns as well as the class to be assigned. This loss is a measure of the cost of making the decision that a pattern belongs to class w_i when the true class is w_j.
The conditional risk of assigning a pattern y to class w_i can be defined as (Webb, 2002)

$$l^i(y) = \sum_{j=1}^{n} \lambda_{ji} p(w_j | y), \tag{15a}$$

where

$$\lambda_{ji} = \text{cost of assigning a pattern y to } w_i \text{ when } y \in w_j \ . \tag{15b}$$

In this case, a reject option can be introduced to establish a Bayes decision rule in terms of the conditional risk, the reject region R^* can be defined by $R^* = \{\min_i l^i(y) > t\}$ (Webb, 2002); the decision rule is to accept a pattern y and assign it to a class w_i if $l^j(y) = \min_j l^j(y) \le t$ and reject y if $l^j(y) = \min_j l^j(y) > t$, this decision is equivalent to make a definition of a region in which is valid a constant conditional risk $l^0(y) = t$, so that the Bayes decision rule is: to assing y to class w_i if $l^i(y) \le l^j(y)$ with j=0, 1...,**n**. This implies that the Bayes decision rule for minimum risk (Webb,2002) gives the minimum risk r^* given by

$$r^* = \int_R t\, p(y)dy + \int_A \min_{i=1,\dots,n} l^i(y) p(y)\, dy. \qquad \textbf{(16)}$$

In this work, the expression in previous equation is used to calculate the minimum conditional risk in terms of the threshold t for the case of three classes. The posterior probability densities $p(w_i| y)$ and the threshold t are used to describe two regions which are disjoined and complete the reject region. The posterior probability densities contribute to the reject options used for the calculation of minimum conditional risk defined in Eq. (16).

Using the Eq. (15), the conditional risk for the case of three classes can be written as:

$$l^1(y) = \lambda_{11}\, p(w_1|y) + \lambda_{21}\, p(w_2|y) + \lambda_{31} p(w_3|y), \qquad \textbf{(17a)}$$

$$l^2(y) = \lambda_{12}\, p(w_1|y) + \lambda_{22}\, p(w_2|y) + \lambda_{32} p(w_3|y), \qquad \textbf{(17b)}$$

$$l^3(y) = \lambda_{13}\, p(w_1|y) + \lambda_{23}\, p(w_2|y) + \lambda_{33} p(w_3|y). \qquad \textbf{(17c)}$$

A loss function that has been used extensively in practice because of its simplicity and sensibility is the symmetrical or zero-one loss function. Then, Eq. (15b) results in:

$$\lambda_{ij} = \begin{Bmatrix} 0 \text{ if } i = j \\ 1 \text{ if } i \neq j \end{Bmatrix}. \qquad \textbf{(18)}$$

This function assigns no loss to a correct decision, and assign a unit loss otherwise, thus, all errors are equally costly. Therefore, minimizing the Bayes risk corresponds to maximizing the posterior probability (Webb, 2002). Considering Eq. (15b) or Eq. (18) it can be found that all the terms in the diagonal of Eq. (17) are zero.

After calculations, the minimum risk r^* for three classes can be calculated with Eq. (16) as shown in Figure 4 in terms of the threshold t, the parameters considered here are $s = 1$, $s_2 = 0.5$ and $s_3 = 2$, $s_2 = 1$. In this Figure 4 it is shown the minimum risk in terms of threshold. The results are consistent with fact that the risk goes to zero when t=0 and is maximum when t=2/3. The best performance of a pattern recognition system is realized when this minimum risk is reached.

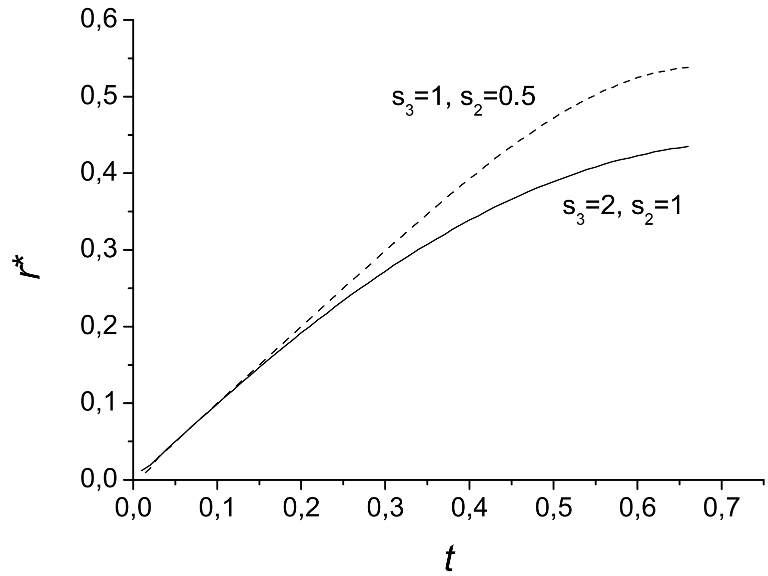


Fig. 4. Minimum risk r^* in terms of threshold t for three classes, with parametrization $s_3 = 1$, $s_2 = 0.5$ (dash) and $s_3 = 2$, $s_2 = 1$ (solid).

4 Conclusion

The Bayes decision rule to obtain the reject rate for a pattern recognition system for the case of three classes was estimated. A simple parametrization to illustrate the error rate as function for a given threshold was proposed. It was showed the validity of the threshold rule $t \leq 1 - 1/n$ in the case of three classes. It was discussed that when a threshold *t*, which partitions the measurement space, is fixed, determination of the minimum risk and error rate is allowed. It was showed that through the definition of the threshold *t* a more efficient pattern recognition system can be reached if the minimum risk is known.

Acknowledgments. The authors appreciate the financial support of CONACYT-SEP through the grant #48085. It is also appreciated to the CIC of the UMSNH for their economic support by means of the projects 9.11 and 9.23.

References

1. Pierson W. E. "Using Boundary Methods for Estimating Class Separability", Ph.D. Thesis, The Ohio State University, 1998.
2. Webb, A. "Statistical pattern recognition", Ed. John Wiley & Sons, LTD, England, 2002.
3. Chow, C. K., "On Optimum Recognition Error and Reject Tradeoff", IEEE Transactions on Information Theory, Vol. IT-16, No. 1, January, 41-46, 1970.
4. Specht, D., "Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification", IEEE Transactions on Neural Networks Vol. 1, No. 1, 525-532, 1990.
5. Baram, Y., "Bayesian classification by iterated weighting", Neurocomputing, 25, 73-79, 1999.
6. Doerman D. and Kang H., "Product Approximation by Minimizing the Upper Bound of Bayes Error Rate for Bayesian Combination of Classifiers", Proceedings of the 17th ICPR'04, IEEE, 1051-4651, 2004.
7. Tumer K. and Ghosh J. "Estimating the Bayes Error Rate Through Classifier Combining", Proceedings of ICPR'96, IEEE, 1015-4651, 1996.

A Study on Assessment Customer Defaults of Mortgage Loans

Yanyou Hao[1, 2], Hongyang Bao[2, 3], Zhongxian Chi[1] and Bo Wang[1]

[1] Department of Computer Science and Engineering,
Dalian University of Technology,
Dalian, 116024,China
[2] Dalian Branch of China Construction Bank,
Dalian, 116011,China
[3] School of Management, Dalian University of Technology,
Dalian, 116024,China
haoyanyou@yahoo.com.cn, haoyanyou@126.com

Abstract. Credit risk is the primary source of risk to financial institutions. As part of the credit risk assessment, the New Basel Accord suggests more granularity in risk-rating classes than currently exists. Credit scoring is one of important tools help financial institutions to hedge the credit risks. Support Vector Machine (SVM) is a new machine learning method based on the idea of VC dimension and Statistical Learning Theory (SLT). It is a good classifier to solve binary classification problem and the learning results possess stronger robustness. In this paper default prediction model of the housing mortgage loan is established by using SVM. We use grid-search method adjusts these penalty parameters to achieve better generalization performances in our application.

Keywords: Credit scoring; Credit risks; Support Vector Machine (SVM); Grid-search; Hoausing mortgage loan.

1 Introduction

Credit risk is the primary risk facing financial institutions. With the proposed guidelines under the New Basel Accord, financial institutions will benefit from better assessing their risks [1]. Credit risk is commonly defined as the loss resulting from failure of obligors to honor their payments. Arguably a cornerstone of credit risk modeling is the probability of default (PD) [2]. The housing mortgage is an important component of bank loan. However, there are huge credit risks that banks take back the principals and interest of loan with default of the consumers. Managers in the bank require the ability to predict the proportion of mortgages that will be defaulted. Many methods have been used to estimate default. Standard & Poors (S&P) and Moody have employed accounting analytic and migration analysis to predict the probability of default. Statistical methods for forecasting default risk include linear discriminant analysis [3], [4], the logistic regression model [5], [6]. The above methods typically require large data to build the forecasting model. However, there are not large data to use in real-life.

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 73-81

Received 02/06/06
Accepted 03/10/06
Final version 10/10/06

There is a credit scoring system for consumer mortgage loan application to produce an internal rating. It is a traditional approach based an empirical model, which takes into account various quantitative as well as subjective factors, such as the consumers' age, household income, interest rate, etc. Through this scoring system, analyzing all the pieces of information in your credit record and summarizing them in a number calculate a credit score of a consumer. A company named Fair, Isaac & Co. (FICO) developed a mathematical way to look at factors in your credit record that may affect your ability and willingness to repay a debt [7]. The problem with this approach is of course the subjective aspect of the prediction, which makes it difficult to make consistent estimates. The credit scoring problems can transform to classification than predict the probabilities of defaults. Recent researches have shown that Artificial Intelligence (AI) methods achieved better performance than traditional statistical methods [8], [9].

This paper applies SVM to estimate the mortgage default. In general, SVM has good generalization performance. However, there are some cases that the numbers of data in different classed are imbalance. The over-fitting of classifier affects the generalization performance of model. The kernel parameters γ and upper bound *C* control the generalization of SVM. Chang and Lin give a grid-search method to find the best parameter for SVM kernel and upper bound *C* [10]. We extend and use this method that adjusts these penalty parameters to achieve better generalization performances in our application.

This paper is organized as follows. Section 2 introduces the Support Vector Machine. In section 3, we described the methodology and procedure in assessment mortgage default, and defined default accuracy and normal accuracy to measure the performance of prediction model. In section 4, a real-life mortgage data set is to test the prediction model. Finally, we summarize the work in section 5.

2 Support Vector Machine and Parameter Selection

In this section, we give a brief review of SVM classification. SVM is a novel-learning machine first developed by Vapnik. It is based on the Structural Risk Minimization (SRM) principle from computational learning theory [11]. We consider a binary classification task with input variables $x_i (i = 1,...,l)$ having corresponding labels $y_i = \pm 1$. SVM finds the hyperplane to separate these two classes with a maximum margin. This is equivalent to solving the following optimization problem:

$$\text{Minimize: } \frac{1}{2} w^T \cdot w$$
$$\text{Subject to: } y_i (w \cdot x_i + b) \geq 1 \tag{1}$$

By introducing Lagrange multiples α_i for the constraints in the (1), the problem can be transformed into its dual form

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) - \sum_{j=1}^{l} \alpha_j$$

$$\textit{Subject to}: \sum_{i=1}^{l} y_i \alpha_i, \qquad \alpha_i \geq 0, i = 1, ..., l \tag{2}$$

Fig. 1 is a sample linearly separable case. Solid points and circle points represent two kind of sample separately. H is the separating hyperplane. H_1 and H_2 are two hyperplane through the closest points (the Support Vectors, SVs). The margin is the perpendicular distance between the separating hyperplane H_1 and H_2.

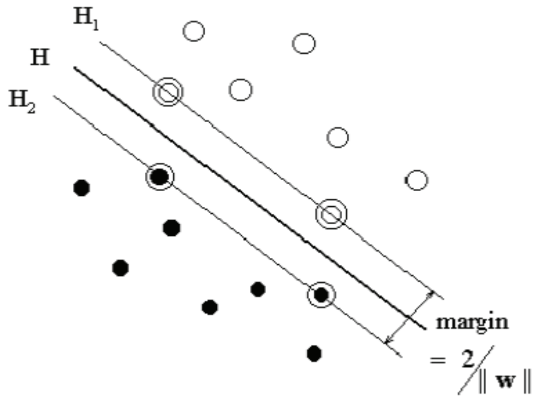


Fig. 1: Optimal Separation Hyperplane

To allow some training errors for generalization, slack variables ξ_i and penalty parameter C are introduced. The optimization problem is re-formulated as

$$\text{Minimize } \frac{1}{2} w^T \cdot w + C \sum_{i=1}^{l} \xi_i$$

$$\text{Subject to } y_i (w \cdot x_i + b) \geq 1 - \xi_i \tag{3}$$

The purpose of $C\sum_{i=1}^{l} \xi_i$ is to control the number of misclassified samples. The user chooses the parameter C so that a large C corresponds to assigning a higher penalty to errors term [12]. In addition, for some classification problems, numbers of data in different classes are imbalanced. The imbalance may come from the unequal proportion of samples between the different classes or the unequal density of the clusters in the feature space even if the populations are the same. Thus, a method using different penalty C_+ and C_- for each class to adjust the penalties on the false positive and false negative [13]. Finding this hyperplane can be translated into the following optimization problem:

$$\text{Minimize } \frac{1}{2}w^T \cdot w + C_+ \sum_{i:y_i=1} \xi_i + C_- \sum_{i:y_i=-1} \xi_i$$

$$\text{Subject to} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i \tag{4}$$

For nonlinear case, we map the input space into high dimension feature space by a nonlinear mapping. With a suitable choice of kernel the data can become separable in feature space despite being non-separable in the original input space. Here are three kinds of kernel function which are most commonly used:

Polynomial: $K_{poly}(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0.$

RBF: $K_{rbf}(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0.$

Sigmoid: $K_{sig}(x_i, x_j) = \tanh(\gamma x_i^T x_j + r).$

Here, γ , r and d are kernel parameters. Every kernel has its advantages and disadvantages [14]. The kernel type, kernel parameters and the penalty parameter C control the generalization of Support Vector Machines. The best choice of kernel of C depends on each other and the art of researcher.

3 Methodology

The estimation of mortgage defaults is a two-class classification task. Accuracy is the typical performance measure for two-class classification schemes. However, two learning algorithms can have the same accuracy, but the one which groups the errors near the decision border is the better one.

In order to appraise the performance of the classifier, Default Accuracy and Normal Accuracy are selected as standard criteria. We define the default accuracy and normal accuracy as:

$$\text{Default Accuracy} = \frac{\text{default samples classified}}{\text{total default samples}} \tag{5}$$

$$\text{Normal Accuracy} = \frac{\text{normal samples classified}}{\text{total normal samples}} \tag{6}$$

The advantage of the default accuracy and normal accuracy is that that they are a good indicator of whether the errors are close to the decision border or not. Given two classifiers with the same accuracy, the one with high default accuracy and normal accuracy is the better one. This definition is equivalent to the definitions of False Alarm and Miss Rate in [15].

Since there are three measures of SVM performance, searching for the optimal solution of SVM parameters is a multi-objective programming. We use RBF kernel to map the input space into high dimension feature space, and use different upper bounder C_+ and C_- for different classes.

The grid-search method use cross-validation on C and γ . Basically pairs of (C, γ) are tried and the one with the best cross-validation accuracy is picked. The grid-search is straight forward but seems stupid. However, there are two motivations why we prefer the simple grid-search approach. One is that psychologically we may not feel safe to use methods which avoid doing an exhaustive parameter search by approximations or heuristics. The other reason is that the computational time to find good parameters by grid-search is not much more than those by advanced methods since there are only two parameters. Furthermore, the grid-search can be easily parallelized because each (C,γ) is independent. The grid-search method in [10] just is used in training data sets; we extend to use the method in training and test data sets. The best parameters not only are of the training data sets but also of test data sets.

4 Experiments Results and Analysis

The available default estimation data set in housing mortgage loan is provided by a major commercial bank of China. This data set contains 18960 samples from January 1998 to December 2004. We defined two classes: “good” and “bad” customers. The “bad” customer is the borrower with at least one defaulted instalment, or more than 3, the one that did not pay one instalment over a period of three months. The “good” customer is the borrower who repayment on time. The data is typically from a sample of applicants who have been granted credit already. The data is imbalanced; the “bad” class is rare class.

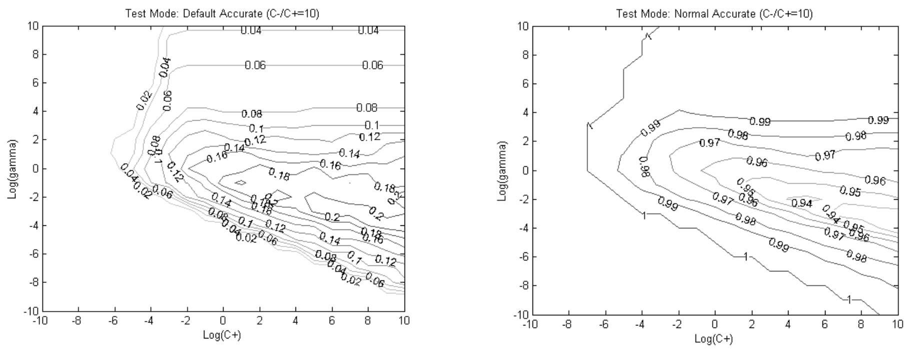
All customers are the bank’s consumer who had applied to the bank for mortgage loan. The Data sets consisted of the customer information and the other information of the loan application form. There are 3 categorical attributes and 11 numerical attributes. Application characteristics available in the data set are summarized in Table 1. The “good” customers are labeled “1” and the “bad” customers are labeled “–1”.

Table 1. The input variables

Index	Indicators
1	Customer Age
2	Educational Level
3	Vocation
4	Working at Industry
5	Years at current work
6	Household income
7	Price of the House
8	Area of House
9	House value at purchase
10	Monthly Payment
11	First Payment Ratio
12	Amount of Contract
13	Loan Terms
14	Balance of Loan

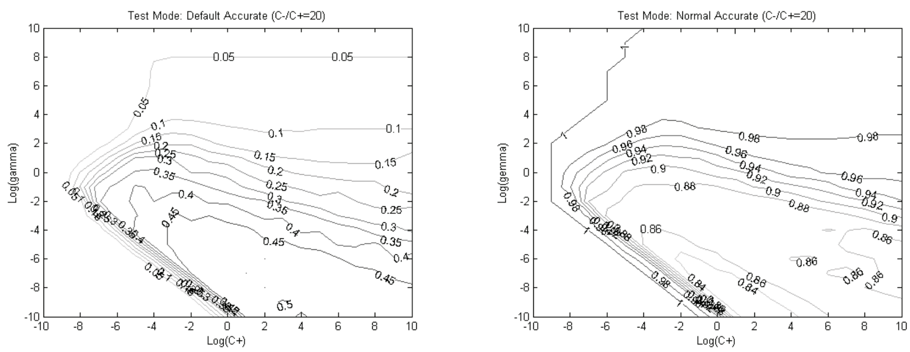
We spitted the data into two sub-datasets evenly, one half for training and the other half for testing, this is referred to as 50-50 split experiment. The training set includes 9480 samples with 241 “bad” customers and 9239 “good” customers. The test set includes 9480 samples with 240 “bad” customer and 9240 “good” customers. The data set is imbalanced between the “bad” and “good” class.

Our SVM for default prediction training code is a modification of LIBSVM [16]. The training tools of LIBVM can set different weight on the parameter *C* of two classes. Since doing a complete grid-search is a time-consuming work, so we parallel do this experiments on 3 PC servers at same time. We give the range of *C* and γ in $2^{-10} \sim 2^{10}$ and C_{-}/C_{+} = {10, 20, 30}. We get 3 group experiment results and there are 421 records in every group. The model’ performance of generalization attracts our more attentions. The Fig. 2 to Fig. 3 show a part of the experiment results.



(a) (b)

Figure 2. Where C_{-}/C_{+} =10, test mode: (a) the highest Default Accuracy is 0.20; (b) the lowest Normal Accuracy is 0.95.



(c) (d)

Figure 3. Where C_{-}/C_{+} =20, test mode: (c) the highest Default Accuracy is 0.50; (b) the lowest Normal Accuracy is 0.84.

Table 2 shows a group of detail results in training sets and test sets.

The predicting total accuracy is around 70% both on training and test phases especially a similar results in Default Accuracy and Normal Accuracy where C_{-}/C_{+} = 30 and Log (C_{+}) = 0 and Log (γ) = -6. We consider that the model is under-fitting on the training set with those parameters. When the penalty parameter *C* becomes too small the error term impact will be decrease on the training process. In general, the real life application is one of nonlinear-classifiable case; we need a relatively big penalty parameter *C* in fact.

Table 2. The grid-search results

			Training sets			Test sets		
C_{-}/C_{+}	Log (C_{+})	Log (γ)	Total Accuracy	Default Accuracy	Normal Accuracy	Total Accuracy	Default Accuracy	Normal Accuracy
10	**10**	**-2**	**98%**	**95%**	**98%**	**93%**	**22%**	**95%**
10	6	-3	95%	63%	96%	92%	**22%**	94%
10	10	-4	95%	70%	96%	91%	**22%**	93%
10	7	-3	95%	71%	96%	92%	**21%**	93%
10	9	-4	95%	65%	96%	92%	**21%**	94%
10	8	-3	95%	77%	96%	91%	**21%**	93%
20	**3**	**-9**	**84%**	**54%**	**85%**	**83%**	**50%**	**84%**
20	1	-7	84%	53%	85%	83%	**50%**	84%
20	2	-8	84%	54%	85%	83%	**50%**	84%
20	2	-6	85%	61%	86%	86%	**50%**	87%
20	0	-5	86%	57%	87%	86%	**50%**	87%
20	8	-10	85%	56%	85%	85%	**50%**	86%
30	**0**	**-6**	**74%**	**76%**	**74%**	**70%**	**69%**	**70%**
30	2	-8	73%	76%	73%	68%	**69%**	68%
30	3	-8	74%	74%	74%	71%	**69%**	71%
30	0	-7	73%	74%	73%	67%	**68%**	67%
30	1	-7	74%	76%	73%	69%	**68%**	69%
30	-3	-4	75%	74%	75%	70%	**68%**	70%
30	-1	-6	73%	73%	73%	68%	**68%**	68%

The predicting total accuracy is above 90% both in training sets and test sets where C_{-}/C_{+} = 10 and Log (C_{+}) = 10 and Log (γ) = -2, but the Default Accuracy is very poor in test phase. Obviously we must to provide more detailed analysis why the results are so poor. We discuss the problem with those experts of the commercial bank. They give some explanations about the imbalance data set. Due to the housing mortgage loan belong to a long-term loan; the period of repayment of credit is between 5 and 30 years in general. In China the loans are developed just before few years ago. The default period is not coming so soon. There is a reason for the poor result that credit information of customers is incomplete and some dirty data in the data sets.

We compare the respectively-penalty SVM (RP-SVM) in this paper with classic SVM and other machine learning algorithms. We select the classic SVM, BP Neural Network and Decision Tree algorithms to training the same data set. Then we predict the same test data set using their classification model respectively. Table 3 shows the comparison of predicting result of those algorithms.

Table 3. The comparison of predicting result

Classification Algorithm	Normal Accuracy	Default Accuracy	Total Accuracy
RP-SVM	95.31%	22.45%	93.47%
Classic SVM	95.60%	16.63%	93.60%
BP Neural Network	95.10%	12.05%	92.50%
Decision Tree	95.29%	14.96%	93.26%

It is obvious in the Table 3 that Default Accuracy of RP-SVM is better than other methods under similar total predicting accuracy, which the predicting accuracy can increase just 5%. Our future direction of the research would focus on how to improve the Default Accuracy especially in the testing data set.

5 Conclusions

As we have shown, SVM are capable of estimation defaults from real life mortgage data. We can adjust just few parameters to obtain the results not very obvious at first glance. This makes SVM particularly well suited as an underlying technique for credit rating and risk management methods applied by financial institution. Further research focus on how to improve the Default Accuracy and Normal Accuracy in the test data set. We believe that deeper data preprocessing and more suitable parameters selection will contribute to improve the performance of generalization. Extending the two-class classification to multi-class classification and introducing fuzzy SVM are also our future research work.

References

1. Allen M. Featherstone, Laura M. Roessler,and Peter J. Barry.: Determining the Probability of Default and Risk-Rating Class for Loans in the Seventh Farm Credit District Portfolio. Review of Agricultural Economics, Volume 28 Page 4 - March 2006
2. Til Schuermann and Samuel Hanson.: Estimating Probabilities of Default. Federal Reserve Bank of New York Staff Reports, no. 190, July 2004
3. Altman, E. I. (1968).: Financial Ratio Discriminant Analysis and the Prediction of Corporate Bankruptcy. Journal of Finance, 23(3), 589-609
4. Altman, E. I., Marco, G., and Varetto, F. (1994).: Corporate Distress Diagnosis: Comparisons Using Linear Discriminant Analysis and Neural Networks (The Italian Experience). Journal of Banking and Finance, 18, 505-529
5. Martin, D. (1977).: Early warring of Bank Failure: A Logic Regression Approach. Journal of Banking and Finance, 1, 249-276
6. Smith, L. D., and Lawrence, E. C. (1995).: Forecasting Losses on a Liquidating Long-term Loan Portfolio. Journal of Banking and Finance, 19, 959-985
7. www.fairisaac.com
8. Hand D.J. and Henley W.E. (1997): Statistical Classification Methods in Consumer Credit Scoring: a Review. Journal of the Royal Statistical Society, A, 160, 523-541
9. Zan Huang and Hsinchun Chen.: Credit rating analysis with support vector machines and neural networks: a market comparative study. Decision Support Systems 37(2004) 543-558
10. Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin.: A Practical Guide to Support Vector Classification. Technical Report, National Taiwan University, Taiwan (2001)
11. C. Cortes and V. Vapnik.: Support-vector networks, Machine Learning. vol 20, no.3, 273-297, 1995
12. Osuna E., Freund R., Girosi F.: Support vector machines: Training and applications. Massachusetts Institute of Technology, AI Memo No. 1602. 1997
13. K. Morik, P. Brockhausen, and T. Joachims.: Combining statistical learning with a knowledge-based approach, a case study in intensive care monitoring, in Proc. 16th International Conference on Machine Learning, Bled, Slovenia, 1999
14. Smits, G.F., Jordaan, E. M. Improved svm regression using mixtures of kernels. Proceedings of the 2002 International Joint Conference on Neural Networks, Vol. 3, 2002, Pages:2785-2790
15. Drucker H., Wu D., V. Vapnik.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks. Vol. 10, Issue 5, Sep 1999
16. Chih-Chung Chang, Chih-Jen Lin.: LIBSVM: a Library for Support Vector Machines. URL: http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf

Optimization Algorithms

Two-Stage Ant Colony System for Solving the Traveling Salesman Problem

Amilkar Puris[1], Rafael Bello[1], Ann Nowe[2] and Yailen Martínez[1]

[1] Department of Computer Science, Universidad Central de Las Villas, Cuba.
[2] Comp Lab, Department of Computer Science, Vrije Universiteit Brussel, Belgium.
{rbellop, ayudier, yailenm}@uclv.edu.cu, ann.nowe@vub.ac.be

Abstract. In this paper we propose a multilevel approach of Ant Colony Optimization to solve the Traveling Salesman Problem. In this case we use the Ant Colony System algorithm. The basic idea is to split the heuristic search performed by ants into two stages. Also, the effect of using local search was analyzed. We have studied the performance of this new algorithm for several Traveling Salesman Problem instances. Experimental results obtained show the Two-stage approach significantly improves the Ant Colony System in terms of the computation time needed.

1 Introduction

Ant Colony Optimization (ACO) is a metaheuristic used to guide other heuristics in order to obtain better solutions than those that are generated by local optimization methods. In ACO a colony of artificial ants cooperates to look for good solutions to discrete problems. Artificial ants are simple agents that incrementally build a solution by adding components to a partial solution under construction. This computational model was introduced in 1991 by M. Dorigo and co-workers [12] and [13]. Information about this metaheuristic can be found in [1], [3] and [5].

Ant System (AS) is the first ACO algorithm; it was introduced using the Traveling Salesman Problem (TSP) [2] and [4]. In TSP, we have a set of N fully connected cities $\{c_1, \ldots, c_n\}$ by arcs (i,j); each arc is assigned a weight d_{ij} which represents the distance between cities i and j, the goal is to find the shortest possible tour visiting each city once before returning to initial city. When ACO is used to solve this problem, pheromone trails (τ_{ij}) are associated to arcs which denote the desirability of visiting city j directly from city i. Also, the function $\eta_{ij}= 1/d_{ij}$ indicates the heuristic desirability of going from i to j. Initially, ants are randomly associated to cities. In the successive steps ant k applies a random proportional rule to decide which city to visit next according to expression (1):

$$p_{ij}^{k} = \frac{(\tau_{ij})^{\alpha} * (\eta_{ij})^{\beta}}{\sum_{l \in N_i^k} (\tau_{il})^{\alpha} * (\eta_{il})^{\beta}} \quad \text{if} \quad j \in N_i^k \quad (\textit{neighborhood of ant } k) \qquad \textbf{(1)}$$

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 85-92

Received 02/06/06
Accepted 03/10/06
Final version 09/10/06

where α and β are two parameters to point out the relative importance of the pheromone trail and the heuristic information, respectively. After all ants have built their tours the values τ_{ij} are updated in two stages. First, τ_{ij} values are decreased by evaporation, $\tau_{ij}=(1-\alpha)*\tau_{ij}$, using the parameter α, where $0<\alpha<1$. This to avoid unlimited accumulation of pheromone. Secondly, all ants increase de value of τ_{ij} on the arcs they have crossed in their tours, $\tau_{ij}=\tau_{ij}+Inc_{ij}$, where Inc_{ij} is the amount of pheromone deposited by all ants which included the arc (i,j) in their tour. Usually, the amount of pheromone deposited by ant k is equal to $1/C_k$, where C_k is the length of the tour of ant k.

Some direct successor algorithms of Ant Systems are: Elitist AS, Rank-based AS and MAX-MIN AS. A more different ACO algorithm is Ant Colony System (ACS). ACS uses the following pseudorandom proportional rule to select the next city j from city i.

$$j = \begin{cases} \arg\max_{l \in N_i^k} \left\{ \tau_{ij} * \left(\eta_{il}\right)^{\beta} \right\} & if \quad q \le q_0 \\ random\ selection\ according\ to\ (1) & otherwise \end{cases} \qquad (2)$$

where q is a random variable uniformly distributed in [0,1], q_0 which is a parameter taken in the interval [0,1], controls the amount of exploration, and $\alpha=1$ in the random selection (expression 1). In ACS, ants have a local pheromone trail update ($\tau_{ij}=(1-\xi)*\tau_{ij}+\xi*\tau_{ij}(0)$) applied after crossing an arc(i,j), where $\tau_{ij}(0)$ represents the initial value for the pheromone, and a global pheromone trail update ($\tau_{ij}=(1-\alpha)*\tau_{ij}+\alpha*Inc_{ij}$) executed only by the best-so-far ant.

In this paper, we propose a new approach to ACO in which the search process developed by ants is splitted into two stages. We have studied the performance of this proposal using the ACS algorithms. In the following, we analyze some related works and introduce the new algorithm. After that, the performance of it is studied in the case of the Travelling Salesman Problem (TSP). Finally we conclude our findings.

2 Related Work

The problem of finding low cost tours in reasonable time rather than solving the problem to optimality in the case of the TSP using a multilevel approach was addressed in [8]. The multilevel idea was first proposed by Bernard and Simon [9] as a method in speeding-up the recursive spectral bisection algorithm partitioning unstructured problems. It has been recognized that an effective way of accelerating search algorithms is to use multilevel techniques. The approach presented in [8] progressively coarsens the TSP, initializes a tour and then employs either the Lin-Kernighan or the Chained Lin-Kernighan algorithms to refine the solution on each coarsened problem; the resulting multilevel algorithm is shown to considerably enhance the quality of tours.

On the other hand, authors in [10] present a multilevel approach to ACO, for solving the mesh partitioning problem in the finite element methods. The multilevel ant colony algorithm performs very well and it is shown to be superior to several classical mesh partitioning methods. Their studies show that ACO was successful to solve the graph-partitioning problem in the case of graphs of smaller size as a result of this, they enhanced the basic ACO with a multilevel technique. That is, a set of the largest independent subgraph is created from the original graph, these are optimized using ACO, and then the optimized partition is expanded.

In the model proposed in this paper we use a multilevel approach to solve the TSP motivated by the fact that it was shown that a multilevel strategy is beneficial in order solve the TSP [14]. In this case, we introduce the multilevel approach to ACO. However, this model differs from the previous approach [10] in some aspects: (i) the search process developed by the ants is divided in two stages instead of on a grouping of cities; (ii) the search process developed by ants is partitioned into two more simple search processes instead of partitioning the problem into more simple subproblems; (iii) the partial solution obtained in the first level is used as initial state for the search of the ants in the second level.

3 Two-Stage ACS Algorithm

The Two-Stage Ant Colony System algorithm (TS-ACS) proposed in this investigation is based on the following idea: to divide the search process made by the ants in two stages, so that, in the first stage preliminary results are reached (partial solutions) that serve as an initial state for the search made by the ants in the second stage. In the case of TSP, this means that tours containing a subset of cities are generated in the first stage, in the second stage, these routes will serve like an initial state for the ants.

The determination of the initial state in which the search process starts has been an interesting problem in heuristic search. It is well known that the initial state has an important effect on the search process. The aim is to be able to approach the initial state to the goal state. Of course, it is necessary to consider an adequate balance between the computational cost of obtaining that initial state and the total cost; in other words, the sum of the cost of approaching the initial state to the goal state plus the cost of finding the solution from that "improved" initial state should not be greater than the cost of looking for the solution from a random initial state.

More formally, the purpose is the following. Let E_i be an initial state randomly generated, or obtained by any other method without a significant computational cost, E_i^* is an initial state generated by some method M that approaches it to the goal state, $C_M(E_i^*)$ denotes the cost of obtaining the state E_i^* from E_i using the method M, and $CC_{HSA}(x)$ is the computational cost to find a solution from state x using a Heuristic Search Algorithm (HSA). Then, the objective is that $C_M(E_i^*) + CC_{HSA}(E_i^*) < CC_{HSA}(E_i)$.

In the TS-ACS proposed here, the procedure to generate E_i^* and the HSA are both the ACS algorithm, so the objective is $C_{ACS}(E_i^*) + CC_{ACS}(E_i^*) < CC_{ACS}(E_i)$. As ACS is used in both stages, the difference between the 2 stages is obtained by giving different values to some parameters of the model in each stage.

A ratio (r) is introduced in order to establish the relative setting of the values of the algorithm's parameters in both stages; the ratio indicates which proportion of the complete search is given to the first stage. For instance, if r=0.3, means that the first stage will cover 30% of the search process and the second stage the remaining 70% (see an example of the application of this ratio in next section).

The setting of the ratio r has a high influence on the overall performance of the algorithm. A high value of r, say almost 1, causes the state E_i^* to be closer to the goal state, by doing so the value of $C_{ACS}(E_i^*)$ may increase and the value of $CC_{ACS}(E_i^*)$ will decrease. But, in addition to this balance between the costs of $C_{ACS}(E_i^*)$ and $CC_{ACS}(E_i^*)$, we have the problem about how much the space search is explored; while more greater is the rate r, the search in the second stage decreases for several reasons: (I) there are less ants working, (II) the amount of cycles decreases, and (III) although the quantity of possible initial states for the second stage must grow when r grows, that amount is already limited by the result of the previous stage.

Therefore, a key point is to study what value of rate r is the best in order to obtain the best balance between the searches in both stages. This value must allow:

- To minimize the value of $C_{ACS}(E_i^*)$ +$CC_{ACS}(E_i^*)$.
- To allow an exploration of the search space that guarantees to find good solutions.

4 Two-Stage ACS in the Travelling Salesman Problem

When applying the ACS algorithm to the TSP, the ants begin the search starting from random initial states; that is, in each cycle an ant begins its tour in a randomly selected city, and chooses the next city to visit using rule (2). In the beginning no pheromone information is available to guide the search, only the heuristic information is present. On the contrary, the TS-ACS constructs partial tours (they do not include all the cities) in the first stage; this information serves as initial state for the ants in the second stage of the search process. In other words, instead of restarting the search from scratch every cycle, ants use the partial tours built in the first stage as the starting point in the second stage.

In TSP the parameters whose values are set depending on the ratio r are: the quantity of ants (m) to use in each stage, the number of cycles to execute in each stage (nc), and the amount of cities (cc) that must be included in the tour in each stage. In the experiments reported below we compare the conventional single stage ACS with the proposed Two Stages ACS. The setting of the parameters is done in the following way. Suppose a setting of the parameters for the conventional ACS algorithm as follows: m=100, nc=100, and cc=30, and a setting of the ratio as r=0.3, then the values of these parameters for the Two Stage ACS were set as follows m_1=100*0.3=30, nc_1=100 * 0.3 = 30 and cc_1=30 * 0.3 = 9 for the first stage; and m_2=100 * 0.7 = 70, nc_2=100 * 0.7 = 70 and cc_2= 30. Meaning that 30 ants execute the ACS algorithm during 30 cycles starting from random cities and constructing tours of 9 cities. In the second stage, 70 ants will execute ACS algorithm during 70 cycles forming tours of 30 cities. This means that in the first stage 30% of the ants, search for solutions which is reduced in size (tours including only 30% of the cities), and this in 30% of the total number of cycles. In the second stage the remaing 70% of the ants are used, they get 70% of the total number of the cycles, in order to find solutions to the complete problem. When the first stage is finished, we select a subset of solutions (denoted by EI) containing a quantity (cs) of the best solutions (tours with shortest distance) found in the first stage.

A refinement algorithm based on a local search strategy is introduced to improve the results of the first stage which explores small regions of the solution space. This means, the partial solutions in the set EI resulting from stage 1 are improved using the well known 2-opt procedure [11]. The TSP 2-exchange neighborhood of a candidate solution s consist of the set of all solutions s^* that can be obtained from s by exchanging two pairs of arcs in any possible way [5].

This improved EI subset, provides the initial states for the second stage. This means that in each cycle of the second stage each ant chooses, in a random way, an element of EI is a tour and positions itself in a randomly selected city belonging to the tour. After that, the ant will add other nc2 cities to this tour using the ACS algorithm. The initial values of the pheromones in the second stage are those attained at the end of the first stage.

The TS-ACS-TSP algorithm is given below:

Given the parameters (beta, rho, epsilon, cc, factor r, number of solutions in EI (cs))

```
P0: Define the quantity of ants (m) either like an
input data or by using some method depending on the
number of cities.

P1: Stage 1.
P1.1: Calculate the parameters for the first stage:
        m1= r*m
        nc1= r*nc
        cc1= r*cc
P1.2: Apply the ACS algorithm, which in the first stage
develops nc1 cycles.
P1.3: Set of tours ← Tours generated by ACS algorithm
in the first stage.

P2: Stage 2.
P2.1: Calculate the parameters for the second stage:
        m2= m-m1
        nc2= nc-nc1
        cc2= cc
        EI ← Selecting the best cs solutions from Set
              of tours.
P2.2: Apply a local search to improve partial solutions
in EI set.
P2.3: Apply the ACS algorithm, which in the second
stage develops nc2 cycles, using the elements of EI like
initial states for the ants in the second stage.
```

5 Experimental Results

A comparative study of ACS and TS-ACS algorithms in the TSP was done by using public available (or benchmark) data base for this problem [7].

For these experiments we use the following parameter setting: α=0.1, ξ=0.1, a quantity of ants m=10 and a number of cycles nc=1000. Firstly, we studied the values {1, 3, 5} and {0.3, 0.6, 0.9} for β and q_0 respectively. Table 1 shows the different values of β and q_0 with which the best solution was obtained for each database. These values β and q_0 were selected to develop the next experiments. The column "Best solution" indicates the best solution reported back for that problem in the corresponding database, and the column "Best solution with ACS" contains the best solution found by our implementation of the ACS algorithm.

Table 1: Best solutions obtained using ACS algorithm.

Data base	Best solution (BS)	Best solution with ACS	Time (ms)	β	q_0
bays29.tsp	2020	2048	1062	5	0.6
berlin52.tsp	7542	7650	3801	5	0.6
st70.tsp	675	762	7632	5	0.6
rd100.tsp	7910	8670	18140	5	0.6
ch150.tsp	6942	6867	54001	5	0.6
kroA200.tsp	29368	32712	112982	5	0.6
tsp225.tsp	3919	4285	148351	3	0.9
a280.tsp	2579	2904	271179	3	0.9
lin318.tsp	42029	47328	398572	3	0.9
pcb442.tsp	50778	58265	973212	3	0.9
rat783.tsp	8806	12255	5081400	3	0.9

Secondly, we developed experiments to determine an adequate value for the ratio. We run the TS-ACS-TSP algorithm using several values for the ratio r = {0.2, 0.25, 0.3, 0.4, 0.5}. Tables 2 and 3 show the results averaged over 6 runs. The quality of the solution and the time cost were measured. Moreover, we made the comparison without using local search and with using the 2-opt procedure to improve the solution generated by ACS and TS-ACS algorithms.

The comparison with respect to the quality of the solution is presented in Table 2. The column "Best solution with ACS plus 2-opt" contains the best solution found by the ACS algorithm plus a local search developed with 2-opt procedure; the column "Best solution with TS-ACS" contains the best solution found by TS-ACS algorithm and the corresponding ratio with which it was found; the column "Best solution with TS-ACS plus 2-opt after first stage" contains the best solution found by TS-ACS algorithm plus a local search developed with 2-opt procedure applied to the solutions obtained after stage 1, and the column "Best solution with TS-ACS plus 2-opt after both stages" contains the best solution found by TS-ACS algorithm plus a local search developed with 2-opt procedure applied to the solutions obtained after the two stage. The time cost to obtain these solutions is presented in Table 3 (measured in seconds).

These experimental results can be summarized in the following way:

- In most of the cases TS-ACS algorithm reaches better solutions than ACS. In case ACS yields the better result, the difference to the results obtained by TS-ACS is not relevant.
- The time in which TS-ACS algorithm gets those results is between 40% and 50% of the time needed by the ACS algorithm, that is, the cost in time is reduced by half.
- A value of rate r between 0.2 and 0.35 produces the best results.
- Greater values for the ratio r decrease the time cost but also the quality (because the search space is not sufficiently covered).

Table 2. A comparisona between ACS and TS-ACS in TSP (quality solution)

Data base	Best solution	BS with ACS plus 2-opt	BS with TS-ACS	BS with TS-ACS plus 2-opt after first stage	BS with TS-ACS plus 2-opt after both stages
bays29.tsp	2020	2041	2058 (0.25)	2045 (r=0.2)	2026 (0.2)
berlin52.tsp	7542	7564	8034 (0.25)	7863 (0.25)	7542 (0.25)
st70.tsp	675	739	785 (0.2)	763 (0.25)	745 (0.25)
rd100.tsp	7910	8345	8668 (0.2)	8658 (0.3)	8159 (0.2)
ch150.tsp	6528	6673	6932 (0.3)	6896 (0.3)	6670 (0.2)
kroA200.tsp	29368	31598	32994 (0.2)	32889 (0.2)	31970 (0.25)
tsp225.tsp	3919	4115	4357 (0.25)	4204 (0.25)	4102 (0.2)
a280.tsp	2579	2827	3002 (0.2)	2944 (0.25)	2878 (0.25)
lin318.tsp	42029	46006	47370 (0.25)	47003 (0.3)	46086 (0.2)
pcb442.tsp	50778	56861	58420 (0.2)	58369 (0.3)	56790 (0.2)
rat783.tsp	8806	11917	12181 (r=0.3)	12202 (0.25)	11931(0.3)

Table 3. A comparison between ACS and TS-ACS in TSP (time cost)

Data base	BS with ACS plus 2-opt	BS with TS-ACS	BS with TS-ACS plus 2-opt after first stage	BS with TS-ACS plus 2-opt after both stages
bays29.tsp	1231	468	606	651
berlin52.tsp	3937	1672	1719	1734
st70.tsp	7851	3484	3609	3687
rd100.tsp	18721	10609	12492	14202
ch150.tsp	57803	20122	21984	31874
kroA200.tsp	123852	64588	66781	74873
tsp225.tsp	155151	74255	76120	78677
a280.tsp	295464	126823	135987	146898
lin318.tsp	430637	154963	172492	190220
pcb442.tsp	1040741	396250	425612	478066
rat783.tsp	5315734	2262624	2490555	2983343

6 Conclusion

We have presented an improvement of the Ant Colony Optimization based in a multilevel approach. It consists on splitting the search process developed by ants into two stages. The study was developed using the Ant Colony System algorithm. In this approach the values of some parameters (number of ants, quantity of cycles, number of cities included in each stage, etc.) are assigned a different value in each stage according to a ratio which indicates what proportion of the complete search corresponds to each stage.

We studied the performance using different ratio values in the Traveling Salesman Problem. The best results were obtained when this value is about 0.3.

This new approach to ACO produces an important reduction of the computation time cost, yet preserving the solution quality.

References

1. Dorigo, M. et al. The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics-Part B, vol. 26, no. 1, pp. 1-13. 1996.
2. Dorigo, M. and Gambardella, L.M.. Ant colonies for the traveling salesman problem. BioSystems no. 43, pp. 73-81, 1997.
3. Dorigo, M. et al. Ant algorithms for Discrete optimization. Artificial Life 5(2), pp. 137-172. 1999.
4. Dorigo, M. and Stützle, T.. ACO Algorithms for the Traveling Salesman Problem, 1999, Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications, John Wiley & Sons., EEUU.
5. Dorigo, M. and Stutzle, T.. Ant Colony Optimization. MIT Press. 2004.
6. Stefanowski, J.. An experimental evaluation of improving rule based classifiers with two approaches that change representations of learning examples. Engineering Applications of Artificial Intelligence 17, pp. 439-445. 2004.
7. The TSPLIB Symmetric Traveling Salesman Problem Instances, http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/index.html
8. Walshaw, C.. A multilevel approach to the traveling salesman problem. Operations Research 50(5), 862-877, 2002.
9. Bernard, S.T. and Simon, H.D.. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. Concurrency: Practice and Experience 6 (2), pp. 101-117. 1994.
10. Korosec, P. et al.. Solving the mesh-partitioning problem with an ant-colony algorithm. Parallel Computing 30, pp. 785-801. 2004.
11. Michalewicz, Z. and Fogel, D.B.. How to solve it: modern heuristics. Springer-Verlag. 2004.
12. Dorigo, M., Maniezzo, V., Colorni, A.: Positive feedback as a search strategy. Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy (1991).
13. Dorigo, M.: Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy (1992).
14. Walshaw, C. et al.. The multilevel paradigm: a generic Meta-heuristic for Combinatorial Optimization Problems?. Technical Report 00/IM/63, Univ. Greenwich, London SE10 9LS, UK. 2000.

GRASP Approach for the Glass Cutting Problem

Abhradeep Guha Thakurta[1], Ashwath Kumar.K[2], and Vijayan Immanuel[3]

[1] National Institute of Technology Karnataka, Surathkal,
Srinivasnagar-575025, India
abhra80@rediffmail.com
[2] National Institute of Technology Karnataka, Surathkal,
Srinivasnagar-575025, India
kashwathkumar@gmail.com
[3] National Institute of Technology Karnataka, Surathkal,
Srinivasnagar-575025, India
vijayan@nitk.ac.in

Abstract. This paper presents a greedy randomized adaptive search procedure(GRASP) for the Glass Cutting Problem. The Glass Cutting Problem deals with the placing of rectangular shapes of different sizes into a rectangular piece having a given width and given length. The length has to be minimized, thereby minimizing the wastage of glass which in turn maximizes the profit. This is an NP-hard problem.
Index Terms:GRASP;Glass Cutting Problem;Heuristics ...

1 Introduction

The industry of glass production [1] is composed of two mainly independent branches: one producing hollow glass (bottles, glasses, and other differently shaped products) and dealing with flat glass (for windows, mirrors, and so on). This paper deals with the flat glass production, where a large glass sheet is first produced, with an area of some tens of square meters, and individual glass pieces of various dimensions are then cut from this sheet.
A glass maker usually starts from a set of desired pieces that have been commissioned to him, and a set of sheets to cut from. Loss is created when a sheet can not be exactly covered by the desired pieces. Loss is constituted by small glass pieces, called scraps, that cant be utilized in any way.
Contemporary algorithms that deal with this problem mainly base on Heuristic approach or Adaptive approach. In this paper we present a GRASP (Greedy Adaptive Reactive Search Procedure) approach to the solution of the problem. We also claim that the settling time of this algorithm to optimal solution is lesser compared to the contemporary approaches.

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 93-102

Received 21/06/06
Accepted 03/10/06
Final version 11/10/06

2 Problem Description

2.1 Geometrical Description

The Glass Cutting Problem consists of finding the best way of placing a given set of n rectangular pieces i = 1, .., n of given heights and widths (hi,wi), without overlapping into a strip of width W and finite height H. We assume that the pieces have fixed orientation. An example is shown in Figure 1, in which several pieces have to be packed into a strip of width W = 40. The problem is NP-hard in the strong sense because the strongly NP-hard one-dimensional bin-packing problem can easily be transformed into the Two Dimensional Glass Cutting problem. Therefore most research effort has been focused on developing heuristic algorithms for this problem.

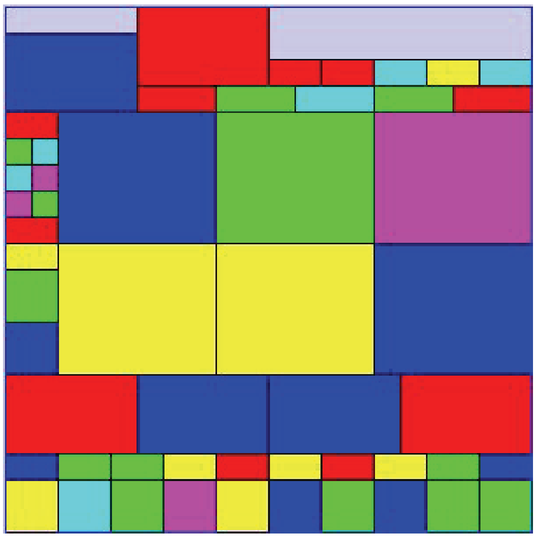


Fig. 1. A typical example of a solution to NP-Glass Cutting Problem

2.2 Glass Cutting Technology

In flat glass cutting [1] two separate steps are needed to extract a set of smaller pieces from a large sheet:

- First, a cutting pattern is scored on the glass sheet using a cutting wheel.
- Second, pieces are separated by using breakout bars, that hit the glass and create a mechanical stress in the neighborhood of one of the lines that were scored in the previous step. Internal glass tension breaks the glass in correspondence with the scored lines. Due to the critical breakout mechanism,

several constraints must be observed when scoring the cutting pattern, the most notable of which is that cuts must always be drawn from end to end of a glass piece.In practice, a cutting pattern is composed of a series of end-to-end vertical cuts (called X-cuts), breaking the sheet in a series of large stripes. Each stripe is then cut end-to-end in the horizontal direction (Y-cuts). This process may be repeated (W- and Z- cuts) in order to obtain the desired piece sizes.

- Third, a minimum distance between adjacent parallel cuts must be observed, in order for breakout bars to operate correctly.
- Fourth, the extreme borders of the sheet cannot be used due to irregularities: pieces must observe a minimum distance from the sheet boundary.
- Fifth, some sheets might be damaged in some internal point: no piece should be placed over these points.
- Sixth, a maximum distance between adjacent vertical parallel cuts must be observed, since the cutting machines have a vertical size smaller than the horizontal one and the pieces need to be rotated during breakout.

Because of all these constraints it is very difficult to make a Non-Adaptive Heuristic algorithm for the problem.

3 Constructive Algorithm

The algorithm adopted to solve the real life glass cutting problem is an iterative process which is called the constructive algorithm [2]. We follow an iterative process in which we combine two elements: a list P of pieces still to be packed, initially the complete list of pieces, and a list L of empty rectangles of infinite height in which a piece can be packed, initially containing only the strip S of width W. The rectangles will be denoted by the pair wi, li, where wi is the width and li is the level of its bottom side. At each step, a rectangle is chosen from L and, from the pieces in P fitting into it, a piece is chosen to be packed. That usually produces new rectangles going into L and the process goes on until all the pieces have been packed. The steps to be followed are as follows-

- Step 0: Initialization
  L = {S}, the set of empty rectangles.
  P = {p1, p2. . . pm}, the set of pieces still to be packed, ordered by non-increasing wi. Ties are broken by non-increasing hi.
  Qi is the number of pieces of type i to be packed.
  C =$\emptyset$; the set of pieces already packed.

- Step 1: Choosing the rectangle in L
  Take R* = {w*, l*}, where l* = min {li | {wi, li} ε L}.
  Ties are broken by minimum wi. If none of the remaining pieces can fit into R*:
  - L is updated by lifting the bottom side of R* to the minimal level li of its adjacent rectangles and merging R* with the rectangle(s) of minimum level

li.
- That leaves a closed rectangle below which is considered waste.
- Go back to select a new R*.
Otherwise, go to Step 2.

– Step 2: Choosing the piece to pack
Once a rectangle R* has been chosen, consider the pieces i of P fitting into R* in order to choose which one to pack. For each of these pieces i we compute mi= max{m ε Z+ | m*wi ≤ w*m ≤ Qi}, the number of copies of piece i fitting into the width of R*, and consider all possible blocks of 1, 2, ..,mi adjacent copies as alternatives to fill R*. The height of a block composed of k copies of piece i is hi and its width k * wi. All these alternatives form the set P*. For the sake of simplicity, all the elements of P* will be called pieces from this point on.
Several criteria have been considered to select the piece to pack:
1. Piece j with maximum width wj , breaking ties by non-increasing hj .
2. Max{wj + 0.1 * hj}
3. Max{wj + 0.5 * hj}
4. The piece j whose height hj is more similar to the difference between the bottom side of R* and the bottom sides of one of the adjacent rectangles.
The first three criteria are based on the width, trying to fill the bottom of rectangle R* as much as possible. Each one of them gives a different importance to the height of the pieces. The fourth criterion tries to maintain a profile of the current solution which is as smooth as possible, avoiding peaks and troughs. However, all these criteria may delay the packing of tall pieces which will cause large increases in the required height H at the end of the process. In order to avoid this situation, we complement these criteria by computing a double estimation of the effect of not packing the tallest remaining piece. When we select a piece, according to the chosen criterion, before packing it we check if it is the highest remaining piece fitting in R*. If that is the case, we place the piece into the strip. Otherwise, we do a double computation:

We put the tallest piece j into the strip and see if that piece increases the current required height H. If it does, we determine the empty area, E, defined by the new height and compare it with the area of the pieces still to be packed, M, plus an estimation of the unavoidable waste involved in the process: $U = (W * LB - A)/4$, where LB is a lower bound on the required length and A is the total area of the pieces. If E > M + U, the tallest piece j is selected for packing and mj copies of it are packed into R*. Otherwise, we compute the second estimation.
We put the selected piece into the strip and then we put the tallest piece j in one of the other rectangles of L or on top of the selected piece, wherever it produces the minimum required length. We repeat the argument of the first estimation, decide if the tallest piece j is preferred for packing and then

pack mj copies of it.

– Step 3: Rotation and optimizing
  Once the step 2 is completed, the rectangle can be positioned in two ways, one with its longer side as the base and other with its smaller side as the base. We have to compute the efficiency in both the cases, and adopt the posture giving more efficiency.
  The Heuristics used can be represented as follows
  1. After finding a rectangle from the strip database, the algorithm checks if the rectangle fits in the upright manner. If it does not fit in, the rectangle has to be flipped and then the optimal placement has to be carried out.
  2. The end result expected after this computation is the height must be minimized.

– Step 4: Choosing a position in R* to pack the piece
  Usually the piece to be packed does not completely fill rectangle R*. Therefore we have to decide its position inside R*. Obviously, the piece will be at the bottom of R*, but its position on the left or on the right hand side of R* has to e determined. Let us denote by Rl and Rr the rectangles of L adjacent to R* on its left and on its right.
  1. If Rl = ⊘ (Rr = ⊘), the piece goes on the left (right) hand side of R *. If Rl = Rr = ⊘, the piece is placed on the right hand side.
  2. Otherwise, we take into account the levels of Rl and Rr, ll and lr:

  If l* + hi = ll (l* + hi = lr), the piece is placed on the left (right) hand side.
  If ll = lr, the piece is placed as near as possible to one strip side.
  Otherwise, the piece is put adjacent to the rectangle with maximum level.

– Step 5: Updating the lists C = C ∪ {pi}
  Make Qi = Qi k, where k is the number of copies of the piece i forming the block chosen to be packed. If Qi = 0, remove piece i from the list P Add the new rectangles to L. Merge two rectangles if they are adjacent and have the same level.

4 GRASP Algorithm

GRASP has a strong intuitive appeal, a prominent empirical track record, and is trivial to efficiently implement on parallel processors [3]. GRASP is an iterative randomized sampling technique in which each iteration provides a solution to the problem at hand. The incumbent solution over all GRASP iterations is kept as the final result. There are two phases within each GRASP iteration: the first intelligently constructs an initial solution via an adaptive randomized greedy function; the the second applies a local search procedure to the constructed solution in hope of finding an improvement. In this case GRASP is an iterative

```
procedure grasp()
    InputInstance();
    for GRASP stopping criterion not satisfied →
        ConstructGreedyRandomizedSolution(Solution);
        LocalSearch(Solution);
        UpdateSolution(Solution,BestSolutionFound);
    rof;
    return(BestSolutionFound)
end grasp;
```

Fig. 2. Generic GRASP Pseudocode

procedure combining a constructive phase and an improvement phase. In the constructive phase a solution is built step by step, adding elements to a partial solution. In order to choose the element to be added, a greedy function is computed, which is dynamically adapted as the partial solution is built. However, the selection of the element is not deterministic but subjected to a randomization process. In that way, when we repeat the process, we can obtain different solutions. After each constructive phase, the improvement phase, usually consisting of a simple local search, tries to substitute some elements of the solution which are there as the result of the randomization, by some others, producing an overall better solution.

4.1 Constructive Phase

In our algorithm the constructive phase corresponds to the constructive algorithm described in Section 3, introducing randomization procedures when selecting the piece to pack. Let si be the score of piece i ε P* on the selection criterion we are using and smax=max{ si| i ε p* },and let δ be a parameter to be determined (δ lies between 0 and 1). We have considered four alternatives:

- 1. Select piece i at random in set
  C = { j | sj = (smin + δ (smax - smin)) }
  (C is commonly called a Restricted Set of Candidates).

- 2. Select piece i at random in set
  C = {j | sj = (smax) }

- 3. Select piece i at random from among the best
  100 (1 - δ)% of the pieces, irrespective of their score.

- 4. Select piece i from among the whole set P* but
  with probability proportional to its score si (pi = si/Σsj).

Using one of these randomization procedures on one of the selection criteria described above, a piece is chosen to be packed at each step of the constructive procedure. Nevertheless, the estimations of the effect of the tallest piece are also taken into account. These estimations are also randomized by using a parameter

γ. If $E > \gamma * (M + U)$, he tallest piece j is selected and a number of copies randomly chosen between 1 and mj is packed. At each iteration the parameter δ is randomly chosen in the interval (0.9, 1.6). A preliminary computational study showed that the term M + U tends to underestimate the total area which will be required by the remaining pieces and therefore the value of γ should oscillate above the value 1, though we also allow it to be slightly lower than 1.

Initialization:

$\mathcal{D} = \{0.1, 0.2, \ldots, 0.9\}$, set of possible values for δ

$S_{best} = \infty$; $S_{worst} = 0$

$n_{\delta^*} = 0$, number of iterations with δ^*, $\forall \delta^* \in \mathcal{D}$.

$Sum_{\delta^*} = 0$, sum of values of solutions obtained with δ^*.

$P(\delta = \delta^*) = p_{\delta^*} = 1/|\mathcal{D}|, \forall \delta^* \in \mathcal{D}$

$numIter = 0$

While $(numIter < maxIter)$

{

Choose δ^* from $\mathcal{D}$ with probability p_δ.

$n_{\delta^*} = n_{\delta^*} + 1$

$numIter = numIter + 1$

Apply Constructive Phase with δ^*, obtaining solution S

Apply Improvement Phase, obtaining solution S'

If $S' < S_{best}$ then $S_{best} = S'$.

If $S' > S_{worst}$ then $S_{worst} = S'$

$Sum_{\delta^*} = Sum_{\delta^*} + S'$

If $mod(numIter, 200) == 0$ (every 200 iterations):

$$eval_\delta = \left(\frac{S_{worst} - mean_\delta}{S_{worst} - S_{best}}\right)^\alpha \quad \forall \delta \in \mathcal{D}$$

$$p_\delta = \frac{eval_\delta}{\left(\sum_{\delta' \in \mathcal{D}} eval_{\delta'}\right)} \quad \forall \delta \in \mathcal{D}$$

}

Fig. 3. The GRASP Procedure for Glass Cutting Problem

4.2 Determining the parameter δ

A preliminary computational experience showed that no value of always produced the best results. Therefore, we considered several strategies basically consisting of changing the value of δ randomly or systematically along the iterations. These strategies were:

- 1. At each iteration, choose δ at random from the interval [0.4, 0.9]
- 2. At each iteration, choose δ at random from the interval [0.25, 0.75]
- 3. At each iteration δ takes one of these 5 values in turn: 0.5,0.6,0.7,0.8,0.9.
- 4. $\delta = 0.75$
- 5. Reactive GRASP
  In Reactive GRASP δ is initially taken at random from a set of discrete values, but after a certain number of iterations, the relative quality of the solutions obtained with each value of δ is taken into account and the probability of values consistently producing better solutions is increased.

5 Experimentation and Comparison with other Contemporary Algorithms

To experimentally evaluate the algorithms performance, in terms of result quality we selected some real benchmarks [5] and a comparison of the results with the set of commercial optimizers and our algorithm is tabulated.

5.1 Benchmarks

The characteristics of benchmarks used are reported in Figure 4. For each benchmark, the number of different piece sized and the total number of pieces are reported. The reported area is the sum of the areas of all the pieces. The bench-

Bench	Length	Breadt h	No. of pieces	Area
1	2	3	17	102
2	7	1	21	147
3	5	12	11	660
4	7	13	29	2639
5	5	13	21	1365
6	15	11	30	4950
7	14	9	22	2772
8	7	9	20	1260
9	11	11	12	1452
10	9	17	5	765
11	1	11	15	165

Fig. 4. Results of the Experimentation

marks selected are regarded as real cases since they represent actual data used by some glass makers. Further, they represent critical data, since they are some cases for which glass makers complained aboutthe optimization efficiency. The total glass area . The glass piece taken given initially is 135*135, the total area of all the pieces is found to be 16277 m2. The area wastage is found to be (3+12+15+14) m^2 , i.e 44 m^2 . The distribution given by the software given is as shown in figure 6. The study of the parameter δ in comparison with other contemporary algorithms give the following results.

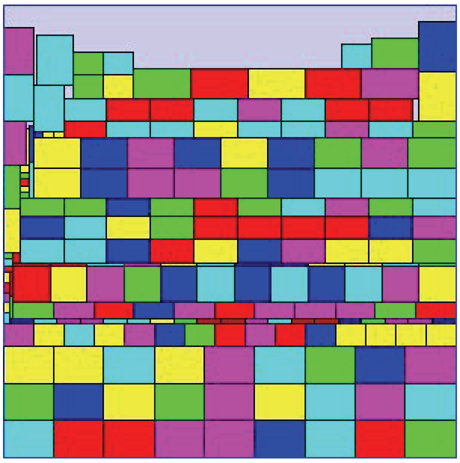


Fig. 5. Distribution Result

	Average deviation from bound		
	Literature	Martello et al. and Berkey et al.	Hopper
Random in [0.4, 0.9]	2.01%	2.15%	2.76%
Random in [0.25, 0.75]	2.03%	2.67%	3.22%
Deterministic from 0.5 to 0.9	1.95%	2.06%	2.83%
Fixed to 0.9	2.95%	2.01%	3.18%
Reactive Grasp	1.93%	2.03%	2.84%

Fig. 6. Delta Comparison

6 Conclusion

We have modified the conventional GRASP and used a Heuristic approach for the Glass Cutting Problem. The complete algorithm obtains good results on large sets of fairly randomized test cases. The algorithm is quite flexible and can be easily adapted to accommodate other conditions or constraints. Future work will involve the design of more efficient procedure to check the existence of a feasible packing layout, in order to take more advantages on the less number of feasibility tests performed.

References

1. The handbook of glass manufacturing, Haslee Publishing, NY USA.
2. BrazilJo.o P. Marques Silva and Karem A. Sakallah, GRASP A New Search Algorithm for Satisfiability.
3. Jakob Puchinger, Gunther R. Raidl,and Gabriele Koller, Solving a Real-World "Glass Cutting Problem
4. R. Alvarez-Valde , F. Parreno and J.M. Tamarit, Reactive GRASP for the Strip Packing Problem.
5. Zhang, D., Kang, Y., Deng, A. (2005) A New heuristic recursive algorithm for the Strip rectangular packing problem, Computers and Operations Research,in press.
6. Youzou Fukagaway, Yuji Shinanoz, Tomoyuki Wadaz, Mario Nakamori, Optimum Placement of a Wafer on a Rectangular Chip Grid. Proceedings of 6th World Congresses of Structural and Multidisciplinary Optimization Rio de Janeiro, 30 May - 03 June 2005.

A Combinated Model of Planning and Scheduling

Ma.de Guadalupe García-Hernández[1], Antonio Garrido[2],
Gerardo Canedo-Romero[1], Sergio Ledesma-Orozco[1] and Donato Hernández-Fusilier[1]

[1]FIMEE-Universidad de Guanajuato
Tampico 912, Col.Bellavista, C.P.36730
Salamanca, Guanajuato, México
Tel (464) 6480911 Ext.224
{garciag,canedo,selo,donato}@salamanca.ugto.mx

[2]DSIC-Universidad Politécnica de Valencia
Camino de Vera S/N, 46022, Valencia, España
Tel. (96) 3877000
agarridot@dsic.upv.es

Abstract. This paper presents an approach to deal with Planning and Scheduling features, which are now required in many real-worls problems, as a model that interleaves a planner and a scheduler, both playing a similar role in the problem solving process. Planning studies action selection for obtaining good quality plans. On the other hand, Scheduling studies the suitable resources application over any plan. Additionally, in the State-of-the-Art, there are two approaches to solve these problems: extended Planning and extended Scheduling. The first approach results in a high computational complexity, and the second approach uses a rigid plan template (embedded planner) avoiding the possibility of re-planning. Hence, this paper analyses the challenging points to work in a collaborative way such as: i) problem modelling, ii) structure definition, iii) functional behaviour and iv) resolution of planning/scheduling conflicts under a mixed initiative in a plan whose actions are very dependent on numeric features.Consequently, this model promises to be efficiently enough to obtain executable and consistent plans.

1 Introduction and Motivation

In Artificial Intelligence (IA), Planning and Scheduling (P&S) processes have been studied and applied separately. The first one studies the selection of actions to be performed to obtain desired goals. The second one considers the requirements for execute those actions.

However, the IA Planning Community has tried to solve real-world problems requiring handle parameters like duration tasks, execution costs, limited resource usage, optimization criteria, temporal constraints, persistent effects, etc., to solve a great variety of problems, such as logistic plan elaboration, crisis management, autonomous spatial vehicle navigation (planetary explorers and observation satellites),

© A. Gelbukh, C.A. Reyes-García. (Eds.) *Received 01/06/06*
Advances in Artificial Intelligence. *Accepted 03/10/06*
Research in Computing Science 26, 2006, pp. 103-112 *Final version 18/10/06*

optimization of touristy routes, airports traffic control, energy distribution systems, etc. [1, 2, 3].

Therefore, it is practically impossible to separate P&S features, because the selection of an action in a plan is usually conditioned to several temporal constraints, resource availability and criteria to be optimized [4]. Although it is clear that real-world problems require features of those IA processes; those processes have been hardly related to each other, mainly because P&S have been considered as disunited problems and, obviously, because this is the most immediate approach.

Consequently, P&S processes complement each other perfectly, which clearly motivates the challenge in designing flexible models to integrate maximal capabilities from both processes, especially if we consider that they use similar techniques (search in graphs, use of heuristics, management and reasoning of constraints, etc. [5].)

Question of what is the best design to interleave these processes still remains open [2,5,6]. In this way, we try to answer that important question proposing an integrated model for these IA study areas in [4]. Through a description of the architecture of this system, the information flow inside this model (where we analyze the way in which both processes cooperate and communicate mutually), its key information to decide which P&S technology is more appropriated to use for solve real-world problems, we contribute in this paper with a detailed analysis of main advantages and challenges obtained of using this system.

2 The Planning and Scheduling Problem

The P&S problem involves the execution of a sequence of actions, which must satisfy several constraints (temporal and resource availability), in order to achieve the problem goals, while trying to optimize a metric function.

Let us assume a problem inspired by a interplanetary explorer's scenario (this problem was introduced in the International Planning Competition 2002, http://ipc.icaps-conference.org). This problem requires that a collection of interplanetary explorers navigate to planet surface, finding samples, taking images, and communicating them back to a control base (lander). Explorer utilization must be coordinated for avoiding the bottleneck of the communication between the explorer and the lander, which must be performed only when the lander is visible from the explorer.

Furthermore, the explorer consumes energy in its various activities and it can only be recharged at locations that receive sunlight (recharging energy actions are very limited and need to be carefully planned to manage energy). There are plans that without energy recharge cannot obtain their goals. This is a clear example of a real-world problem that makes it necessary to include temporal constraints and resource availability into planning, as they modify the plan structure, because the plan generation is highly influenced by the scheduling constraints.

The requirements to model a P&S problem, i.e. in Rovers Domain, contain:

i) **Initial state** (IS) and problem goals (G), with the propositions that are true at the beginning of the problem and the facts that need to be achieved, respectively. i.e. the initial state may be: sunlight locations, the initial quantity of energy of the explorer; whereas the goals may be: communicated and received data of the samples (i.e., rock, soil and images of the explored planet).

ii) **Actions**, with their durations, conditions and effects for the study of different alternatives to achieve the goals, i.e. camera calibrating, image taking, etc.

iii) **Resources**, their availability to execute actions in the plan. They are implicit in action definition (conditions and/or effects). Actions can modify their propositional state (calibrated, available, etc,) or their numeric value (decrease energy in 15 units).

iv) **Metric function** that needs to be optimized as a multi-criteria function that allows finding solutions with the consideration of several weighted criteria that plays an important role in the plan, i.e. minimize the energy used and the plan makespan (total duration).

v) **Resources and/or time constraints**. They can model the requirements of the real-world problem. They can be of the obligatory type (strong constraints) or of the preference type (soft constraints), i.e. deadlines (they limit the duration time to obtain something) or time windows (sunlight is available from 5' to 20').

3 An Integrated Planning and Scheduling Model

Once we had established the dependency between P&S processes for solving real-world problems, we reviewed the State-of-the-Art about their integration. Furthermore, we proposed an integrated model for these IA processes in [4]. En this way, different approaches have been studied [2, 5, 8, 9]. From a planning point of view, a planner is extended to handle time and resources (this is known as the temporal planning approach) [8,12,13]. From a scheduling point of view, a scheduler is extended to insert a planning component, by providing a previously planned set of activities and, usually, the order in which these activities must be respected [5]. Those approaches have high computational complexity, making intractable some problems, and re-planning was not possible [14] over the rigid plans, respectively.

However, the architecture of this integrated approach provides a general and flexible model. It is a general model because each process is solved using existing techniques. Additionally, it is a flexible model because both P&S processes have a homogeneous role in problem solving. Then, this model creates a collaborative and strongly coupled integration structure for both IA processes.

3.1 Description of the Integrated Model

The architecture of this integrated model for P&S processes is depicted in Figure 1; it has two modules, one for each process.

The planning module acts as a plan repairer or re-planner [14] when is found a time/resource conflict (adding or deleting actions to make the plan executable) or as a simple planner when it needs to achieve some problem goals. The scheduling module checks and validates the feasibility of the plan, considering its constraints. The scheduler allocates actions satisfying resources/time constraints; it also informs and collaborates with the planner when it detects scheduling conflicts that require re-planning techniques [14]. However, this structure needs a data structure called Action Network (AN) [4], that is shared between both modules (see Figure 2).

AN follows the philosophy of Temporal Constraint Networks (TCN) [10] and Consumable Resource Networks (RCN) [11] to represent the plan [4], whose nodes represent time points where actions begin/end. Labeled edges represent: i) the usage of a resource (note that time is a resource too), ii) causal links between different actions, and iii) temporal constraints between nodes.

This architecture requires an input plan, with the corresponding domain and problem. This model could start from an empty plan, but it becomes such a complex task when there are many State-of-the-Art planners that generate plans efficiently. We can use a classical planner to tackle that task. Obviously, this type of planner gives executable plans, but if we provide a non-executable plan, there will not present conflicts, because one of the objectives of this model is precisely to repair a given plan and make it executable.

3.2 Used Technology in the Integrated Model

Planning finds executable plans without considering resource requirements; when planning conflicts are presented, the planning module inserts actions to support propositions or new sub-goals. Different planning technology is used, i.e. flaw repair [14] and decomposition [12]. Otherwise, the scheduling module requires to validate and guarantee the feasibility about time and resources of the AN. A feasible plan is a consistent plan. From the AI point of view, scheduling algorithms are used for checking and maintaining the plan arc-consistency, through MAC and AC-3 algorithms [20], with the involved resources and their constraints in the AN, this is a special case of the Problem of Satisfaction of Constraints (CSP) [10].

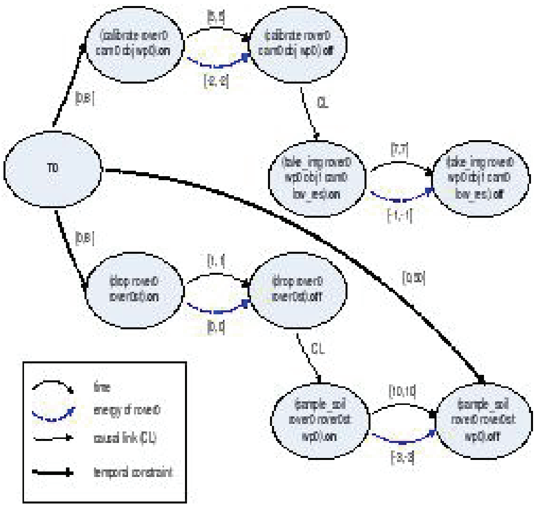


Figure 1. Integrated model of P&S processes

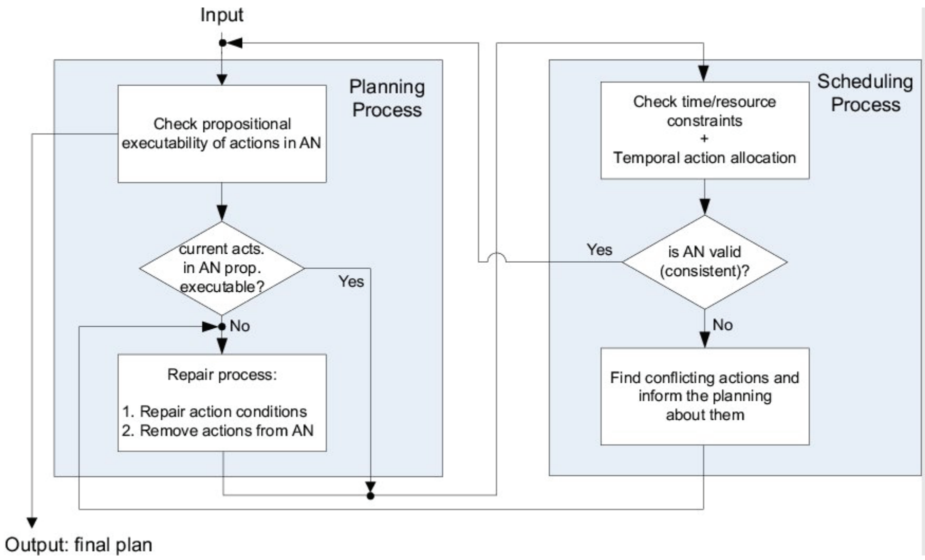


Figure 2. Action Network

3.3 Information Flow in the Integrated Model

Figure 3 and 4 depict the information flow inside the integrated model. The objective is to verify all actions in the AN to find a plan completely free of conflicts. This verification is performed from the planning and scheduling point of view (rev_PLN and rev_SCH) applying next algorithm,

```
1:  rev_PLN ← IS
2:  rev_SCH ← ∅
3:  while ∃ a ∈ AN ∧ a ∉ rev_SCH do
4:   {Planning part; reasoning on causal link}
5:   a ← earliest action that can be planned (executed) in AN
6:   rev_PLN ← rev_PLN ∪ {a}
7:   if number_of_conflicts (rev_PLN) > 0 then
8:    solve_PLN_conflicts (rev_PLN)
9:    {Scheduling part; reasoning on time and resources}
10:    for all ai ∈ rev_PLN ∧ ai ∉ rev_SCH do
11:     if ∃ a consistent allocation of ai in rev_SCH then
12:      rev_SCH ← rev_SCH ∪ {ai}
13:      else
14:     solve_SCH_conflicts (rev_PLN, heuristic parameters)
```

Figure 3. General scheme for integrating P&S

First, one action is selected (step 5), and then two main tasks are performed. One is to solve the planning conflicts that make the plan no executable, using solve_PLN_conflicts (step 8). This type of conflict is basically caused by unsupported action conditions or by mutually exclusive relations between plan actions [4]. When the planning conflicts have been solved, the other task is to eliminate scheduling conflicts. This type of conflict is caused by insufficient time and/or resources, using solve_SCH_conflicts (step 14). In both cases, the planner performs the same task (see Figure 3) with the technology before mentioned. During the process of solving the planning problem, the planner eventually calls the scheduler to obtain advice and make the most convenient decision [4].

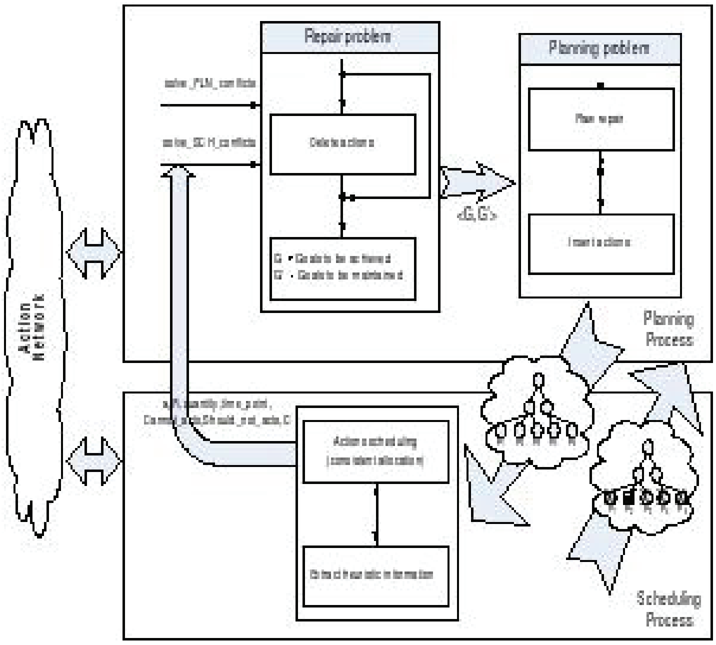


Figure 4. Information flow in the integrated model [4]

About the key points of the information flow in this model, there are two calls (step 8 and 14 of Figure 3) to the planning module (which is the executor):

i) solve_PLN_conflicts, where planner calls itself in order to fix planning conflicts in rev_PLN. It tries to solve the planning repairing problem [14].
ii) solve_SCH_conflicts implies solving a scheduling conflict through planning, because the scheduler detected an over-subscripted resource [19] and this one is a constraint violation. Then, the planner investigates other actions or the consistent allocation of resources from rev_PLN. If planner finds a solution to the conflict, then it allocates the solution using rev_SCH. Additionally, the scheduler provides important heuristic information to the planner on which actions cannot (and should not) be removed from the AN and other heuristic parameters. The first case is the set

of actions that only gives support to problem goals or they provide a unique way to attain the sub-goals or goals. The second case is the set of actions that helps to obtain sub-goals, however, they are not unique. Furthermore, the scheduler supplies a set of constraints in order to guide the planning search as much as possible, that planner must hold the constraints while it tries to solve the problem. Moreover, the planner shows the scheduler the planning decision points (where different plans are obtained). The last one decides which is the best plan.

It is obvious that the planner has a central role in this collaborative task, with the intelligent guidance given by the heuristic information provided by the scheduler.

4 Main Advantages of the Integrated Planning and Scheduling Model

Any system is evaluated for the advantages it provides, in this sense we have concluded that the proposed system is a very viable system because of the quality and quantity of its advantages, which will be described next,

a) **Flexibility**, both modules effectively work in a strongly coupled form, where one module takes into account the relevant information and valuable advice provided by the other module. Then, both processes knowingly communicate and collaborate each other to successfully build a partial plan.
b) **Modularity**, in every module there is only one process that maintains its remarkable work capacity at any time so that it can apply all its techniques when they are required.
c) **Generality**, the proposed architecture considers using this model to solve problems in any domain, because this model expressly applies existing techniques in the State-of-the-Art for each mentioned process.
d) **Conflict earlier detection**, both processes build and verify the execution and consistency of the plan step-by-step due to the mutual collaboration, in other words action-by-action, labeling the allocated actions once these were revised for each module. Consequently, the modules will opportunely detect the presence and type of a conflict; this could be for unsupported action conditions, mutually exclusive actions/propositions (planning problems) or insufficient resources (scheduling problem).
e) **Conflict immediate solution**, as soon as this model detects a conflict, it tries to instantly solve it, applying the P&S technology before mentioned.
f) **Available operative information** between both modules, through the use of two physical key points that interchange information, i.e., an advice when a conflict is detected or the system needs to find out which one is the best plan. One module cannot work without the appropriate guidance from the other one.
g) **Common heuristic information**, the scheduler extracts heuristic parameters from the planning process, and uses its own heuristic parameters in order to be able to choose the best plan.
h) **Common optimizing criteria**, each process has its own optimizing criteria. They

communicate each other and combine their criteria to make the final decision.

i) **Flexible interface**, this model may contain many sub-modules in order to finally achieve an effective communication.

j) **Architecture simplicity**, this is one of the best advantages of our system, because it freely allows inserting other required modules in order to obtain better plans without making significant changes.

5 Main Challenges of the Integrated Planning and Scheduling Model

Because there is not a common language between P&S processes, one of the main challenges is the design of an interface for mutual communication. From a planning perspective, there is a well-accepted language to define planning domains, which is actually called PDDL3.0 [7]. Another challenge of this system is that the mentioned language has some limitations for describing scheduling features, such as finite persistent of actions effects (the camera remains calibrated only during 20') and temporal constraints between actions (image taking must be done 6' before finishing the calibrated effect).

6 Conclusions and Related Work

From the 90's, combining P&S capabilities during resolution of real-world problems is a hot topic of research in IA [2,6,8,15,18]. There are two approaches to solve these problems: extended Planning and extended Scheduling. The first approach results in high computational complexity [8,12,13], and the second approach uses a rigid plan template (embedded planner) avoiding the possibility of re-planning [5]. Additionally, there have been some successful integration models, such as HSTS [16] and ASPEN [17], however, these systems only work in the domains they were designed. Thus, the search of a general and flexible design that integrates the P&S processes still remains open [2, 5, 6].

Otherwise, the architecture of our novel integrated approach proposed in [4] provides a general and flexible model. It is called a general model because each process uses existing technologies. On the other hand, it is flexible because both processes have a homogeneous role in problem solving. Then, this model creates a collaborative and strongly coupled integration structure of both IA processes.

Notoriously, this model has many advantages and they are considerably important, as they have been studied in this paper. We have concluded that the proposed system is a very viable system because of the quality and quantity of its advantages This model is based on the common idea of the dynamic interleaving of processes, where both are playing similar roles, not embedded one inside each other. Thus, the planner has the central role in this collaborative task (with the intelligent guidance given by the heuristic information provided by the scheduler).

One of their main challenges is the design of an interface for mutual processes understanding (note that they use different languages). Additionally, the planning language has some limitations for describing scheduling features, such as finite persistent of actions effects and temporal constraints between actions. Its modularity makes easy to face these challenges, because it successfully helps to insert required modules. Thus, its flexible architecture allows improvement. Because each process keeps its own maximal capabilities for early detection and immediate conflict solving, this integrated model guarantees obtaining the best solution plan. Consequently, this model promises to be efficiently enough to obtain executable and consistent plans.

References

1. Long, D., Fox, M., Progress in AI Planning Research and Applications, The European Online Magazine for the IT Professional, 3(5), pages 10-24, 2002
2. Smith, D., Frank, J., Jónsson, A., Bridging the gap between Planning and Scheduling, Knowledge Engineering Review, 15 (1), pages 47-83, 2000
3. Weld, D., Recent advances in AI Planning, AI, Magazine, 20(2), pages 93-123, USA, 1999.
4. Garrido, A., Garcia-Hernández, M.de G., Onaindía, E., “Towards an efficient Integration of Planning and Scheduling”, Proc. of the 24th Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2005), pages 58-65, London, England, December, 2005.
5. Smith, S., Zimmerman, T., “Planning tactics within scheduling problems”, Proc. ICAPS 2004, Workshop on Integrating Planning and Scheduling, pages 83-90, British Columbia, Canada, 2004.
6. Pecora, F., and Cesta, A., “The role of different solvers in Planning and Scheduling integration”, Proc. Congress of the Italian Association for Artificial Intelligence (AI*IA 2003), pages 362-374, Italy, 2003.
7. Gerevini, A. Long, D., ”Plan constraints and preferences in PDDL3”, Technical Report, University of Brescia, Italy, 2005.
8. Ghallab, M., Nau, D., Traverso, P., Automated Planning. Theory and Practice. Morgan Kaufmann, CA, USA, 2004
9. Myers, K., Smith, S., “Issues in the integration of planning and scheduling for enterprise control”, Proc. of the DARPA-JFACC Symposium, Sn Diego, CA, USA, 1999.
10. Dechter, R., Meiri, I. and Pearl, J., Temporal Constraint Networks, Artificial Intelligence, 49, pp. 61-95. USA, 1991.
11. Wallace, R.J., Freuder, E.C., “Supporting dispatchability in schedules with consumable resources”, Journal of Scheduling, 8(1), 7-23, 2005.
12. Chen, Y., Hsu, C.W. and Wah, B.W., “SGPlan: Subgoal Partitioning and Resolution in Planning”, Proc. International Conf. on Automated Planning and Scheduling (ICAPS-2004) - International Planning Competition, pages 30-32, British Columbia, Canada, 2004.
13. Gerevini, A., Saetti, and Serina, I., “Planning with numerical expressions in LPG”, ECAI-2004, pages. 667-671, Valencia, Spain, 2004.
14. van der Krogt, R., de Weerdt, M., “Plan repair as an extension of planning”, Proc. International Conference on Automated Planning and Scheduling (ICAPS-2005), pages 161-170, Monterey, CA, USA, 2005.
15. Bartak, R., “Integrating planning into production scheduling: a formal view”, in Proc. of ICAPS-2004, Workshop on Integrating Planning Into Scheduling, pages 1-8, British Columbia, Canada, 2004.
16. Muscettola N., “HSTS: Integrating planning and scheduling”, Zweben & Fox Editor, Intelligent Scheduling, pages 169-212, Morgan Kaufmann, CA, USA,1994.

17. Chien, S., Rabideau, G., Knight, R., Sherwood, R., Engelhardt, B., Mutz, D., Estlin, T., Smith, B., Fisher, F., Barrett, T., Stebbins, G., Tran, D., ASPEN-Automating Space Mission Operations using automated Planning and Scheduling, Proc. SpaceOps2000,USA, 2000
18. Do, M.B., Kambhampati, S.,"Planning as constraint satisfaction: Solving the planning graph by compiling into CSP", Artificial Intelligence, 132 (151-182), USA, 2001.
19. Benton, J., Do, M., Kambhampati, S., " Over-subscription Planning with Numeric Goals", 19th International Joint Conference on Artificial Intelligence, IJCAI 2005, Scotland, Aug 2005.
20. Buettner , D., "Applying Constraint Satsifaction Techniques to AI Planning Problems", Thesis, University of Nebraska, USA, Dec. 2003

Computer Vision

A Circle Detection Task For Performance Evaluation of Edge Detection Algorithms

Victor Ayala-Ramirez, Gerardo Trejo-Caballero, Raul E. Sanchez-Yanez, Carlos H. Garcia-Capulin and Francisco J. Montecillo-Puente

Universidad de Guanajuato FIMEE
Tampico 912, Col. Bellavista
36730 Salamanca, Mexico
e-mail: {*ayalav, trejocg, sanchezy, carlosg, frajavimopu*} @laviria.org

Abstract. In this paper, we propose to use a circle detection task to evaluate the performance of several edge detection algorithms. The purpose of using this kind of task is to measure the algorithm performance when edges are present in any possible orientation. We use a GA based circle detector to ensure sub-pixellic accuracy in the detection of the circle. Edge detection algorithms are applied to synthetic images with a single circle with known parameters on them. We analyze the edge operators response according to three criteria: spurious pixel detection, fitness value criterion and the center position criterion. These criteria are equivalent in some sense to Canny criteria for edge detection optimality. We present the tests and results performed with Canny, Roberts, Sobel, Marr-Hildreth, Prewitt edge detectors and on a custom developed NN-based edge detector.

1 Introduction

Robotic tasks like navigation, localization and reconstruction require computer vision methods [1]. Examples of these methods are object recognition [2], tracking objects and image understanding [3]. Some of these methods require the extraction of features from images like color blobs, texture images or edge information. This paper deals with methods where the main information is detected from edges. There are some widely used standard edge detectors like the ones by Canny, Roberts, Marr-Hildreth, Sobel and Prewitt [4].

In particular, to get a good performance in an object recognition task when using edges as information, we need to select an appropriate edge detector [5]. But, how to select it? We know that the goals of the object recognition task are the detection and localization of the object. In this order, the ideal detector is the one that optimizes the performance of the edge detector based on the following criteria: good detection, good localization and a unique response to a single edge [6]. Even if Canny proposes mathematical definitions for these criteria, we propose to use a specific task that includes implicitly all of these criteria. In this work, we present the evaluation of several edge detection algorithms based on a circle detection task. This task uses edges to detect circles and gives us the circle

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 115-123

Received 03/06/06
Accepted 02/10/06
Final version 10/10/06

center coordinates in the image with sub-pixellic accuracy. The position of the center varies if there are more or less edges and also varies according to where they are located. The comparison is made between the standard algorithms enumerated above and a neural network-based edge detector proposed in [7].

The paper is organized as follows. A description of the edge detectors under comparison is given in Section 2. Our proposed procedure to evaluate the performance of edge detectors algorithms is described in Section 3. A description of the tests and of the results are in Section 4. A discussion of the evaluation and the results is presented in Section 5.

2 Edge Detection Algorithms

Edges are used to represent objects in tracking tasks or object recognition tasks. Among the classical edge detectors, we have chosen the following ones to compare their performance for the circle detection task: Canny , Sobel, Prewitt, Roberts and the Marr-Hildreth edge detectors . Our choices were selected because we can test optimal implementations available in software packages like Matlab. We have also included in our tests a neural-network based edge detector [7]. The purpose of this inclusion is to compare performance when the edge detector is designed by using training samples taken under conditions where most of the classical edge detectors face difficulties to satisfy the criteria specified above.

2.1 Neural Network-based Edge Detector

We employed in this work the neural network-based edge detector that was previously introduced in [7]. For the sake of self-completeness, we describe here this edge detector. Neural networks are useful for systems where we have training examples of the required performance of a system [8], specially when there is an underlying complex model as in the edge detection problem [9] [10]. The neural network-based edge detector uses the intensities of the neighborhood pixels for each pixel of the image (a 3×3 pixel window) as inputs for the neural network to determine if the window center pixel has to be classified as an edge pixel. The architecture of the neural network is a multi-layer perceptron with 9 neurons in the input layer, 6 neurons in the hidden layer and only one neuron in the output layer. The neural network-based edge detector does not need tuning of any parameter because this step is performed during the neural network training phase. As a result of the training step, synaptic weights of the neural networks converge toward the optimal coefficients for the edge detection problem.

3 Circle Detection for Performance Evaluation

In this section we present our proposed procedure to evaluate the performance of edge detectors algorithms based on a circle detection task. Firstly, we present the circle detection algorithm and then we describe the procedure and criteria that give us the information used to select the best algorithm for such a task.

3.1 GA-based circle detector

The circle detection task consists in detecting a circle with known parameters on an input image that is the result of the application of one of the edge detection algorithms under test. The algorithm chosen for the circle detection is a method using genetic algorithms to detect circles [11]. This algorithm features a subpixellic accuracy in the determination of circle parameters. We consider this task specially suited for performance evaluation of edge detection algorithms because present edges have orientations in the full range from 0 to 2π radians.

The detected circle is represented by the triad (x_0, y_0, r) with (x_0, y_0) being the center of the circle in image coordinates and r is the radius of the circle also in pixels. Ground truth images used for input to edge detectors were constructed by software, that is, pixels were drawn for a circular region by considering if they satisfy the locus of a circle.

$$I(x,y) = \begin{cases} 1 & \text{if } \sqrt{(x-x_0)^2 + (y-y_0)^2} \leq R \\ 0 & \text{elsewhere} \end{cases}$$

The fitness function is a measure that relates the number of expected edge pixels and the pixels that actually are present in the edge image. A fitness value for the best circle is provided at each run of the GA-based circle detector. Good detection and good localization criteria are satisfied when the actual circle coordinates obtained using the edge image from the edge detection phase result are very close to the ground truth parameters of the synthetic images. Uniqueness of the response to edges is correlated to the fitness value of the detected circle. A greater value for the fitness function of the detected circle implies that edge points are unique in the detected circle.

The GA-based circle detector was evolved until there were no changes in fitness above a 0.01 % of the total number of edge points of the ideal circle.

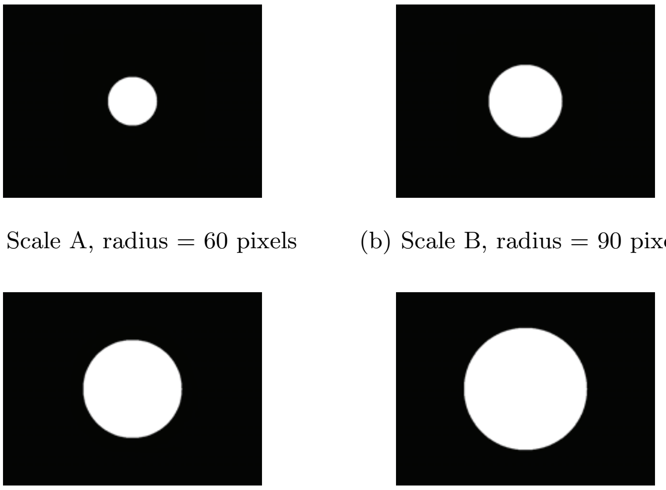
3.2 Procedure to Evaluate Edge Detection Algorithms Performance Using a Circle Detection Task

For the procedure, we need a synthetic image including a circle with known center coordinates and radius. In Fig. 1, we show four of the test images. We apply then the edge detector under test to each of these images and save the center coordinates of the best circle detected and also the corresponding fitness function value. These results are then compared to find the optimal edge detector for the circle detection task.

For all the tests, the parameters employed for the circle detector algorithm are show in Table 1.

4 Tests and Results

To evaluate performance of edge detection algorithms, we use three criteria. The first criterion concerns spurious edge pixels, that is, the detected edge pixels not



(a) Scale A, radius = 60 pixels (b) Scale B, radius = 90 pixels

(c) Scale C, radius = 120 pixels (d) Scale D, radius = 150 pixels

Fig. 1. Four test images, they present the same center, to know $(318.5, 238.5)$ but different radii (60, 90, 120 and 150 pixels respectively).

Table 1. Parameters of the circle detector

Generations	Population size	Crossover probability	Mutation probability	Elite individuals
50	1000	0.80	0.10	2

belonging to the detected circle. The second criterion is the fitness value of the best detected circle. The last criterion concerns accuracy of the position of the center of the detected circle.

In Fig. 2, we show the edges extracted by the edge detection algorithms under test. By using only visual inspection, it is difficult to find differences in their outputs.

4.1 Spurious Pixels Criterion

Ground truth data for the spurious pixels criterion is obtained by generating a circle using the parameters found by the GA circle detector. We define spurious pixels as the ones that are present in an edge image but they are not part of the edge ground truth image. Table 2 shows the total number of edge points in the image, the number of detected edges not belonging to the ground truth image and the ratio of these quantities. Values in this table are the average of 30 tests. The lower the spurious pixel ratio, the best we can rate the performance

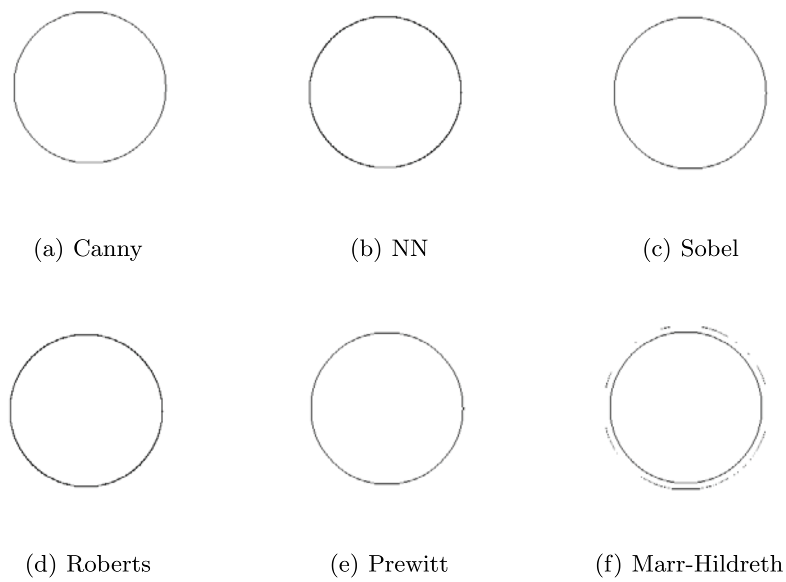


Fig. 2. Output edge images for the test image Scale B (radius = 90 pixels) obtained by the application of six edge detectors under test.

of the edge detector. In our tests, we have found that Canny is the best detector according to this criterion, followed in the second place by the NN detector.

Table 2. Spurious and detected pixels in the edge image.

Edge Detector	Detected Pixels	Spurious Pixels	Ratio
Sobel	282	20	0.070
Canny	284	6	0.021
NN	404	17	0.042
Roberts	404	18	0.044

4.2 Fitness Value Criterion

The fitness function value gives the fraction of pixels present both in the output edge image and in the ground truth image. It is used as a indicator of the correct response and the good localization of the edge points. This value will tend to 1 when the edge detector provides a good localized response.

We show the behavior of the fitness function value for all the detectors under test at scale B ($r = 90$ pixels) in Fig. 3. Best performance is obtained by the Roberts edge detector. The second place is for the NN based edge detector.

We also show the behavior of the fitness function value through the different scales for each edge detection algorithm under test. For all of them, we have run 30 tests at each scale. We can see the behavior for Canny, Roberts and the NN based edge detectors in Fig. 4 (a), (b) and (c) respectively. We can see that Roberts edge detector decreases its performance for greater scales (C and D). In the other hand, NN based detector shows a good performance at all scales. We conclude that for this criterion, the NN based edge detector is the best one. Canny detector has shown to be the worst edge detector for all scales That is explained by the fact that it detects a lower number of edges than the other methods.

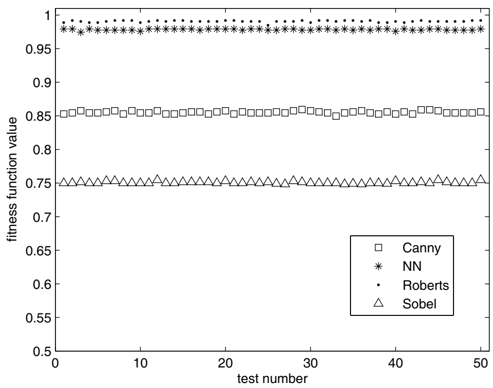
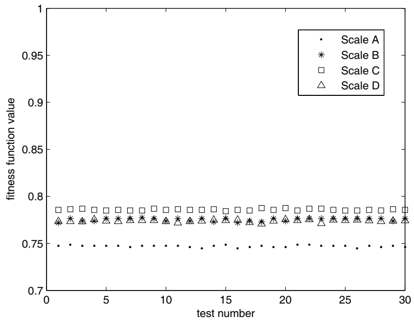


Fig. 3. The fitness function value of the detected circles shown for scale B.

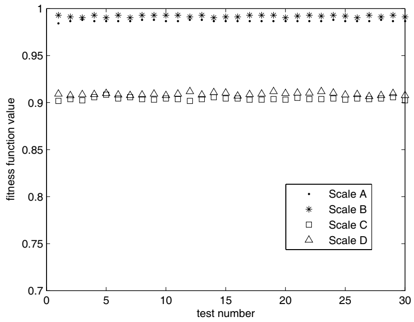
4.3 Circle Center Criterion

Another test we have performed to evaluate edge detectors was to compare the center position of the detecte circle in the output edge image against the known center of the circle. The ground truth center is $(318.5, 238.5)$ for test images at all scales.

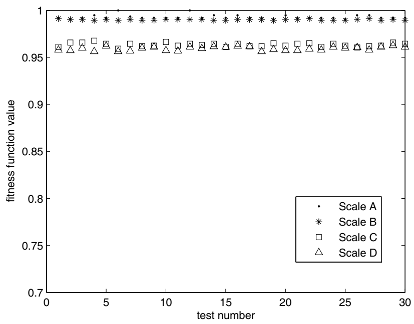
We show in Fig. 5 (a), (b) and (c) the position of the center of the detected circle for Canny, Roberts and NN edge detectors, respectively. We have run 30 tests at each scale. Graphs share the same axis limits to ease comparison. We can conclude that the NN edge detector presents a smaller bias from the ground truth circle center. According to this criterion, Canny detector comes in second place.



(a) Canny



(b) Roberts

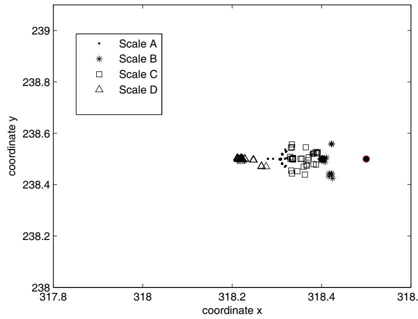


(c) NN

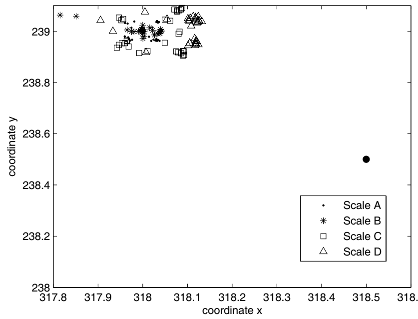
Fig. 4. Fitness function values found for the Canny, Roberts and NN algorithms at different scales.

5 Conclusions and Perspectives

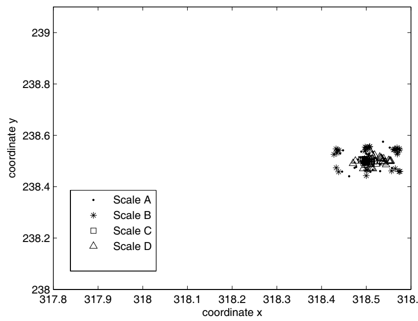
The results of our tests are only significant for a task like the one used here. Detecting circles and ellipses could be useful in some camera calibration procedures where an edge detection step is needed.



(a) Canny



(b) Roberts



(c) NN

Fig. 5. Bias of the center position for the circles detected on outputs from the Canny, Roberts and NN-based algorithms.

We have presented three criteria where a circle detector may be used to evaluate the performance of several edge detectors. There is no an only winner edge detector for all tests. Additionally we have compared a custom NN edge detector and the results show that its performance is good enough as the classical ones. Tests show the dependence on scale of the classical edge detector performance. However, the custom NN edge detector maintains its performance through the different scales used for testing.

As a future work, we plan to evaluate the performance on real-time systems that uses edges as main information, in order to face real problems like illumination and noise.

Acknowledgements

This work has been partially funded by the University of Guanajuato project "Funcionalidades visuales para la navegación topológica de robots móviles" and the Concyteg project "Herramientas Mecatrónicas para la Implementación de Entornos Virtuales".

References

1. Souza, G.D., Kak, A.C.: Vision for mobile robot navigation: a survey. IEEE Trans. on Pattern Analysis and Machine Intelligence **24** (2002) 237–267
2. Hunter, I., Soraghan, J.J., McDonagh, T.: Fully automatic left ventricular boundary extraction in echocardiographic images. In: Proc. Computers in Cardiology'95. (1995) 741–744
3. Tabb, M., Ahuja, N.: Multiscale image segmentation by integrated edge and region detection. IEEE Transactions On Image Processing **6** (1997) 642–655
4. Parker, J.R.: Algorithms for image processing and computer vision. John Wiley and Sons (1997)
5. Heath, M.D., Sarkar, S., Sanocki, T., W.Bowyer, K.: A robust visual method for assessing the relative performance of edge-detection algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence **19** (1997) 1338–1359
6. Canny, J.: A computational approach to edge detection. IEEE Trans. on Pattern Analysis and Machine Intelligence **8** (1986) 679–698
7. Trejo-Caballero, G., Ayala-Ramirez, V., Perez-Garcia, A., Sanchez-Yanez, R.E.: A neural network approach to edge detection in real images. In: Proc. RFMPIM 2005. (2005)
8. Lippmann, R.P.: An introduction to computing with neural nets. IEEE ASSP Magazine (1987) 4–25
9. Kulkarni, A.D.: Computer Vision and Fuzzy-Neural Systems. Prentice-Hall, Upper Saddle River, NJ (2001)
10. Perry, S.W., Wong, H.S., Guan, L.: Adaptive Image Processing: A Computational Intelligence Perspective. CRC PRESS, Boca Raton, Florida (2002)
11. Ayala-Ramirez, V., Garcia-Capulin, C.H., Perez-Garcia, A., Sanchez-Yanez, R.E.: Circle detection on images using genetic algorithms. Pattern Recognition Letters **27** (2006) 652–657

Agents

Application of Software Agents Technology to Create Knowledge

Juan Pablo Soto, Aurora Vizcaíno, Javier Portillo and Mario Piattini

Alarcos Research Group, University of Castilla-La Mancha
Paseo de la Universidad, 4-13071, Ciudad Real, Spain
jpsoto@proyectos.inf-cr.uclm.es
{aurora.vizcaino, mario.piattini}@uclm.es
javier.portillo@alu.uclm.es

Abstract. Agent technology has shown promise in the area of information management and is therefore a good candidate for knowledge management solutions. Intelligent agents have thus played diverse roles in knowledge management systems, as assistants, recommenders, user trackers, etc. Intelligent agents have also been used in frameworks or architectures whose development was intended to make the implementation of knowledge management systems easier. It is on this point that we are going to focus in this paper. Many of these architectures and frameworks were developed from a technical point of view, forgetting the knowledge tasks that they should support. Bearing this in mind, we designed a knowledge life cycle which can be taken into account when developing knowledge management systems. Moreover, in this paper we describe a generic multi-agent architecture (based on the knowledge cycle) to help developers to implement knowledge management systems.

1 Introduction

In the last decades, Knowledge Management (KM) has captured the attention of companies as being one of the most promising ways to reach success in this information era [9]. A shorter life-cycle of products, globalization, and strategic alliances between companies demand a deeper and more systematic organizational knowledge management [1]. One way to assess an organization's performance is to determine how well it manages its critical knowledge.

KM can be defined as a discipline that enables an organization to take advantage of its intellectual capital, in order to reuse it and learn from previous experience [16]. Skyrme [19] suggests that KM is the purposeful and systematic management of vital knowledge, along with its associated processes of creating, gathering, organizing, diffusing, using, and exploiting that knowledge. Due to the importance of knowledge management, tools which support some of the tasks related to KM have been developed. Different techniques have been used to implement these tools. One of them, which is proving to be quite useful, is that of intelligent agents [22]. Software agent technology can monitor and coordinate events or meetings and disseminate information [2]. Furthermore, agents are proactive; this means they act automatically when it

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 127-136

Received 06/07/06
Accepted 03/10/06
Final version 09/10/06

is necessary. The autonomous behaviour of the agents is critical to the goal of this research; reducing the amount of work that employees have to perform when using a KM system. Another important issue is that agents can learn from their own experience. Because of these advantages, different agent-based architectures have been proposed to support activities related to KM [6]. Some architectures have even been designed to help in the development of KM systems. However, most of them focus on a particular domain and can only be used under specific circumstances. What is more, they do not take into account the cycles of knowledge when proposing to use knowledge management in the system itself. For these reasons, in this paper we propose a generic multi-agent architecture for creating knowledge in organizations that use knowledge management techniques. The agents used in this architecture have been designed to support the different phases of the knowledge life cycle and foster the creation, storage and dissemination of knowledge.

This paper is structured as follows. In section two we describe different knowledge models as they have been proposed in the relevant literature. In addition, we put forward our own knowledge model. In section three, previous works based on multi-agent architectures are outlined. Section four presents our proposal. Finally, conclusions are explained in section five.

2 Knowledge Processes

One of the most widely discussed approaches is the SECI process [12], where the interaction between tacit and explicit knowledge emerges as a spiral (knowledge spiral) that includes four layers of knowledge conversion: *socialization* involves the sharing and exchanging of tacit knowledge between individuals, to create common mental models and abilities, most frequently through the medium of shared experience; *externalization* is the process of articulating tacit knowledge into comprehensive forms that can be understood by others; *combination* involves the conversion of explicit knowledge into a more complex set of explicit knowledge, and *internalization* is the process of adding explicit knowledge to tacit new knowledge by experimenting in various ways, through real life experience or simulations.

As there is no consensus in defining the stages that form a KM life cycle, Davenport and Prusak identify three tasks of knowledge management: (generation, codification/coordination and transfer) [4]. Wiig, in [24] observes five KM processes: (knowledge creation, knowledge storing, knowledge use, knowledge leverage, knowledge sharing). The similarities found in the stages of the models described previously helped us to define a process that is able to integrate the different proposals. For example, all the proposals consider stages which create, acquire, transfer, distribute, and disseminate knowledge. Another important aspect to take into account is that the capture or storing, along with the access or retrieval of knowledge, also form an important part of a KM life cycle model.

We have therefore chosen those stages that we believe should be supported by an architecture for developing KM systems (see Figure 1) and which are critical for this kind of systems. In section four we will explain how our multi-agent architecture follows this knowledge life cycle model.

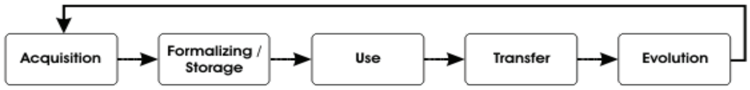


Fig. 1. Knowledge Management Life Cycle Model proposed

3 Related works

Traditional KM systems have received certain criticism, since they are often implanted in the company itself. Employees are consequently overloaded with extra work, as they have to introduce information into the KM systems and worry about updating this information. One proposal to avoid this extra burden was to add software agents to perform this task in place of the employees. Later, intelligent agent technology was also applied to other different activities, bringing several benefits to the KM process [15, 18].

The benefits of applying agent technology to knowledge management include distributed system architecture, easy interaction, resource management, reactivity to changes, interoperation between heterogeneous systems, and intelligent decision making. The set of KM tasks or applications in which an agent can assist is virtually unlimited, for example: to provide support to cooperative activities [23], to help people find information or experts which/who can assist them in their daily work [3] and to exploit an organizational memory [6].

Besides these works, we found others which focused on document classification [13], mailing list management [11], or data mining [7].

These and other existing systems were often developed without considering how knowledge flows and what stages may foster these flows. Because of this, they often support only one knowledge task, without taking into account that KM implies giving support to different process and activities. On the other hand, KM systems often focus on the technology, without taking into account fundamental problems that this kind of systems is likely to support.

4 A Multi-agent architecture to develop KM systems

As one of our goals is to develop a multi-agent architecture with KM itself, we shall start this section by describing each stage of our knowledge life cycle, explaining the type of agent that supports that stage.

4.1 Knowledge Model

- *Knowledge acquisition* is the stage responsible for making the organization knowledge visible. This stage considers the activities necessary to create organizational knowledge. Furthermore, the acquisition stage determines the organization skills for importing knowledge from external sources. The definition of the knowledge to be acquired can be assisted by classifying types of knowledge and knowledge sources [5]. To support this stage we propose to include in our architecture an agent called *Captor Agent*. The Captor Agent is responsible for collecting the information (data, models, experience, etc) from the different knowledge sources. It executes a proactive monitoring process, to identify the information and experiences generated during the interaction between the user and groupware tools (email, consulted web pages, chats, etc.). In order to accomplish this, the Captor Agent uses a knowledge ontology which defines the knowledge to be taken into account in a domain. Another useful ontology is the source ontology, which defines where each type of knowledge might be found [20]. Both of these are based on Rodriguez's ontologies for representing knowledge topics and knowledge sources [15].

The Captor Agent communicates with another agent (the *Constructor Agent*) which is in charge of creating knowledge. For example, when the Captor Agent acquires information that should be converted into knowledge it sends this information to the Constructor Agent.

One advantage of this architecture is that the Captor agent can work in any domain, since by changing these ontologies the Captor knows what key knowledge should be found and where it might be.

- *Knowledge formalizing/storing* is the stage that groups all the activities that focus on organizing, structuring, representing and codifying the knowledge, with the purpose of facilitating its use [4]. To help carry out these tasks we propose a *Constructor Agent*. This agent is in charge of giving an appropriate electronic format to the experiences obtained so that they can be stored in a knowledgebase to aid retrieval. Storing knowledge helps to reduce dependency on key employees, because at least some of their expert knowledge has been retained or made explicit. In addition, when knowledge is stored, it is made available to all employees, providing them with a reference as to how processes must be performed, and how they have been performed in the past. Moreover, the Constructor Agent compares the new information with old knowledge that has been stored previously and decides whether to delete it and add new knowledge or to combine both of them. In this way, the combination process of the SECI model is carried out, producing new knowledge as the end-result of the merging of explicit knowledge with explicit knowledge.
- *Knowledge use* is one of the main stages, since knowledge is useful when it is used and/or reused. The main enemy of knowledge reuse is ignorance. Employers often complain because employees do not consult knowledge sources and do not take advantage of the knowledge capital that the company has [21]. KM systems should offer the possibility of searching for information; they can even give recommendations or suggestions, with the goal of helping users to perform their tasks by reusing lessons that have already been learnt, and previous experiences. In our framework the agent in charge of this activity is the *Searcher Agent*, which searches in the knowledgebase for information that is needed. The result of the search will be sent to the *Interface Agent.* This will be explained in the next section. This agent could be implemented with different retrieval tech-

niques. Since this architecture is proposed at a high level, these aspects will not be dealt with in this paper. However, we would like to emphasize that the *Searcher Agent* fosters the internalization process of the SECI model, since the employees have the opportunity of acquiring new knowledge, by using the information that this agent suggests.

- *Knowledge transfer* is the stage in charge of transfering tacit and explicit knowledge. Tacit knowledge can be transfered if it has been previously stored in shared means, for example: repositories, organizational memories, databases, etc. The transfer stage can be carried out by using mechanisms to inform people about the new knowledge that has been added. For this stage we propose a *Disseminator Agent*, which must detect the group of people, or communities, who generate and use similar information: for example, in the software domain, the people who maintain the same product or those who use the same programming language. So this agent fosters the idea of a community of practice in which each person shares knowledge and learns, thanks to the knowledge of the other community members. Disseminated information may be of different types; it may be information linked to the company's philosophy, or specific information about a given process. Finally, the Disseminator agent needs to know exactly what kind of work each member of the organization is in charge of, together with the knowledge flows linked to their jobs. In order to do this, the Disseminator Agent contacts a new type of agent called the *Personal Agent,* which gives him information about the profiles of the user. Comparing this stage with the SECI model we can say the *Disseminator Agent* fosters the socialization process, since it puts people who demand similar knowledge in touch with each other. Once in contact, they can share their experience, thus increasing their tacit knowledge.
- *Knowledge Evolution*. This stage is responsible for monitoring the knowledge that evolves on a daily basis. To carry out this activity we propose a *Maintenance Agent*. The main purpose of this agent is to keep the knowledge stored in the knowledgebase updated. Information that is not often used is therefore considered by the Maintenance Agent as information that could possibly be eliminated.

4.2 Multi-agent Architecture

Once the knowledge model which the architecture is based on and the agents which support the different stages are defined, we can then define how the agents are organized into different agencies. Figure 2 shows that the architecture is formed of two agencies, made up of several agents.

The *Knowledge Agency* consists of the Constructor Agent, the Captor Agent, the Searcher Agent, the Disseminator Agent and the Maintenance Agent, previously described in Section 4.1. Therefore, the agency is in charge of giving support to the KM process.

The *User Agency* consists of the *Interface Agent* and the *Personal Agent*. The *Interface Agent* acts as an effective bridge between the user and the rest of the agents. Thus, if any agent wants to give a message to the user, the agent needs to send it to the Interface agent, which is the only one allowed to "talk" to the user. The Interface Agent also communicates with the *Personal Agent,* which obtains user profiles and information that are relevant to users' knowledge and which help to determine the expertise level and knowledge that each person has, or that a given person may need.

Another component used in this architecture is the *Shared Ontology*, this ontology is shared by all agents and provides a conceptualization of the knowledge domain. The Shared Ontology is used for consistent communication of the agencies.

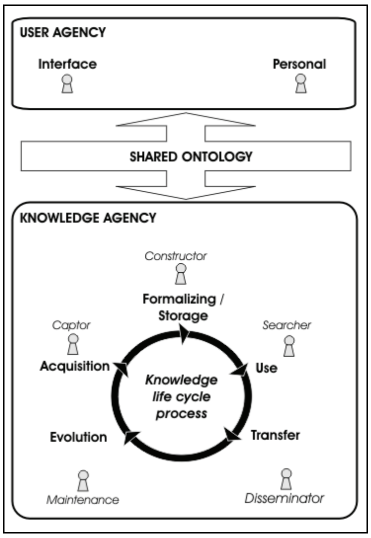


Fig. 2. Multi-agent Architecture for KM Systems

4.3 Agents Collaboration

As was mentioned before, the agents must collaborate with other agents. In order to show how they collaborate, we are going to describe a possible scenario that can take place in an organization.

Scenario

Let us imagine that a person is writing a mail and that the agents start to work, in order to check whether the mail contains information that should be stored in the data base (we are supposing that the employees know that the mails are being looked at in this way and that they agree to this). As Figure 3 shows, the Interface Agent captures each event that is triggered by the Employee. In this case the employee sends an email. Then the Interface Agent advises the Captor Agent that an event has been triggered. Afterwards, the Captor Agent determines the type of groupware tool used (email) to identify and obtain information topics about related tasks. In order to obtain information from the mail, a new agent can be added to the system (it would not form part of our architecture) but would be an agent that has been already developed to assist in this task. There are several implemented agents in existence for dealing with email [8]. Most of the current implementations are text classifiers [17] or keyword extractors [10]. The Captor Agent would study whether the information sent by the "email agent" should be transformed into knowledge. Finally, the Constructor

Agent receives the information which is structured in the form of, for instance, cases, for its later storage.

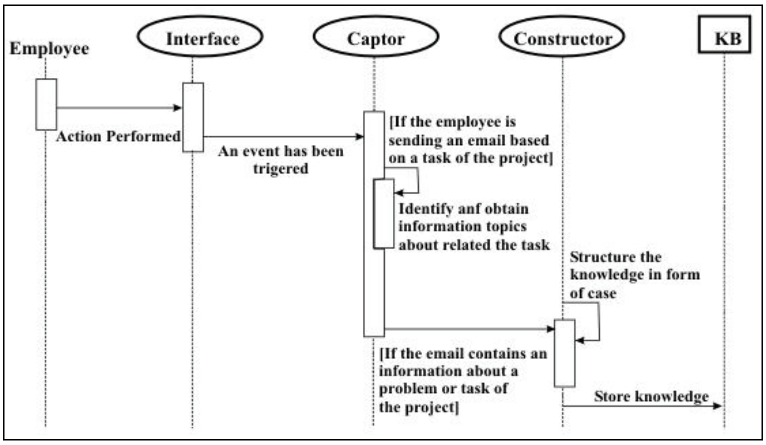


Fig. 3. Scenario of Agent Collaboration

4.4 Design of the Multi-agent Architecture

Methodology

This architecture has being designed by using INGENIAS [14] which provides meta-models to define MAS, and support tools to generate them. Using meta-models facilitates the development of the system enormously, since they are oriented to visual representations of concrete aspects of the system. Below, we are going to use the agent meta-model diagrams to describe the roles and tasks of each agent proposed in our architecture.

Figure 4 shows the goal, roles and tasks performed by the Captor Agent. The goal of this agent is to obtain information that should be stored. Its role is “filter” since it must decide what information should be transformed into knowledge; the purpose being to use it in future projects.

In the following lines, we describe each one of the task carried out by this agent.

- *CaptureInfo*: The agent must capture information.
- *IdentifyIS*: This task consists of identifying available information sources in the system.
- *SendToConstructor*: Once the suitability of storing the information has been analyzed, the Captor sends it to the Constructor Agent.

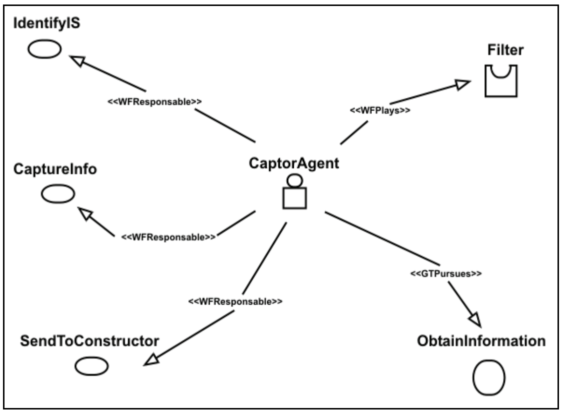


Fig. 4. Captor Agent diagram

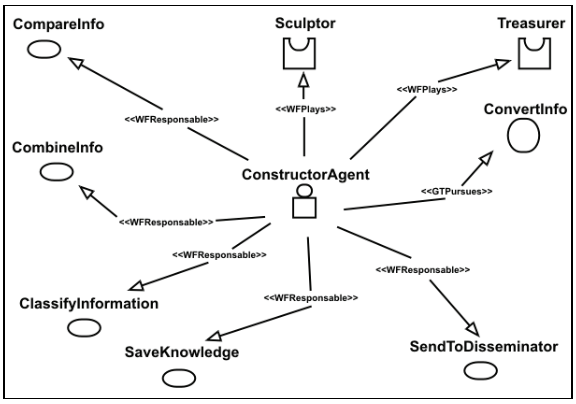


Fig. 5. Constructor Agent diagram

Figure 5 shows the role and tasks performed by the Constructor Agent whose roles are: sculptor and treasurer since it is in charge of giving an appropriate electronic format to the information (sculptor) and of storing it in the knowledgebase (treasurer). The tasks developed by Constructor Agent are:

- *CompareInfo*: The agent is in charge of comparing the new information with the previously stored knowledge.
- *CombineInfo*: The agent is also in charge of combining the new information with the previously stored knowledge. In this way, the combination process of the SECI model is carried out, producing new knowledge which comes about as a result of the merging of explicit knowledge with explicit knowledge.
- *ClassifyInformation*: Another task is to classify the information received by the Captor Agent (for instance: models, structures, files, diagrams, etc.).
- *SendToDisseminator*: This is a critical task which consists of sending knowledge to the Disseminator Agent.

- *SaveKnowledge*: One of the most important tasks is to store the new knowledge in the knowledgebase.

In this paper, only the Captor and Constructor Agent diagram are shown due to space constraints.

5 Conclusions

KM is a hot topic nowadays, as companies have realized that it offers them a competitive advantage. Because of this, various knowledge management systems have been developed, to support specific knowledge tasks. Moreover, different frameworks have been proposed, in the quest to make the development of certain knowledge management systems easier. Surprisingly, however, most of these frameworks have not taken into account how knowledge is created and how it flows through companies. In this paper we have described a knowledge life cycle to be taken into account when KM systems are being developed. Moreover, we have described a multi-agent architecture to support the different stages of that cycle. The main feature of the architecture is that it is so generic that it can be used to develop KM systems for almost any domain. Furthermore, it has been designed according to the INGENIAS methodology whose meta-models help future developers to understand how the different agents work.

Acknowledgement. This work is partially supported by ENIGMAS (PIB-05-058), and MECENAS (PBI06-0024) project, Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, both in Spain.

References

1. Ale, M., Chiotti, O., Galli, M.R.: Agent-Supported Ontology-Based Knowledge Management Systems. Proceedings of ASIS (Simposium Argentino de Sistemas de Información) and JAIIO (33º Jornadas Argentinas de Informática e Investigación Operativa) (2004).
2. Balasubramanian, S., Brennan, R., Norrie, D.: An Architecture for Metamorphic Control of Holonic Manufacturing Systems. Computers in Industry, Vol.46 (2001) 13-31.
3. Crowder, R., Hughes, G., Hall, W.: An Agent Based Approach to Finding Expertise in the Engineering Design Environment. Proceedings of the 14th International Conference on Engineering Design, (2003).
4. Davenport, T.H., Prusak, L.: Working Knowledge: How Organizations Manage What They Know. Project Management Institute. Harvard Business School Press, Boston, Massachusetts (1997).
5. Dickinson, A.: WP3 - KM Framework. Enhancing Knowledge Management in Enterprises (ENKE) IST Project, IST-2000-29482, http://www.ist-enke.com (2000).
6. Gandon, F.: Multi-Agent System to Support Exploiting an XML-based Corporate Memory. Proceedings PAKM, Basel, Switzerland (2000).
7. Giannella, C., Bhargava, R., Kargupta, H.: Multi-agent Systems and Distributed Data Mining. Cooperative Information Agents VIII: 8th International Workshop, CIA 2004, Vol. 3191/2004. Springer-Verlag, Erfurt, Germany, (2004) 1-15.

8. Maes, P.: Agents that reduce work and information overload. Communications of the ACM, Vol.37 (1994) 31-40.
9. Malone, D.: Knowledge Management: A Model for Organizational Learning. International Journal of Accounting Information Systems, Vol.3 (2002) 111-123.
10. Mock, K.: An Experimental Framework for Email Categorization and Management. 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, USA (2001).
11. Moreale, E., Watt, S.: An Agent-Based Approach to Mailing List Knowledge Management. Proceedings of the Agent-Mediated Knowledge Management (AMKM) (2003) 118-129.
12. Nonaka, I., Takeuchi, H.: The Knowledge Creation Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press (1995) 304.
13. Novak, J., Wurst, M., Fleischmann, Strauss, W.: Discovering, Visualizing and Sharing Knowledge through Personalized Learning Knowledge Maps. Proceedings of the Agents-Mediated Knowledge Management (AMKM) (2003) 213-228.
14. Pavón, J., Gómez-Sanz, J.: Agent Oriented Software Engineering with INGENIAS. Multi-Agent Systems and Applications II, LNAI 2691. Springer-Verlag (2003) 394-403.
15. Rodríguez, O., Martinez-García, Ana., Vizcaíno, A., Favela, J., Piattini, M.: Identifying Knowledge Flows in Communities of Practice. In Coakes, Elayne and Steve Cleark: Encyclopedia of Communities of Practice in Information and Knowledge Management. IDEA Group (2005) 210-217.
16. Rus, I., Lindvall, M.: Knowledge Management in Software Engineering. IEEE Software, Vol. 19 (2002) 26-38.
17. Segal, R.B., Kephart, J.O.: MailCat: An Intelligent Assistant for Organizing E-Mail. Proceedings of the 3rd International Conference on Autonomous Agents, Seattle, WA, USA (1999).
18. Silverman, B.G., Bedewi, N., Morales, A.: Intelligent Agents in Software Reuse Repositories. CIKM Workshop on Intelligent Information Agents, Baltimore, USA (1995).
19. Skyrme, D.: Knowledge Management: Making Sense of an Oxymoron. Management Insight, Vol.22 (2003).
20. Soto, J.P., Vizcaíno, A., Piattini, M., Rodríguez, O.: A Multi-Agent Architecture to Develop Knowledge Management Systems. 1er. Congreso Internacional de Ciencias Computacionales (Cicomp'06), Ensenada México (2006).
21. Szulanski, G.: Intra-Firm Transfer of Best Practices Project. American Productivity and Quality Centre, Houston Texas (1994) 2-19.
22. van Elst, L., Dignum, V., Abecker, A.: Agent-Mediated Knowledge Management. International Simposium AMKM 2003. Springer, Stanford, CA, USA (2003) 1-30.
23. Wang, A., Reidar C., Chunnian, L.: A Multi-Agent Architecture for Cooperative Software Engineering. Proceedings of the Eleventh International Conference on Software Engineering and Knowledge Engineering (SEKE'99), Germany (1999) 162-169.
24. Wiig, K.M.: Knowledge Management: Where Did it Come from and Where Will it Go? : Expert Systems with Applications, Vol. 13 (1997) 1-14

FCPx: A Tool for Evaluating Teams' Performance in RoboCupRescue Simulation League

João Certo[1], Nuno Cordeiro[1], Francisco Reinaldo[1,2], Luís Paulo Reis[1] and Nuno Lau[3]

[1] LIACC, Faculty of Engineering, University of Porto, Portugal
[2] GIC, Department of Computer Science, UnilesteMG, Brazil
[3] IEETA-DET, University of Aveiro, Portugal
reifeup,lpreis,joao.certo{@fe.up.pt}, nuno-cordeiro@hotmail.com, lau@det.ua.pt

Abstract. In the RoboCupRescue environment, one of the biggest problems in choosing the best team strategy is the inability to properly compare the advantages and limitations each strategy provides. This paper presents a new tool for the performance evaluation of Rescue Simulation agents, overcoming the limitations imposed by using solely the overall score. By extracting and processing a wide set of parameters from the simulation log files, and making a comparative analysis in spreadsheets, our tool facilitates the evaluation of the different agent types. By integrating our software into a widely used viewer, we expect further development to be continued by the Rescue community, allowing it to become a standard choice for the evaluation and analysis of new team strategies and approaches.

1 Introduction

RoboCup Rescue Simulation is an international joint project [1] that promotes research in distributed artificial intelligence and intelligent robotics. Through the use of a comprehensive urban disaster simulator, heterogeneous agents' teams try to minimize damage to both people and property. Every year, a RoboCup international competition is organized, where, in a competitive but constructive environment, researchers from all over the world can test their agents against other teams. Comparing approaches and exchanging ideas, progress is made at an amazing rate due to the open source nature of the project. After each competition, the source code for every team is released, so that work may be done on top of the best ideas and implementations.

The simulations take place in the three days following a major calamity in an urban scenario. Burning buildings, people trapped under debris, and blocked roads, are just some of the challenges rescue teams must overcome, coordinating as many as up to forty agents of six different types. Fire Brigade, Police Force, and Ambulance Team agents act in a coordinated manner, assisted by their

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 137-148

Received 02/06/06
Accepted 03/10/06
Final version 20/10/06

respective Control Centres, attempting to eliminate fires and save lives. Centre agents coordinate platoons in their collaborative tasks, by trading messages of world state and defining strategies. Still, platoons have the autonomy to learn, and to decide on the best course of action, on unexpected states. Police agents have the responsibility to clear roads, blocked by debris, allowing other agents to circulate and reach their targets; Ambulance Team agents search for buried citizens, rescuing and transporting them to refuges; and Fire Brigade agents extinguish buildings on fire. All tasks performed by mobile agents are centred on saving lives and reducing city damage [2][3].

The team success is measured as follows:

1. Let *Sint* be the total amount of health points of all agents at start and *Bint* be total undamaged area at start
2. At any given time step, the following values are obtained: the number of living agents (P), the remaining total health points of all agents (S) and the total undamaged area of buildings (B).
3. Finally, the simulation score V is calculated using Eq. 1 [4].

$$V = \left(P + \frac{S}{Sint}\right) * \sqrt{\frac{B}{Bint}} \quad [1]$$

Every year, new strategies and algorithms allow successful Rescue operations on harder scenarios. A major obstacle in the development of Rescue Agent Teams is the inability to objectively measure the performance of an agent. This proves to be a big obstacle to the process of implementing minor improvements and tweaking existent parameters. The overall score, although being a single numeric value, can be affected by so many variables and starting conditions, that it becomes very difficult to describe the improvement or deterioration of the agents' performance, in each small task. Furthermore, due to the stochastic nature of the simulator, it is nearly impossible to realize if minor changes in the code improve the overall behaviour of the agents, without running several simulations using different maps. It becomes therefore essential to extract further data from the simulation. Existing tools provide a limited amount of data, while some are currently outdated and, consequently, not working. Comparison can only be presented visually, and on a limited set of parameters, with no further options to record and thoroughly analyze the observed data. It is also, usually, limited to two teams at a time, restricting the user's ability to analyze a certain aspect in a wide variety of teams.

There are two possible paths to the extraction of simulation data, which are not, in any way, mutually exclusive. The required data can be logged from the simulator or from the agents and, although some figures may be common, the different nature of both kinds of data gives each one its own perspective on the simulation. Information extracted from the agents is extremely flexible, depending, both in amount and in format, on the implementation of the code for each individual team. Due to the lack of standardization in the Rescue community it is extremely difficult to use this data to compare different teams, as terminology and information extracted can be extremely diverse. Also, since the code is developed not only by different people, with different strategies and tactics in mind, but also in different development environments and different programming languages, the challenge of creating and maintaining a tool capable of mining the relevant data in every different team, and to present it in a comparable form, is a gigantic task - one that would require a large effort from every team involved. Still, the figures obtained from the agents are of extreme importance to complement the ones extracted from the simulator, providing the programmers with valuable insight into the behavioural process of each individual agent. It is, nonetheless, a path impossible to take if one of the main objectives is inter-team comparison, due to the above mentioned lack of standardization between teams. The chosen path is, therefore, to collect data from the simulator, benefiting from the fact that a single, common, program is used as the staging grounds for the agents' activities. This enables the gathering of information in a common form to all teams, allowing objective comparisons in any of the obtained parameters.

In order to visualize a simulation in a graphic way, a viewer tool must be used. This tool is responsible for converting the simulated information into an easily perceptible form, usually showing a plant of the city, along with the position and status of every agent during the simulation. On each run, a log containing information pertaining to the entire simulation is generated for future use. An important feature, present on most viewers, is their ability to replay simulations using those log files, allowing a detailed analysis to be performed.

In this paper we present FCPx [5], a new tool for the performance evaluation of RoboCupRescue Simulation agents. By extracting a wide set of parameters from the simulation logs and allowing a comparative analysis in spreadsheets - where the data is displayed in an organized structure - our tool facilitates the evaluation of the different agent types in various aspects. The simulation figures saved in our files are either directly acquired from the log files, or inferred from that information, using appropriate algorithms. This way, researchers are provided with enough data to allow for an easier, faster and informed decision concerning the best strategies in a number of different situations. Furthermore, this plethora of information can be used as a benchmarking environment, allowing the assessment and comparison of Rescue teams in a myriad of different factors, situations and time intervals. The data extracted may also be used to feed other software applications, from simple performance evaluators to advanced learning modules [3].

In the development of this tool two main objectives were kept in mind: flexibility and expandability. These factors are utterly important due to the constant evolution of the simulator package. By integrating our software into a widely used viewer, we expect further development to be continued by the Rescue community, allowing it to become the standard choice for the evaluation and analysis of new team strategies and approaches.

The rest of this paper is organized as follows. The next section discusses related work. In Section 3 we introduce our tool and describe it in detail. Sections 4 and 5 provide some results by presenting analysis and comparison of Rescue teams using FCPx. Finally, in Section 6 we present our conclusions and future work.

2 Related Work

Log viewers are tools used to view the evolution and result of rescue simulations. Some viewers have been written so far, enabling different viewing perspectives of the simulation, but all of them lack the functionality of a good debugging viewer [6]. Some teams have performed work in this area, but the need for a more comprehensive tool was only made more visible [6]. Also, there have been some recurring requests for good team analyzers and log viewers for some time now - both in the official RoboCupRescue mailing list as well as in other Rescue documentation.

The Freiburg team, amongst others, has developed its own viewer, releasing it to the rescue community. Freiburg's 3D viewer [7] is one of the most used by the community, second only to Morimoto's 2D viewer [8] [9], which is included in the official simulator package. Freiburg's 3D viewer tool has several built-in features. A statistic view option is available to measure and evaluate a team's performance in various aspects, such as the team's efficiency in exploring, rescuing civilians or extinguishing fires. This enables an easy comparison of two teams, since it is possible to see their actual differences, by simultaneously parsing two log files[4]. Also possible is the post-processing of simulation log files, allowing the continuous, strenuous, evaluation of time-dependant parameters.

Team Damas[5] also made some work in this area, basing their effort on Morimoto's 2D viewer and adding code to enhance the statistic interface, making it especially useful when used in conjunction with their own "autorun" program. When both these tools were used together, several simulations would be run consecutively. At the end of each, the viewer would save the relevant statistics into a file, killing itself subsequently, thus enabling a new simulation to begin. This tool provided a limited set of statistics. It is now outdated, no longer supporting recent versions of simulator log files.

3 FCP eXtended Freiburg 3D viewer

3.1 Overview

The FCPortugal eXtended Freiburg 3D viewer (FCPx) tool was developed over the simple logging features on Freiburg's 3D viewer, extending its ability to analyze and compare different teams. A critical aspect in evaluating and comparing

[4] `http://kaspar.informatik.uni-freiburg.de/~rescue3D/index.html`

[5] `http://www.damas.ift.ulaval.ca/projets/RobocupRescue/Tools.php`

performance is having the information stored in a flexible, adaptable, form. If the data can be easily imported into a spreadsheet, it becomes easy to build, and analyze, tables and comparative charts. From herein, the versatility of this solution gives the programmer endless possibilities. Still, the FCPx tool provides some customizable spreadsheets to reach a few elaborate outcomes, both as a proof of concept, and as a means to help users analyze their data in a predefined way.

Our initial effort went into acquiring as much data as possible from the simulation. A large amount of parameters is directly obtained from the log file, while some other are calculated using those previously gathered. Some values are kept in both absolute and percentage form, so that users may have a relative idea of the action developments, simplifying the comparison between maps and teams. Some of the numbers are, therefore, redundant. However, the effortless comparison achieved justifies that. Also worth mentioning is the focus on expandability. It is possible to add a new parameter with little effort, besides the one required for its characterization.

The information is then saved into a file with the same base name as the log file - therefore using the "date-time-team-map" filename, but with a different extension. As was previously mentioned, versatility was the main factor when considering file formats. Keeping that goal in mind, and considering essential the ability to easily export the data into a spreadsheet, CSV (Comma Separated Values) was the chosen file type. Some other advantages arise from this choice, like the potential to reduce the amount of hard disk space occupied by log files. To properly evaluate a strategy there's a need to run dozens of simulations comparing the attained results with data from previous approaches. That generates dozens of logs, which take around 5 to 15 megabytes of disk space, each. Also, in order to examine a team's evolution through time, older logs are stocked for reference. If you add to that the need of comparison to other teams' results, therefore keeping some of their logs in store, the amount of disk space used can quickly amount to dozens, or even hundreds, of gigabytes. A CSV statistical file with our chosen parameters takes around 15 to 50 kilobytes, which is less than one hundredth of the disk space used by each log. Keeping only the statistical files, and deleting the logs, allows for an extremely larger amount of data to be stored and used, improving the ability to evaluate different teams and strategies. Still, with the deletion of the log file, the ability to reproduce the simulation is lost.

From the CSV files, the data can be effortlessly imported into a spreadsheet. This creates a space where, either through tables or charts, evolutions can be perceived, behaviors studied and conclusion drawn. By this means, the user acquires the extra flexibility necessary for the analysis, and evaluation, of minor changes in the code, and their direct, and indirect, repercussions on the agents' performance.

3.2 Operation

The open-source FCPx viewer can be fully downloaded at its official website[6]. The next section presents an experiment which was created by using FCPx Tool.

When the FCPx tool is executed with the WRITE-STATS flag set to FALSE, the program behaves in the same way as the unmodified Freiburg's 3D viewer. The only exception is in the computation of the number of civilians that die due to fire, or buried under debris, where a small bug was corrected. In contrast, when the mentioned flag is set to TRUE, CSV files are created with the additional statistical data. An extra mode was created for the fast extraction of data from previously ran simulations, allowing for extra flexibility.

But, in WRITE-STATS mode, CSV files are created with the additional statistical data, using the same base name of the log file which allows the correct correlation of the statistical files. Besides that, in "WRITE-STATS, FAST" mode, no windows are opened and, after the parsing, the viewer kills itself which allows an outside program to run the parsing, and the final score is displayed which allows fast easy access to this value. This solves another issue noted during the RoboCup 2005 competition, in which the referee would run the 2D viewer with the log file - sometimes more than once - in order to obtain the score of the competing team. The "WRITE-STATS, FAST" method takes less than 5 seconds on modern machines.

4 Evaluating different types of Agents

FCPx extracts data to a flexible organized file, presenting several opportunities for the analysis of data. As a proof of concept and as a means to show the usefulness of the tool, a comprehensive spreadsheets was created, allowing the detailed analysis of rescue teams. In this section, we will present an analysis of the FCPortugal's [3] FireBrigades' performance in the KobeHard map presented during RoboCup 2005 in Osaka. In this simulation the team results were bellow expectations, which warrants a closer study.

The Fire Brigade relevant data obtained is displayed on Table 1, Table 2 and Table 3. Table 1 presents building data. The most important figure presented is the amount of undamaged buildings, but all the other values are useful in strategic analysis. The amount of burning buildings is a good indication of whether the Fire Brigades have fires under control - looking at the displayed data we can easily see that they don't. Two important records, in tactical terms, are the number of re-ignited buildings and of those preemptively flooded that caught fire. With a perfect strategy, when a building is flooded it is done with the right amount of water. Too much will waste both water and time - extremely precious resources; while too little is even worse, allowing the building to catch fire and disrupting the prevention strategy. The same hold true for re-ignition of extinguished buildings. When a building is extinguished, the Fire Brigades should take their surroundings into account in order to decide whether they should keep

[6] `http://www.fe.up.pt/~rescue/FCPx`

pouring water into the building to further lower its temperature or focus their efforts on a different building.

Table 1. Fire Brigades - Building related data.

Building Related Data		
Legend		Total Number of Buildings
Burning Buildings	Flooded Unburned Buildings	740
Extinguished Buildings	Number of Times Buildings Reignited	Initial Undestroyed Area
Completely Burned Buildings	Perc. of Bldgs. Undamaged by Fire nor Water	14895169
Preemptively Watered Buildings that Caught Fire		Number of Refuges
		1

Time	Burning	Extinguished	Burned	Flooded	Re-ignited	Flooded That Caught Fire	Bldgs. Undamaged by Fire nor Water
0	N/A	N/A	N/A	N/A	0	0	N/A
1	6	0	0	0	0	0	99,19%
2	6	0	0	0	0	0	99,19%
3	6	0	0	0	0	0	99,19%
4	6	0	0	0	0	0	99,19%
5	6	0	0	0	0	0	99,19%
6	7	0	0	0	0	0	99,05%
7	8	0	0	0	0	0	98,92%
8	8	0	0	0	0	0	98,92%
9	6	2	0	0	0	0	98,92%
10	7	2	0	0	0	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	2	191	[illegible]	9	0	64,73%
290	61	2	209	0	9	0	63,24%
300	53	2	229	0	9	0	61,62%

Table 2 exhibits the Civilian data relevant to Fire Brigades. This includes de amount of Civilians killed by fire, which is an extremely important figure, due to the high value of every live. By the end of the simulation twelve people were killed by flames, which means over twelve points were lost this way.

Agent related data is displayed on Table 3. Analyzing these numbers we should notice two important facts. Firstly, there were two Fire Brigade casualties. Rescue agents' casualties are avoidable and should never take place. In order to accomplish that, agents must be tweaked in order to ensure that only necessary risks are taken. Secondly, there are two buried Fire Brigades since the beginning of the simulation and only one is rescued. With the uncontrolled blaze present in this map, one more active Fire Brigade could prove extremely helpful.

Like tables, charts are also generated automatically in FCPx spreadsheets. In Chart 1a it can be seen that only 42% of the total score was kept and most was lost due to casualties. In chart 1b it is shown that most Civilians died due to sustained injuries from collapsed buildings, but a rather significant part (17%) were burned alive. As can be seen, using FCPx it becomes possible to analyze a single type of agent.

5 Comparing Different Teams and Strategies

The tables from the previous section provide us only rough guidelines of the team's flaws and do not make it obvious what the coding priorities should be. In

Table 2. Fire Brigades - Civilian related data.

	Civilian Related Data				
	Legend				
	Percentage of Civilians Discovered		Civilians Killed by Fire		
	Total Number of Civilians Discovered		Perc. of Dead Civilians Killed by Fire		Total Number of Civilians
	Fraction of Those Discovered by Firebrigades				70
	Civilians Discovered				
Time	Perc. of Civs. Discovered	No. Civs. Discovered	Discovered by FBs.	Civilians Killed by Fire	Perc. of Dead Civilians Killed by Fire
0	5,71%	4	2	0	0,00%
1	5,71%	4	2	0	0,00%
2	5,71%	4	2	0	0,00%
3	5,71%	4	2	0	0,00%
4	7,14%	5	2	0	0,00%
5	7,14%	5	2	0	0,00%
6	7,14%	5	2	0	0,00%
7	12,86%	9	4	0	0,00%
8	22,86%	16	8	0	0,00%
9	22,86%	16	8	0	0,00%
10	22,86%	16	8	0	0,00%
20	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
290	82,86%	58	10	12	30,77%
300	82,86%	58	10	12	30,77%

Table 3. Fire Brigades - Agent related data.

	Agent Related Data				
	Legend				
	Burried Police Forces Killed by Fire			FireBrigades Killed by Fire (Not Burried)	
	Burried Ambulances Killed by Fire			Number of Buried FireBrigades (Alive)	Total Number of FireBrigades
	Burried FireBrigades Killed by Fire				13
	Buried Agents Killed by Fire				
Time	Police	Ambulance	FireBrigade	FireBrigades Killed by Fire (Not Burried)	No. Buried FireBrigades (Alive)
0	0	0	0	0	N/A
1	0	0	0	0	N/A
2	0	0	0	0	2
3	0	0	0	0	2
4	0	0	0	0	2
5	0	0	0	0	2
6	0	0	0	0	2
7	0	0	0	0	2
8	0	0	0	0	2
9	0	0	0	0	2
10	0	0	0	0	2
20	[illegible]	0	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	0	2	1
270	0	0	0	2	1
280	0	0	0	2	1
290	0	0	0	2	1
300	0	0	0	2	1

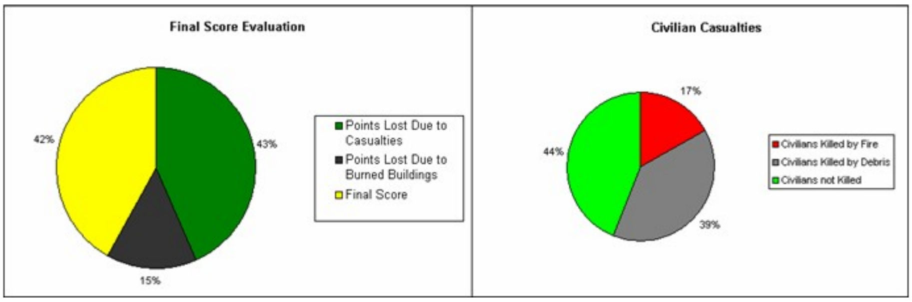


Fig. 1. a) Final score evaluation; b) Civilian casualties.

order to acquire that knowledge, we should compare our team with the current RoboCupRescue World Champions - team Impossibles, from Iran . This will allow us to perceive the most important differences. Since the competition has an open source policy, it also allows any team to analyze its weaknesses and, finding a team which performs better, adapt their strategies. For this purpose another custom spreadsheet was developed - this one oriented towards team comparison.

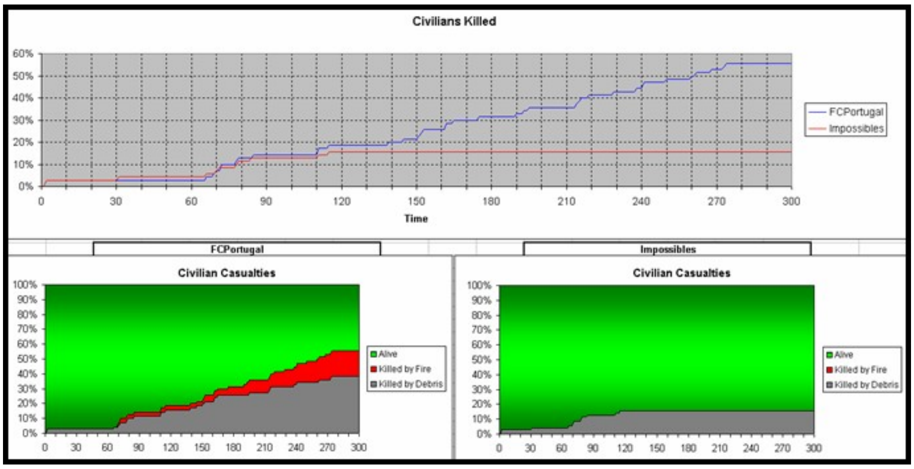


Fig. 2. a) Civilians Killed; b) FC Portugal - Civilians casualties (stacked); c) Impossibles - Civilians casualties (stacked).

The enormous difference in Civilian casualties becomes clear on Chart 2. While no Civilians were burned to death on the Impossibles' simulation, around 17% of the Civilians present in FC Portugal's simulation died by exposure to fire. The Impossibles' Ambulances were also undoubtedly more productive as the number of casualties caused by collapses was also extremely lower.

As seen on Chart 3, our opponents are able to control the fire within the first sixty simulation cycles. Besides reducing score loss from building destruction and burned civilians, this allows their Fire Brigades to pursue other tasks: namely exploration towards Civilian discovery.

Fig. 3. a) Buildings on fire; b) Building area destroyed.

Table 4 provides us with a summary of team's comparison. Some of the facts commented on before are assembled here, allowing for some conclusions to be consolidated. The necessity to improve the Fire Brigades becomes obvious for three primary reasons: reduce the amount of points lost to burned buildings; reduce the amount of points due to burned Civilians; and allow the Fire Brigades to act as scouts when they are able to rapidly control the fires.

Table 4. Comparison summary.

Display Name:	FCPortugal	Impossibles
Team:	FCPortugal	Impossibles
Map:	KobeHard	KobeHard
Date:	07|15	07|15
Time:	10:29:20	10:58:55

*You may change the Display Names as you please

Score Data	FCPortugal	Impossibles
Total Number of Agents	99	99
Number of Agents not Killed	58,59%	88,89%
Percentage of Building Area Destroyed	31,19%	0,39%
Initial Score	100,00	100,00
Points Lost Due to Civilian Casualties	43,18%	11,29%
Points Lost Due to Burned Buildings	14,98%	0,19%
Final Score	41,84%	88,51%

Civilian Casualties	FCPortugal	Impossibles
Total Number of Civilians	70	70
Civilians Killed by Fire	17,14%	0,00%
Civilians Killed by Debris	38,57%	15,71%
Civilians not Killed	44,29%	84,29%

Civilians Discovered	FCPortugal	Impossibles
Civilians Discovered by FireBrigades*	10	43
Civilians Discovered by Police Forces*	44	26
Civilians Discovered by Ambulances*	5	4
Civilians not Discovered*	12	0

*Note: As explained in the civilians tab, these parts may not add up to the total number of civilians.

Civilians Discovered (Normalized by Agent Type)	FCPortugal	Impossibles
Total Number of FireBrigades	13	13
Total Number of Police Forces	10	10
Total Number of Ambulances	6	6
Civilians Discovered by Each FireBrigade	0,77	3,31
Civilians Discovered by Each Police Force	4,40	2,60
Civilians Discovered by Each Ambulance	0,83	0,67

6 Conclusions and Future Work

The FCPx tool usefulness was proved just after its introduction in the rescue community. Several teams use it now to compare their different strategies and to compare their best strategies with other teams' strategies. Also, its open source nature and its integration into the known Freiburg's 3D viewer makes it a very easy to extend tool.

The tool enables to analyse team strategies in a very easy way and conclusions can be drawn and directions set by a simple analysis of tables and charts. What took before several simulations, detailed manual analysis and great attention, to be determined, can now be done in minutes. Using the files generated, it is a simple endeavour to create personalized spreadsheets, to evaluate a single set of parameters in different teams.

We hope our tool can solve or simplify some of the issues associated with improving a rescue team. By using a different tool to run a set of consecutive simulations it becomes possible to leave a machine running agents with different strategies, for an extended period of time, while the developer focus on different tasks. Later analysis can be done in an easy, systematic fashion, allowing faster and better informed decisions, concerning development paths.

The tool's webpage has seen quite some traffic for such a restricted community. The main page had over four hundred visits and the download page has been seen over a hundred times. It was, nevertheless, expected, since FCPx fills a gap which had been previously mentioned several times in on-line discussion.

Being the first release of this tool, only groups of agents are analyzed. Future development will be oriented into analyzing the team agents as individuals, and as such the tool will allow a greater perception off exceptional situations that led to a poorer performance.

Future work is also related with the use of our tool to enable the use of machine learning techniques in the context of (removed for blind review) team. Having a high-level configurable strategy and the capability to evaluate it, automatically, also at a high-level, using different parameters, enables us to select the best strategy for each type of map, using common machine learning techniques.

Still, even in its present form its usefulness is hard to be disputed and it had a crucial part in our achievements in RoboLudens 2006. Using FCPx's abilities to correct our weaknesses, we were able to win the competition and became the current European champions.

7 Acknowledgements

This project is a collaboration between the Universities of Aveiro and Porto. The authors wish to thank FCT/PT for the fellowship granted. This research was supported by FCT under grant # POSI/EIA/63240/2004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the sponsor.

We would also like to thank Alexander Kleiner for creating Freiburg's 3D Viewer, a great piece of software that allowed us to develop the tool focused on this paper.

References

1. Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara. Robocup: A challenge problem for ai and robotics. In *RoboCup*, pages 1–19, 1997.
2. H. Kitano, S. Tadokor, H. Noda, I. Matsubara, T. Takhasi, A. Shinjou, and S. Shimada. Robocup-rescue: Search and rescue for large scale disasters as a domain for multi-agent research, 1999.
3. Francisco Reinaldo, Joao Certo, Nuno Cordeiro, Luís Paulo Reis, Rui Camacho, and Nuno Lau. Applying biological paradigms to emerge behaviour in robocup rescue team. In Carlos Bento, Amílcar Cardoso, and Gaël Dias, editors, *EPIA*, volume 3808 of *Lecture Notes in Computer Science*, pages 422–434. Springer, 2005.
4. H. Akin, C. Skinner, J. Habibi, T. Koto, and S. L. Casio. *Robocup 2004 Rescue Simulation League Rules V1.01.* The RoboCup Rescue Technical Committee, 2004.
5. Nuno Cordeiro, Francisco Reinaldo, Jo ao Certo, Luís Paulo Reis, and Nuno Lau. Fcpx home page. `http://www.fe.up.pt/~rescue/FCPx`, Feb 2006.
6. Arian Team. Official arian home page. `http://ce.sharif.edu/~arian/`, Nov 2005. Sharif University of Technology.
7. Alexander Kleiner and Moritz Gbelbecker. Rescue3d: Making rescue simulation attractive to the public public.
8. Takeshi Morimoto. *How to Develop a RoboCupRescue Agent for RoboCupRescue Simulation System version 0.* The RoboCup Rescue Technical Committee, Nov. 2002.
9. Takeshi Morimoto. Viewer for robocuprescue simulation system. `http://ne.cs.uec.ac.jp/morimoto/rescue/viewer/`, 2002.

Intelligent Tutoring Systems

Towards an Integral Architecture for Semantic Web, Adaptive and Intelligent WBE Systems

Alejandro Canales Cruz[1], Rubén Peredo Valderrama[1], J. Humberto Sossa Azuela[1] and Agustín Fco. Gutiérrez Tornés[2]

[1] Computer Science Research Center of National Polytechnic Institute, Mexico City, Mexico
[2] National Bank of Mexico - BANAMEX, Mexico City, Mexico
+ (52) 55 57296000 Ext. 56611
{acc, peredo, hsossa}@cic.ipn.mx; agutornes@banamex.com.mx

Abstract. A new Agents and Components Oriented Architecture for the development of Web-Based Education systems is presented. This architecture is based on the IEEE 1484 LTSA (Learning Technology System Architecture) specification and the software development pattern of the Intelligent Reusable Learning Components Object Oriented (IRLCOO). IRLCOO are a special type of Sharable Content Object (SCO) according to the Sharable Content Object Reference Model (SCORM). SCORM is used to create reusable and interoperable learning content. The architecture, communication model (LMS API, Web Services, AJAX, and Struts Framework) and Semantic Web Platform that is described in this paper is used to develop authoring and evaluation tools oriented to offer application level interoperability under the philosophy of Web Services.

1 Introduction

The use of WBE as a mode of study is due to the increase in the number of students and limited learning content resources available to meet a wide range of personal needs, backgrounds, expectations, skills, levels, etc. Therefore, the purpose of the delivery process is very important, because it means to produce learning content and to present it to the learner in multimedia form. Nowadays, there are approaches over this process that focus on new paradigms to produce and deliver quality content for online learning experiences. These approaches try to develop, revise and upgrade the learning content in an efficient way. The work described in this paper is based on a special type of labeled materials called IRLCOO, developed by Peredo et al [1]. The IRLCOO represent a kind of learning content characterized by rich multimedia, high interactivity and intense feedback that is supported by means of a standard interface and functionality.

The IRLCOO are part of a new Agents and Components Oriented Architecture (ACOA) based on IEEE 1484 LTSA specification [2] and open standards such as XML [3] as a bar coding system and to make sure that the learning content is interoperable, the Global IMS Learning Consortium [4], Advanced Distributed Learning (ADL), and SCORM [5]. This paper is organized as follows: in Section 2, ACOA and

© A. Gelbukh, C.A. Reyes-García. (Eds.)
Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 151-162

Received 03/06/06
Accepted 02/10/06
Final version 19/10/06

IRLCOO are described; in Section 3, the authoring system based on ACOA and Conceptual Maps is presented; furthermore in Section 4, the evaluation system based on ACOA and Semantic Web Platform is showed; finally, the conclusions are discussed.

2 Agents and Components Oriented Architecture

Between the key issues of software engineering is the aim for developing quality software. Thus, components are widely seen by software engineers as a main technology to address the "software crisis." The Industrial Software Revolution is based upon component software engineering. Between the reasons that explain the relevance of the Component-Oriented Programming (COP) are: the high level of abstraction of-fered by this paradigm and the current trends for authoring reusable component librar-ies, which support the development of applications for different domains. In addition, according to Wang [6] three major goals pursued by COP are considered: conquering complexity, managing change, and reusability.

According to Szyperski [7] a software component is "a unit of composition with contractually specified interfaces and explicit context dependencies. A software component can be deployed independently and is object to composition by third parties." Although in most cases this definition is acceptable, its meaning is quite generic, so it is not surprising that the term is used to mean rather different concepts.

Our ACOA is based on layer 3 of IEEE 1484 LTSA specification. This architecture is presented in Fig. 1, and consists in four processes: learner entity, evaluation, coach, and delivery process; two stores: learner records and learning resources; and fourteen information workflows.

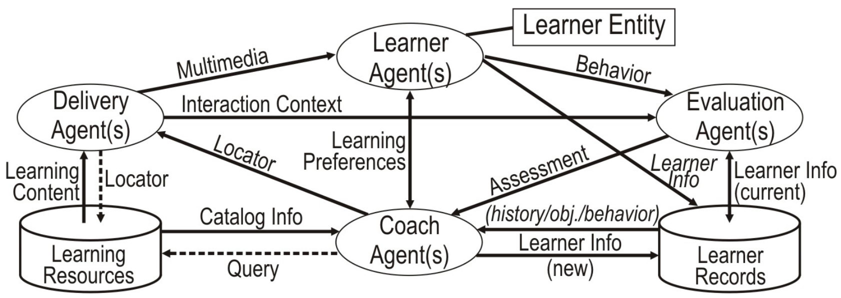


Fig. 1. Components Oriented Architecture.

First, the coach process has been divided in two subprocesses: coach and virtual coach. The reason is because we considered that this process has to adapt to the learners' individual needs in a quick way during the learning process. For this, some decisions over sequence, activities, examples, etc., can be made manually for the coach but in others cases these decisions can be made automatically for the virtual coach.

Briefly, the overall operation has the following form: (1) the learning styles, strategies, methods, etc., are negotiated among the learner and other stakeholders and are communicated as learning preferences; (2, new proposal) the learner information (behavior inside the course, e.g., trajectory, times, nomadicity, etc.) is stored in the learner records; (3) the learner is observed and evaluated in the context of multimedia interactions; (4) the evaluation produces assessments and/or learner information; (5) the learner information (keyboard clicks, mouse clicks, voice response, choices, written responses, etc., all over learner's evaluation) is stored in the learner history database; (6) the coach reviews the learner's assessment and learner information, such as preferences, past performance history, and, possibly, future learning objectives; (7, new proposal) the virtual coach reviews the learner's behavior and learner information, and automatic and smartly he makes dynamic modifications on the course sequence (personalized to learner's needs) based on the learning process design; (8) the coach/virtual coach searches the learning resources, via query and catalog info, for appropriate learning content; (9) the coach/virtual coach extracts the locators (e.g., URLs) from the available catalog info and passes the locators to the delivery process, e.g., a lesson plan or pointers to content; and (10) the delivery process extracts the learning content and the learner information from the learning resources and the learner records respectively, based on locators, and transforms the learning content to an interactive and adaptive multimedia presentation to the learner. In the section 4 describes ACOA from agents viewpoint.

2.1 IRLCOO platform

IRLCOO were developed with Flash. Flash is an integrator of media and have a powerful programming language denominated ActionScript 2.0 [8]. This language is completely Object Oriented and enables the design of client components that allows multimedia content. At Run-Time, the components load media objects and offer a programmable and adaptive environment to the student's necessities. Flash already has Smart Clips for the learning elements denominated Learning Interactions. The aim is to generate a multimedia library of IRLCOO for WBE systems with the purpose to separate the content from the control. Thus, the components use different levels of code inside the Flash Player. With this structure, it is possible to generate specialized components which are small, reusable, and suitable to integrate them inside a bigger component at Run-Time. The liberation of ActionScript version 2.0 inside Flash MX 2004 allows the implementation of the Object Oriented paradigm. With these facilities IRLCOO are tailored to the learners' needs. In addition, this IRLCOO development platform owns certain communication functionalities inside the Application Programming Interface with LMS, Multi-Agent System (MAS), and different frameworks, as AJAX [9], Hibernate [10], Struts [11], etc.), and dynamic load of Assets in Run-Time.

IRLCOO are meta-labeled with the purpose of complete a similar function as the product bar codes, which are used to identify the products and to determine certain characteristics specify of themselves. This contrast is made with the meta-labeled Resource Description Framework (RDF-XML) [12], which allows enabling certain grade inferences on the materials by means of the Semantic Web Platform.

2.2 Communication between IRLCOO and Web Services

ActionScript 2.0 adds the component WebServiceConnector to connect to Web Services (WS) from the IRLCOO. The WebServiceConnector component enables the access to remote methods offered by a LMS through SOAP protocol. This gives to a WS the ability to accept parameters and return a result to the script, in other words, it is possible to access and join data between public or own WS and the IRLCOO. It is possible to reduce the programming time, since a simple instance of the WebServiceConnector component is used to make multiple calls to the same functionality within the LMS. The components discover and invoke WS using SOAP and UDDI, via middleware and a JUDDI server. Placing a Run-Time layer between a WS client and server dramatically increases the options for writing smarter, more dynamic clients. Reducing the needs for hard-coded dependencies within WS clients. It is only necessary to use different instances for each one of the different functionalities. WS can be unloaded using the component and deployed within an IRLCOO.

3 SiDeC

In order to facilitate the development of learning content, it was built an authoring system called SiDeC (Sistema de Desarrollo de eCursos - eCourses Development Sys-tem). SiDeC is a system based on ACOA to facilitate the authoring content to the tu-tors who are not willing for handling multimedia applications. In addition, the Struc-ture and Package of content multimedia is achieved by the use of IRLCOO, as the lowest level of content granularity.

SiDeC is used to construct Web-based courseware from the stored IRLCOO (Learning Resources), besides enhancing the courseware with various authoring tools. Developers choose one of the SiDeC lesson templates and specify the desired compo-nents to be used in each item. At this moment, the SiDeC lesson templates are based on the cognitive theory of Conceptual Maps (CM) [13], but in the future we will con-sider others theories such as: Based-Problems Learning (BPL), the cases method, etc.

The inclusion of cognitive elements, as CM, obeys to the instructional design pattern for the development of the courses. Thus, the courses do not only have theoretical or practical questions, but rather they include a mental model about individual thought process. CM is a schema to structure concepts with the purpose of helping the learners to maximize the knowledge acquisition. A CM is a graphical technique used during the teaching-learning process, among other forms as instructional and learning strat-egy, and as schematic resource or navigation map.

A metadata tool supports the generation of IRLCOO to provide on-line courses. This courseware estimates learners' metrics with the purpose to tailor their learning experiences. Furthermore, the IRLCOO offer a friendly interface and flexible functionality. These deliverables are compliance with the specifications of the IRLCOO and with learning items of SCORM 1.2 Models (Content Aggregation, Sequencing and Navigation, and Run Time Environment) [5]. Metadata represent the specific description of the component and its contents, such as: title, description, keywords, learning objectives, item type, and rights of use. The metadata tool provides templates for entering metadata and storing each component in the SiDeC or another IMS/IEEE standard repository.

SiDeC proposes a course structure based on the idea of a compound learning item as a collection of Reusable Learning Atoms (RLA) and Reusable Information Atoms (RIA) [14]. These atoms are grouped together to teach a common task based on a single learning objective, as is depicted in Fig. 2. A RLA is an elementary atomic piece of learning that is built upon a single learning objective. Each RLA can be classified as: concept, fact, process or procedure. The RLAs provide the information of learner's behavior within the course, e.g., trajectory, times, and assessments. This information is stored in the learner history database (learner records).

On the other hand, a RIA is an atomic piece of information that is built upon single information object. It may contain up to seven different content items, such as: overview, introduction, importance, objectives, prerequisites, scenario, and outline.

In Fig. 2, the SiDeC implements the CM as a navigation map or instructional and learning strategy allowing to the learner to interact with content objects along the learning experiences. These experiences follow an instructional-teaching strategy. These kinds of strategies carry out modifications of the learning content structure. Such modifications are done by the designer of the learning experience with the objective of provide significant learning and to teach the learners how to think [15]. The learning content can be interpreted in a Learning Content Tree.

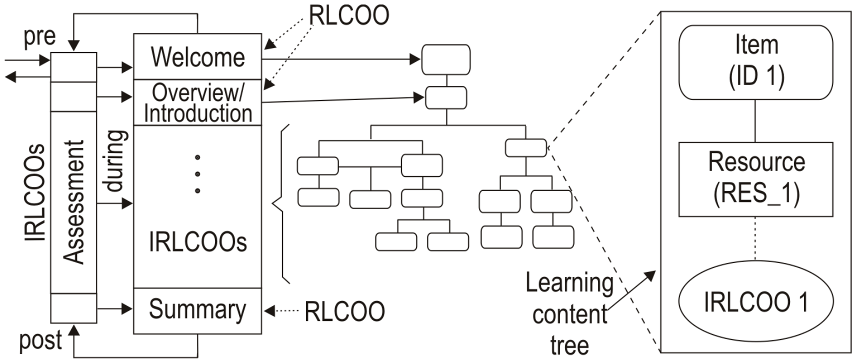


Fig. 2. Learning content generated for the SiDeC.

3.1 Communication between IRLCOO and LMS

Our communication model uses an asynchronous mode in Run-Time Environment (RTE) and joins to LMS communication API of ADL [5], and AJAX (Asynchronous JavaScript And XML) [9] and Struts Framework [11] for its implementation. The LMS communication API of ADL consists of a collection of standard methods to let the Client to communicate with the LMS.

AJAX is a Web development technique to create interactive applications that it is executed in client side, in other words, the Web browser maintains the asynchronous communication with the server in backstage. This way it is possible to carry out changes in the same page without necessity of reload it. This increases the interaction speed.

On the other hand, the Struts Framework is a tool for Web application development under the Java MVC (Model-View-Controller) architecture; with this Framework is defined the independent implementation of the Model (business object), the View (in-terface with the user or another system) and the Controller (controller of the applica-tion workflow). This Framework provides the advantage of maintainability, perform-ance (tags pooling, caching, etc.), and reusability (contains tools for the field validation that it is executed in client or server sides). The browser-based communica-tion model is depicted in Fig. 3.

According to Fig. 3, the communication model starts: (1) when an IRLCOO generates an event. (2) Form the browser interface is made a JavaScript call to the function FileName_DoFSCommand(command,args), which handles all the FSCommand messages from IRLCOO, LMS communication API, and AJAX and Struts methods. Next, a fragment of this code is showed:

```
function FileName_DoFSCommand(command,args)  {
 doInitialize();
 doSetValue(name,value); // i.e. (StudentName,name)
 doTerminate();
 useAjaxStruts();  }
```

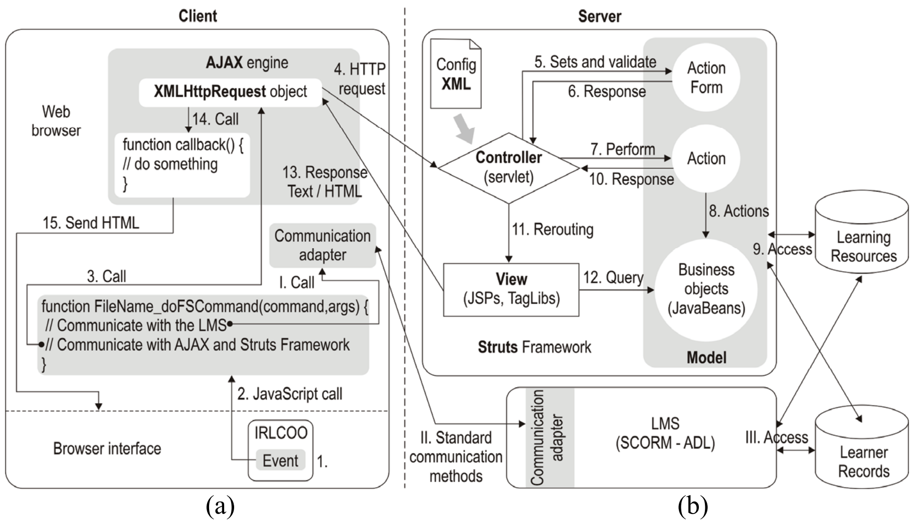


Fig. 3. Communication model between IRLCOO, LMS and AJAX and Struts Framework.

The communication with the LMS starts when: (I) the standard methods call to the Communication Adapter (written in JavaScript). (II) The communication adapter implements the bidirectional communication ADL´s API between the Client and the LMS. (III) The LMS realizes the query-response handling and the business logic, i.e., the access to the database.

The communication with AJAX and Struts Framework begins when AJAX-Struts method is called. (3) An instance of the XMLHttpRequest object is created. Using the open() method, the call is set up, the URL is set along with the desired HTTP method, typically GET or POST. The request is actually triggered via a call to the send() method. This code might look something like this:

```
function useAjaxStruts ()  {
  createXMLHttpRequest();
  var url = "register.do?";
  var urlString = createUrlString();
  xmlHttp.open("POST",url,true);
  xmlHttp.onreadystatechange = processStateChange;
  xmlHttp.setRequestHeader("Content-Type",
    "application/x-www-form-urlencoded;");
  xmlHttp.send(urlString);  }
```

(4) A request is made to the server, this might be a call to a servlet or any server-side technique. (5) The Controller is a servlet which coordinates all applications activities, such as: reception of user data, (6) data validations, and control flow. The Controller is configured for a XML file. (7) The Controller calls to perform method of Action, it passes to this method the data values and the Action reviews the characteris-tic data that correspond to the Model. (8) The business objects (JavaBeans) realize the business logic, (9) usually a database access. (10) The Action sends the response to the Controller. (11) The Controller reroutes and generates the interface for the results to the View (JSPs). (12) The View makes the query to the Business objects based on the correspondent interface. (13) The request is returned to the browser. The Content-Type is set to text/xml, the XMLHttpRequest object can process results only of the text/ html type. In more complex instances, the response might be quite involved and include JavaScript, DOM manipulation, or other technologies. (14) The XMLHttpRe-quest object calls the function callback() when the processing returns. This function checks the readyState property on the XMLHttpRequest object and then looks at the status code returned from the server. (15) Provided everything is as expected, the callback() function sends HTML code and it does something interesting on the client, i.e. advanced dynamic sequence.

This communication model provides new wide perspectives for the WBE systems development, because it provides the capabilities of communication, interaction, interoperability, security, and reusability, between different technologies. For example, the LMS communication API allows us to make standard database queries of learners' information such as personal information, scores, assigned courses, trajectory, etc. While the communication with AJAX and Struts Framework provides the capability of modify the learner's trajectory according to variables from the learner records in RTE (advanced dynamic sequence), components management (IRLCOO) – remember that these components are built and programming with XML – then, this model provides the way to write, load, change and erase XML files in the Server side.

4 Evaluation System

The Evaluation System for WBE is designed under the same philosophy used for the SiDeC. The functionality of the Evaluation System lays on the analysis of the learner's profile, which is built during the teaching-learning experiences. The profile is based on metrics that elicited from the learner's behavior at Run-Time. These measures are stored into the learner records that compose the profile. The generation of new sequences of courses is in function of the results obtained, besides the account of the adaptation level.

The Evaluation System combines IRLCOOs, additional meta-labels, and a Java Agent platform. Also, some technologies of the Artificial Intelligence field are considered in order to recreate a Semantic Web environment. Semantic Web aims for assisting human users to achieve their online activities. Semantic Web offers plenty of advantages, such as: reduction of the complexity for potential developers, standardization of functionalities and attributes, definition of a set of specialized APIs, and deployment of a Semantic Web Platform.

All resources have a Universal Resource Identifier (URI). An URI can be a Unified Resource Locator (URL) or some other type of unique identifier. An identifier does not necessarily enable access to a resource. The XML layer is used to define the SCORM metadata of IRLCOO that are used to interchange data over the Web. XML Schema tier corresponds to the language used to define the structure of metadata [3]. RDF level is represented by the language used for describing all information and metadata sorts. RDF Schema layer is carried out by the Framework that provides meaning to the vocabulary implemented [12]. The Ontology tier is devoted to define the semantics for establishing the usage of words and terms in the context of the vocabulary [16]. Logical level corresponds to the reasoning used to establish consistency and correctness of data sets and to infer conclusions that are not explicitly stated [17]. The Proofs layer explains the steps of logical reasoning. The Trust tier provides authentication of identity and evidence of the trustworthiness of data, services and agents.

In resume, the components and operation of the SiDeC and Evaluation System are outlined in Fig. 4. Basically the Evaluation System is fulfilled through two phases. The first phase is supported by the LMS, and is devoted to present the course and its structure. All the actions are registered and the presentation of the contents is realized with IRLCOO content. The evaluations are done by evaluating IRLCOO and in some cases by simulators based on IRLCOO. These processes are deployed by the Framework of Servlets, Java Server Pages and JavaBeans.

The second phase analyzes the learner's records carried out by the Server based on JADE MAS. This agent platform owns seven agents: Snooper, Buffer, Learner, Evaluation, Delivering, Coach, and Info. The fundamental idea is to automate the learner's analysis through the coach/virtual coach, and to give partial results that can be useful for the learner's final instruction. These agents are implemented as JavaBeans programs, which are embedded in the applications running both at the client and server sides. The Snooper Agent works as a trigger by means of the INFORM performative, which activates the MAS server's part. This agent is deployed into a Java Server Page that uses a JavaBean. During the lesson or once evaluation is finished, the graphical user interface activates the Snooper Agent and sends it the behavior or evaluation metrics (using Agents Communications Language [18]) to be analyzed at the server-side of the MAS. The Snooper Agent activates the system, whereas the Buffer Agent manages the connection and all the messages from the client. Both tasks are buffered and send them to the Coach Agent. Then the Coach Agent requests to the learner records for the preferences learner, trajectory, previous learner monitor-ing information, etc. The Coach Agents analyzes this information to determine if the learner needs help. If this situation is true, the Coach Agent requests to the learning resources the needful learning content (URLs) and it sends the learning contents (URLs) to the Delivery Agent. The Delivery Agent sends the learning content to the Learner and Evaluation Agents for its presentation. These agents employ the dynamic sequencing to change the course or assessment sequence. The sequencing is defined for the instructional strategy based on CM and it employs the SCORM Sequenc-ing/Navigation. Once the necessary information is received (sequence, kind of IRLCOO and localization, etc.), this is represented as a string, which is constructed dynamically by the rule-based inference engine known as JENA [19] and JOSEKI server [20], to generate dynamic feedback.

Fig. 5 illustrates an overview of how ours systems are integrated with ACOA. In general, the software architecture is divided into four layers: application, agents & components, database, and server layers. The application layer includes an administration system, which is the ADL platform, to allow system administrators, instructors, and learners to manage learner records and curriculum. On the left side below the administration system, asynchronized ours systems are incorporated. Thus, structure authoring systems are separated from learning content.

4.2 Semantic Web Platform

The overall architecture of Semantic Web Platform, which includes three basic engine representing different aspects, is provided in Fig. 4.

1. The query engine receives queries and answers them by checking the content of the databases that were filled by info agent and inference engine.

2. The database manager is the backbone of the entire systems. It receives facts from the info agent, exchanges facts as input and output with the inference engine, and provide facts to the query engine.
3. The inference engine use facts and ontologies to derive additional factual knowledge that is only provided implicated. It frees knowledge providers from the bur-den of specifying each fact explicitly.

Again, ontologies are the overall structuring principle. The info agent uses them to extracts facts, the inference engine to infer facts, the database manager to structure the database, and query engine to provide help in formulating queries.

JENA was selected as the inference engine. It is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine [19].

While JOSEKI was selected as Web API and server. It is an HTTP and SOAP engine supports the SPARQL Protocol and the SPARQL RDF Query language. SPARQL is developed by the W3C RDF Data Access Working Group [20].

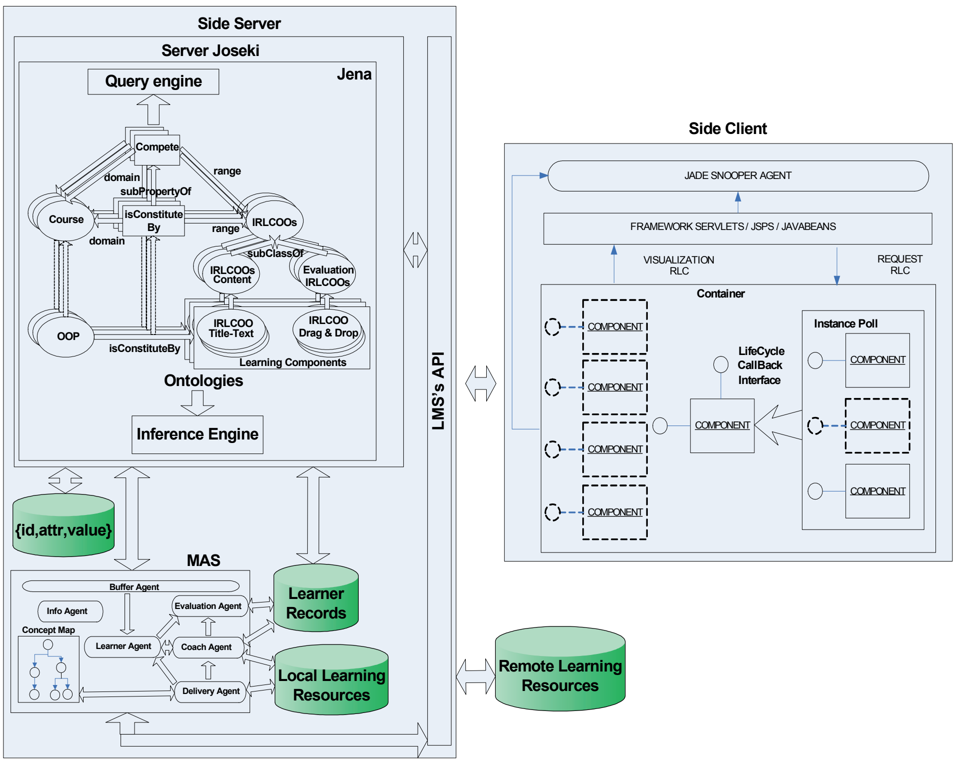


Fig. 4. Semantic Web Platform for WBE.

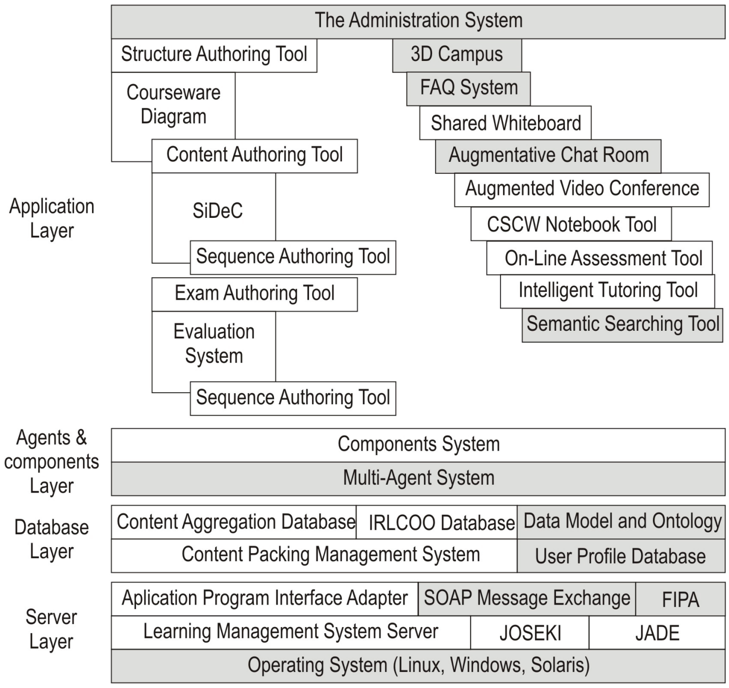


Fig. 5. An integral architecture for WBE systems.

5 Conclusions

ACOA, IRLCOO and Semantic Web Platform allow developing authoring and evaluation systems to create adaptive and intelligent WBE. Our approach focus on: reusability, accessibility, durability, and, interoperability of the learning contents, which are built as IRLCOO, as the main component for delivering learning and evaluation content.

The communication model composes for the LMS communication API, AJAX and Struts Framework, IRLCOO, WS, Semantic Web, and JUDDI. It provides new development capabilities for WBE systems, because their integrant technologies are complementary. SiDeC and the Evaluation System were developed under this model to help in the automation and reduce of the complexity of the learning content process.

The incorporation of Web Semantic Platforms helps us to create intelligent and adaptive systems (bidirectional communication), according to the users needs.

The ADL-Schema manages dynamic sequencing, composition, content separation, and navigation in RTE for development learning and evaluation content in Web. While, our ACOA has the same ADL advantages and adds the capacity of generates desk and Web CASE tools using the same learning and evaluation components generated.

References

1. Peredo, R., Balladares, L., & Sheremetov, L.: Development of intelligent reusable learning objects for web-based education systems. Expert Systems with Applications. 28(2). (2005) 273-283
2. IEEE 1484.1/D9 LTSA: Draft Standard for Learning Technology - Learning Technology Systems Architecture (LTSA). New York, USA (2001). URL: http://ieee.ltsc.org/wg1
3. XML specifications (2006). URL: http://www.w3.org/XML/
4. Global IMS Consortium (2005). URL: http://www.imsproject.org
5. Advanced Distributed Learning Consortium (2006). URL: http://www.adlnet.org
6. Wang, A., & Qian, K.: Component-Oriented Programming. John Wiley & Sons, Inc., Publication. Georgia, USA. (2005) 3-5
7. Szyperski, C.: Component Software. Beyond Object-Oriented Programming. Addison-Wesley Editorial. USA (1998)
8. Macromedia, Inc. (2006). URL: http://www.macromedia.com
9. Grane. D., Pascarello, E., & James, D.: Ajax in Action. Manning Publications. Greenwich, USA (2006)
10. Peak, P. & Heudecker, N.: Hibernate Quickly. Manning Publications. Greenwich, USA (2006)
11. Holmes, J.: Struts: The Complete Reference. Mc Graw Hill – Osborne Publications.Edited by Herbert Schild. California, USA (2004)
12. RDF specification (2005). URL: http://www.w3.org/RDF/default.htm
13. Novak, J. & Gowin, B.: Learning How to Learn. Cambridge University Press. Cambridge, USA (1984)
14. Vladimir, U., & Maria, U.: Reusable learning objects approach to Web-Based Education. International Journal of Computer and Applications, 25(3). (2003)
15. Díaz-Barriga, F.: Educational strategies for a significant learning. Mc Graw Hill Publication. Second edition. D.F., México (2002)
16. Passin, T.: Explorer's Guide to Semantic Web. Manning Publications Co. USA (2004)
17. Antoniou, G. & Van Harmelen, F.: A Semantic Web Primer. The MIT Press. USA (2004)
18. FIPA XC00061 (2001). FIPA ACL message structure specification
19. JENA (2006). URL: http://jena.sourceforge.net/
20. JOSEKI server (2006). URL: http://www.joseki.org/

Putting Artificial Intelligence Techniques into Distance Education

Pedro Salcedo L., M. Angélica Pinninghoff J., Ricardo Contreras A.

Dept. Ing. Informática y Ciencias de la Computación
Facultad de Ingeniería
Universidad de Concepción, Chile
psalcedo@udec.cl

Abstract. This paper reflects on how some Artificial Intelligence Techniques may positively affect the operation of a distance education platform. It addresses to a particular distance education platform at which specific characteristics are incorporated, such as the adaptation to different users profiles as much to diagnose as to determine a plan for the most adequate teaching strategies. The use of Bayes nets and neural networks are mentioned in the evaluation process and a summarized example is included in the use of the platform.

1 Introduction

In recent years, computer aided teaching has been strongly impacted by incorporating artificial intelligence techniques, as a form of approaching the strategies that artificial systems get closer to the way in which humans take decisions and solve problems in education. This idea is not new, Papert and Minsky regarded computer sciences as an instrument for positively modifying the way in which people learn and not only as a form that allows to explain this process [6].

The very first artificial intelligence application in the field of education has been the implementation of intelligent tutorials, i.e., systems with the capability of adapting the instruction process to student's characteristics. This goal is achieved by creating a relationship between the four basic system components: expert module, student model, pedagogical module and interface. The expert module level is the one that solves the problems intended to be taught; consequently it must have a broad knowledge and an ample structure of the subject to allow for an optimum management of them. The student model is a database that represents the student knowledge at any given time. The pedagogical module is the one in charge of managing in which sequence the subjects are to be presented according to the information gathered from the student model determining when and what kind of help the student needs. The interface supplies backing to the student activities and methods used to carry out such activities. The interface must be easy to use in such a way that the student waste a minimum time in learning on how to use it and concentrate in the learning process of the subject.

Advances in Artificial Intelligence.
Research in Computing Science 26, 2006, pp. 163-173

Received 02/06/06
Accepted 03/10/06
Final version 10/10/06

Many specific systems like these, their goals and structure have been discussed in detail by different authors [11, 10, 9]. Today, some efforts are focused on incorporating artificial intelligence techniques into Adaptive Hypermedia Systems (AHS), which appear to be a promising option in computer aided education, mainly due to Internet and WWW. Brusilovsky defines it as "a system based in Hypertext and Hypermedia that reflects some user characteristics and applies this model to adapt some visible aspects of the system to user" [1].

This work introduces artificial intelligence techniques at a distance education platform and whose novelty lies in considering aspects like learning strategies adaptability, communication tools and student profile evaluation: their learning styles and knowledge, these issues have been found partially only in revised platforms both in literature and through Internet.

2 The Distance Education Platform: MISTRAL

This platform shares common features with many similar projects in use today, but it differs in the way on how it adapts to all sorts of student's profile, suggesting activities, providing tools and more appropriate mechanisms for correcting detected misunderstandings.

The course structure adapts to a particular student features such as prior knowledge configuring activities and sequence of contents. In addition, the course structure adapts to the student's learning style. The platform automatically measures different students' styles and then chooses the best strategy for each one of them. Once students have developed the proposed sequence of suitable activities the course structure itself adapts to the student's learning capability. If students do not reach the required qualification, the platform proposes a series of complementary activities different from the previous ones that allows it to enhance the learning process. In section 5 a course example is shown to clarify the platform operation.

In summary, different functionalities of the platform can be highlighted as follows:

- It allows for the management and automatic generation of different distance courses in multi-user modality.
- It makes the knowledge acquisition easier in contents and varied in activities for a particular course.
- The platform has the necessary knowledge for detection and diagnosis of the learning capabilities in a specific unit, and the necessary activities required for improving the learning process.
- It generates a Virtual Classroom, personalized for each student, with discussion groups that adapts to different learning styles and a mailing system that includes a tracking mechanism for warning if the interaction level of a student in some activity is lower than required.

A general scheme for MISTRAL is shown in figure 1.

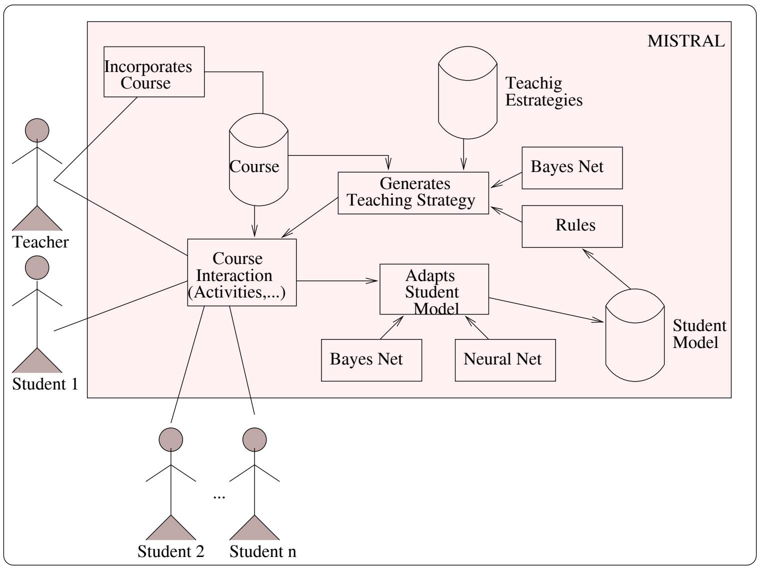


Fig. 1. MISTRAL: The basic components

3 Using the Platform

From the instructor's point of view the use of this platform can be explained through two main concepts: the course development process and the platform operation process. The course development process describes the facilities that help the instructor when creating a new course; this is the way and sequence in which the material will be handled out according to the student particular profile. The platform operation process describes the way in which both the instructor and the student will use this platform and tools given by it during the course development.

3.1 The Course Development Process

This process is carried out through two phases: the first phase is the **identification of requirements** and the second one is the **creation of contents**. In the identification of requirements phase it is necessary a previous planning of the work that must be done to get an effective teaching. An adequate plan has to answer questions like what, how and when to teach. The answers to these questions highlight different issues taking into account: the student, his environment and psychosocial features (the user profile where the learning style is considered), the goals to reach, the contents, the activities (didactic resources) and the correct sequence considering these steps along with evaluation instruments. In computer programs in general and in a knowledge-based distance course development in particular, the planning process is related to analysis, which has been traditionally developed in manual form. The analysis required for developing a course using this platform will affect two components in the system: the student model and the pedagogical module.

In order to analyze the learning-teaching material both books and expert knowledge are fundamental from which a clear view about contents are to be handled out, also the possible sequences to consider and different ways to submit and evaluate are taken. This phase aims to structure the acquired knowledge about the learning strategy, the sequence of contents for different profiles and the sequence of contents for different learning styles. This arrangement can be observed in figure 2. The learning styles taken into consideration here are four: Concrete Experience (CE), Concepts Abstraction (CA), Active Experimentation (AE) and Reflexive Observation (RO) [3].

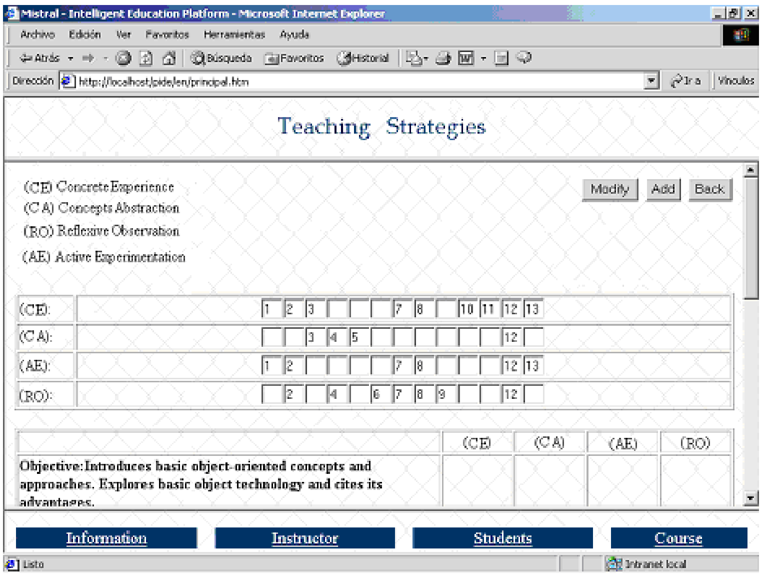


Fig. 2. Sequence of contents for different styles

For the creation of contents the learning-teaching material mainly relies on the expert and in books. To this end, the material representation (frames) and the used mechanisms (semantic networks) are needed. In figure 3, a possible sequence of contents determined for different user profiles is shown, allowing the establishment in a semantic network, of the sequence of contents (previous and further knowledge) for users that present particular characteristics and knowledge.

In order to automatically build a distance course the characteristics of domain's knowledge are required to be provided. Most courses have a structure at a pedagogic level that attempts to represent contents and skills to be delivered to students and the relationship between contents necessary to ensure an effective learning, i.e., in the pedagogic organization of the teaching material.

For instance in the course Object Oriented Analysis and Design, the pedagogic level features are:

- Instructional goals to be accomplished in the course (general and specific objectives).

- In this platform the task of basic learning units is placed as activities, allowing storing of theoretical contents, as the definition of polymorphism, or practical contents or procedures such as the modeling of a problem using object orientation.

- In order for the platform to evaluate the student, different tests and exercises are created.

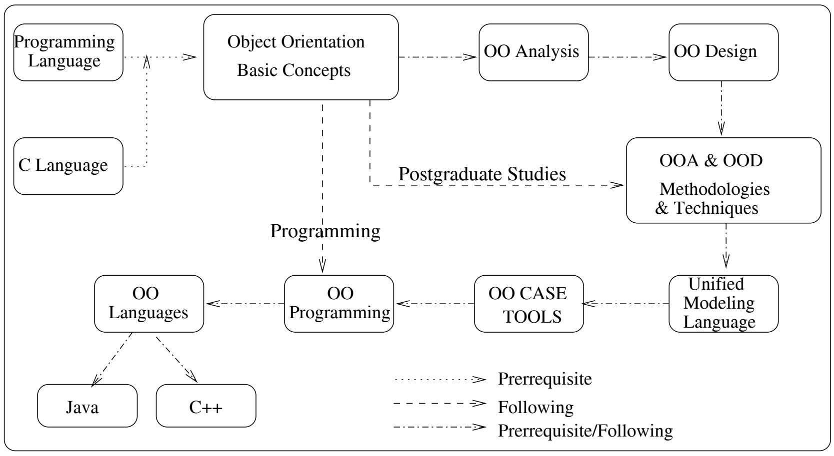


Fig. 3. Sequence of contents for the Object Oriented Analysis and Design Course

3.2 The Platform Operation Process

As the instruction process goes on the system will use the guided method in which this platform after providing a series of activities suitable for a user profile will allow the students to develop these activities by placing the results into a personal portfolio (a repository to store the results of activities accomplished by the student), so that the system generates another sequence of activities suitable for the student. Different authors characterize learning styles in many diverse ways, but in general they locate people between two points in a continuous space. In this work the learning style of David Kolb has been used [3], to discover the learning style of users using this platform, aimed to adapting the more appropriate activities for students to increase one style and support the others. The reason for using Kolb's work is that it represents the classification suggested by the educational authorities in our country.

4 Artificial Intelligence into the Platform

Tasks that need to be performed in order to accomplish a distance course are:

i) Didactic material preparation adjusting contents from the Expert Module, where the experienced person decides the most suitable materials and activities to tackle such content for different students' profile.

ii) Teaching learning strategies contained in the Pedagogical Module that determines the most appropriate sequence of activities to the different student's profile.

iii) Evaluation, assigned to determine the amount of acquired knowedge by the student which will modify the pattern within the Student Model.

iv) Dialogue among students, between student and instructor.

v) System administration.

Different techniques used in AI have been incorporated into the platform. Following, there is a summary of ways in which these techniques are inherent to the didactical material preparation, evaluation and teaching-learning strategies.

Frames. Most all the knowledge available from the subject (didactical material) and about the student (learning style, amount of knowledge acquired) is structured into frames. The arrangement of didactical material in the platform consists of a semantic net of frames. For the actual course construction using this platform the didactical material must be classified in: *General Objective* (course guide), *Specific Objectives* (leading to specific topics) and relative relevance to each objective, *Contents* (components of the specific objectives) and relative relevance to each content. Activities according to different profiles and difficulty levels, and impact ratio for each activity. Difficulty levels are defined by the instructor, based on his/her experience in teaching the particular topics and impact is rated by considering how the specific topic is related to the general objective for the proposed activity. There is a frame type where the information associated to each of these types of material is stored. An example of a frame to model the subject matter is the one presented below. Here we find a collection of attributes identifying the entity "Activity" that stores knowledge of contents of a course to develop.

Node Type: [reading/solving/programming/modeling/simulation]
Code Activity: [n]
Description: [text]
Scoring: [integer]
Probability to answer to the activity since its content is already known
Probability to answer to the Activity since its contents is not known
Previous Knowledge: [integer]
Activity Type: [analogous/empiric]
Learning Style: RO, CE, AE, CA [text]
Guessing: [real], Impact: [real], Difficulty: [real], Oversight: [real]

In this frame, among other slots we can identify:

- *Learning Style*, this activity may favor a specific learning style more that others or a particular one (RO, CE, AE, CA); this slot will store knowledge for each entity and allow further down to adapt a specific strategy related to this information.

- Several slots are used to store the necessary information to generate an activity similar to Computerized Adaptive Tests (CAT).

In order to adapt this platform to a specific student a model of this particular student must be established. The student model may be separated into two parts; the first one being the psychosocial with the psychological and social features to be taken into account in the teaching-learning process; and the second part of the student model is the knowledge model that allows to establish which are the contents that the student is learning and the objectives that is accomplishing. Same with the didactical material the information related to the student model is stored within a structure of frames.

Semantic Networks. Relationship between contents used by this platform are made explicit through a semantic network with predefined terms (is, uses, is-part-of, etc.). In order to organize the didactic material the instructor must first define the concepts as a whole with which he/she will be building the supporting frame of knowledge. These concepts and their relationships will be the groundings from which the student will learn according to his/her profile and previous knowledge. Semantic networks are a promising way to model static knowledge as they allow identifying concepts and their relationships in a given course. In addition, this allows students to be aware of the previous concepts before carrying on with the next step.

Production System. The course is built by selecting or generating a set of necessary rules and objects in terms of specified knowledge. This can be accomplished by an expert (the instructor) through tables that allow him/her to establish the sequence of activities based on the students' profile.

Teaching strategies are carried out in order to induce changes to the student behavior through well structured processes involving the use of didactic materials. This kind of particular teaching strategy for a certain learning style of each student falls into a sequence of activities already defined in that particular didactic material. Selecting the most suitable sequence of activities is carried out by using a rule-based production system that incorporates its own pedagogical rules as such or from psycho-pedagogy. These rules are applied to the available structure knowledge under the frame format.

Neural Networks. Psychological and social variables of the student model are used by an artificial neural network previously trained to predict the students' probability of success or failure. This information is stored in a slot in the appropriate frame in order to be used at a later time by the production system that makes recommendations about the necessary activities required for success. This prediction supplies the necessary information for the different models of the student and the teaching strategy being able to split into two steps, first the one where the psychological features are encountered and second, the periodical that is carried out at the interaction process between the student and the platform [7].

Bayes Networks. The knowledge model of the student is updated through evaluations taken to the students using a Bayes net. The Bayes net is used to assess a degree of knowledge accomplished by a student in order to recommend the most adequate activities from the actual level of knowledge learning style and the student profile.

Diagnosis of the learning level of a student and the sequence of activities that are to be used for a particular student's profile are predicted through the probability theory using Bayes nets. The net consists of four levels or layers; the first layer contains the general objectives for the course; in the second are the specific objectives for the course, i.e., groups of contents aimed to particular purposes, that are a decomposition for the general objective. In the third layer, the contents and the minimal units for knowledge decomposition are located. Finally, in the fourth layer the activities for each content and for each learning style are located. These activities are prepared to generate knowledge: the instructor needs to specify a difficulty level, and an impact ratio for each activity. The platform needs this information to compute the conditioned probabilities that relate learning degree and activities to be accomplished, by using Item Response Theory [5, 8].

On the other hand it is possible to periodically diagnose through diverse techniques such as portfolios or computerized adaptive tests. The later, that uses Bayes nets allows choosing the best activity or questioning the needs of the student for knowledge.

CommonKADS. As a general methodology to model in a structured manner the knowledge of tasks, methods, inferences, rolls and concepts of expertise in each of the phases, necessary for the global distance teaching/learning function. This is a method used for analysis and organization of knowledge-based systems in an analogous way to the methods used in software engineering [2].

5 The MISTRAL Platform for a Course on Object Oriented Analysis and Design

A specific course has been developed, *Object oriented Analysis and Design* (OOAD) for students in a third year engineering, using this platform. The course structure adapts to particular student's features such as the previous knowledge in programming languages and modeling tools. The instructor has decided to offer the capability for choosing the language to use in the course or to begin with a specific programming language and choose the modeling tool (Rational Rose, Microsoft Visual Modeler, GDPro, etc.). The OOAD is a course developed in a semi-present modality, so those students can achieve the objectives in restricted deadlines but according to their own learning capabilities.

Determining the general objective for the OOAD course. The objective for the OOAD course was established in the following manner: *Understanding the concepts of analysis, methodologies, design and programming pointing to objects, in order to apply this knowledge to the analysis software of systems development and using through the UML (Unified Modeling Language) and JAVA language*. From the preceding statement we conclude that for this particular case the possibility of choosing a programming language to use was not taken into account. Following the general objective it is necessary to specify the main objectives with a level of relevance in order to accomplish the general objective.

Determining specific objectives. The process associated to a specific objective finishes when the learning level of this objective is 100% accomplished since the instructor established that each objective must be satisfied completely. Thus, although the *relevance* to each objective may appear not to be important, the indexing of this factor was maintained to allow the student be aware of his/her progress. Once specific objectives are settled it is necessary to determine the contents that the student must learn in order to accomplish each one of them. Contents for the first objective are shown:

Contents	Relevance (%)
Objective 1	
1 Software quality	
1.1 Introduction	20
1.2 Software quality	50
1.3 Modularity, coupling, cohesion	30

Determining activities. This phase was the most difficult in knowledge acquisition. Even though a great variety of activities are used by different instructors it is necessary that a reduced number of activities for each content be worked out within a great amount o time frame in order to establish certain activities for different learning styles and also determine the appropriate parameters. These parameters have been explained in [4] ant its meaning are as follows:

LS: Learning Style (figure 1)

IR: Impact Ratio (one of the following values: 0.2, 1.2, 2)

D: Difficulty Level (1 to 10)

G: Guess Factor (0 to 100 %)

(Texts, exercises an examples referred to in the following table can be consulted in http://152.74.11.209/pide/, a course demo). Some of the activities associated to the content *Software Quality* are shown as follows:

Cont.	Activities	LS	G	IR	D
1	Software Quality				
1.1	Reading the text *1.1 Introduction.* Underline in the text whatever you consider more important and place it in your files folder as Act1-1	RO	0	0.2	1
	...				
1.1	Search in the Internet what conferences or seminars are at this time fully dedicated to object orientation, describe the subjects considered. Place this activity in your file folder named Act1-4	AE	10	1.2	3
	...				

In summary, it is necessary to structure and classify lots of activities that from the instructor's point of view and the Internet sources become important when the OOAD course is learned in order to meet proposed goals for each objective and different students. This course was improved incorporating new activities not considered in previous courses. This is the result of considering different learning styles and different teaching strategies. The strategy selection is done by the production system. Each activity created must be evaluated; therefore the simplest planned activity will be useful for the students. This evaluation is accomplished using Bayes nets. The platform allows to generate complementary activities automatically which are suggested through the neural net.

6 Conclusions

In this paper we have presented how the use of techniques of artificial intelligence can influence the teaching learning process when a platform for distance education is used. The use of neural nets as a mean for prediction related to the potential success in specific learning matters, proved to be an adequate mechanism considering that were also used to predict more appropriate activities to accomplish in case the students presented a sign of failure. The use of Bayes nets allowed automating the student range knowledge degree in the evaluation process thus different student profiles lead to a sequence of personalized activities that suit each particular learning profile, considering the previous knowledge acquired by the student. The instructor may verify through his experience whether it is possible to reach a real personalized education using Internet or otherwise. Private discussion groups put students of different learning styles together. This aim strengthens these different styles so that successful strategies can be imitated. The ability to store different kinds of work in the portfolio allows the student and the instructor to check the learning process progress that constitutes an additional communication mechanism among the actors.

Acknowledgments. This work is supported by Project DIUC 205.164.001-1.0, Universidad de Concepción.

References

1. P. Brusilovsky. Methods and Techniques of adaptive hypermedia. In *Hypertext and Hypermedia. P. Brusilovsky and A. Kobsa and J. Vassileva (Eds)*, pages 1–43. Dordrecht: Kluwer Academic Publishers, 1998.
2. G. Schreiber; H. Akkermans; A. Anjewierden; R. De Hoog; N. Shadbolt; W. Van de Welde and B. Wielinga. *Knowledge Engineering and Management: The CommonKADS Methodology.* The MIT Press, Cambridge, Mass., 1999.
3. D. Kolb. *The Learning Style Inventory.* Boston, MA. Hay Resources Direct, 2005.
4. F. Lord. *Application of Item Response Theory to Practical Testing Problems.* Hillside, N.J.: Earlbaum, 1980.
5. E. Millán. Thesis: Bayesian system for student modeling. *AI Commun.*, 13(4):277–278, 2000.
6. S. Papert. *Mindstorms: Children, Computers and Powerful Ideas.* New York Basic Books, 1980.
7. P. Salcedo, M.A. Pinninghoff, and R. Contreras. Mistral: A knowledge-based system for distance education that incorporates neural networks techniques for teaching decisions. *Lecture Notes in Computer Science, 2687, Springer*, 2003.
8. P. Salcedo, M.A. Pinninghoff, and R. Contreras. Computerized adaptive tests and item response theory on a distance education platform. *Lecture Notes in Computer Science, 3562, Springer*, 2005.
9. R Schank and C. Cleary. *Engines for Education.* Lawrence Erlbaum Associates. Hillsdale, N.J., 1995.
10. D. Sleeman and J. S. Brown. *Intelligent Tutoring Systems.* New York: Academic Press, 1982.
11. E. Wenger. *Artificial Intelligence and Tutoring Systems.* Los Altos: Morgan and Kaufmann, 1987.

Dynamic Systems

Lessons Learned in Modeling Dynamic Systems using Genetic Programming

Juan J. Flores and Mario Graff

División de Estudios de Posgrado
Facultad de Ingeniería Eléctrica
Universidad Michoacana
Mexico

Abstract. This article describes a set of experiences in modeling dynamic systems using Genetic Programming and Gene Expression Programming. We have made experiments in linear systems, non-linear systems and times series. The experiments in linear systems include linear pendulum, coupled mass-spring, electrical circuits, etc. In non-linear systems we have modeled the Van der Pol Equation, non-linear pendulum with friction, etc. The models have been represented as an ordinary differential equation, system of ordinary differential equations, and using a time series approach.

1 Introduction

Modeling a dynamic system is a process that generally has been made by an engineer. This work describes the ability of Genetic Programming (GP) [6] and Gene Expression Programming (GEP) [2] to model a dynamic system.

It is presented a set of examples where GP and GEP have found good models. Examples included in this work are: a mass spring shock absorber, a coupled mass-spring system, the Van der Pol Equation, a non-linear pendulum with friction, and a wind time series.

This article shows three different ways of representing a model in GP and GEP. It is used an ordinary differential equation, a system of ordinary differential equations and a time series approach.

Section 2 is an introduction to system identification. Section 3 shows the representation of a dynamic system in GP. Section 4 is the experiments scenarios. Section 5 presents examples in of linear systems. Section 6 shows examples of non-linear systems. Section 7 presents an example of time series. Section 8 presents a summary of the obtained results. Section 9 presents the conclusions.

2 System Identification

System identification (SID) is the process of deriving a mathematical model from the internal dynamics of a system of observations of its outputs. Modeling is the essential first step in a variety of engineering problems. For example, if an

Advances in Artificial Intelligence.
-Research in Computing Science 26, 2006, pp.177-186

Received 14/07/06
Accepted 03/10/06
Final version 13/10/06

engineer is going to control a dc-motor, first he needs to model it. The model is made by applying a given input to the system and observing its behavior. From this behavior, a mathematical model is built and tested against the dc-motor. Based on the obtained mathematical model, the controller is made.

Linear system identification methods have been widely studied (see [7]). However, these methods involve a complicated process that usually can only be followed by an expert. Nonlinear system identification remains a difficult task, because frequently there is not enough information about the system (i.e. the structure system's is unknown).

3 Representation of Dynamic Systems

In order to make the experiments, a computer program called ECSID (Evolutionary Computation based System Identification) was implemented. ECSID uses GP or GEP to build a mathematical model from observed data. ECSID has three different ways to represent a model; it can represent the model as an ODE (Ordinary Differential Equation), a System of ODEs, or using a time series approach.

All the models represented as ODE have the general form of Equation 1.

$$x^{(n)} = f(t, x, x', x'', \ldots, x^{(n-1)}) \tag{1}$$

ECSID evolves only the right part of Equation 1. The order of the system is determined by the order of the highest order element whose coefficient is not zero. Figure 1 shows an ODE represented in ECSID. The ode represented in the Figure is $\frac{d^2x}{d^2t} = 7\frac{dx}{dt} + 10x + 12$. We can observe that this system is a second order system, because it has a first order element.

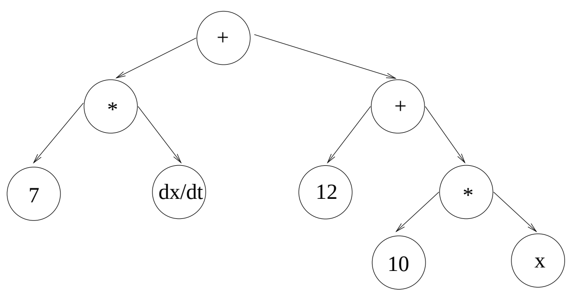


Fig. 1. ODE represented in ECSID.

In order to get the behavior of the system, Equation 1 is integrated. First this equation is transformed to Equation 2 and a 4th order Runge-Kutta method is used to integrate the latter.

$$\begin{cases} y'_1 &= y_2 \\ y'_2 &= y_3 \\ \vdots \\ y_n &= f(t, y_1, y_2, \cdots, y_n) \end{cases} \tag{2}$$

Equation 2 is formed by replacing the following variables $y_1 = y, y_2 = y', y_3 = y'', \cdots, y_n = y^{(n-1)}$.

ECSID can evolve a system of ODEs. It uses a multi-gene chromosome to represent the system, where each gene has the same structure of the right part of Equation 1. An operation similar to crossover was implemented; it is called gene-recombination. Gene-recombination receives two individuals and randomly chooses one gene from each individual and swaps them.

ECSID can also represent its models using a time series approach, where the output is modeled as a function of past values of the input(s) and output(s). Equation 3 shows this approach where k is the current time and τ is the maximum time shift.

$$y_k = f(y_{1,k-1}, \ldots, y_{1,k-\tau}, y_{n,k-1}, \ldots, y_{n,k-\tau}) \tag{3}$$

Another characteristic of ECSID is that it can evolve linear systems. In order to evolve linear systems we need to disable product and division. Product and division can receive any s-expression and multiply or divide these s-expressions. For example, product can give an individual whose form is $\frac{dx}{dt} = x^2$ which is non-linear. An operation called "coefficient" is implemented to replace product. Coefficient receives any s-expression and a constant, and multiplies the constant with the s-expression. It was also implemented a function that punishes individuals that are non-linear. One way of punishing a non-linear individual is to set its fitness to its original fitness plus the average fitness of the population.

4 Experiments Scenarios

In order to compare experiments from different domains, we use the correlation coefficient (Equation 4). The correlation coefficient gives a number between -1 and 1 where 1 means that the curves are equal.

$$r = \frac{n\sum xy - \sum x \sum y}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \tag{4}$$

ECSID uses the standard evolutionary computation procedure, random initial population, fitness proportional selection and elitism. The fitness function is the absolute difference of the errors $\Sigma|e|$. Each experiment had a population of 500 individuals and was run for 500 generations The termination criteria is met when the correlation coefficient is ≥ 0.99. All models presented here are selected from the best individuals of 20 independent runs. Table 1 shows the parameters used in ECSID .

Genetic Operator	Probability
Mutation	0.2
Crossover	0.8
is-transposition	0.1
ris-transposition	0.1
gene-transposition	0.1
one-point recombination	0.3
two-point recombination	0.3
GP and GEP gene-recombination	0.1

Table 1. Genetic operators' parameters

5 Linear Systems

There are a lot of problem that are modeled as linear systems. E.g. Electrical Circuits, Electrical Machines, Mechanical Systems, etc. If it is known that the system is linear, it is important that ECSID finds a linear model. A system is linear if it can be represented as Equation 5.

$$a_n(t)\frac{d^n x}{dt^n} + \ldots + a_1(t)\frac{dx}{dt} + a_0(t)x = g(t) \tag{5}$$

In order to obtain a linear model, we use two methods: the first one uses the operation "coefficient" and disables the operations "product" and "division"; the second one uses a function that punishes those individuals that are non-linear. The function set used to obtain the following two examples is $\{+, -, *, /\}$ and the terminal set is $\{x, \frac{dx}{dt}, \Re\}$. The symbol $\Re$ means a random floating point number in the range $[0, 100]$. In the following example a function that punishes those individuals that are non-linear is used.

The first example is a mass spring shock absorber. Equation 6 shows its model. Its initial conditions are $x(0) = 1$ and $x'(0) = 0$.

$$\frac{d^2x}{dt^2} = -\frac{dx}{dt} - 3x \tag{6}$$

GP and GEP evolved the same equation that Equation 6. Figure 2 shows the graphics of Equation 6.

In the second example we model a coupled mass-spring system. Equation 7 shows this system. The initial conditions for the system are $x(0) = 1, x'(0) = 0, y' = 2, y'(0) = 0$.

$$\begin{aligned} \frac{d^2x}{dt^2} &= -5x + 2y \\ \frac{d^2y}{dt^2} &= 2x - 2y \end{aligned} \tag{7}$$

In this example GEP and GP obtained the same model. Equation 8 shows this model. It can be observed that the obtained model is linear.

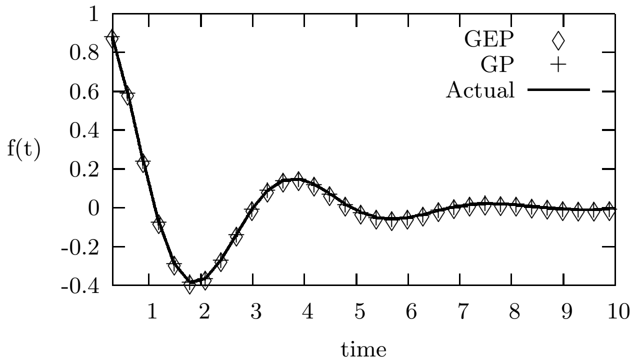


Fig. 2. Mass Spring Shock-absorber

$$
\begin{aligned}
\frac{d^2x}{dt^2} &= -x \\
\frac{d^2y}{dt^2} &= -y
\end{aligned} \tag{8}
$$

Figure 3 shows the behavior exhibited by the obtained models and the real system (there are four graphics in the figure because each model has two equations).

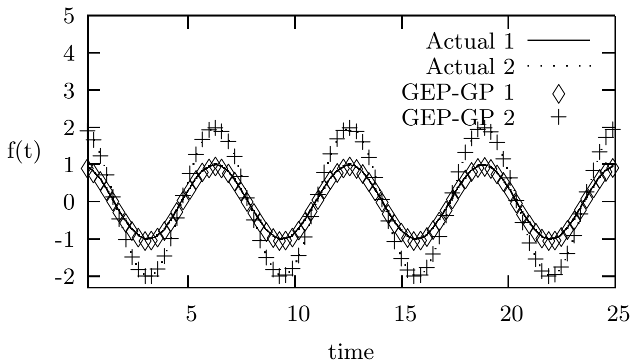


Fig. 3. Mass springs coupled.

ECSID has obtained models for a linear pendulum, an electric circuit (two branches), and a DC machine; all of models obtained exhibit a similar behavior that the ones presented in this section. All the models that have been obtained have a similar structure than the real models. Therefore this methodology builds good models and also the models obtained can be understood by an engineer.

Hinchliffe [4] states that it is unlikely to evolve models that can provide any insight into the underlying physical processes of a dynamic system. In all experiments done so far we have found models that have a similar structure than the real models. Therefore this procedure can produce not only good models but also models that provide some insight into the underlying physical processes.

The results obtained using both methods ("coefficient" and punish function) were comparable.

We have not found any work that can evolve linear systems.

6 Non-linear Systems

Non-linear systems are those systems that cannot be represented using Equation 5. These kind of systems are very important; e.g. mechanical systems, chemical processes, electrical circuits, etc.

This example evolves the Van der Pol Equation, which is shown in Eq. 9. The experiment was done with initial conditions $x(0) = 1.5$ and $x'(0) = 0$. The function set is $\{+, -, *, /\}$ and the terminal set is $\{\frac{dx}{dt}, x, \Re\}$

$$\frac{d^2x}{dt} = (1 - x^2)\frac{dx}{dt} - x \tag{9}$$

Both GEP and GP obtained the same model that Equation 9. Figure 4 shows the real model and the obtained by GEP and GP.

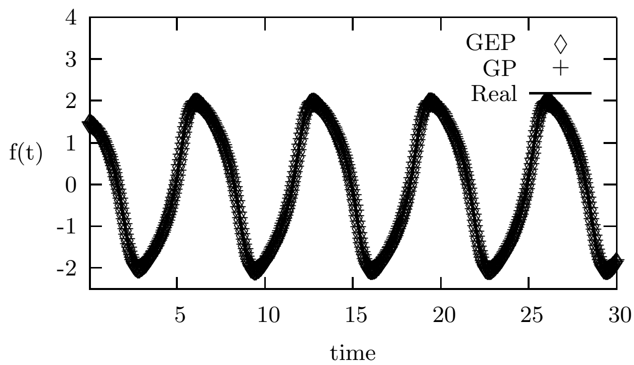


Fig. 4. Van der Pol Equation

The next example is a non-linear pendulum with friction; equation 10 non-linear pendulum shows the model. The initial conditions for the experiment are $\theta(0) = 1$ and $\theta'(0) = 0$, the function set is $\{+, -, *, /, \sin, \cos, \exp\}$ and the terminal set is $\{\frac{d\theta}{dt}, \theta, \Re\}$.

$$\frac{d^2\theta}{d^2t} = -2\frac{d\theta}{dt} - 19.6sin(\theta) \tag{10}$$

Equations 11 and 12 show the result using GEP and GP, respectively. It is observed that the models obtained are linear but it is not implemented any restriction about non-linearity. Figure 5 shows the behavior exhibited by those models.

$$\frac{d^2\theta}{d^2t} = -2\theta\frac{d\theta}{dt} - 2\frac{d\theta}{dt} - 20\theta \tag{11}$$

$$\frac{d^2\theta}{d^2t} = -2.3048492976\frac{d\theta}{dt} - 19.34372929760\theta \tag{12}$$

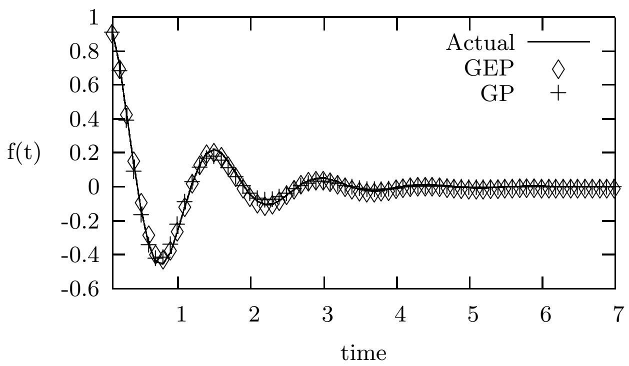


Fig. 5. Pendulum with friction.

Gray et al [3], Weinbrenner [11] and Cao et al [1] use a similar procedure to evolve non-linear systems. All of them represent the system as an ODE. Their procedure uses GP to find the structure of the system and another one to find the parameters of the system.

In all the experiments done we have not found evidence that it is necessary to use a different procedure to optimize the parameters of the ODE. We have found that GP does a good job in finding a model that behaves close to the observed data. Furthermore all the models obtained are simple enough to be understood by an engineer.

7 Time Series Prediction

This example models a wind time series using a slide window prediction method. In order to compare the obtained model, we developed an ARIMA model (see

[8]) using the standard ARIMA procedure. This model was obtained using the software Minitab [9]. Equation 13 shows the model obtained using the ARIMA procedure. Where e means the prediction errors.

$$
\begin{aligned}
f(n) = {} & f(n-1) + f(n-2) - f(n-13) - 0.997e(n-1) \\
& - 0.7976e(n-12) + 0.7956e(n-13)
\end{aligned} \tag{13}
$$

The function set is $\{+, -, *, /, \sin, \cos, \log\}$ and the terminal set is $\{y_{n-1}, \ldots, y_{n-16}, \Re\}$ Equation 14 shows the best model. Figure 6 shows the time series and the obtained model.

$$
\begin{aligned}
f(n) = {} & 0.1408 \cos(\frac{\sin(\ln(f(n-12)))}{\sin(f(n-14))} + f(n-2)) \\
& + \cos(\frac{8.5209}{\frac{6.8594}{\ln(\ln(f(n-3)))} - 5.6555} + f(n-12) - 6.8962) \\
& + \frac{(0.7323f(n-12) + 4.1514)\cos(2\sin(f(n-12)))}{\frac{f(n-9)}{f(n-14)} - f(n-2)} \\
& + 0.5986f(n-12)
\end{aligned} \tag{14}
$$

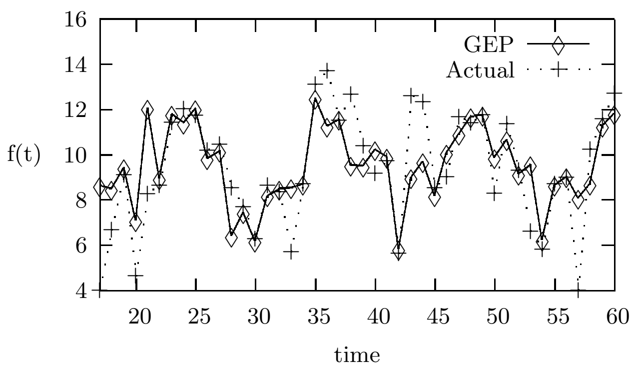


Fig. 6. Wind Time Series

The correlation coefficient of Equation 13 is $r = 0.51933$ meanwhile for Equation 14 is $r = 0.7313$. It is clear that GP found a better model than the one obtained by the ARIMA procedure.

Lie et al [5], and Szpiro [10] use a slide window prediction method (Equation 3) to model a time series. Hinchliffe [4] used this method to find a model of a dynamic system.

8 Results

Table 2 presents a summary of the results of the experiments. ECSID found good models in all the the experiments represented as ODEs and slide window regressors. In the last experiment the model found is no as good as the others but the observed data is more complex.

Method	Problem	r
GEP	Mass Spring Shock-absorber (Eq. 6)	1
GP	Mass Spring Shock-absorber (Eq. 6)	1
GEP	Coupled mass-spring (Eq. 8)	1
GP	Coupled mass-spring (Eq. 8)	1
GEP	Van der Pol (Eq. 9)	1
GP	Van der Pol (Eq. 9)	1
GEP	Pendulum with friction (Eq. 11)	0.9982
GP	Pendulum with friction (Eq. 12)	0.9969
GEP	Wind Prediction (Eq. 14)	0.7313

Table 2. Results

Table 3 shows the computational resources needed to find a good model with a probability of 0.99 (see [2] Chapter 8). This information was acquired experimentally. Each experiment was run 20 times. $I(M, i, z)$ is the number of individuals that needs to be processed in order to obtain a model with a correlation coefficient ≥ 0.99. The last column is maximum number of generations needed to set a correlation coefficient ≥ 0.99.

Column $I(M, i, z)$ and *Gen.* give us an idea of the problem complexity and the computational resources needed by GP or GEP. It is observed that the most complex problem is the "Pendulum with friction" for both cases and the simplest experiment is the "Coupled mass-spring" again for both cases.

From this table we can say that GP generally needs to process less individuals than GEP, therefore it also needs less generations. Ferreira [2] states that GEP is better than GP. We have not found any evidence that supports her assertion. Another characteristic that can be inferred from this table is the number of generations needed, we can say that 100 generations are acceptable.

The wind time series model is not included in this table because there is not information to calculate $I(M, i, z)$, instead an ARIMA model was used to compare the GP model.

9 Conclusions

In this work we have found experimentally that GP is better than GEP because it processes less individuals to obtain comparable results. ECSID has found good models and those models can provide an insight into the underlying physical process of a dynamic system.

Method	Problem	$I(M, i, z)$	Gen.
GEP	Mass Spring Shock-absorber	$110,162$	77
GP	Mass Spring Shock-absorber	$9,000$	18
GEP	Coupled mass-spring	$41,275$	4
GP	Coupled mass-spring	$42,760$	8
GEP	Van der Pol	$2,872,995$	64
GP	Van der Pol	$197,769$	37
GEP	Pendulum with friction	$262,011$	137
GP	Pendulum with friction	$48,000$	48

Table 3. Computational effort

We provide the reader with a set of experiences that show evidence about what results a scientist or engineer might expect from using GP for modeling Dynamic Systems.

ECSID is a free software, written in Lisp, and can be downloaded from http://sf.net/projects/ecsid.

References

1. H. Cao, L. Kang, Y. Chen, and J. Yu. Evolutionary modeling of systems of ordinary differential equations with genetic programming. *Genetic Programming and Evolvable Machines*, 1(4):309–337, Oct. 2000.
2. C. Ferreira. Gene expression programming: A new adaptive algorithm for solving problems. In *Complex Systems*, number 2, pages 87–129, 2001.
3. G. J. Gray, D. J. Murray-Smith, Y. Li, and K. C. Sharman. Nonlinear model structure identification using genetic programming. In J. R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1996 Conference Stanford University July 28-31, 1996*, pages 32–37, Stanford University, CA, USA, 28–31 1996. Stanford Bookstore.
4. M. Hinchliffe. *Dynamic Modelling Using Genetic Programming.* PhD thesis, University of Newcastle upon Type, 2001.
5. Z. Jie, T. Changjie, L. Chuan, C. Anlong, and Y. Chang'an. Time series prediction based on gene expression programming. International Conference for Web Information (WAIM04), Springer Verlag, 2004.
6. J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems).* The MIT Press, 1992.
7. L. S. Ljung. *System Identification: Theory for the User.* Prentice Hall, 1987.
8. S. Makridakis, S. C. Whellwright, and R. J. Hyndamn. *Forecasting Methods and Applications.* John Wiley & Sons, Inc., 3 edition, 1992.
9. Minitab. http://www.minitab.com/.
10. G. G. Szpiro. Forecasting chaotic time series with genetic algorithms. Physical Review E., 1997.
11. T. Weinbrenner. Genetic programming techniques applied to measurement data. Diploma Thesis, 1997.

Author Index

Índice de autores

Editorial Board of the Volume

Comité editorial de volumen

Additional Reviewers

Árbitros adicionales

Mohamed Abdel Fattah
Juan Carlos Acosta
Marco Alberti
José Arrazola
Miguel A. Alonso
Hector Hugo Avilés
J. C. Acosta Guadarrama
Ernesto Luiz Andrade Neto
Prasanna Balaprakash
Alejandra Barrera
María Lucía Barrón Estrada
Tristan Marc Behrens
Edgard I. Benítez Guerrero
Daniel Le Berre
Francesca Bertagna
Viktor de Boer
Dario Bottazzi
Olivier Buffet
Nils Bulling
Davide Buscaldi
Felix Calderon Solorio
Jose Antonio Calderón Martínez
Francesco Calimeri
Antonio Camarena Ibarrola
Sergio D. Cano Ortiz
Francisco J. Cantu Ortiz
Carlos Cares
Giuseppe Carignani
Jesús Ariel Carrasco Ochoa
Tommaso Caselli
María José Castro Bleda
Debrup Chakraborty
Zenon Chaczko
Federico Chesani
Marco Chiarandini
George Coghill
Miguel Contreras
Sylvie Coste Marquis
Juan Jose del Coz
Nareli Cruz Cortés
Víctor Manuel Darriba Bilbao
Michael Dekhtyar
Jorge Díez
Héctor Díez
Luigi Dragone
Florence Dupin de St Cyr
Gemma Bel Enguix
Arturo Espinosa
Bruno Feres de Souza
Antonio J. Fernández
Óscar Ferrández
Reynaldo Félix
Leticia Flores Pulido
Per-Erik Forssen
José E. Gallardo
Juan Manuel García
Mario García
Miguel García
Luciano García Bañuelos
Karen A. García Gamboa
Paola García Perera
Luca Di Gaspero
Marco Gavanelli
Ma. del Pilar Gómez
Jesús A. González
Miguel González Mendoza
Fernando Godínez
Mario Graff
Carlos Guillen Galván
Scott Helmer
Adrian Horia Dediu
Marc Hull
Luca Iocchi
Wojtek Jamroga
Dolores Jiménez López
Sophia Katrenko
Ingrid Kirschning A.
Sébastien Konieczny
Jerzy Kotowski
Evelina Lamma
Ricardo Landa Becerra
João Leite
Domenico Lembo
Andrzej Lingas
Miguel López
Alejandra López
Juan Carlos López Pimentel
Oscar Luaces

Ana Carolina Lorena
Fernando Magán Muñoz
Michael Maher
Marco Maratea
Alejandro Marcus
Patricio Martínez Barco
Francesco Masulli
Stefano Mattoccia
Yana Maximova
Bertrand Mazure
Patricia Melin
Paola Mello
Marco Mesiti
Efrén Mezura Montes
Mónica Monachini
Marco Montali
Manuel Montes
Marco A. Montes de Oca
Oscar Montiel
Jaime Mora Vargas
Eduardo Morales
Emmanuel Morin
Rafael Muñoz
Rafael Murrieta Cid
Gonzalo Navarro
Niels Netten
Pascal Nicolas
Juan Carlos Nieves
Peter Novak
Slawomir Nowaczyk
Kenji Okuma
Alberto Oliart
Oscar Olmedo Aguirre
Ivan Olmos Pined
Magdalena Ortíz
Mauricio Osorio
Odile Papini
Isaac Parra Ramirez
Nelio Pastor Gómez
Bernhard Peischl
Marco Otilio Peña Díaz
Carlos Mex Perera
Carlos F. Pfeiffer-Celaya
David Pinto
José Ramón Quevedo
J. Federico Ramírez Cruz
Jorge Adolfo Ramírez Uresti
Raúl V. Ramírez Velarde
Orion F. Reyes Galaviz
Francisco José Ribadas Pena
Fabrizio Riguzzi
Homero V. Rios-Figueroa
Norma F. Roffe Samaniego
Leonardo Romero Muñoz
Roberto Rossi
Olivier Roussel
Stefano Rovetta
Robert Santana
Estela Saquete
Roberto Sepúlveda
Jasper Snoek
Alejandro Sobrino
Volker Sorge
Claudia Soria
Tristram Southey
Oleg Starostenko
Gerald Steinbauer
Armando Suárez Cueto
Ewa Szlachcic
Gian Diego Tipaldi
Paolo Torroni
Gregorio Toscano Pulido
Mars Valiev
Fernando Velasco
Vincent Vidal
Mario A. Villalobos-Arias
Julia Vogel
Christel Vrain
Jörg Weber
Joseph N. Wilson
Bob Woodham
Pinar Yolum
Gabriel Zachmann
Ramón Zatarain Cabada
Claudia Zepeda
Yingqian Zhang

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2006.
Printing 500 / Edición 500 ejemplares.