ISSN : 2165-4069(Online)
ISSN : 2165-4050(Print)

IJARAI

International Journal of Advanced Research in Artificial Intelligence

Volume 1 Issue 7

www.ijarai.thesai.org

A Publication of
The Science and Information Organization

IJARAI

INTERNATIONAL JOURNAL OF
ADVANCED RESEARCH IN ARTIFICIAL INTELLIGENCE

SAI

THE SCIENCE AND INFORMATION ORGANIZATION

www.thesai.org | info@thesai.org

Editorial Preface

From the Desk of Managing Editor…

"The question of whether computers can think is like the question of whether submarines can swim." ― Edsger W. Dijkstra, the quote explains the power of Artificial Intelligence in computers with the changing landscape. The renaissance stimulated by the field of Artificial Intelligence is generating multiple formats and channels of creativity and innovation.

This journal is a special track on Artificial Intelligence by The Science and Information Organization and aims to be a leading forum for engineers, researchers and practitioners throughout the world.

The journal reports results achieved; proposals for new ways of looking at AI problems and include demonstrations of effectiveness. Papers describing existing technologies or algorithms integrating multiple systems are welcomed. IJARAI also invites papers on real life applications, which should describe the current scenarios, proposed solution, emphasize its novelty, and present an in-depth evaluation of the AI techniques being exploited. IJARAI focusses on quality and relevance in its publications.

In addition, IJARAI recognizes the importance of international influences on Artificial Intelligence and seeks international input in all aspects of the journal, including content, authorship of papers, readership, paper reviewers, and Editorial Board membership.

The success of authors and the journal is interdependent. While the Journal is in its initial phase, it is not only the Editor whose work is crucial to producing the journal. The editorial board members , the peer reviewers, scholars around the world who assess submissions, students, and institutions who generously give their expertise in factors small and large— their constant encouragement has helped a lot in the progress of the journal and shall help in future to earn credibility amongst all the reader members.

I add a personal thanks to the whole team that has catalysed so much, and I wish everyone who has been connected with the Journal the very best for the future.

Thank you for Sharing Wisdom!

Managing Editor
IJARAI
Volume 1 Issue 7 October 2012
ISSN: 2165-4069(Online)
ISSN: 2165-4050(Print)

Editorial Board

Reviewer Board Members

CONTENTS

Evacuation Path Selection for Firefighters Based on Dynamic Triangular Network Model

Qianyi Zhang
College of Information Science and Technology, Donghua University
Shanghai, china

Demin Li
College of Information Science and Technology, Donghua University
Shanghai, china

Yajuan An
College of Information Science and Technology, Donghua University
Shanghai, china

***Abstract*— Path selection is one of the critical aspects in emergency evacuation. In a fire scene, how to choose an optimal evacuation path for firefighters is a challenging aspect. In this paper, firstly, a dynamic triangular network model formed by robots is presented. On the basis of this model, directed graph is established in order to calculate direct paths. Then multi-parameter information fusion which includes smoke density, temperature and oxygen density is discussed in detail for environment safety evaluation. Based on the discussions, a new way has been proposed for optimal path selection, taking into consideration the safety-factor of the path. The objectives of the method are to minimize the path lengths, at the same time, to protect firefighters from the dangerous regions. In the end, numerical simulation results prove the feasibility and superiority of this method.**

Keywords- Evacuation Path Selection; Information Fusion; Dynamic Triangular Network Model.

I. INTRODUCTION

In recent years, frequent fire disasters have brought great casualties and huge property loss. Firefighter becomes a high-risk profession. When firefighters finish the search and rescue missions, the situation of the fire scene usually gets worse, meanwhile, the oxygen reserves the firefighters carried may become exhausted. Therefore the firefighters must evacuate to safety areas as soon as possible. Due to the complex, variable and uncertain situation, it is difficult for firefighters to orientate themselves and find a suitable path to the exit. How to choose an optimal evacuation path for firefighters is an important and challenging problem.

There are various researches regarding the problem of path selection in emergency situations. Marina Yusoff provides a good review on the mathematical algorithm and model of evacuation problems [1]. The ant colony optimization algorithm is applied to find the shortest path for emergence evacuation [2-4]. Tianyu Wang established a fire evacuation system and combined it with the Dijkstra Algorithm [5]. A multi-objective optimization model based genetic algorithm is adopted to solve the proposed evacuation routing problem [6]. A multi-objective approach is presented to identify evacuation paths and the location of shelters for urban evacuation planning [7].

By now, most researches only take the shortest path as the objective. However, the optimal path does not necessarily mean the shortest path. It should also take the path safety into consideration. Ulf Witkowski[8] put forward a dynamic triangular ad-hoc network formed by multi-robot systems to provide robust communication links for human and robots in a fire scene. Base on the dynamic triangular network model, Demin Li, etc. [9] proposed a method for establishing smoke density gradient based directed graph and selecting the shortest directed path for firefighters, but did not take the multiple parameter information fusion into account.

In this paper, we establish a new source-destination based directed graph in the dynamic triangular network model. Besides, Smoke density, temperature and oxygen density are considered in the information fusion for environment safety evaluation. Our objective is to minimize the path length, meanwhile, to make assure the firefighters keep away from the dangerous regions.

The paper is organized as follows: In section II, a dynamic triangular network model formed by multi-robots is established in a fire scene. In section III, the source-destination directed graph in the dynamic triangular network model is presented and described in detail. In section IV, firstly, multiple parameters consisting of temperature, smoke density and oxygen density are considered in information fusion process. The results of fusion are used to classify the environment into three safety levels. Secondly, based on the discussion all above, a new path selection method with safety-factor considered is proposed and verified. The last section makes the conclusion.

II. DYNAMIC TRIANGULAR NETWORK MODEL

The modern buildings tend to be large-scale and complex at present. When a fire happens, all communication patterns that rely on infrastructures might be useless and cannot supply enough surrounding information for firefighters. Firefighters usually equip with multiple sensors and could get real-time information.

However, some regions are too dangerous for human to approach, the information collection is difficult. Paper [8] proposes a dynamic triangular method for robots distribution in a fire scene. All the robots are equipped with sensors to collect surrounding information and position data. The main purpose of the method is to deploy robots in a way with the largest coverage for facilitate communication and environment exploration.

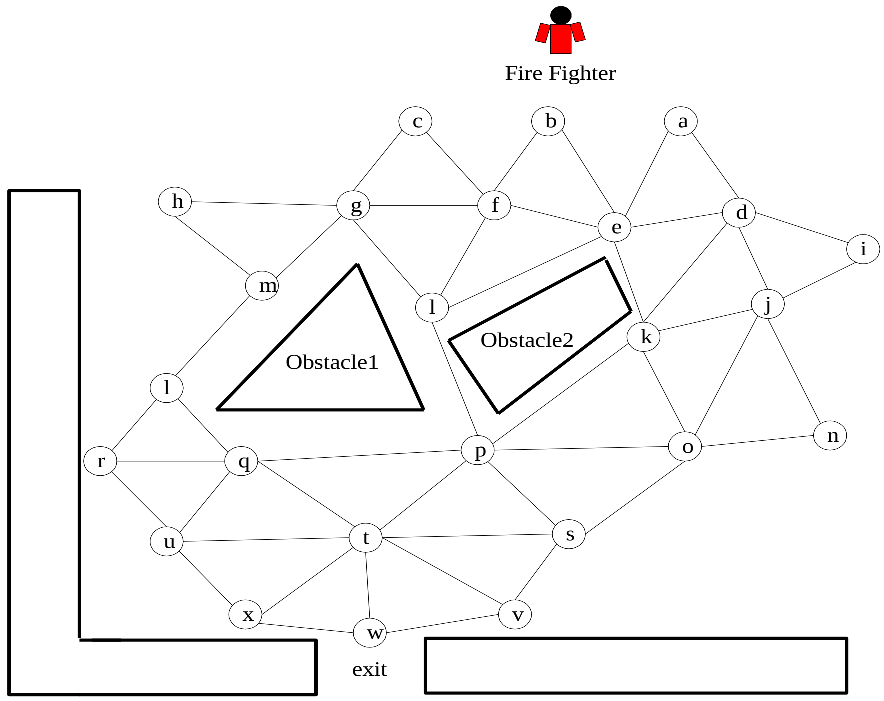


Figure.1. Dynamic triangular network model graph [9]

First of all, an ad-hoc network of robots needs to be built up. Ulf Witkowski provides a dynamic triangular network schema [8], based on this scheme, Demin Li, etc. proposed the dynamic triangular network model [9], it is shown in Figure 1.

The Figure 1 shows an environment covered by the ad-hoc network of robots using dynamic triangular network model. Robots act as the nodes of the network. Robots are represented as circles, marked by English letters. The communication links of the robots are indicated by solid line segments. The thicker solid lines in the figure represent the obstacles. The dynamic triangulation method provides positioning of beacons used as reference points at the vertices of equilateral or nearly equilateral triangles.

Some robots can be placed as beacons at the important position such as entrances, doors, corners and so on. Other robots might be distributed as uniformly as possible in order to gather reliable information about the environment. The robots will form a partition of the environment, separating it in regions, which will represent a triangulation in the absence of obstacles.

As shown in Figure 1, a fire fighter is going to evacuate to the exit. There are obstacles in the building, so it is not safe for him to run directly to the exit. Besides, the smoke and flames are also great threat to him. Since obstacles are surrounded by robots, and every robot could get the information of neighboring environment, it is wise to evacuate along the connection line between the robots. In this way, firefighters could avoid obstacles and high-risk region.

III. ESTABLISHING DIRECTED GRAPH BASED ON DYNAMIC TRIANGULAR NETWORK MODEL

To make the optimal path selection, firstly, directed graph in the dynamic triangular network model must be set up, so that all the feasible direct paths could be found out. The sub graph Figure 2 of Figure 1 shows 7 nodes in the network including the exit.

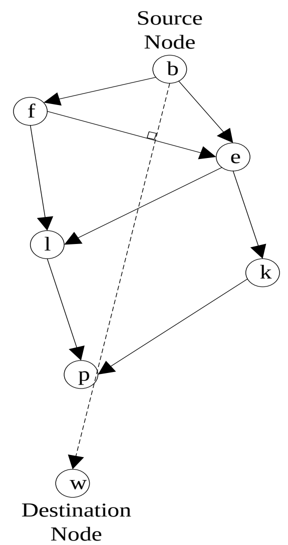
Node Ri is defined as a two dimension coordinate (Xi, Yi). Xi and Yi are horizontal coordinate and vertical coordinate of the node, respectively. Suppose Rj is the neighbor node of Ri, Rs is the source node and Ro is the destination node (exit). We define that:

(1) If |arctan[(Yj-Yi)/(Xj-Xi)]-arctan[(Yo-Ys)/(Xo-Xs)]|<90°, we could set the direction from Ri to Rj;

(2) If |arctan[(Yj-Yi)/(Xj-Xi)]-arctan[(Yo-Ys)/(Xo-Xs)]|>90°, we set the direction from Rj to Ri;

(3) If |arctan[(Yj-Yi)/(Xj-Xi)]-arctan[(Yo-Ys)/(Xo-Xs)]|=90°, the direction can be set from the further node(the distance between the node and the exit) to the nearer node.

In this way we make sure that every movement of the firefighter is getting closer to the exit. According to the principle, we establish all the possible paths from node b to node p. As shown in the figure, Vector Vef is perpendicular to Vbw, but Re is closer to Rw , so the direction should be set from Rf to Re.



.

Figure 2. Directed graph from node b to w

The directed graph can be converted into a path matrix Cv . If the directed arc is between the ith vertex and the jth vertex, the element of the ith row and jth column C (i, j) v is Eij=exp(dij) (dij is the length of arc between Ri and Rj); Otherwise it is 0. The path matrix of Fig.2 can be defined as:

$$V = \begin{bmatrix} b \\ e \\ f \\ k \\ l \\ p \end{bmatrix}, \qquad C_V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ E_{be} & 0 & E_{fe} & 0 & 0 & 0 \\ E_{bf} & 0 & 0 & 0 & 0 & 0 \\ 0 & E_{ek} & 0 & 0 & 0 & 0 \\ 0 & E_{el} & E_{fl} & 0 & 0 & 0 \\ 0 & 0 & 0 & E_{kp} & E_{lp} & 0 \end{bmatrix}$$

$$C^2_V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ E_{bf}E_{fe} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ E_{be}E_{ek} & 0 & E_{fe}E_{ek} & 0 & 0 & 0 \\ E_{be}E_{ei} + E_{bf}E_{fi} & 0 & E_{fe}E_{el} & 0 & 0 & 0 \\ 0 & E_{ek}E_{kp} + E_{el}E_{lp} & E_{fl}E_{lp} & 0 & 0 & 0 \end{bmatrix}$$

$$C^3_V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ E_{bf}E_{fe}E_{ek} & 0 & 0 & 0 & 0 & 0 \\ E_{bf}E_{fe}E_{el} & 0 & 0 & 0 & 0 & 0 \\ E_{be}E_{ek}E_{kp} + E_{be}E_{el}E_{lp} + E_{bf}E_{fl}E_{lp} & 0 & E_{fe}E_{ek}E_{kp} + E_{fe}E_{el}E_{lp} & 0 & 0 & 0 \end{bmatrix}$$

$$C^4_V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ E_{bf}E_{fe}E_{ek}E_{kp} + E_{bf}E_{fe}E_{el}E_{lp} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C^5_V = C^6_V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

If C^r_V （6,1）=0, we conclude that there is no path from Rb to Rp taking num=r hops; while if C^r_V （6,1）>0, there is at least one path exist between Rb and Rp with num=r hops. So all the possible paths of Fig.2 can be obtained by the formula as follows:

$$C^{\Sigma}_V(6,1) = \sum_{n=1}^{6} C^n_V(6,1) = E_{be}E_{ek}E_{kp} + E_{be}E_{el}E_{lp} + E_{bf}E_{fl}E_{lp} + E_{bf}E_{fe}E_{ek}E_{kp} + E_{bf}E_{fe}E_{el}E_{lp} \quad (1)$$

The value of the first factor of equation(1), $E_{be}E_{ek}E_{kp} = \exp(d_{be} + d_{ek} + d_{kp})$, fully indicates the length of the path : Rb→Re→Rk→Rp.

So the smallest factor of the formula corresponds to the shortest path. However, the optimal path does not mean the shortest path. It should also take the safety of the path into consideration.

IV. The Optimal Evacuation Path Selection

In fire scenes, there are many factors that may threaten the safety of firefighters, such as high temperature, smoke, poisonous gas and so on. Therefore, environmental situation around the selected path is essential.

Firstly, in section 3, we proposed a method to establish source-destination based directed graph, so that all feasible paths can be found out; Secondly, information fusion is used to evaluate the safety of nodes in each path, the nodes safety level reflects surrounding environment situation; Finally, safety-factor considered optimal path selection method is presented.

A. Evaluation of the environment safety based on D-S evidence theory

The environment situation of the fire scene is a complex situation with the characteristics of dynamic, nonlinear and uncontrolled. At the same time, due to the inaccuracy of sensor and environmental noise, the information gathered by single sensor may be not reliable and complete. So the multi-sensor information fusion is very necessary, which is used to increase the data using rate and to advance the information reliability and error tolerance.

Currently, there are several methods of multi-sensor information fusion, such as Bayesian test, Kalman filter, expert system, neural networks, fuzzy sets, D-S evidence theory and so on.

However, there are many shortages which are difficult to overcome in the process of fusion by these methods, the large amount of data calculation limits the application of Bayesian test and Kalman filter, the shortcoming of fuzzy sets is being more sensitive to the object of noise in the fusion process. Compared to this algorithm, D-S evidence theory [10-14] is better than the traditional methods to grasp the unknown and uncertainty of the problems, meanwhile, it does not require the priori probability, resulting in having been widely used in multi-sensor information fusion.

There are many factors that may threaten the safety of firefighters in fire scenes; we choose three most representative parameters to evaluate the environment situation: smoke density, temperature and oxygen density. According to the results of the fusion, environment safety levels are divided into three levels (I, II, III), representing safe, and medium, risky respectively.

The table 1 below shows the each probability of environment situation, it obtained by the expert system [11].

TABLE.I. Each Probability Of Environment Situation [11]

Parameter	Range	Safe (I)	Medium (II)	Risky (III)
Temperature	>=800	0.2	0.1	0.7
	200<T<800	0.3	0.4	0.3
	<=200	0.7	0.1	0.2
Smoke Density	>=1000	0.2	0.1	0.7
	600<S<1000	0.3	0.3	0.4
	<=600	0.7	0.1	0.2
Oxygen Density	<=14%	0.2	0.1	0.7
	14%<O2<21%	0.3	0.4	0.3
	>=21%	0.7	0.1	0.2

If the information sensor gathered at the certain time is: T=400°c, S=1200ppm, O_2=10%, Mi means different kinds of sensors, M(i) means different levels of environment situation. The probability distribution is as follow:

TABLE.II. Probability Distribution At A Certain Time

Probability	M(I)	M(II)	M(III)
M1	0.3	0.4	0.3
M2	0.2	0.1	0.7
M3	0.2	0.1	0.7

We obtain the results of information fusion with the algorithm of D-S evidence theory. $H(1)=M(1)\oplus M(2)$, $H(2)=M(1)\oplus M(2)\oplus M(3)$. Papers [13-14] provide a detailed definition and fusion steps of D-S evidence theory. Learning from that, we can obtain the fusion results shown in table III.

TABLE.III. Results Of Information Fusion

Probability	M(I)	M(II)	M(III)
H1	0.2564	0.0256	0.718
M3	0.2	0.1	0.7
H2	0.0662	0.0009	0.9329

According to all the tables above, the probabilities of level I, II, III are 0.0662 、 0.0009 、 0.9329, max{M(I),M(II),M(III)} is M（III）, obviously, the surrounding situation of the path is very dangerous, it warns the firefighters to avoid the high-risk node in the evacuation process. We can obtain environment safety level of each node by information fusion in order to provide for optimal path selection.

B. Safety-factor considered optimal path selection

Based on the results in section A, environment safety level of each node can be classified into a certain safety level through information fusion.

We assume that green, yellow and red circles represent safe, medium and risky nodes respectively. The directed graph with safety-factor weight is shown in Figure 3 as follows.

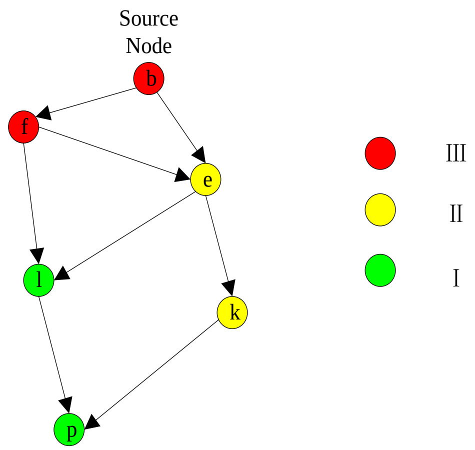


Figure 3. Directed graph with safety-factor weight

The node Ri can be set as a three dimension coordinate (Xi, Yi, αi). αi represents the safety level of the node obtained by information fusion above. αi adjusts the distance between nodes according to the safety level of the destination node. The coefficient αi is larger when the safety level of destination node is higher. The value of αi is assigned as a, 2a, 3a for level I (safe), II (medium) and III (risky) respectively. So we define:

Dij =αi *dij

Eij=exp(Dij), written as $E_{ij}^{\alpha_i}$

The path matrix evolves as follows:

$$C_V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ E_{be}^{\alpha_e} & 0 & E_{fe}^{\alpha_e} & 0 & 0 & 0 \\ E_{bf}^{\alpha_f} & 0 & 0 & 0 & 0 & 0 \\ 0 & E_{ek}^{\alpha_k} & 0 & 0 & 0 & 0 \\ 0 & E_{el}^{\alpha_l} & E_{fl}^{\alpha_l} & 0 & 0 & 0 \\ 0 & 0 & 0 & E_{kp}^{\alpha_p} & E_{lp}^{\alpha_p} & 0 \end{bmatrix}$$

According to formula (1), we can deduce that the safety-factor path collections are as follows:

$$C^{\Sigma}_V(6,1)=\sum_{n=1}^{6} C^{n}_V(6,1) = E^{\alpha_e}{}_{be}E^{\alpha_k}{}_{ek}E^{\alpha_p}{}_{kp} + E^{\alpha_e}{}_{be}E^{\alpha_l}{}_{el}E^{\alpha_p}{}_{lp} + E^{\alpha_f}{}_{bf}E^{\alpha_l}{}_{fl}E^{\alpha_p}{}_{lp}$$

$$+E^{\alpha_f}{}_{bf}E^{\alpha_e}{}_{fe}E^{\alpha_k}{}_{ek}E^{\alpha_p}{}_{kp} + E^{\alpha_f}{}_{bf}E^{\alpha_e}{}_{fe}E^{\alpha_l}{}_{el}E^{\alpha_p}{}_{lp}$$

C. Numerical simulation

In order to verify the method presented above, we define the parameter of Figure 3 in Table IV & Table V. The safety level of each node have obtained by information fusion has shown in table IV. Each path length we assumed and the comparison are listed in Table V.

TABLE.IV. SAFETY LEVEL OF EACH NODE

Node	Safety Level	αi
b	III	3a
e	II	2a
f	III	3a
k	II	2a
l	I	a
p	I	a

TABLE.V. THE DISTANCE BETWEEN NODES

Arc	Length/m	Adjust Arc	length/m
dbe	5	Dbe=αe dbe	2a*5
dbf	5	Dbf =αf dbf	3a*5
dek	4	Dek=αk dek	2a*4
del	8	Del=αl del	a*8
dfe	7	Dfe=αe dfe	2a*7
dfl	5	Dfl =αl dfl	a*5
dkp	8	Dkp=αp dkp	a*8
dlp	5	Dlp=αp dlp	a*5

From Table V, we can learn that the distance between nodes have changed after adjustment. The lengths of all feasible paths can be obtained by adding distances between nodes. The results are shown in Table VI as follows:

TABLE.VI. THE COMPARISON OF PATH LENGTHS BEFORE AND AFTER ADJUSTMENT

Path	Before Adjust/m	After adjust/m
b→e→k→p	17	26a
b→e→l→p	18	23a
b→f→l→p	15	25a
b→f→e→k→p	24	45a
b→f→e→l→p	25	42a

According to Table VI, without safety consideration, the shortest path is b→f→l→p, considering safety factor, the shortest path changes to b→e→l→p. Noted that f is a high risk node, it is reasonable to avoid it. Therefore the Safety-factor considered optimal path selection method is superior to the old one.

V. CONCLUSIONS

We have proposed a method of optimal evacuation path selection for firefighters based on the dynamic triangular network model. The network consists of robots who can work in unknown environment where information gather is difficult. Robots in the network are equipped with sensors to provide real-time information for firefighters. In order to find out all the feasible direct paths, we set up the directed graph by calculating the included angle between the vector of source-destination and the vector of two random nodes. In this way we make sure every movement is closer to the exit. For the multi-parameter information fusion, D-S evidence theory is applied to make accurate estimation of the environment, avoiding the misjudgment of single sensor. The objectives of the method are to minimize the path length for shortening evacuation time, meanwhile, to keep the firefighters from the dangerous regions.

In this paper, we assumed that there is only one exit in the dynamic triangular network model. Evacuation problem of multi-exit should also be explored. Besides, the positions of robots who gather information in the model are fixed. Actually, the robots may move with the firefighters in order to provide real-time information. A new model and an improved algorithm for evacuation path selection will be the future research directions of the authors.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (NSFC) granted number 71171045.

REFERENCES

[1] Marina Yusoff, Junaidah Ariffin, Azlinah Mohamed, “Optimization Approaches for Macroscopic Emergency Evacuation Planning,” A Survey, Information Technology, pp. 1-7, March 2008.

[2] Yuan Yuan, Dingwei Wang, “Multi-Objective Path Selection Model and Algorithm Emergency Evacuation,” Proceedings of the IEEE International Conference on Automation and Logistics, 2007.

[3] Xinlu Zong, Shengwu Xiong., “Multi-objective Ant Colony Optimization Model for Emergency Evacuation,” International Conference on Natural Computation, 2010.

[4] Zhixiang Fang, Xinlu Zong, Qingquan Li, QiupingLi, Shengwu Xiong. “Hierarchical multi-objective evacuation routing in stadium using ant colony optimization approach,” Journal of Transport Geography. vol.19, pp. 443–451, 2007.

[5] Wang Tian-yu , Huang Rui, “The Application of the Shortest Path Algorithm in the Evacuation System,” International Conference of Information Technology, Computer Engineering and Management Sciences, 2011.

[6] Qiuping Li, Zhixiang Fang, Qingquan Li, “Multi-objective Evacuation Route Assignment Model Based on Genetic Algorithm,” Geoinformatics, 18th International Conference, 2010.

[7] João Coutinho-Rodrigues, Lino Tralhão, Luís Alçada-Almeida, “Solving a location-routing problem with a multi-objective approach: the design of urban evacuation plans,” Journal of Transport Geography. vol.22, pp. 206–218, 2012.

[8] Ulf Witkowski, Mohamed EI-Habbal, “Ad-hoc net- work communication infrastructure for multi-robot systems in disaster scenarios,” Proceedings of the EURON/IARP International Workshop on Robotics for Risky Interventions and Surveillance of the Environment, 2008.

[9] Demin Li, Jie Zhou, Jiacun Wang, Yun Wang, Qianyi Zhang, “ Information Fusion and Path Selection for Firefighters Based on Ad Hoc Dynamic Triangular Network,” International Journal of Advancements in Computing Technology, unpulished..

[10] Rombaut M, Zhu YM, “Study of Dempster-Shafer for image segmentation applications,” Image Vis Computer. vol. 20, pp.15-23, 2002.

[11] D.L.Hall, “Mathematical Techniques in Multi-sensor Data Fusion,” Boston, London: Artech House, 1992.

[12] Li Xiangli, Lv Jianping, “Improvement of D-S Theory of Evidence in Multi-sensor Information Fusion System,” Sensor Technology, 2005.

[13] Li Yi-Bo, Wang Ning, “Based on D-S Evidence Theory of Information Fusion Improved Method,” International Conference on Computer Application and System Modeling, 2010.

[14] Jia-nan Chu, Yunhua Li, “Study on the Coal Mine Safety Assessment Based on Rough Set-Neural Network-Evidence Theory.” International Conference on Future Information Technology and Management Engineering, 2010.

The Classification of the Real-Time Interaction-Based Behavior of Online Game Addiction in Children and Early Adolescents in Thailand

Kongkarn Vachirapanang
Technopreneurship and Innovation Management Program
Graduate School, Chulalongkorn University
Bangkok, Thailand

Sakoontip Tuisima
Faculty of Computer Engineering
Chulalongkorn University
Bangkok, Thailand

Sukree Sinthupinyo
Faculty of Computer Engineering
Chulalongkorn University
Bangkok, Thailand

Puntip Sirivunnabood
Faculty of Psychology
Chulalongkorn University
Bangkok, Thailand

***Abstract*— This paper aims to study actual behaviors of Thai children and early adolescents with different levels of game addiction while playing online games from an angle of the interaction between a user and computer. Real-time interaction-based behavior data from a program agent installed in personal computers in 20 sample houses were screened along with consent given by children and their parents. Collection of data about game-playing periods, frequency, game-playing times, text-based chatting, mouse click and keyboard typing during the game was carried out over two months and four case study in-depth interviews for addicted players and their parents. The results revealed a novel method to classify online game addiction level of children and early adolescents by mouse click and keyboard typing data and also found relationship between the playing data recorded and game addiction risk conditions and risk behaviors as explained in the article.**

Keywords- online game addiction; classification; intelligent agent; real time.

I. Introduction

From the previous research about online game addiction, we found that most researches relating to behaviors and the factors or mechanisms behind game addiction commonly used self-report methodology [1]. But Achab et al.[2] argued that this method may not provide realistic results because respondents may give inaccurate answers to protect themselves. So some researchers attempted to introduce technology as a tool to provide explanatory information during the game such as the measurement of electrodermal activity (EDA) and heart rate (HR) in order to describe the player's experience, cognition and emotions [3, 4], or the real-time emotion diagnosis system which monitor the facial or vocal expressions occurring while playing the game [5].

These are all studies of physically effects on the body when users are playing an online game taking into account mental or psychological conditions. At the same time, Caplan et al.[6] proposed a method which collects and analyzes the data from actual playing periods while playing an online game via the server of the game provider to analyze the relationship between time and problematic Internet use (PIU). In fact, the players interact with their computers not only during the playing period but also with the character controller via basic devices such as the mouse or keyboard as well as playing some online games with an opportunity for conversation via chat room. In previous researches, a relationship was also found between game chat and externalizing aggressive behavior and game addiction [7, 8].

Therefore, this research intends to study actual behavior from various interactions between player and computer in the period, time spent, frequency, chat via the keyboard including the amount of mouse clicking and keyboard typing. By comparing children and youths in Thailand with different levels of addiction – Addicted level, High engagement level and Normal level – for defining the behavior patterns which can be used for data evaluation and also to predict and further protect them from game addiction in Thailand.

II. Materials And Methods

A computer program constructed for collecting data of online game-playing behavior was designed and developed in order to store data consisting of the name of the game, the playing period, frequency, playing time, quantity of mouse click and keyboard typing and also behaviors occurring during play the game such as using harsh or unsuitable words while chatting [9]. Our limitation is that this computer must be regularly used by only one agent; it collected the data of online game-playing behavior based on real-time interaction-based behavior (RTIB) and monitored and recorded the behaviors assigned throughout the period the computer was used, counting units for recording the data as milliseconds, and daily sending all data automatically to the server via the Internet from May - July 2011 as system design shown in Figure 1.

The playing period, frequency, and playing time were analyzed the differentiation values of each addiction level by

ANOVA and quantity of mouse click and keyboard typing in four game type based on playing characteristics; long term game, casual game, real time game, and turn base game [10].Those data were constructed the user model using Waikato Environment for Knowledge Analysis (WEKA)[11] by features shown in Table 1.Then using 10-fold cross validation method by Decision Tree (DT) and Backpropagation Neural Networks (BNN). For more details, the semi-structure in-depth interviews were conducted with all parents of addicted players.

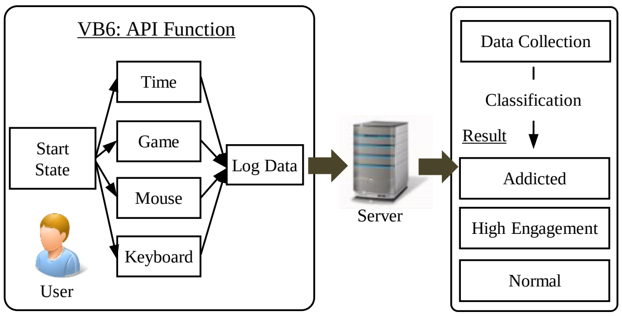


Figure 1. The system design.

TABLE I. FEATURES LIST

Features
1) Weekday or holiday.
2) Game playing duration per day. (minutes)
3) Duration of mouse click in game per day. (minutes)
4) Duration of keyboard pressing in game per day. (minutes)
5) Total number of mouse click and keyboard pressing in game playing per day.
6) Total number of mouse click in game playing per day.
7) Total number of keyboard pressing in game playing per day.
8) Total number of switching between mouse and keyboard use.
9) Maximum number of mouse click before switching to keyboard using in game playing per day.
10) Maximum number of keyboard using before switching to mouse click in game playing per day.
11) Average time of mouse click and keyboard pressing in game playing per minute.
12) Average number of mouse click in game playing per minute.
13) Average number of keyboard using in game playing per minute.
14) Three computer game addition level; Normal, High engagement, and Addiction

III. RESULTS

A. Subjects

By screening and randomizing the sample of children and early adolescents (12- to 14-year-olds) from Thailand who play online games.

All sample stay in Bangkok and are not part of the treatment program for game addiction symptoms and Attention Deficit Hyperactivity Disorder (ADHD). We screen the sample by Game Addiction Screening Test (GAST)[12] of the Center of Game Addiction Prevention at the Institute of Child and Adolescent Ratchanakharin Mental Health, Department of Mental Health, Ministry of Public Health, Thailand, along with the equipment quality test having a Cronbach's alpha value of 0.804.

A total of 31 children passed the screen test and consented to be the sample for the collection of data about online game-playing behavior. After eliminating the player data recorded for less than seven days and the recorded data of more than one player using the same computer, the remaining data were from 20 players classified according to risk level of online game addiction, gender and age, as shown in Table 2.

TABLE II. DE MOGRAPHICS

Quantity by / Addiction Level	Gender		Age		
	Boy	**Girl**	**12**	**13**	**14**
Normal Level	10	-	2	4	4
High engagement Level	6	-	2	3	1
Addicted Level	3	1	0	2	2
Total	19	1	4	9	7
	20				

B. Game playing period and frequency

Table 3 indicates that the average values of the playing periods of all players were: 60.00 minutes (SD = 16.71 minutes) for the maximum value of time spent per period is the Addicted level player, 53.50 minutes (SD = 17. minutes) for the Normal level player and 47.33 minutes (SD = 11.48 minutes) for the player at High engagement level.

The research defined a weekday as studying day and a holiday as a day off with no studying even in the middle of the week. We checked the parents and each player for the real value. The average value of all level players on holiday was higher than on a weekday.

The results were as follows: the maximum average value on a weekday was from the Addicted level player at 136.75 minutes (SD = 56.52 minutes); from the Normal level player at 87.90 minutes (SD = 25.66 minutes); and from the High engagement level player at 85.00 minutes (SD = 35.52 minutes).

For the average value on the holiday, we found that the maximum value was from the High engagement level player at 189.33 minutes (SD = 65.16 minutes); from the Normal level player, it was 181.40 minutes (SD = 59.69 minutes); and from the Addicted level, it was 155.25 minutes (SD = 53.80 minutes).

Furthermore, the maximum value of game playing per day was on the Addicted level at 132.75 minutes (SD = 31.60 minutes); the High engagement level was 126.66 minutes (SD = 38.68 minutes); and the Normal level was 126.30 minutes (SD = 38.11 minutes).

The average value of playing per week of the High engagement level was the highest value at 601.00 minutes (SD = 270.84 minutes); the Normal level at 576.30 minutes (SD = 280.31 minutes) and Addicted level at 572.75 hr. (SD = 252.70 minutes).

The average value of game-playing frequency per day of three addiction level players were quite similar at three times per day as shown in Table 3 too.

TABLE III. DESCRIPTIVE STATISTIC VALUE

		N	Mean	Std. Deviation
Time spent per period	Normal Level	10	53.5000	17.36695
	High Engagement Level	6	47.3333	11.48332
	Addicted Level	4	60.0000	16.71327
	Total	20	52.9500	15.56472
Time spent on weekday	Normal Level	10	87.9000	25.66645
	High Engagement Level	6	85.0000	35.51901
	Addicted Level	4	136.7500	56.52359
	Total	20	96.8000	39.62535
Time spent on holiday	Normal Level	10	181.4000	59.69403
	High Engagement Level	6	189.3333	65.16032
	Addicted Level	4	155.2500	53.80443
	Total	20	178.5500	58.46049
Time spent per day	Normal Level	10	126.3000	38.11693
	High Engagement Level	6	126.6667	38.68161
	Addicted Level	4	132.7500	31.60564
	Total	20	127.7000	35.30484
Time spent per week	Normal Level	10	576.3000	280.30898
	High Engagement Level	6	601.0000	270.84608
	Addicted Level	4	572.7500	252.70718
	Total	20	583.0000	258.36958
Playing frequency per day	Normal Level	10	2.6590	.97001
	High Engagement Level	6	2.8550	1.11783
	Addicted Level	4	2.4650	1.22449
	Total	20	2.6790	1.01533

TABLE IV. ONE-WAY ANOVA TESTING

		Sum of Squares	df	Mean Square	F	Sig.
Time spent per period	Between Groups	391.117	2	195.558	.789	.470
	Within Groups	4211.833	17	247.755		
	Total	4602.950	19			
Time spent on weekday	Between Groups	8011.550	2	4005.775	3.121	.070
	Within Groups	21821.650	17	1283.626		
	Total	29833.200	19			
Time spent on holiday	Between Groups	2950.467	2	1475.233	.405	.674
	Within Groups	61984.483	17	3646.146		
	Total	64934.950	19			
Time spent per day	Between Groups	128.017	2	64.008	.046	.955
	Within Groups	23554.183	17	1385.540		
	Total	23682.200	19			
Time spent per week	Between Groups	2813.150	2	1406.575	.019	.981
	Within Groups	1265528.850	17	74442.874		
	Total	1268342.000	19			
Playing frequency per day	Between Groups	.373	2	.187	.165	.849
	Within Groups	19.214	17	1.130		
	Total	19.587	19			

Table 4 shown that time spent per period (F=0.789, *p*=0.47), time spent on weekday (F=3.121, *p*=0.70), time spent on holiday (F=0.450, *p*=0.67), time spent per day (0.46, *p*=0.95), time spent per week (F=0.019, *p*=0.981), and playing frequency per day (F=0.165, *p*=0.85) of each addiction level have no significantly different.

C. *Game playing time*

We divided the time in a day into eight periods, with a period of new morning (0.01-6.00a.m.), before entering a study period (6.01-8.00a.m.), morning study period (8.01-11.00a.m.) lunch period (11.01a.m.-1.00p.m.), afternoon study period (1.013.30p.m.), after-study period (3.31-6.00p.m.), before-sleeping period (6.01-8.00p.m.) and resting period (8.01-12.00p.m.).

Table 5 shows that most players of all levels have similar behavior patterns for game playing, i.e. on a weekday they play after coming back from school until going to bed, except for some players who play until the following day, i.e. Normal-02 and Addict-02. Some players wake up early in the morning to play before going to school, such as Normal-04, Normal-08, High engagement-01 and High engagement-06.

During online computer game playing in a holiday period shown in Table 6, it is found that there is a distribution of game playing across all periods of the day. Also, it is noticed that there are some players who play until the following morning, especially in a holiday period, i.e. Normal-03, Normal-04 and High engagement-02, while Addict-02 is found playing until the following morning both on a weekday and during a holiday. From these results, it is not possible to conclude that the online computer game-playing times of day of players in each level are different.

TABLE V. PERCENTAGE OF PLAYING FREQUENCY IN EACH PERIOD ON WEEKDAY OF ALL SUBJECTS DIVIDED ACCORDING TO ADDICTION LEVEL

Player ID	Playing frequency in period of Weekday (%)								Total (%)
	0.01-6.00 a.m.	6.01-8.00 a.m.	8.01-11.00a.m.	11.01 a.m.-1.00 p.m.	1.01 - 3.30 p.m.	3.31-6.00 p.m.	6.01-8.00p.m.	8.01-12.00 p.m.	
Normal-01	-	-	-	-	-	**100.00**	-	-	100.00
Normal-02	1.37	-	-	-	2.74	8.22	21.92	**65.75**	100.00
Normal-03	-	-	-	-	14.28	4.76	14.28	**66.68**	100.00
Normal-04	-	2.80	-	-	-	-	31.78	**65.42**	100.00
Normal-05	-	-	-	-	-	-	30.77	**69.23**	100.00
Normal-06	-	-	-	-	2.13	**48.94**	38.30	10.64	100.00
Normal-07	-	-	-	-	-	8.00	44.00	**48.00**	100.00
Normal-08	-	7.32	-	-	-	29.27	**41.46**	21.95	100.00
Normal-09	-	-	-	-	-	5.17	37.93	**56.90**	100.00
Normal-10	-	-	-	-	-	19.23	38.46	**42.30**	100.00
HE-01	-	5.89	-	-	-	14.70	35.29	**44.12**	100.00
HE-02	-	-	-	-	-	3.63	**52.73**	43.64	100.00
HE-03	-	-	-	-	-	24.81	33.08	**42.11**	100.00
HE-04	-	-	18.18	-	-	-	**63.64**	18.18	100.00
HE-05	-	-	-	-	7.41	7.41	37.03	**48.15**	100.00
HE-06	-	3.84	-	-	9.61	27.88	**49.04**	9.61	100.00
Addicted-01	-	-	-	-	-	30.51	**49.15**	20.34	100.00
Addicted-02	33.33	-	-	-	-	16.67	-	**50.00**	100.00

Addicted-03	-	-	-	-	**33.33**	33.33	25.00	8.34	100.00
Addicted-04	-	-	-	-	-	-	45.45	**54.55**	100.00

Note: HE denotes High engagement Level

TABLE VI. PERCENTAGE OF PLAYING FREQUENCY IN EACH PERIOD ON HOLIDAY OF ALL SUBJECTS DIVIDED ACCORDING TO ADDICTION LEVEL

Player ID	Playing frequency in period of Holiday (%)								Total (%)
	0.01-6.00 a.m.	6.01-8.00 a.m.	8.01-11.00a.m.	11.01 a.m.-1.00 p.m.	1.01 - 3.30 p.m.	3.31-6.00 p.m.	6.01-8.00p.m.	8.01-12.00 p.m.	
Normal-01	-	-	-	**66.67**	11.11	22.22	-	-	100.00
Normal-02	-	3.53	18.82	20.00	14.12	4.71	-	**38.82**	100.00
Normal-03	8.89	3.33	3.33	3.33	22.22	10.00	6.67	**42.22**	100.00
Normal-04	2.05	0.51	11.28	13.33	10.26	19.49	**23.08**	20.00	100.00
Normal-05	-	-	**25.00**	20.00	10.00	10.00	15.00	20.00	100.00
Normal-06	-	-	12.20	17.07	26.83	4.88	**36.59**	2.44	100.00
Normal-07	-	11.43	**31.43**	17.14	17.14	8.57	2.86	11.43	100.00
Normal-08	-	11.11	**25.93**	14.81	18.52	11.11	14.81	3.71	100.00
Normal-09	-	2.35	2.35	7.06	**30.59**	10.59	17.65	29.41	100.00
Normal-10	-	-	**33.33**	4.17	-	16.67	20.83	25.00	100.00
HE-01	-	-	11.11	**37.04**	22.22	3.70	14.82	11.11	100.00
HE-02	1.19	9.53	**29.76**	16.67	13.10	10.71	7.14	11.90	100.00
HE-03	-	5.19	19.48	10.39	10.39	**23.38**	11.69	19.48	100.00
HE-04	-	-	21.43	7.14	14.28	**42.86**	3.57	10.72	100.00
HE-05	-	2.86	4.28	20.00	17.14	**31.43**	12.86	11.43	100.00
HE-06	-	14.28	**38.89**	14.28	10.33	4.76	7.94	9.52	100.00
Addicted-01	-	-	11.54	**26.92**	21.15	13.47	9.62	17.30	100.00
Addicted-02	7.14	14.29	7.14	7.14	7.14	21.43	7.14	**28.58**	100.00
Addicted-03	-	-	13.33	33.34	20.00	**20.00**	13.33	-	100.00
Addicted-04	-	3.45	10.34	6.90	**17.24**	31.03	13.79	**17.24**	100.00

Note: HE denotes High engagement Level

D. Text-based chat during game playing

Table 7 shows that almost all players have conversations with other players during online computer game playing except Normal-06, Normal-07, High engagement-02, High engagement-04 and High engagement-05. Normal-02, Normal-04 and Normal-05 have conversations both with players they already know and with new players in order to make friends. Most conversations are to do with game matters, five players at Normal level and two players at both of High-engagement and Addicted level; matters of general life are less frequently discussed and only Normal-02, Normal-09 and High engagement-01 have conversations about studying.

Moreover, it is found three players at Normal level and one player at Addicted level who have rude or violent conversations, i.e. Normal-04, Normal-05, Normal-11 and Addict-03. In the case of Normal-04, conversations concerning pornographic matters have also been found, including requests for sexual relationships with other players. Normal-05 has been found challenging other players they did not know before with severe words. However, from Table 7 we cannot conclude that there is any difference between types of conversation during the online computer game playing of players in all three levels.

TABLE VII. CONVERSATIONS DURING ONLINE COMPUTER GAME PLAYING OF ALL SUBJECTS DIVIDED ACCORDING TO ADDICTION LEVEL.

Player ID	Conversation with		Contents of conversation			Found rude or violent conversations
	Already know players	New players	General life matters	Studying matters	Game matters	
Normal-01	-	✓	✓	-	-	-
Normal-02	✓	✓	✓	✓	✓	-
Normal-03	✓	-	-	-	✓	-
Normal-04	✓	✓	✓	-	-	✓
Normal-05	✓	✓	-	-	✓	✓
Normal-06	-	-	-	-	-	-
Normal-07	-	-	-	-	-	-
Normal-08	-	✓	-	-	✓	-
Normal-09	✓	-	✓	✓	-	-
Normal-10	✓	-	-	-	✓	✓
Total in Normal Level	6	5	4	2	5	3
HE-01	✓	-	✓	✓	-	-
HE-02	-	-	-	-	-	-
HE-03	✓	-	-		✓	-
HE-04	-	-	-	-	-	-
HE-05	-	-	-	-	-	-
HE-06	✓	-	✓	-	✓	-
Total in High engagement Level	3	0	2	1	2	0
Addicted-01	-	✓	-	-	✓	-
Addicted-02	-	✓	-	-	✓	-
Addicted-03	✓	-	✓	-	-	✓
Addicted-04	-	-	-	-	-	-
Total in Addicted Level	1	2	1	0	2	1

Note: HE denotes High engagement Level

E. Mouse click and keyboard typing

The agent can be classifying each addiction level players by the data of mouse click and keyboard typing in any game type. Table 8 shown the validation and indicates that a result from DT is higher percentage of classification's accuracy than BNN in case of casual game at 92.63. While the results from BNN are higher than DT in case of turn base game, real time game, and long term game at 97.73, 90.00, and 97.75 respectively.

TABLE VIII. THE PERCENTAGE OF ACCURACY COMPARISON BY BNN AND DT FOR EACH GAME TYPE IN ANY ADDICTED LEVEL

Type of game	Percentage of classification's accuracy		Percentage of classification's error	
	BNN	**DT**	**BNN**	**DT**
Casual game	91.35	92.63	8.65	7.37
Turn base game	97.73	90.91	2.27	9.09
Real time game	90.00	87.50	10.00	12.50
Long term game	97.75	88.76	2.25	11.24

F. In-depth Interview with Parents, Self-Report and RTIB of Addicted Players

Addicted-01: The parents provided information that Addicted-01 uses a computer to play online computer games in a private room and they can hear the noise of the games being played all the time. They felt that the child talks or takes part in events with the family less often, although they cannot monitor the other risk behaviors because they did not have enough time to observe the behavior of the child. Table 9 shows the comparison between the risk behavior and its impact according of the Self-Report and the RTIB recorded.

TABLE IX. COMPARISON BETWEEN SELF-REPORT AND RITB OF ADDICTED-01

Self-Report	RTIB
- Usually play the game without caring for others as forget homework, rest, or eating, etc.	- The data showed that playing during dinner without resting occurred in 50% of total behavior recorded.
-Often play the game for more than the intended time (two hours per day).	-The data showed that playing more than two hours per day occurred in 95% of total behavior recorded.
-There was school absence, because of playing the game until late, and cannot wake up.	The data showed that continuous playing in the evening until 10.00 pm occurred in 30% of total behavior recorded.

Addicted-02: The parents have an agreement with the child about restricting the playing time for online computer games to less than two hours per day. But the child usually plays for longer than the agreement time. The parents usually warn the child to stop playing every day. However, the Addicted-02 accepts the request without any resistance. Regarding the observation by parents, the child often talks with friends about the game, but there were no negative effects or other negative behaviors. The comparison between the behavior which the parents observed, the Self-Report, and the RTIB of Addicted-02 is shown in Table 10.

TABLE X. COMPARISON BETWEEN IN-DEPTH INTERVIEW WITH PARENT AND SELF-REPORT AND RITB OF ADDICTED-02

In-depth Interview	Self-Report	RTIB
- The children always play the game more than the promised time.	- Often play the game over the intended time.	- The data showed that playing more than two hours per day occurred in 67% of total behavior recorded.
-	- Felt that the relationship with others family members is worse.	- The data showed that using almost all free time to play the game occurred in 100% of the total behavior recorded.

Addicted-03: The parents provided the information that previously Addicted-03 played for up to 3–4 hours per day, and there were side effects on many functions as a result. After that, the parents closely controlled playing by setting an agreement with Addicted-03 to play computer games online for no more than 1 hour 30 minutes a day on weekdays and not over 2 hours per day in the holidays. But this was not as successful as it should have been, and some the risk behaviors were still found. Some behaviors that are recorded by the RTIB but which do not appear from the In-depth Interview or Self-Report are shown in Table 11.

Addicted-04: The parents set the playing period at not more than 1 hour 30 minutes a day, but Addicted-04 often could not follow this and greatly lost interest in the other activities. The parents have tried to provide other activities after school in both the weekday and holiday, so the playing time could be decreased. However, after the other activities were completed, the child spent the remaining time playing the game again (see Table 12).

TABLE XI. COMPARISON BETWEEN IN-DEPTH INTERVIEW FROM PARENT AND SELF-REPORT AND RITB OF ADDICTED-03

In-depth Interview	Self-Report	RTIB
- The child always plays the game more than the promised time.	- Often play the game more than the intended time.	- The data showed that playing more than the promised time on weekdays and in holidays occurred in 67% and 17% of total behavior recorded.
- Found the children play the game almost all the time.	- Use almost all free time to play the game.	- The data showed that using free time to play the game occurred in 100% of total behavior recorded.
- Usually play the game without care for others as forget school assignment, and housework.	- Less responsibility.	-
-	-	- The data showed that rude word chatting occurred in 50% of total behavior recorded.

TABLE XII. COMPARISON BETWEEN IN-DEPTH INTERVIEW FROM PARENT AND SELF-REPORT AND RITB OF ADDICTED-04

In-depth Interview	Self-Report	RTIB
- The child sometimes plays the game more than the promised time.	- Often plays the game more than the intended time.	- The data showed that playing more than 1hour 30 minutes per day occurred in 58% of total behavior recorded.
- The child uses the remaining free time from other activities to play online games, and cannot stop playing the game.	- Use almost all free time to play the game.	- The data showed that using free time to play the game occurred in 100% of total behavior recorded.

IV. DISCUSSION

The study of the game-playing period found three discrepancies with previous researches: 1) The average game-playing per periods and per week from the sample of the Addicted level were less than the results from the self-report method [13-15]; 2) The average game-playing period per week of the Addicted level player was not higher than the High engagement level player as found in some research [16-18]; 3) The average value on holiday of the Addicted level was not more than on a weekday as Kim et al.[19] have stated.

Besides, when interviewing the parents of the sample group, it showed that some of the sample, especially Addict-04 and 05, even had a high value of playing behavior on holiday. But agreement between the children and the parents, as well as interest in other activities, may cause a decrease in the playing period on holiday [20], and the playing period per day and week to be lower than ever.

Moreover, the interview with the parents of the sample with the playing period average of higher than 10 hours per week indicated that these players were affected by the various dimensions of online game playing, while Normal-05, 08 and 10 and Addicted-01 had less interest in other activities. Normal-08, Normal-10, High engagement-06 and Addicted-01 spent most of their free time with the game when the relationships between Normal-08, High engagement-03 and High engagement-06 and their families were worse and everybody lost their temper when the parents stopped their games. However, the playing period data were unable to indicate the online game addicted condition in the children and adolescents [19], and we should consider this together with the negative consequences that occurred [21, 22].

While the above results included the positive relationship between the playing period and game addiction from the previous research [16, 23, 24], it could be said that the playing period average of higher than 10 hours per week could be an indicator for the risk of game addiction, especially in Thailand.

Furthermore, the next topic is about the frequency and time. We found that the sample played the online game on a weekday from after school (about 3.30p.m.) onwards and had a high value between 8.00 and 12.00p.m., which is different from the results of Thamawipat and Sawakejun [25], who found that the children from the south of Thailand commonly played the game between 4.00 and 8.00p.m. This may be caused by the difference in lifestyles between the people in Bangkok and those in the provinces.

In the case of continuous playing through the next day and playing in the morning before class, after the self-report was checked by GAST, and an additional interview with the parents was conducted, and it was found that Normal-02, Normal-03 and Normal-04 were unable to control the playing period and continued playing as they forgot the time. This was similar to High engagement-02 and Addict-02 who always forgot the time and was unable to stop playing when required. Furthermore, Normal-08, High engagement-01 and High engagement-06 were found to play every time except the class that conform to the self-evaluation of always spending their free time on playing the game. Both of the playing tendencies to forget the time and be unable to stop playing when they wanted were indicators of the loss of control that was related with the online game addiction [26-28], including playing from 10.00 to 12.00 p.m. This most affected vulnerable children and adolescents [29], so it can be said that the period of game playing may reflect the risk behavior and lead to loss of control and vulnerability.

The next point is the conversation during the game. The parents of players Normal-04, Normal-05, Normal-10 and Addicted-03 who saw the conversation with harsh or unsuitable words were interviewed and we found that Normal-04, Normal-05, Normal-10 and Addicted-03 were easily irritated and felt annoyed. Sometimes they expressed aggressive words or behavior after playing the online game, but Normal-05 did not show such behavior in real life. Chat during the game was a factor which increased the desire of the player to play it [30].

Meanwhile, even though a direct relationship between chat with aggressive words and the game addicted condition was not found, a relationship between aggressive behavior and the online game addicted condition was found [19, 28, 31], so it could be said that following the behavior of the text-based chat in the game can reflect the risk behavior that indicates aggression and relates to the online game addicted condition. Furthermore, the unsuitable risk behavior can be confirmed by what is found in the conversation on sex.

Even this had no direct relationship with the game addicted condition but may affect the lives of children.

The last and very important point is the amount of mouse click and keyboard typing while playing the online computer game. It is an only one result in this paper that can classify an online computer game addiction level of players. Besides, it is a novel method to classify the game addiction level by classification technique in machine learning field. Meanwhile, this method used the real-time interaction-based behavior collected from the interaction between a user and computer while playing games. So, it can reduce errors by the over or underestimate when the children do the addiction screening test by their subjective thinking. At the same time, it can reduce errors from inappropriate behavior observation by the parents that cannot monitor their child all the time. Then, this new addiction classification method can help the parent to have more accuracy data and warning them before their child get into the addiction level.

From the discussion, we can conclude that the development of an intelligent agent could help parents to be aware the addiction level of their children by real-time and more accurate data. The agent can classify not only the addiction level but also shows the risk behavior of self-control, the risk of vulnerability and aggression, and other unsuitable behaviors. Those accuracy real-time data can lead to efficient protection or cure children and early adolescents form the online game addiction disaster.

V. LIMITATIONS

It had 20 subjects for this paper then the case studies by semi-structure in-depth interviews of addicted players were conducted for compared with the real-time interaction behaviors. So it may not be used as a general reference but only as a case study and a guideline for further research.

ACKNOWLEDGMENT

This study was funded by THE 90th ANNIVERSARY OF CHULALONGKORN UNIVERSITY FUND (Ratchadaphiseksomphot Endowment Fund)

REFERENCES

[1] Kuss, D. and M. Griffiths, Internet Gaming Addiction: A Systematic Review of Empirical Research. International Journal of Mental Health and Addiction, 2012. 10(2): p. 278-296.
[2] Achab, S., et al., Massively multiplayer online role-playing games: comparing characteristics of addict vs non-addict online recruited gamers in a French adult population. BMC Psychiatry, 2011. 11: p. 144.
[3] Drachen, A., Nacke, L.E., Yannakakis, G. and Pedersen A.L., Correlation between heart rate, electrodermal activity and player experience in first-person shooter games, in Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games2010, ACM: Los Angeles, California. p. 49-54.
[4] Kuikkaniemi, K., Laitinen, T., Turpeinen, M., Saari, T., Kosunen, L., and Ravaja, N., The influence of implicit and explicit biofeedback in first-person shooter games, in Proceedings of the 28th international conference on Human factors in computing systems2010, ACM: Atlanta, Georgia, USA. p. 859-868.
[5] Nosu, K., Kurokawa, T., Korita, H., Ohhazama, Y., and Takeda, H., Real Time Emotion-Diagnosis of Video Game Players from their Facial Expressions and its Applications to Voice Feed-Backing to Game Players. in International Conference on Machine Learning and Cybernetics, 2007.

[6] Caplan, S., D. Williams, and N. Yee, Problematic Internet use and psychosocial well-being among MMO players. Comput. Hum. Behav., 2009. 25(6): p. 1312-1319.

[7] Holtz, P. and M. Appel, Internet use and video gaming predict problem behavior in early adolescence. Journal of Adolescence, 2011. 34(1): p. 49-58.

[8] Moore, R.J., N. Ducheneaut, and E. Nickell, Doing Virtually Nothing: Awareness and Accountability in Massively Multiplayer Online Worlds. Comput. Supported Coop. Work, 2007. 16(3): p. 265-305.

[9] Inrak, P. and S. Sinthupinyo, Applying latent semantic analysis to classify emotions in Thai text. in 2nd International Conference on Computer Engineering and Technology (ICCET), 2010.

[10] Game Genre. [cited 2011 6 October]; Available from: http://wiki.thaigamedevx.com/index.php?title=GameGenre.

[11] I. H. Witten, E. Frank, and M. E. Hall, Data Mining: Practical Machine Learning Tools and Techniques. Third Edition ed2011: Morgan Kaufmann Publishers.

[12] Pornnoppadol, C., Sornphaisan, B., Kumkleang, K., and Pattana-amorn, S., Game Addiction Screening Test Development. in The Annual Conference of The Royal College of Psychiatrists of Thailand., 2006. Thailand.

[13] Griffiths, M., A 'components' model of addiction within a biopsychosocial framework. Journal of Substance Use, 2005. 10(4): p. 191-197.

[14] Griffiths, M., Relationship between gambling and video-game playing: a response to Johansson and Gotestam. Psychol Rep, 2005. 96(3 Pt 1): p. 644-6.

[15] Chappell, D., Eatough, V. Davies, M. and Griffiths, M., EverQuest —It's Just a Computer Game Right? An Interpretative Phenomenological Analysis of Online Gaming Addiction. International Journal of Mental Health and Addiction, 2006. 4(3): p. 205-216.

[16] Hsu, S.H., M.-H. Wen, and M.-C. Wu, Exploring user experiences as predictors of MMORPG addiction. Computers & Education, 2009. 53(3): p. 990-999.

[17] Lee, M.S., et al., Characteristics of Internet use in relation to game genre in Korean adolescents. Cyberpsychol Behav, 2007. 10(2): p. 278-85.

[18] Charlton, J.P. and I.D.W. Danforth, Distinguishing addiction and high engagement in the context of online game playing. Computers in Human Behavior, 2007. 23(3): p. 1531-1548.

[19] Kim, E.J., Namkoon, K., Ku. T., and Kim, S.J., The relationship between online game addiction and aggression, self-control and narcissistic personality traits. Eur Psychiatry, 2008. 23(3): p. 212-8.

[20] Xu, Z., O. Turel, and Y. Yuan, Online game addiction among adolescents: motivation and prevention factors. Eur J Inf Syst, 2012. 21(3): p. 321-340.

[21] Griffiths, M., The Role of Context in Online Gaming Excess and Addiction: Some Case Study Evidence. International Journal of Mental Health and Addiction, 2010. 8(1): p. 119-125.

[22] Skoric, M.M., L.L. Teo, and R.L. Neo, Children and video games: addiction, engagement, and scholastic achievement. Cyberpsychol Behav, 2009. 12(5): p. 567-72.

[23] Smahel, D., L. Blinka, and O. Ledabyl, Playing MMORPGs: connections between addiction and identifying with a character. Cyberpsychol Behav, 2008. 11(6): p. 715-8.

[24] Peters, C.S. and L.A. Malesky, Problematic usage among highly-engaged players of massively multiplayer online role playing games. Cyberpsychol Behav, 2008. 11(4): p. 481-4.

[25] Thamwipat, K. and T. Sawakejun. The study of behaviors and effects of internet technology focus on online game exposure by Thai high school students. in 2nd International Conference on Computer Technology and Development (ICCTD), 2010 2010.

[26] Liu, M. and W. Peng, Cognitive and psychological predictors of the negative outcomes associated with playing MMOGs (massively multiplayer online games). Comput. Hum. Behav., 2009. 25(6): p. 1306-1311.

[27] Vammen, C. and J. Perkins. An Unobserved Societal Issue, Video Game Addiction in the 21st Century. in IEEE International Symposium on Technology and Society (ISTAS), 2007. 2007.

[28] Mehroof, M. and M.D. Griffiths, Online gaming addiction: the role of sensation seeking, self-control, neuroticism, aggression, state anxiety, and trait anxiety. Cyberpsychol Behav Soc Netw, 2010. 13(3): p. 313-6.

[29] Lemola, S., et al., Habitual computer game playing at night is related to depressive symptoms. Personality and Individual Differences, 2011. 51(2): p. 117-122.

[30] Yee, N., Motivations for play in online games. Cyberpsychol Behav, 2006. 9(6): p. 772-5.

[31] Charlton, J.P. and I.D.W. Danforth, Validating the distinction between computer addiction and engagement: online game playing and personality. Behav. Inf. Technol., 2010. 29(6): p. 601-613.

Provenance and Temporally Annotated Logic Programming

Anastasia Analyti
Institute of Computer Science, FORTH-ICS,
Heraklion, Greece

Ioannis Pachoulakis
Dept. of Applied Informatics & Multimedia, TEI of Crete,
Heraklion, Greece

***Abstract*— In this paper, we consider provenance and temporally annotated logic rules (pt-logic rules, for short), which are definite logic programming rules associated with the name of the source that they originate and the temporal interval during which they are valid. A collection of pt-logic rules form a provenance and temporally annotated logic program P, called pt-logic program, for short. We develop a model theory for P and define its maximally temporal entailments of the form A:<S, ti>, indicating that atom A is derived from a set of sources S and holds at a maximal temporal interval ti, according to S. We define a consequence operator that derives exactly the maximally temporal entailments of P for a set of sources. We show that the complexity of the considered entailment is EXPTIME-complete.**

Keywords- Annotated logic programming; provenance and temporal information; model theory; consequence operator.

I. Introduction

Definite logic programming rules traditionally are not associated with the name of the source (provenance information) from which they originate and the temporal interval during which are valid. However, logic programming rules are usually derived from different sources that interact. Additionally logic programming rules may not always be valid, but be valid only for a specific temporal interval.

A temporal interval has the form [*t*,*t'*], where *t*,*t'* are time points and $t \leq t'$. In this paper, time points are years. However, this assumption can be generalized and we may assume any set of time points that can be mapped one-to-one to the set natural numbers. We consider definite logic programming rules associated with a source name and a validity temporal interval, called *provenance and temporally annotated logic rules*, or *pt-logic rules* for short.

We assume that *pt*-logic rules are applied similarly to definite logic programming rules but at each application new provenance and temporal information is derived for the derived atom *A*. In particular, derived atoms (*pt*-atoms) have the form *A*:<*S*, *ti*>, where *ti* is the temporal interval at which *A* is valid, as derived from a set of sources *S*.

Obviously, if *A*:<*S*, *ti*> is true then *A*:<*S'*, *ti*> is true, where $S \subseteq S'$ and *S'* is a subset of a set of considered source names. Additionally, if *A*:<*S*, *ti*> and *A*:<*S'*, *ti'*> are true, where temporal intervals *ti, ti'* are overlapping or consecutive then *A*:<$S \cup S'$, *ti''*> is true, where *ti''* is the combination of *ti* and *ti'*.

A collection of *pt*-logic rules form a *provenance and temporally annotated logic program,* called *pt*-logic program, for short. The models of *pt*-logic programs *P* are defined, as well as, the simple entailments and temporally maximal entailments of *P*. We show that *P* has a minimal model containing exactly the temporally maximal entailments of *P*.

A set of three operators are defined which are applied on *pt*-atoms such that the closure of their composition derives the minimal model of *P*. We define a query language that consists of simple and composite queries, querying provenance and temporal information of derived *pt*-atoms based on certain conditions.

We show that the complexity of simple entailment of a *pt*-atom or temporally maximal entailment of a *pt*-atom from *P* is EXPTIME-complete.

The rest of the paper is organized as follows: In Section II, we define *pt*-logic rules, *pt*-logic programs, *pt*-atoms, and the instantiation of a *pt*-logic program. In Section III, we provide a model theory for *pt*-logic programs *P* and the minimal model of *P* is defined.

In Section IV, we define a consequence operator deriving the minimal model of *P*. In Section V, a query language for *pt*-logic programs is defined. Section VI contains related work. Finally, Section VII contains directions for future work.

II. Provenance And Temporally Annotated Logic Programs

In this Section, we define provenance and temporally annotated logic programs *P*. Additionally, we define the rules based on which the models of *P* are defined. We consider a set of variables *Var*, all preceded by the question mark symbol "?".

Definition 1. A *provenance and temporally annotated logic* rule, called *pt-logic rule* for short, is a definite logic programming rule *r* without function symbols, associated with a source name *nam* and a temporal interval *ti*. In particular, it has the form <*nam*, *ti*>: *r*.

Definition 2. A provenance and temporally annotated logic program P, called pt-logic program for short, is a set of pt-logic rules.

Example 1. The following is a *pt*-logic program *P*.

<*Person*, [1990,1994]>: *has_job*(*Mary*,*Hairdresser*).
<*Person*, [1995,2002]>: *has_job*(*Mary*,*Secretaty*).
<*Person*, [2006,2009]>: *has_job*(*Mary*,*Hairdresser*).
<*Person*,[1999,2002]> : *extra_vacation*(*Mary*).
<*Person*, [2001,2003]>: *has_job*(*Peter*, *Garbage_collector*).
<*Person*, [2005,2008]>: *has_job*(*Peter*,*Bulider*).
<*Job*, [1980,1992]>: *heavy_job*(*Hairdresser*).
<*Job*, [1980, 2001]>: *heavy_job(Garbage_collector*).
<*Job*, [2006, 2012]>: *heavy_job(Builder*).
<*Vacation*, [1988,2012]: *extra_vacation*(?*x*) ←
has_job(?*x*,?*y*), *heavy_job*(?*y*).

Predicate *has_job* indicates the job of a person. Predicate *heavy_job* indicates that a job is heavy and unhealthy. Finally, predicate *extra_vacation* indicates that a person gets for some reason extra vacation days.

C*onvention*: In the following by *P*, we will denote a *pt*-logic program.

We define $Names_P$ to be the set of source names appearing in *P*. Additionally, we define *definite*(*P*) to be the definite logic program derived from *P* after ignoring the provenance and temporal annotations of *P*. Further, we denote by $t_P^{\min}$ the minimum temporal point appearing in *P* and by $t_P^{\max}$ the maximum temporal point appearing in *P*.

Definition 3. A *pt-atom* of *P* is an atom *A* built using predicates and constants appearing in *definite*(*P*) and associated with (i) a set of source names that is a subset of $Names_P$ or a provenance variable *S* and (ii) a temporal interval within $[t_P^{\min}, t_P^{\max}]$ or a temporal variable *ti*. In particular, it has the form *A*: <*S, ti*>.

Below, we define the set of rules [*P*], built from ground *pt*-atoms of *P*, based on which the models of *P* are defined.

Definition 4. Let $r \in P$ be a *pt*-rule $\langle nam, ti\rangle: A_0 \leftarrow A_1, \ldots, A_n$. We define $[r]_P$ to be the rule $B_0{:}\langle S_0, ti_0\rangle \leftarrow B_1{:}\langle S_1, ti_1\rangle, \ldots, B_n{:}\langle S_n, ti_n\rangle$, where (i) B_0 is derived from A_i by replacing all the variables in A_i by constants in *definite*(*P*), (ii) $S_i \subseteq Names_P$ s.t. S_0 is the union of the sets $S_1, \ldots, S_n$ and {*nam*}, and (iii) ti_i is a temporal interval within $[t_P^{\min}, t_P^{\max}]$ s.t. i_0 is the intersection of the temporal intervals $ti_1, \ldots, ti_n$ and *ti*. We define the *instantiation* of *P*, as follows:

$$[P] = \bigcup_{r \in P} [r]_P \ .$$

III. Model Theory of pt-Logic Programs

In this Section, we present the model theory of *pt*-logic programs and entailment of *pt*-atoms.

First, we present three auxiliary definitions. We say that two temporal intervals *ti* and *ti'* are *overlapping* if they have at least one common time point. We say that two temporal intervals $[t_1,t_2]$ and $[t_3,t_4]$ are *consecutive* if $t_3=t_2+1$. We say that a temporal interval $[t_1,t_2]$ is *included* in a temporal interval $[t_3,t_4]$ if $t_3 \le t_1$ and $t_2 \le t_4$.

Definition 4. An *interpretation I* of *P* is a set of ground *pt*-atoms of *P* s.t. (i) if $A{:}\langle S, ti\rangle, A{:}\langle S, ti'\rangle \in I$ then the temporal intervals *ti, ti'* are neither overlapping nor consecutive and (ii) if $A{:}\langle S, ti\rangle \in I$ then, for all *S'* s.t. $S \subseteq S' \subseteq Names_P$, there exists $A{:}\langle S', ti'\rangle \in I$ s.t. *ti* is included in *ti'*.

Below, we define entailment of a *pt*-atom for an interpretation *I* of *P*.

Definition 5. Let *I* be an interpretation of *P* and let *A*:<*S, ti*> be a ground *pt*-atom of *P*. We say that *I* (*simply*) *entails A*:<*S, ti*>, denoted *by I* |= *A*:<*S, ti*>, if there exists an *A*:<*S, ti'*> $\in I$ s.t. *ti* is included in *ti'*.

Additionally, we define an ordering on the interpretations of *P*. Let *I, I'* be interpretations of *P*. We say that $I \le I'$ iff for each $A{:}\langle S, ti\rangle \in I$ there exists an $A{:}\langle S, ti'\rangle \in I$ s.t. *ti* is included in *ti'*. It is easy to see that if $I \le I'$ and $I' \le I$ then $I= I'$.

Below, we define the models if a *pt*-logic program *P*,

Definition 6. Let *M* be an interpretation of *P*. We say that *M* is a *model* of *P* if for each $r= A_0{:}\langle S_0, ti_0\rangle \leftarrow A_1{:}\langle S_1, ti_1\rangle, \ldots, A_n{:}\langle S_n, ti_n\rangle \in [P]$, it holds that: if $M \models A_i{:}\langle S_i, ti_i\rangle$, for all $i=1,\ldots,n$, then $M \models A_0{:}\langle S_0, ti_0\rangle$. We denote the set of models of *P* by $\mathcal{M}_{\mathcal{P}}$.

Let *ti* = [*t,t'*] be a temporal interval. We define *start*(*ti*)=*t* and *end*(*ti*)=*t'*.

Below, we define simple entailment of a ground *pt*-atom from a *pt*-logic program *P*, as well as maximally temporal entailment.

Definition 7. We say that P (simply) entails a ground pt-atom A:<S, ti>, denoted by P |= A:<S, ti>, iff for each $M \in \mathcal{M}_P$, M |= A:<S, ti>. We say that P maximally temporally entails a ground pt-atom A:<S, ti>, denoted by $P \models^{\max}$ A:<S, ti> iff (i) P |= A:<S, ti>, (ii) there exists $M \in \mathcal{M}_P$ s.t. M does not entail A:<S, [end(ti)+1, end(ti)+1]>, and (iii) there exists $M \in \mathcal{M}_P$ s.t. M does not entail A:<S, [start(ti)-1, start(ti)-1]>.

Below, we define the minimal model of a *pt*-logic program *P* w.r.t. ≤.

Definition 8. Let $M_1,..,M_n$ be all the models of *P*, we define the interpretation $M_{min}=\{ A{:}\langle S, ti\rangle \mid$ there exist $A{:}\langle S, ti_i\rangle \in M_i$, for all $i=1,\ldots,n,$ and *ti* is the intersection of $ti_1,\ldots, ti_n\}$.

Below, we show that M_{min} is an interpretation of *P*.

Proposition 1. M_{min} is an interpretation of *P*.

Proof: Obviously, M_{min} is a set of *pt*-atoms of *P*. Let $M_1,..,M_n$ be all the models of *P*. If $A{:}\langle S, ti\rangle \in M_{min}$ then there exists $A{:}\langle S, ti_i\rangle \in M_i$, for all $i=1,\ldots,n,$ and *ti* is the intersection of $ti_1, \ldots, ti_n$. Since $M_1,..,M_n$ interpretations of *P*, it holds that if $A{:}\langle S, ti\rangle \in M_{min}$ then, for all *S'* s.t. $S \subseteq S' \subseteq Names_P$, there exists $A{:}\langle S', ti'\rangle \in M_{min}$ s.t. *ti* is included in *ti'*. Additionally, since $M_1,..,M_n$ interpretations of *P*, there are no $A{:}\langle S, ti\rangle, A{:}\langle S, ti'\rangle \in M_{min}$ s.t. temporal intervals *ti, ti'* are overlapping or consecutive.

Obviously, $M_{min} \le M$, for all $M \in \mathcal{M}_P$.

Below, we show that M_{min} is a minimal model of *P*.

Proposition 2. M_{min} is a minimal model of P.

Proof: Let $M_1,..,M_k$ be all the models of P and let $r = A_0{:}\langle S_0, ti_0\rangle \leftarrow A_1{:}\langle S_1, ti_1\rangle, \ldots, A_n{:}\langle S_n, ti_n\rangle \in [P]$. Assume that $M_{min} \models A_i{:}\langle S_i, ti_i\rangle$, for all i=1,..n. Then, $M_j \models A_i{:}\langle S_i, ti_i\rangle$, for all j=1,..,k and i=1,…,n. Thus, $M_j \models A_o{:}\langle S_0, ti_0\rangle$, for all j=1,…,k. Therefore, $M_{min} \models A_o{:}\langle S_0, ti_0\rangle$.

In Proposition 3, below, we show that P and M_{min} have the same simple entailments.

Proposition 3. Let $A{:}\langle S, ti\rangle$ be a ground *pt*-atom. It holds that: $P \models A{:}\langle S, ti\rangle$ iff $M_{min} \models A{:}\langle S, ti\rangle$.

Proof:

=>) Let $P \models A{:}\langle S, ti\rangle$. Then, for each $M \in \mathcal{M}_P$, $M \models A{:}\langle S, ti\rangle$. This means that for each $M \in \mathcal{M}_P$, there exists $A{:}\langle S, ti'\rangle \in M$ s.t. ti is included in ti'. Thus, by the definition of M_{min}, there exists $A{:}\langle S, ti''\rangle \in M_{min}$ s.t. ti is included in ti''. Thus, $M_{min} \models A{:}\langle S, ti\rangle$.

<=) Assume that $M_{min} \models A{:}\langle S, ti\rangle$. Then, there exists $A{:}\langle S, ti'\rangle \in M_{min}$ s.t. ti is included in ti'. Thus, by the definition of M_{min}, for all $M \in \mathcal{M}_P$, there is $A{:}\langle S, ti''\rangle \in M$ s.t. ti is included in ti''. Therefore, $P \models A{:}\langle S, ti\rangle$.

In Proposition 4, below, we show that M_{min} contains exactly the maximally temporal entailments of P.

Proposition 4. Let $A{:}\langle S, ti\rangle$ be a ground *pt*-atom. It holds that: $P \models^{max} A{:}\langle S, ti\rangle$ iff $A{:}\langle S, ti\rangle \in M_{min}$.

Proof:

=>) Let $P \models^{max} A{:}\langle S, ti\rangle$. This means that for each $M \in \mathcal{M}_P$, there exists $A{:}\langle S, ti'\rangle \in M$ s.t. ti is included in ti'. Additionally, there exists $M \in \mathcal{M}_P$, that does not contain $A{:}\langle S, ti'\rangle \in M$ s.t. time point $end(ti)+1$ is contained in ti' and there exists $M \in \mathcal{M}_P$, that does not contain $A{:}\langle S, ti'\rangle \in M$ s.t. time point $start(ti)-1$ is contained in ti'. Therefore, $A{:}\langle S, ti\rangle \in M_{min}$.

<=) Assume that $A{:}\langle S, ti\rangle \in M_{min}$. This means that for each $M \in \mathcal{M}_P$, there exists $A{:}\langle S, ti'\rangle \in M$ s.t. ti is included in ti'. Additionally, there exists $M \in \mathcal{M}_P$, that does not contain $A{:}\langle S, ti'\rangle \in M$ s.t. time point $end(ti)+1$ is contained in ti' and there exists $M \in M_P$, that does not contain $A{:}\langle S, ti'\rangle \in M$ s.t. time point $start(ti)-1$ is contained in ti'. Therefore, $P \models^{max} A{:}\langle S, ti\rangle$.

IV. COMPUTING THE MINIMAL MODEL OF A PT-PROGRAM

In Section, we provide three operators s.t. the closure of their composition provides the minimal model of a *pt*-logic program P.

We denote by Q_P the ground *pt*-atoms of P. We define the operator W_P from the powerset of Q_P to the powerset Q_P, as follows:

$$W_P(Q)=\{ A{:}\langle S, ti\rangle \mid \text{it exists } A{:}\langle S, ti\rangle \leftarrow A_1{:}\langle S_1, ti_1\rangle, \ldots, A_n{:}\langle S_n, ti_n\rangle \in [P] \text{ s.t. } \langle S_i, ti_i\rangle \in Q, \text{ for all } i=1,\ldots,n\}$$

We now define the operator Z_P from the powerset of Q_P to the powerset of Q_P, as follows:

$$Z_P(Q)=\{ A{:}\langle S, ti\rangle \mid \text{it exists } A{:}\langle S', ti\rangle \in Q \text{ and } S' \subseteq S \subseteq Names_P\}$$

Before, we define the third operator, we provide a few definitions. Let $Q \subseteq Q_P$. We define $intervals(Q,S,A)=\{ti \mid \text{it exists } A{:}\langle S, ti\rangle \in Q\}$. Let TI be a set of temporal intervals. The *maximal subset* of TI w.r.t. a temporal interval $ti \in TI$ is a set B that (i) contains ti and (ii) if $ti' \in B$ and there exists a $ti'' \in TI$ s.t. ti' and ti'' are overlapping or consecutive then $ti'' \in B$. The *maximal interval* of TI w.r.t. a temporal interval $ti \in TI$, denoted by $max\_interval(TI,ti)$, is the temporal interval $[t,t']$ formed by the minimum time point t and maximum time point t' appearing in a maximal subset of TI w.r.t. ti.

We are now ready to define the operator R_P from the powerset of Q_P to the powerset of Q_P as follows:

$$R_P(Q)=\{ A{:}\langle S, ti\rangle \mid \text{it exists } A{:}\langle S, ti'\rangle \in Q \text{ and } ti= max\_interval(intervals(Q,S,A),ti')\}$$

Finally, we define the consequence operator T_P from the powerset of Q_P to the powerset of Q_P as the composition of W_P, Z_P, and R_P. In particular, $T_P(Q)= R_P(Z_P(W_P(Q)))$, for $Q \subseteq Q_P$.

It can be easily seen that the consequence operator T_P is monotonic with respect to $\leq$. That is, if $Q,Q' \subseteq Q_P$ s.t. $Q \subseteq Q'$ then $T_P(Q) \leq T_P(Q')$. We will show that the closure of operator T_P coincides with M_{min}.

Proposition 4. Let M be an interpretation of P. It holds that M is a model of P iff $T_P(M) \leq M$.

Proof:

=>) Let M be a model of P. Consider the rules $A{:}\langle S, ti\rangle \leftarrow A_1{:}\langle S_1, ti_1\rangle, \ldots, A_n{:}\langle S_n, ti_n\rangle \in [P]$ s.t. $A_i{:}\langle S_i, ti_i\rangle \in M$, for all $i=1,\ldots,n$. Then, $M \models A{:}\langle S, ti\rangle$. Since M is an interpretation of P, M satisfies each *pt*-atom in $T_P(M)$. Thus, $T_P(M) \leq M$.

<=) Assume that $T_P(M) \leq M$. We need to show that if $A{:}\langle S, ti\rangle \leftarrow A_1{:}\langle S_1, ti_1\rangle, \ldots, A_n{:}\langle S_n, ti_n\rangle \in [P]$ s.t. $M \models A_i{:}\langle S_i, ti_i\rangle$, for all i=1,…,n, then $M \models A{:}\langle S, ti\rangle$. Note that there exists $A{:}\langle S, ti'\rangle \leftarrow A_1{:}\langle S_1, ti'_1\rangle, \ldots, A_n{:}\langle S_n, ti'_n\rangle \in [P]$ s.t. $A_i{:}\langle S_i, ti'_i\rangle \in M$, for all i=1,…,n.. Note that $T_P(M)$ is an interpretation of P and that $T_P(M) \models A{:}\langle S, ti'\rangle$. Since $T_P(M) \leq M$, it follows that $M \models A{:}\langle S, ti'\rangle$. Now, since ti is included in ti', it follows that $M \models A{:}\langle S, ti\rangle$. Thus, M is a model of P.

Below, we show that M_{min} can be computed as the closure of the T_P operator.

Proposition 5. $M_{min}=T_P^{\uparrow\omega}(\{\})$.

Proof: From Proposition 4, it follows that $M_{min}= minimal_{\leq}(\{M \mid M \text{ is an interpretation of } P \text{ and } T_P(M) \leq M\})$. Since M_{min} is a model of P, it follows from Proposition 4 that $T_P(M_{min}) \leq M_{min}$. Further note that since the operator T_P is monotonic, it follows that $T_P(T_P(M_{min})) \leq T_P(M_{min})$. Additionally, $T_P(M_{min})$ is an interpretation of P. Thus, $M_{min} \leq T_P(M_{min})$. Therefore, $T_P(M_{min})=M_{min}$. Thus, $M_{min}=T_P^{\uparrow\omega}(\{\})$. Note that $T_P^{\uparrow\omega} \subseteq Q_P$.

Proposition 6. Simple and maximally temporal entailment of a ground *pt*-atom from a *pt*-logic program P is EXPTIME-complete w.r.t. the size of P.

Proof: *Membership*) Note that the computation of $M_{min}=T_P^{\uparrow\omega}(\{\})$, involves the application of at most an exponential number of rules applied at most exponential number of times. Obviously, the computation of *max_interval*(*intervals*(*Q*,*S*,*A*)*,ti'*) is polynomial w.r.t. the size of *Q*. Therefore, the computation of M_{min} is in EXPTIME.

Hardness) It follows directly by the fact that datalog is program-complete for EXPTIME [1].

V. A Query Language for pt-logic Programs

Below, we define the queries that can be applied to a *pt*-logic program *P*.

A simple *pt*-query of type 1 has the form *SQ*=*A*:<?*S*, *ti*>, where *A*:<?*S*, *ti*> is a *pt*-atom of *P*, *ti* is a temporal interval, and ?*S* is a provenance variable. The answers of *SQ* w.r.t. *P*, denoted by $Ans_P(SQ)$, is the set of mappings *v* from the variables of *A* to the constants of *definite*(*P*) and from ?*S* to a subset of $Names_P$ s.t. *P* |= *v*(*A*:<?*S*, *ti*>).

A simple *pt*-query of type 2 has the form *SQ*=*A*:<?*S*, ?*ti*>, where *A*:<?*S*, ?*ti* > is a *pt*-atom of *P* (note that ?*S* is a provenance variable and ?*ti* is a temporal variable). The answers of *SQ* w.r.t. *P*, denoted by $Ans_P(SQ)$, is the set of mappings *v* from the variables of *A* to the constants of *definite*(*P*), from ?*S* to a subset of $Names_P$, and from ?*ti* to a temporal interval s.t. *P* $|=^{max}$ *v*(*A*:<?*S*, ?*ti*>).

A simple *pt*-query of type 3 has the form *SQ*=A_1:<?S_1, ?*ti*> $\wedge \ldots \wedge A_n$:<?S_n, ?*ti*> where each A_i:<?S_i, ?*ti* > is a *pt*-atom of *P*. The answers of *SQ* w.r.t. *P*, denoted by $Ans_P(SQ)$, is the set of mappings *v* s.t. if $u_i \in Ans_P(A_i$:<?S_i, ?*ti*>) s.t. $u_1,\ldots,u_n$ coincide on the common variables of A_i and ?S_i then *v* coincides with u_i on the variables of A_i and ?S_i and *v*(?*ti*) is the intersection of u_1(?*ti*),…, u_n(?*ti*), if such intersection exists.

A simple *pt*-query of type 4 has the form *SQ*=A_1<?S_1, ?*ti*> $\wedge \ldots \wedge$ A_n:<?S_n, ?*ti*> $\wedge$ *included*(?*ti*, [*t*,*t'*]), where each A_i:<?S_i, ?*ti* > is a *pt*-atom of *P* and [*t*,*t'*] is a temporal interval. The answers of *SQ* w.r.t. *P*, denoted by $Ans_P(SQ)$, is the set of mappings *v* s.t. if $u_i \in Ans_P(A_i$:<?S_i, ?*ti*>) s.t. $u_1,\ldots,u_n$ coincide on the common variables of A_i and ?S_i then *v* coincides with u_i on the variables of A_i and ?S_i and *v*(?*ti*) is the intersection of u_1(?*ti*),…, u_n(?*ti*) and [*t*,*t'*], if such intersection exists.

A complex *pt*-query has the form $CQ=SQ_1 \wedge \ldots \wedge SQ_n \wedge$ *filter*, where SQ_i are simple *pt*-queries, each having a different temporal variable, and *filter* is an expression of the following EBNF grammar:

term:=*duration*(?*ti*) | *start*(?*ti*) | *end*(?*ti*) | *c*, where ?*ti* is a temporal variable and *c* is a decimal.

complex_term:= term | complex_term (+ | - | * | /) complex_term

temp_comparison:= complex_term (< | > | = | ≤ | ≥ | ≠) complex_term

prov_comparison:= (?*S* (⊂ | ⊃ | ⊆ | ⊇ | =) ?*S'*) | (?*S* (⊂ | ⊃ | ⊆ | ⊇ | =) *S''*) , where ?*S*, ?*S'* are provenance variables and $S'' \subseteq Names_P$.

filter:= temp_comparison | prov_comparison | *filter* (∧ | ∨) filter,

such that each provenance and temporal variable appearing in *filter* appears in $SQ_1 \wedge \ldots \wedge SQ_n$.

The answers of *CQ* w.r.t. *P*, denoted by $Ans_P(CQ)$, is the set of mappings *v* s.t. (i) if $u_i \in Ans_P(SQ_i)$ s.t. $u_1,\ldots,u_n$ coincide on the common variables of SQ_i then *v* coincides with u_i on the variables of SQ_i and (ii) *v*(*filter*) holds.

Consider a simple or complex query *CQ*. Let *v*,*u* ∈ $Ans_P(CQ)$. We say that answer *v* is *more informative* than answer *u*, denoted by $v \leq u$, if (i) *v*, *u* coincide on their no provenance variable mappings, and (ii) for each provenance variable ?*S* ∈ *domain*(*v*), it holds that *v*(?*S*) ⊆ *u*(?S). We define $Answer_P(CQ)=minimal_{\leq}(Ans_P(CQ))$ and we consider that these are the *desired answers*. This is because if *P* |= *A*:<*S*, *ti*> then *P* |=*A*:<*S'*, *ti*>, for each $S \subseteq S' \subseteq Names_P$.

Example 2. Consider the simple query *SQ* = *extra_vacation*(*Peter*): <?*S*, ?*ti*>. Then, the answers to this query is (i) the mapping *v*, where *v*(?*S*)={*Person*, *Job*, *Vacation*} and *v*(*?ti*)= [2001,2001], and (ii) the mapping *u*, where *u*(?*S*)={*Person*, *Job*, *Vacation*} and *u*(*?ti*)=[2006,2008].

Consider now the simple query *SQ* = *extra_vacation*(*Mary*):<?*S*, ?*ti*>. Then, the answers to this query is (i) the mapping *v*, where *v*(?*S*)={*Person*} and *v*(*?ti*)= [1999,2002], and (ii) the mapping *u*, where *u*(?*S*)={*Person*, *Job*, *Vacation*} and *u*(*?ti*)=[1900,1992].

Consider now the simple query *SQ* = *extra_vacation*(*Mary*):<?*S*, ?*ti*> ∧ *extra_vacation*(*Peter*):<?*S'*,?*ti*>, requesting the common temporal intervals that Mary and Peter get extra vacation days, independently of the set of sources that this information is derived. Then, the answer to this query is the mapping *v*, where *v*(?*ti*)=[2001,2001], *v*(?*S*)={*Person*} and *v*(?*S'*)={*Person*, *Job*, *Vacation*}.

Consider now the simple query *SQ* = *extra_vacation*(*Mary*):<?*S*, ?*ti*> ∧ *extra_vacation*(*Peter*):<?*S*, ?*ti*>, requesting the common temporal intervals that Mary and Peter get extra vacation days, as derived from the same set of sources. Then, the answer to this query is the mapping *v*, where *v*(?*ti*)=[2001,2001] and *v*(?*S*)={*Person*, *Job*, *Vacation*}.

Consider now the complex query *CQ*= *has_job*(*Mary*, ?*x*):<*?S*, *?ti*> ∧ *included*(*?ti*, [1993,2008]) ∧ ?*S*={*Person*} , requesting the temporal intervals that Mary has a job from 1993 to 2008, as derived from the source *Person*. The answers to this query is (i) the mapping *v*, where *v*(?*ti*)=[1993, 2002] and *v*(?*S*)={*Person*} and (ii) the mapping *u*, where *u*(?*ti*)=[2006,2008] and *u*(?*S*)={*Person*}.

Consider now the complex query CQ = extra_vacation(Mary): <?S, ?ti> ∧ extra_vacation(Peter):<?S, ?ti'> ∧ start(?ti) ≤ end(?ti') ∧ start(?ti') ≤ end(?ti') , requesting the temporal intervals that Mary and Peter get extra vacation days, as long as these overlap, and as derived from the same set of sources.

Then, the answer to this query is the mapping v, where v(?ti)=[2001,2001], v(?ti')=[1999,2002], and v(?S)={Person, Job, Vacation}.

VI. Related Work

In this Section, we discuss related work.

Flouris et al. [2] add provenance information to RDF theory [3], [4]. In particular, they extend RDF triples to RDF quadruples, where the fourth element is the set of graph names that participated in the derivation of the RDF triple through a limited subset of the RDFS entailment rules. They also discuss atomic update operations (i.e. inserts and deletes) of the RDF quadruples. Comparing to this paper, [2] does not present a model theory and does not consider the temporal domain. Note that our approach can also be applied to RDFS, as RDFS inference rules can be expressed through definite logic programming rules [5].

In [6], we extend extended logic programming rules [6] with their validity temporal intervals. We consider derivations based on temporal time points and Answer Set Programming [7], and provide an algorithm that returns the maximal temporal intervals that a literal is true. Present work has a different model theory and implementation than [6], as we consider only definite logic programming rules and on the other hand we also consider the provenance domain. Yet, the query language presented here is an extension of the query language presented in [6].

In [8], the authors present a general framework for representing, reasoning, and querying with RDFS annotated data on the Semantic Web. They show that their formalism can be instantiated on the temporal, fuzzy, and provenance domain. The authors can associate RDF triples with their validity temporal intervals and supportive sources and apply the RDFS inference rules (which are always valid). Yet, [8] does not support simple queries of type 4. Moreover, our query answering is more efficient, since during query answering, we directly work on maximal temporal intervals. In [8], all entailed temporal intervals returned by the query are considered and then the maximal ones are returned. Further, our semantics is different than [8]. For example, consider the annotated RDF triples *p*(*a*,*b*) :<*n*, [1990, 2000]> and *q*(*c*,*d*): <*n*, [1995, 2010]>. Then, according to [8], the answer to query *p*(*a*,*b*):<?*S*, ?*ti*> ∧ *q*(*c*,*d*): <?*S*, ?*ti'*> ∧ *end*(?*ti*) < *start*(?*ti'*) will provide the mapping *v* s.t. v(?*ti*)=[1990,1994] and *v*(?*ti'*)=[1995, 2010]. In our case, we will provide no answers, since 2000 > 1995.

In [9], the authors present a framework to incorporate temporal reasoning into RDFS. The authors associate RDF triples with their validity temporal interval and apply the RDFS inference rules (which are always valid). Unlike our work, their semantics is based on time points and not on temporal intervals. Additionally, [9] does not consider the provenance domain. Further, it does not support simple queries of type 3 and type 4 and the filter condition is limited.

In [10], the authors extend RDF graphs with temporal information, by associating RDF triples with their validity interval. They consider any entailment regime that can be expressed through definite rules $A_0 \leftarrow A_1, \ldots, A_n$, where A_i is an RDF triple. Each such rule is replaced by the temporal rule A_0:[$max(t_1,..., t_n)$, $min(t'_1,..., t'_n)$] $\leftarrow A_1$:[t_1, t'_1],..., A_n:[t_n, t'_n]. These rules are applied recursively, until a fixpoint is reach. Then, maximal validity temporal intervals for each derived RDF triple are produced. Yet, this work does not present a model theory based on temporal intervals and does not consider the provenance domain. Additionally, it does not support simple temporal queries of type 4 and the filter condition is left unspecified.

Work in [11] provides a framework to support spatial and temporal analysis over RDFS data. With respect to the temporal component, [11] is similar to [10], as it also computes the maximal validity temporal intervals of derived RDF triples, using the RDFS entailment rules. Yet, [11] does not consider the provenance domain.

Finally, we would like to note that our theory cannot be considered as a special case of annotated logic programming [12], as the model theory and the operational semantics are different there.

VII. Conclusion

In this paper, we have presented a model theory and the operational semantics of a *pt-logic* program *P*, that is a set of definite logic programming rules, annotated with the source that have been derived and their validity temporal interval. We have defined the simple and maximally temporal entailments of *P*, showing that there exists a minimal model M_{min} that contains exactly these maximal entailments. Additionally, we defined a consequence operator whose closure coincides with M_{min}. Further, we showed that simple and maximally temporal entailment from a *pt*-logic program *P* is EXPTIME-complete w.r.t. the size of *P*. A query language for our framework is proposed.

As future work, we plan to extend our theory to extended logic programs. Further, we plan to add additional parameters to definite logic programming rules such that space and trust.

References

[1] Evgeny Dantsin, T. Eiter, G. Gottlob, and A. Voronkov: “Complexity and expressive power of logic programming”. ACM Comput. Surv. 33(3),2001,pp. 374-425.

[2] G. Flouris, I. Fundulaki, P. Pediaditis, Y. Theoharis , and V. Christophides}, "Coloring RDF Triples to Capture Provenance", 8th International Semantic Web Conference (ISWC-2009), 2009, pp. 196-212.

[3] G. Klyne and J. J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax”, W3C Recommendation, 10 February 2004, available at http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[4] Patrick Hayes, "RDF Semantics", W3C Recommendation, 10 February 2004, available at http://www.w3.org/TR/2004/REC-rdf-mt-20040210/.

[5] G. Ianni, A. Martello, C. Panetta, and G. Terracina, "Efficiently Querying RDF(S) Ontologies with Answer Set Programming", Journal of Logic and Computation, 19(4), 2009, pp. 671-695.

[6] A. Analyti and I. Pachoulakis, “Temporally Annotated Extended Logic Programs”, accepted to International Journal of Advanced Research in Artificial Intelligence (IJARAI), 2012.

[7] M. Gelfond and V. Lifschitz, "Classical Negation in Logic programs and Disjunctive Databases", New Generation Computing, 9, 1991, pp. 365-385.

[8] A. Zimmermann, N. Lopes, A. Polleres, and U. Straccia, "A General Framework for Representing, Reasoning and Querying with Annotated Semantic Web Data", Journal of Web Semantics , 11, 2012, pp. 72–95.

[9] C. Gutierrez, C. A. Hurtado, and A. A. Vaisman, "Introducing Time into RDF", IEEE Transactions on Knowledge and Data Engineering, 19(2), 2007, 207-218.

[10] B. Motik, "Representing and Querying Validity Time in RDF and OWL: A Logic-Based Approach", 9th International Semantic Web Conference (ISWC-2010), 2010, pp. 550-565.

[11] M. Perry, A. P. Sheth, F. Hakimpour, and P. Jain, "Supporting Complex Thematic, Spatial and Temporal Queries over Semantic Web Data", 2nd International Conference on GeoSpatial Semantics (GeoS-2007), 2007, pp. 228-246.

[12] Michael Kifer and V. S. Subrahmanian, "Theory of Generalized Annotated Logic Programming and its Applications", Journal of Logic Programming, 12(3&4), 1992, pp. 335-367.

AUTHORS PROFILE

Anastasia Analyti earned a B.Sc. degree in Mathematics from University of Athens, Greece and a M.Sc. and Ph.D. degree in Computer Science from Michigan State University, USA. She worked as a visiting professor at the Department of Computer Science, University of Crete, and at the Department of Electronic and Computer Engineering, Technical University of Crete. Since 1995, she is a principal researcher at the Information Systems Laboratory of the Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH-ICS). Her current interests include reasoning on the Semantic Web, modular web rule bases, non-monotonic-reasoning, faceted metadata and semantics, conceptual modelling, contextual organization of information, information integration and retrieval systems for the web, interoperability of heterogeneous and distributed information bases, and biomedical information systems. She has participated in several research projects and has published over 55 papers in refereed scientific journals and conferences.

Ioannis Pachoulakis received a B.Sc. in Physics (1988) at the University of Crete, Greece, and a Ph.D. in Astrophysics (1996) and an M.Sc. in Engineering (1998), both from the University of Pennsylvania in the U.S.A. Since 2001, he serves as an Assistant Professor at the Department of Applied Informatics and Multimedia at TEI of Crete with mainstream interests in realistic multimedia applications, virtual reality and multimedia applications for science.

Temporally Annotated Extended Logic Programs

Anastasia Analyti
Institute of Computer Science, FORTH-ICS,
Heraklion, Greece

Ioannis Pachoulakis
Dept. of Applied Informatics & Multimedia, TEI of Crete,
Heraklion, Greece

***Abstract*— Extended logic programs (ELPs) are a set of logic rules with strong negation ¬ allowed in the bodies or head of the rules and weak negation ~ allowed in the bodies of the rules. ELPs enable for various forms of reasoning that cannot be achieved by definite logic programs. Answer Set Programming provides a widely acceptable semantics for ELPs. However, ELPs do not provide information regarding the temporal intervals that derived ELP literals or weakly negated ELP literals are valid. In this paper, we associate ELP rules with their validity temporal interval, resulting in a *temporally annotated logic program*. A ground temporal literal has the form *L*:*i*, where *L* is a ground ELP literal or weakly negated ELP literal and *i* is a temporal interval. We define (simple) entailment and maximal entailment of a ground temporal literal *L*:*i* from a temporally annotated logic program *C*. Both kinds of entailment are based on Answer Set Programming. Additionally, we provide an algorithm that for an ELP literal or a weakly negated ELP literal *L* returns a list with all temporal intervals *i* such that a temporally annotated logic program *C* maximally entails *L*:*i*. Based on this algorithm, the answer of various kinds of temporal queries can be provided.**

Keywords- Extended logic programs; validity temporal intervals; temporal inference; query answering.

I. INTRODUCTION

Extended logic programs (ELPs) are a set of logic rules with strong negation ¬ allowed in the bodies or head of the rules and weak negation ~ allowed in the bodies of the rules [1]. ELP rules are assumed to be valid at the time of their evaluation but no historical information is derived as it is not known if these rules were valid in the past. However, each ELP rule is usually valid at a certain interval of time. In this paper, we associate ELP rules with their validity temporal interval. Thus, derived ELP literals and weakly negated ELP literals are associated with the temporal intervals at which they are valid. A temporal interval has the form $[t_1,t_2]$, where t_1,t_2 are time points s.t. $t_1 \leq t_2$ and it includes all time points t s.t. $t_1 \leq t \leq t_2$ In this paper, time points are years but they can also be dates, date times, etc. In general, we assume that the set of time points is a discrete linearly ordered domain that can be mapped to the set of integers through a bijective mapping.

In particular, ELP logic rules, associated with their validity temporal interval, form a *temporally annotated logic program*. A ground temporal literal has the form *L*:*i*, where *L* is a ground ELP literal or weakly negated ELP literal and *i* is a temporal interval. We define (*simple*) *entailment* of a ground temporal literal *L*:*i* from a temporally annotated logic program *C* expressing that according to *C*, *L* is true during all time points within *i*. We define *maximal entailment* of a ground temporal literal *L*:*i* from a temporally annotated logic program *C*, expressing that according to *C*, *L* is true during all time points within *i* but *L* is not true at the time point before *i* and at the time point after *i*. Both kinds of entailment are based on Answer Set Programming [1]. The complexities of simple and maximal entailment of ground temporal literals are provided showing that simple temporal entailment is not harder than entailment of a ground ELP literal from an ELP logic program, based on Answer Set Programming.

We provide an algorithm that for an ELP literal or a weakly negated ELP literal *L* returns a list with all temporal intervals *i* such that a temporally annotated logic program *C* maximally entails *L*:*i*. Based on this algorithm, the answer of various kinds of temporal queries can be provided. For example, the user may request the maximal temporal intervals that a number of ELP literals or weakly negated ELP literals hold concurrently within a temporal interval of interest. Additionally, the user may request the maximal temporal intervals that a number of ELP literals or weakly negated ELP literals hold, provided that these intervals are associated with complex relations concerning their duration, start and end points. In particular, we define four types of simple temporal queries. Additionally, we define a complex temporal query as a conjunction of simple temporal queries and a *filter* condition that provides for various kinds of checks regarding the duration, start and end points of the temporal intervals returned by the query. The filter condition can express any combination of Allen's interval algebra relations [2].

To the best of our knowledge, there is no work concerning entailment from ELP rules associated with their validity temporal intervals. The rest of the paper is organized as follows: In Section II, we define temporally annotated logic programs and entailment of temporal literals. Additionally, we provide an algorithm that for an ELP literal or weakly negated ELP literal *L* returns a list with all temporal intervals *i* such that a temporally annotated logic program *C* maximally entails *L*:*i*. In Section III, we define various kinds of temporal queries and their answers. In Section IV, we present related work. Finally, Section V concludes the paper.

II. TEMPORALLY ANNOTATED LOGIC PROGRAMS & ENTAILMENT

In this section, we define temporally annotated logic programs and entailment of temporal literals.

We consider a vocabulary *V*=<*Pred*,*Const*>, where *Pred* is a set of predicate symbols and *Const* is a set of constants. We consider a set of variable symbols *Var*. Variables are preceded by the question mark symbol "?". Additionally, we consider a maximal temporal interval $[t_{min}, t_{max}]$, within which all temporal inferences are made.

Definition 1 [temporally annotated logic rule]

A *temporally annotated logic rule* over a vocabulary V is an ELP rule over V associated with a temporal interval i, i.e. it has the form i:r where r is an ELP rule and $i=[t_1,t_2]$, where $t_1 \leq t_{min}$ and $t_2 \leq t_{tmax}$.

Definition 2 [temporally annotated logic program]

A *temporally annotated logic program* C over a vocabulary V is a set of temporally annotated logic rules over V.

Example 1 An example temporally annotated logic program C over V=<*Pred*, *Const*>, where *Pred* are all predicates appearing in C and *Const* are all the constants appearing in C, is the following (we consider t_{min}=1988 and t_{max}=2012):

[1990,1994]: *has_job*(*Mary, Hairdresser*).
[1995,2002]: *has_job*(*Mary, Secretary*).
[2006,2009]: *has_job*(*Mary, Hairdresser*).
[2001,2003]: *has_job*(*Peter, Garbage_collector*).
[2005,2008]: *has_job*(*Peter, Builder*).
[1980,1992]: *heavy_job*(*Hairdresser*).
[1980,2000]: *heavy_job*(*Garbage_collector*).
[1999,2012]: *heavy_job*(*Builder*).
[1988, 1991]: *vacation_days*(?*x*,29) ← *has_job*(?*x*,?*y*), *heavy_job*(?*y*).
[1992, 2012]: *vacation_days*(?*x*,27) ← *has_job*(?*x*,?*y*), *heavy_job*(?*y*).
[1988, 1991]: *vacation_days*(?*x*,25) ← *has_job*(?*x*,?*y*), ~ *heavy_job*(?*y*).
[1992, 2012]: *vacation_days*(?x,22) ← *has_job*(?*x*,?*y*), ~ *heavy_job*(?*y*).

Predicate *has_job*(x,y) expresses that x has as job y. Predicate *heavy_job*(x) expresses that x is a heavy and unhealthy job. Predicate *vacation_days*(x,y) expresses that x is entitled to y vacations days per year.

Convention: In the sequel, by C, we will denote a temporally annotated logic program and, by V=<*Pred*,*Const*>, we will denote the vocabulary of C.

Below, we define the temporal projection of C at a certain time point t.

Definition 3 [temporal projection]

The temporal projection of C at a time point t is the extended logic program $C(t)=\{r \mid i{:}r \in C \text{ and } t \in i\}$.

Example 2 Consider the temporally annotated logic program C of Example 1. Then, C(1994) is the following extended logic program:

has_job(*Mary, Hairdresser*).
heavy_job(*Garbage_collector*).
vacation_days(?*x*,27) ← *has_job*(?*x*,?y), *heavy_job*(?*y*).
vacation_days(?*x*,22) ← *has_job*(?*x*,?*y*), ~ *heavy_job*(?*y*).

A temporal literal over V has the form L:i, where L is an ELP literal or a weakly negated ELP literal over V and i a temporal interval or variable, called *temporal variable*.

Let P be an extended logic program of a vocabulary V=<*Pred*,*Const*> and L be an ELP literal or a weakly negated ELP literal over V, we write $P \models^{ASP} L$ if the grounded version of P over the constants in *Const* entails L under Answer Set Programming [1].

Definition 4 [temporal entailment]

We say that C (*simply*) *entails* a ground temporal literal L:i, denoted by $C \models L{:}i$, if for all $t \in i$, $C(t) \models^{ASP} L$. We say that C *maximally entails* a temporal literal $L{:}[t_1,t_2]$, denoted by

$C \models_{\max} L{:}[t_1,t_2]$, if

1. $C \models L{:}[t_1,t_2]$,
2. if $t_1 > t_{min}$ then $C(t_1\text{-}1) \not\models^{ASP} L$, and
3. if $t_2 < t_{max}$ then $C(t_2+1) \not\models^{ASP} L$.

Example 3 Consider the temporally annotated logic program C of Example 1. Then, $C \models$ *vacation_days*(*Mary*,22): [2006,2008] and $C \models_{\max}$ *vacation_days*(*Mary*,22):[2006, 2009]. Additionally, $C \models$ *vacation_days*(*Peter*,27):[2006, 2007] and $C \models_{\max}$ *vacation_days*(*Peter*,27): [2005,2008].

Proposition 1 Let L:i be a ground temporal literal over V. Deciding if $C \models L{:}i$ is:

1. co-NP-complete, in the case that C does not contain variables or the number of variables of each rule of C is less than a constant.
2. co-NEXPTIME-complete, in the general case.

Note that entailment of a ground ELP literal from an ELP program P, under Answer Set Programming is (i) co-NP-complete, in the case that P does not contain variables or the number of variables of each rule of P is less than a constant and (ii) co-NEXPTIME-complete, in the general case [3]. Therefore, entailment of a ground temporal literal from a temporally annotated logic program does not increase the computational complexity over Answer Set Programming.

Now, we provide a few definitions. Let $i=[t_1, t_2]$ be a temporal interval. We define $start(i)=t_1$ and $end(i)=t_2$. Let L be an ELP literal or a weakly negated ELP literal, we denote by *pred*(L), the predicate appearing in L. Let r be an ELP rule. We denote by *Head*(r), the head of r.

Below, we present the algorithm *FindMaximalIntervals*(C, L) that, for an ELP literal or a weakly negated ELP literal L, returns a list with all temporal intervals i such that $C \models_{\max} L{:}i$. This algorithm calls the algorithm *GetIntervals*(C,p) which returns a list of the maximal temporal intervals i that define predicate p, i.e. for all $t \in i$, p appears in the head of a rule of $C(t)$. The list of returned temporal intervals i is sorted by *start*(i).

Algorithm 1 *GetIntervals*(C,p), where $p \in$ *Pred*, first gets all intervals i s.t. there exist rule $i{:}r \in C$ with *pred*(*Head*(r)) =p. Then, it orders these intervals i based on *start*(i) and puts them in a list *IL*.

Afterwards, it fetches intervals from *IL* in the stored order and combines the intervals that overlap or are consecutive, creating maximal intervals that puts them in a new list *IL'*. Finally, it returns *IL'*.

Algorithm 1 GetIntervals(C,p)

Input: a temporally annotated logic program *C* and a predicate $p \in Pred$

Output: a sorted list of the maximal temporal intervals *i* that define *p*

Let $S=\{i \mid i{:}r \in C$ and $p=pred(Head(r))\}$.
If $S=\{\}$ then return(*empty list*).
Order intervals $i \in S$ by *start*(*i*) and put them in list *IL*.
Let *IL'* be the empty list.
Get first interval *i* from *IL*.
current_start=*start*(*i*).
current_end=*end*(*i*).
For each interval *i* taken in sequence from the list *IL* do
If *start*(*i*) ≤ *current_end*+1 and *end*(*i*)>*current_end* then
current_end=*end*(*i*).
If *start*(*i*) > *current_end*+1 or *i* is the last interval in
IL then
Add at the end of *IL'* the temporal interval
[*current_start*, *current_end*].
current_start=*start*(*i*).
current_end=*end*(*i*).
If *start*(*i*) > *current_end*+1 and *i* is the last interval in *IL*
then
Add at the end of *IL'* the temporal interval
[*start*(*i*), *end*(*i*)].
EndFor
return(*IL'*).

Example 4 Consider the temporally annotated logic program *C* of Example 1. Then, *GetIntervals*(*C*, *has_job*) returns the list [[1990,2003], [2005,2009]].

Algorithm 2 *FindMaximalIntervals*(*C*,*L*), where *L* is an ELP literal or a weakly negated ELP literal over *V*, first calls algorithm *GetIntervals*(*C*, *pred*(*L*)) which returns a list *IL* of maximal intervals *i* that define *pred*(*L*), ordered by *start*(*i*) (line 2). Assume that *IL* is the empty list (lines 3-7). If *L* is a weakly negated literal then the algorithm returns the list containing $[t_{min}, t_{max}]$. This is because in this case $C \models_{max} L{:}[t_{min}, t_{max}]$. Otherwise, the algorithm returns the empty list. This is because in this case, there is no interval *i* s.t $C \models_{max} L{:}i$. If *IL* is not the empty list then the algorithm fetches the first interval *i* of *IL* and if *L* is a weakly negated ELP literal and $start(i) \neq t_{min}$ then it puts the interval $[t_{min}, start(i)\text{-}1]$ in a new list *IL'* (line 11). This is because for all time points $t \in [t_{min}, start(i)\text{-}1]$, it holds that $C(t) \models^{ASP} L$. Then, it fetches each interval $[t_s, t_e]$ from *IL* in the stored order.

Afterwards, it examines each time point $t \in [t_s, t_e]$ and creates maximal temporal intervals *i* within $[t_s, t_e]$ such that for each $t' \in i$, it holds that $C(t) \models^{ASP} L$ (lines 13-21). These intervals are stored in order in *IL'* with the difference that if the first of these intervals is consecutive with the previous interval in *IL'* then it combines these. Note that if $[t_s, t_e]$ is the last temporal interval in *IL*, *L* is a weakly negated literal, and $t_e < t_{max}$ then, for each $t \in [t_e+1, t_{max}]$, it holds that $C(t) \models^{ASP} L$. Thus, if $C(t_e) \models^{ASP} L$ then it combines interval $[t_e+1, t_{max}]$ with the last interval in *IL'* (line 25), otherwise it adds interval $[t_e+1, t_{max}]$ to the end of *IL'* (line 27). If $[t_s, t_e]$ is not the last temporal interval in *IL* and *L* is a weakly negated literal then (i) if $C(t_e) \models^{ASP} L$ and $[t'_s, t_e]$ is the last interval in *IL'* then it replaces it by the interval $[t'_s, t''_s\text{-}1]$ (line 31) and (ii) if $C(t_e) \not\models^{ASP} L$ then it adds $[t_e+1, t''_s\text{-}1]$ to *IL'* (line 33), where $[t''_s, t''_e]$ is the next to $[t_s, t_e]$ interval in *IL*. This, is because for all $t \in [t_e+1, t''_s\text{-}1]$, it holds that $C(t) \models^{ASP} L$. Then, this process continues until all intervals $[t_s, t_e]$ from *IL* are fetched. In the latter case, the temporal interval list *IL'* is returned.

Algorithm 2 FindMaximalIntervals(C,L)

Input: a temporally annotated logic program *C* and an

ELP literal or a weakly ELP literal *L* over *V*

Output: a sorted list of maximal temporal intervals *i* s.t.

for each $t \in i$, $C(t) \models^{ASP} L$

(1) *flag*=TRUE.
(2) *IL*= *GetIntervals*(*C*, *pred*(*L*)).
(3) If *IL* is the empty list then
(4) If *L* is a weakly negated ELP literal then
(5) return(*a list containing* $[t_{min}, t_{max}]$).
(6) else
(7) return(*empty list*).
(8) Let *IL'* be the empty list.
(9) Let *i* be the first item in *IL*.
(10) If *L* is a weakly negated ELP literal and $start(i) \neq t_{min}$ then
(11) Add $[t_{min}, start(i)\text{-}1]$ to the end of *IL'*.
(12) For each interval $[t_s, t_e]$ taken in sequence from the list *IL* do
(13) For each time point $t=t_s, \ldots, t_e$ do
(14) If $C(t) \models^{ASP} L$ then
(15) If *IL'* is not empty and *flag*=TRUE then
(16) Take the last item $[t'_s, t'_e]$ from *IL'* and replace it by $[t'_s, t]$.
(17) else
(18) Add interval $[t,t]$ to the end of *IL'*.
(19) else /* $C(t) \not\models^{ASP} L$ */
(20) *flag*=FALSE.
(21) EndFor
(22) If $[t_s, t_e]$ is the last interval in *IL* then
(23) If *L* is a weakly negated ELP literal and $t_e \neq t_{max}$ then
(24) If *IL'* is not empty and *flag*=TRUE then
(25) Take the last item $[t'_s, t_e]$ from *IL'* and replace it by $[t'_s, t_{max}]$.
(26) else
(27) Add interval $[t_e+1, t_{max}]$ to the end of *IL'*.
(28) else /* $[t_s, t_e]$ is not the last interval in *IL* */
(29) If *L* is a weakly negated ELP literal then
(30) If *IL'* is not empty and *flag*=TRUE then
(31) Take the last interval $[t'_s, t_e]$ from *IL'* and replace it by $[t'_s, t''_s\text{-}1]$, where $[t''_s, t''_e]$ is the next interval to $[t_s, t_e]$ in *IL*.
(32) else
(33) Add $[t_e+1, t''_s\text{-}1]$ to the end of *IL'* where $[t''_s, t''_e]$ is the next interval to $[t_s, t_e]$ in *IL*.
(34) *flag*=TRUE.

```
else /* L is not a weakly negated ELP literal */
    flag=FALSE.
EndFor
return(IL').
```

Example 5 Consider the temporally annotated logic program C of Example 1. Then, *FindMaximalIntervals*(C *has_job*(*Mary*, *Hairdresser*)) returns the list [[1990,1994], [2006, 2009]] and *FindMaximalIntervals*(C, ~*has_job*(*Mary*, *Hairdresser*)) returns the list [[1988, 1989], [1995,2005], [2010,2012]].

Based on Algorithm 2, the following complexity results are derived.

Proposition 2 Let L:i be a ground temporal literal over V. Deciding if $C \models^{max} L{:}i$ is:

1. in P^{NP}, in the case that C does not contain variables or the number of variables of each rule of C is less than a constant, and
2. in $P^{NEXPTIME}$, in the general case.

We would like to note that we have investigated the case to define, for a temporally annotated logic program C, interpretations I containing temporal literals $L{:}i$ s.t. if $L{:}i$, $L{:}i' \in I$ then temporal intervals i,i' have no common time points and they are not consecutive. We can decide if I is an answer set of C, using techniques similar to Answer Set Programming, but now we work on temporal intervals and not on time points. Then, $C \models L{:}[t_1,t_2]$, if for all answer sets M of C it holds that $M \models L{:}[t_1,t_2]$. Additionally, $C \models_{max} L{:}[t_1,t_2]$, if (i) for all answer sets M of C, it holds that $M \models L{:}[t_1,t_2]$, (ii) there is an answer set M of C such that $M \not\models L{:}[t_1\text{-}1,t_1\text{-}1]$, and (iii) there is an answer set M of C such that $M \not\models L{:}[t_2+1,t_2+1]$. However, this technique has higher complexity and does not provides answer sets in the case that it exists a time point t such that $C(t)$ has no (consistent) answer set.

III. QUERY ANSWERING

Below, we define the queries that can be imposed to a temporally annotated logic program C over a vocabulary V=<*Pred*, *Const*> and their answers.

First, we define the intersection of a set of temporal intervals. The intersection of the temporal intervals $[t_1, t'_1],...,[t_n, t'_n]$ is the interval $[max(\{t_1,...,t_n\}), min(\{t'_1,..., t'_n\})]$, if $max(\{t_1,...,t_n\}) \leq min(\{t'_1,...,t'_n\})$. Otherwise, it is undefined.

A simple temporal query of type 1 has the form SQ=$L{:}i$, where $L{:}i$ is a temporal literal with i being a temporal interval. The answers of SQ w.r.t. C, denoted by $Ans_C(SQ)$, is the set of mappings v from the variables of L to *Const* s.t. $C \models v(L{:}i)$.

A simple temporal query of type 2 has the form SQ=L:?i, where L:?i is a temporal literal with ?i being a temporal variable. The answers of SQ w.r.t. C, denoted by $Ans_C(SQ)$, is the set of mappings v from the variables of L to *Const* and from ?i to the set of temporal intervals s.t $C \models_{max} v(L{:}?i)$.

The answer to this type of queries can be provided using Algorithm 2.

A simple temporal query of type 3 has the form SQ=L_1:?i $\wedge ... \wedge L_n$:?i, where L_i:?i is a temporal literal with ?i being a temporal variable. The answers of SQ w.r.t. C, denoted by $Ans_C(SQ)$, is the set of mappings v s.t. if $v_i \in Ans_C(L_i{:}?i)$, for i=1,...,n, s.t. $v_1,...,v_n$ coincide on the common variables of $L_1,...,L_n$ then v coincides with v_i on the variables of L_i, for i=1,...,n, and maps ?i to the intersection of the temporal intervals $v_1(?i),...,v_n(?i)$, if such intersection exists.

A simple temporal query of type 4 has the form SQ=L_1:?i $\wedge ... \wedge L_n$:?i $\wedge$ included(?i, $[t_1,t_2]$), where L_1:?i $\wedge ... \wedge L_n$:?i is a simple temporal query of type 3 and t_1,t_2 are time points. The answers of SQ w.r.t. C, denoted by $Ans_C(SQ)$, is the set of mappings v s.t. if u $\in Ans_C(L_1{:}?i \wedge ... \wedge L_n{:}?i)$ then v coincides with u on the variables of L_i, for i=1,...,n, and maps ?i to the intersection of the temporal intervals u(?i) and $[t_1,t_2]$, if such intersection exists.

Example 6 Consider the temporally annotated logic program C of Example 1. Consider the simple temporal query of type 1 SQ=*vacation_days*(*Mary*,?x):[1994,2001], requesting the vacations days that Mary is entitled to continuously during the temporal interval [1994,2001]. Then, $Ans_C(SQ)$ is the mapping v s.t. v(?x)=22.

Consider the simple temporal query of type 2 SQ= *vacation_days*(*Mary*,?x):?i, requesting the vacations days that Mary is entitled to and their maximal temporal intervals. Then, $Ans_C(SQ)$ is the set of mappings (i) v_1 s.t. v_1(?x)=29 and v_1(?x)=[1990,1991], (ii) v_2 s.t. v_2(?x)=27 and v_2(?i)= [1992, 1992], (iii) v_3 s.t. v_3(?x)=22 and v_3(?i)= [1993, 2002], and (iv) v_4 s.t. v_4(?x)=22 and v_4(?x)=[2006, 2009].

Consider the simple temporal query of type 3 SQ=*has_job*(*Mary*,?x):?i $\wedge$ *has_job*(*Peter*,?y):?i, requesting the jobs of Mary and Peter and their common validity temporal intervals. Then, $Ans_C(SQ)$ is the set of mappings (i) v_1 s.t. v_1(?x)=*Secretary*, v_1(?y)=*Garbage_collector*, and v_1(?i)=[2001,2002] and (ii) v_2 s.t. v_2(?x)=*Hairdresser*, v_2(?y)= *Builder*, and v_2(?i)=[2006,2008].

Consider the simple temporal query of type 4 SQ=*vacation_days*(*Mary*,?x):?i $\wedge$ *included*(?i,[1991, 2001]), requesting the vacations days that Mary is entitled to and their maximal temporal intervals, within the temporal interval of interest [1991, 2001]. Then, $Ans_C(SQ)$ is the set of mappings (i) v_1 s.t. v_1(?x)=29 and v_1(?x)=[1991,1991], (ii) v_2 s.t. v_2(?x)=27 and v_2(?i)=[1992,1992], and (iii) v_3 s.t. v_3(?x)=22 and v_3(?i)=[1993,2001].

A complex *pt*-query has the form CQ=$SQ_1 \wedge ... \wedge SQ_n \wedge$ *filter*, where SQ_i are simple temporal queries, each having a different temporal variable, and *filter* is an expression of the following EBNF grammar:

term:=*duration*(?i) | *start*(?i) | *end*(?i) | c,
where ?i is a temporal variable and c is a decimal.
complex_term:= term | complex_term (+ | - | * | /) complex_term
comparison:= complex_term (< | > | = | ≤ | ≥ | ≠) complex_term
filter:= comparison | *filter* (∧|∨) filter,

such that each temporal variable appearing in *filter* appears in $SQ_1 \wedge \ldots \wedge SQ_n$. The answers of *CQ* w.r.t. *C*, denoted by $Ans_P(CQ)$, is the set of mappings *v* s.t. (i) if $u_i \in Ans_P(SQ_i)$, for i=1,…,n, s.t. $u_1,\ldots,u_n$ coincide on the common variables of $SQ_1,\ldots, SQ_n$ then *v* coincides with u_i on the variables of SQ_i, for i=1,…,n, and (ii) *v*(*filter*) holds.

Note that all Allen's interval algebra relations and their combinations can be expressed by a *filter* expression. For example, the Allen's algebra relation *overlaps*(?*i*,?*j*) can be expressed by the filter (*start*(?*i*) < *start*(?*j*)) ∧ (*start*(?*j*) < *end*(?*i*)).

Example 7 Consider the temporally annotated logic program *C* of Example 1. Consider the complex query *CQ*= *has_job*(*Mary*,?*x*):?*i* ∧ *has_job*(*Peter*,?*y*):?*i′* ∧ (*start*(?*i*) ≤ *end*(?*i′*)) ∧ (*end*(?*i*) ≥ *start*(?*i′*)), requesting the job of Mary and its validity temporal interval and the job of Peter and its validity temporal interval, provided that these intervals have common time points. Then, $Ans_C(SQ)$ is the set of mappings (i) v_1 s.t. v_1(?*x*)=*Secretary*, v_1(?*i*)=[1995,2002], v_1(?*y*)= *Garbage_collector*, and v_1(?*i′*)=[2001, 2003] and (ii) v_2 s.t. v_2(?*x*)=*Hairdresser*, v_2(?*i*)=[2006,2009], v_2(?*y*)=*Builder*, and v_2(?*i′*)=[2005,2008].

Consider the complex query

$$
\begin{aligned}
& CQ = has\_job(Mary,?x):?i \wedge has\_job(Peter,?y):?i' \wedge \\
& ((start(?i) > start(?i')) \wedge (end(?i) < end(?i')) \wedge (end(?i) - start(?i) > 2) \vee \\
& (start(?i') \geq start(?i)) \wedge (end(?i') \leq end(?i)) \wedge (end(?i') - start(?i') > 2) \vee \\
& (start(?i) \leq start(?i')) \wedge (start(?i') \leq end(?i)) \wedge (start(?i') - end(?i) > 2) \vee \\
& (start(?i') \leq start(?i)) \wedge (start(?i) \leq end(i')) \wedge (start(?i) - end(?i') > 2))
\end{aligned}
$$

requesting the job of Mary and its validity temporal interval and the job of Peter and its validity temporal interval, provided that these intervals have more than 2 common time points. Then, $Ans_C(SQ)$ is the mapping *v* s.t. *v*(?*x*)= *Hairdresser*, *v*(?*i*)=[2006,2009], *v*(?*y*)=*Builder*, and *v*(?*i′*)= [2005,2008].

IV. Related Work

Below, we review related work.

In [4], the authors present a framework to incorporate temporal reasoning into RDFS [5][6]. The author associate RDF triples with their validity temporal interval and apply the RDFS inference rules (which are always valid). Like our work, their semantics is based on time points and not on temporal intervals. Yet, [4] does not consider strong and weak negation and validity intervals on logic rules. Additionally, it does not support simple queries of type 3 and type 4 and the filter condition is limited.

Note that our approach can also be applied to RDFS, as RDFS inference rules can be expressed through definite rules [7].

In [8], the authors present a general framework for representing, reasoning, and querying with annotated data on the Semantic Web. They show that their formalism can be instantiated on the temporal, fuzzy, and provenance domain. The authors associate RDF triples with their validity temporal intervals and apply the RDFS inference rules (which are always valid). Unlike our work, their semantics is based on temporal intervals. Yet, [8] does not consider strong and weak negation and validity intervals on logic rules. Additionally, it does not support simple queries of type 4. Moreover, our query answering is more efficient, since during query answering, we directly work on maximal temporal intervals. In [8], all temporal intervals returned by the query are considered and then the maximal ones are returned. Further, our semantics are different than [8]. For example, consider the temporal RDF triples *p*(*a*,*b*) : [1990, 2000] and *q*(*c*,*d*): [1995, 2010]. Then, according to [8], the answer to query *p*(*a*,*b*):?*i* ∧ *q*(*c*,*d*):?*i′* ∧ *end*(?i) < *start*(?*i*′) will provide the mapping *v* s.t. *v*(?*i*)=[1990, 1994] and *v*(?*i′*)=[1995, 2010]. In our case, we will provide no answers, since 2000 > 1995.

In [9], the authors extend RDF graphs with temporal information, by associating RDF triples with their validity interval. They consider any entailment regime that can be expressed through definite rules $A_0 \leftarrow A_1,\ldots, A_n$, where A_i is an RDF triple. Each such rule is replaced by the temporal rule A_0:[$max(t_1,\ldots,t_n)$,min($t'_1,\ldots,t'_n$)] $\leftarrow A_1$:[t_1,t'_1] ,..., A_n:[t_n, t'_n]. These rules are applied recursively, until a fixpoint is reach. Then, maximal validity temporal intervals for each derived RDF triple are produced. Yet, [9] does not consider strong and weak negation and validity intervals on logic rules. Additionally, it does not support simple temporal queries of type 4 and the filter condition is left unspecified.

Work in [10] provides a framework to support spatial and temporal analysis over semantic web data. With respect to the temporal component [10] is similar to [9], as it also computes the maximal validity temporal intervals of derived RDF triples, using the RDFS entailment rules. Yet, [10] does not consider strong and weak negation and validity intervals on logic rules.

In [11], [12], the authors extend the RDFS and ter-Horst entailment rules [13] (which extend RDFS with terms from the OWL [14] vocabulary) with temporal information. In particular, they support inference rules having the general form of these, supported by [9]. However, they dot consider a query language. Additionally, they do not consider strong and weak negation and validity intervals on logic rules.

In [15], we presented semantics for provenance and temporally annotated definite logic programs. However, [15] does not consider strong and weak negation, and reasons based on temporal intervals and not time points. The query language presented here is a restriction of the query language presented in [15] on the temporal component, with the difference that negated atoms in the queries are supported in the present work.

In [16], the authors present a temporal algebra, where the validity temporal interval of two joined relational tuples with associated temporal intervals i_1 and i_2 is the intersection of i_1 and i_2. This temporal algebra operation is also adopted by TSQL2 [17]. In general, TSQL2 is an extension of SQL that supports temporal and non-temporal tables. It also provides a temporal relational algebra that can undertake temporal selection of data and temporal joins based on temporal intersection.

V. CONCLUSION

In this paper, we considered extended logic programming rules, associated with their validity temporal intervals, forming a *temporally annotated logic program*.

We defined (simple) entailment and maximal entailment of a ground temporal literal *L*:*i* from a temporally annotated logic program *C*. Both kinds of entailment are based on Answer Set Programming. The complexities of simple and maximal entailment of ground temporal literals are provided. Additionally, we provided an algorithm that for an ELP literal or a weakly negated ELP literal *L* returns a list with all temporal intervals *i* such that a temporally annotated logic program *C* maximally entails *L*:*i*. Based on this algorithm, the answer of various kinds of temporal queries can be provided.

Note that we do not support operations, such as *next*, *until*, *since*, *sometimes*, and *always*, supported by temporal logic (for an overview, see [18]). Additionally, we do not support inferences such that ``if something is true in one temporal interval then something else is true in another temporal interval", as supported by [19]. Yet, these works do not support the inferences made by our own model.

As future work, we plan to consider logic programs annotated over multiple domains and not just the temporal domain.

. **Appendix: Proof of Propositions**

Proof of Proposition 1

Hardness: Let *P* be an extended logic program over a vocabulary *V*=<*Pred*,*Const*>. Consider the temporally annotated logic program *C* over *V* that is derived from *P* by associating all rules with the validity temporal interval [*t*,*t*]. Let *L* be an ELP literal over *V*. Then, $P \models^{ASP} L$ iff $C \models L:[t,t]$. In [3], it is shown that deciding if $P \models^{ASP} L$ is (i) co-NP-complete, in the case that *P* does not contain variables or the number of variables of each rule of *P* is less than a constant and (ii) co-NEXPTIME-complete, in the general case. Therefore, deciding if a temporally annotated logic program *C* |= *L*:*i*, for a temporal literal *L*:*i*, is (i) co-NP-hard, in the case that *C* does not contain variables or the number of variables of each rule of *C* is less than a constant and (ii) co-NEXPTIME-hard, in the general case.

Membership: Guess a time point t within the temporal interval i and an interpretation I of C(t) over constants in Const. Deciding if I is an answer set of C(t) and I |≠ L is in (i) P, in the case that C(t) does not contain variables or the number of variables of each rule of C(t) is less than a constant and (ii) EXPTIME, in the general case [3]. Thus, deciding if C |≠ L:i is in (i) NP, in the case that C does not contain variables or the number of variables of each rule of C is less than a constant and (ii) NEXPTIME, in the general case. Therefore, deciding if C |= L:i is in (i) co-NP, in the case that C does not contain variables or the number of variables of each rule of C are less than a constant and (ii) co-NEXPTIME, in the general case.

Proof of Proposition 2

In [3], it is shown that entailment of an ELP literal or a weakly negated ELP literal *L* from an extended logic program *P*, under Answer Set Programming, is (i) co-NP-complete, in the case that *P* does not contain variables or the number of variables of each rule of *P* is less than a constant and (ii) co-NEXPTIME-complete, in the general case. Note that Algorithm 2 runs in polynomial time by calling oracles deciding if $C(t) \models^{ASP} L$. Therefore, the complexity of Algorithm 2 is in (i) P^{NP}, in the case that *C* does not contain variables or the number of variables of each rule of *C* is less than a constant and (ii) $P^{NEXPTIME}$, in the general case.

REFERENCES

[1] M. Gelfond and V. Lifschitz, "Classical Negation in Logic programs and Disjunctive Databases", New Generation Computing, 9, 1991, pp. 365-385.
[2] J. F. Allen, “Maintaining Knowledge about Temporal Intervals", Communications of the ACM, 1983, pp. 832–843.
[3] Evgeny Dantsin, T. Eiter, G. Gottlob, and A. Voronkov: “Complexity and expressive power of logic programming”. ACM Comput. Surv. 33(3),2001,pp. 374-425.
[4] C. Gutierrez, C. A. Hurtado, and A. A. Vaisman, "Introducing Time into RDF", IEEE Transactions on Knowledge and Data Engineering, 19(2), 2007, 207-218.
[5] G. Klyne and J. J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax”, W3C Recommendation, 10 February 2004, available at http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.
[6] Patrick Hayes, "RDF Semantics", W3C Recommendation, 10 February 2004, available at http://www.w3.org/TR/2004/REC-rdf-mt-20040210/.
[7] G. Ianni, A. Martello, C. Panetta, and G. Terracina, "Efficiently Querying RDF(S) Ontologies with Answer Set Programming", Journal of Logic and Computation, 19(4), 2009, pp. 671-695.
[8] A. Zimmermann, N. Lopes, A. Polleres, and U. Straccia, "A General Framework for Representing, Reasoning and Querying with Annotated Semantic Web Data", Journal of Web Semantics , 11, 2012, pp. 72–95.
[9] B. Motik, "Representing and Querying Validity Time in RDF and OWL: A Logic-Based Approach", 9th International Semantic Web Conference (ISWC-2010), 2010, pp. 550-565.
[10] M. Perry, A. P. Sheth, F. Hakimpour, and P. Jain, "Supporting Complex Thematic, Spatial and Temporal Queries over Semantic Web Data", 2nd International Conference on GeoSpatial Semantics (GeoS-2007), 2007, pp. 228-246.
[11] H.Krieger, “A Temporal Extension of the Hayes and ter Horst Entailment Rules for RDFS and OWL”, AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning, 2011.
[12] H.Krieger, “A Temporal Extension of the Hayes/ter Horst Entailment Rules and a Detailed Comparison with W3C’s N-ary Relations”, Deutsches Forschungszentrum fur Kunstliche Intelligenz GmbH Technical Report, RR-11-02, 2011.
[13] H. J. ter Horst, “Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary, Journal of Web Semantics, 3(2-3), 2005, pp. 79-115.
[14] G. Antoniou and F. van Harmelen, A semantic web primer, MIT Press, 2004.
[15] A. Analyti and I. Pachoulakis, “Provenance and Temporally Annotated Logic Programming”, accepted to the International Journal of Advanced Research in Artificial Intelligence (IJARAI), 2012.
[16] D. Dey, T. M. Barron, and V. C. Storey, "A Complete Temporal Relational Algebra", VLDB Journal, 5(3), 1996, pp. 167-180.
[17] R.T. Snodgrass , The TSQL2 Temporal Query Language, Springer, 1995.
[18] M. A. Orgun and W. Ma, "An Overview of Temporal and Modal Logic Programming", First International Conference on Temporal Logic (ICTL-1994), 1994, pp. 445-479.
[19] T.W. Fruhwirth, "Temporal Annotated Constraint Logic Programming", Journal of Symbolic Computation, 22(5/6), 1996, pp.555-583.

AUTHORS PROFILE

Anastasia Analyti earned a B.Sc. degree in Mathematics from University of Athens, Greece and a M.Sc. and Ph.D. degree in Computer Science from Michigan State University, USA. She worked as a visiting professor at the Department of Computer Science, University of Crete, and at the Department of Electronic and Computer Engineering, Technical University of Crete. Since 1995, she is a principal researcher at the Information Systems Laboratory of the Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH-ICS). Her current interests include reasoning on the Semantic Web, modular web rule bases, non-monotonic-reasoning, faceted metadata and semantics, conceptual modelling, contextual organization of information, information integration and retrieval systems for the web, interoperability of heterogeneous and distributed information bases, and biomedical information systems. She has participated in several research projects and has published over 55 papers in refereed scientific journals and conferences.

Ioannis Pachoulakis received a B.Sc. in Physics (1988) at the University of Crete, Greece, and a Ph.D. in Astrophysics (1996) and an M.Sc. in Engineering (1998), both from the University of Pennsylvania in the U.S.A. Since 2001, he serves as an Assistant Professor at the Department of Applied Informatics and Multimedia at TEI of Crete with mainstream interests in realistic multimedia applications, virtual reality and multimedia applications for science.

Brain Computer Interface

Boulevard of Smarter Thoughts

Sumit Ghulyani , Yashasvi Pratap, Sumit Bisht, Ravideep Singh
Department Of Computer Science & Engineering
Dronacharya College of Engineering
Gurgaon, Haryana, India

***Abstract* — The Brain Computer Interface is a major breakthrough for the technical industry, medical world, military and the society on a whole. It is concerned with the control of devices around us such as computing gears & even automobiles in the near future without really the physical intervention of the user. It helps bridge the communication gap between the society and the disabled. This mainly lays its focus on people suffering from brainstem stroke, going through a spinal cord injury or even blindness. BCI helps such patients to retain or restore communication with the outside world through intelligent signals from the brain due to the high risk of paralysis under such circumstances. This is achieved by a signal acquisition technique and converting these signals available from the sensors placed on the scalp into real-time computer commands that can be visually operated and understood. It has nothing to do with the natural neural transmission of brain signals but extracts them with the help of sensors to be processed and direct the outputs to an external device.**

This may also prove to be a major military gadget where troops may communicate their thoughts in highly stressed situations without breaking the hush. But, as every technology have some merits and demerits, so does BCI.

Keywords- Brain Computer Interface (BCI); Blood Oxygen Level Dependent (BOLD); Electrocorticography (ECoG); Practical Electrical Stimulation (PES); Electroencephalograph (EEG); Magnetic Resonance Imaging (MRI),Functional Magnetic Resonance Imaging (fMRI) brain scythe, motor cortex.

I. Overview

A plethora of thoughts is ready to open up and make things work just fine as they did ever. In the human brain, the thoughts and emotions are transmitted in the form of electric signals with the help of those millions of neurons. The following figure below displays a wireless neuro-headset developed by Emotiv Systems which detects brainwave signals and categorizes them specifically into conscious &non-conscious thoughts

This makes possible selected processing of only appropriate signals that are meant to be transmitted by the user [1]. This device makes use of the same set of electrodes as sensors which are used in the medical examination known as electroencephalograph, or in short EEG. The display is maintained by an advanced software program than the old EEG paper & pen tracker scheme.

Figure 1. BCI Developed by Emotiv Systems

MRI or Magnetic Resonance Imaging technique is under research to be used in BCIs as an effective approach to map actual part of the brain with the task to be performed. With Brain Computer Interface, the control of the world, on a whole, lays in the hands of thoughts & emotions. The power behind the advent of this technology is the working of the human brain itself. Though a person maybe disabled with speech, hearing or a total “locked-in”, the brain never stops functioning. This feature is made use of in BCI using external peripheral gears to detect these signals and convert them into visual commands that can be read and understood. In the course of this, the actions of the brain are manipulated to control the computing gears and communication devices instead of generating signals using the traits of a motor movement.

The figure 2 is a depiction of the BrainGate technology which is a part of the research carried on at the Brown Institute for Brain Science. It aims at implanting a baby sensor in the gray matter which will read the electric signals and transmit them to the computer system. Evenly, the research in the field of Human Computer Interaction has achieved impressive results in the rapid & heavy transmission of signals from the computer to the human brain using sensory channels and vice versa. A phase of perceptual computing is about to begin which studies user’s psychology and operating environment to process the information about a user’s intent.

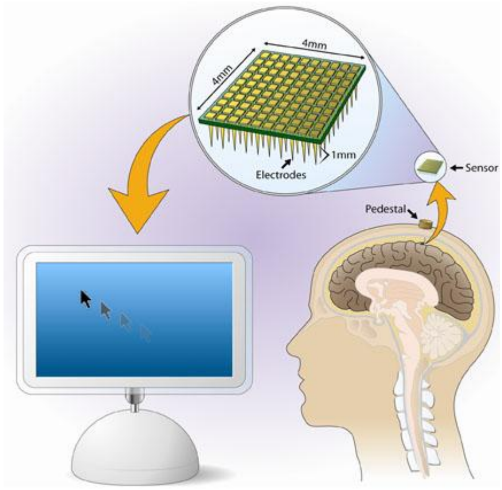


Figure 2. Chip Implantation

In this way, a computer can adapt itself according to the user and the task to be performed thereby giving phenomenal results. If extensive research is done and one succeeds in recognizing the signals transmitted when light falls on the retina it may lead to a major breakthrough of BCI in the medical world which may grant artificial vision to the blind. And as we see, possibilities are endless.

II. History

The past of the Brain Computer Interface can be dated back to the time when Electroencephalography was in his early years of birth. This became possible with the successful research of German scientist Hans Berger, who in 1924, succeeded in recording the electrical signals in the human brain. He successfully analyzed the oscillatory activity of the brain and discovered the Alpha wave, also popularly known as the Berger's wave. Though his instrumentation, which consisted of Lippmann capillary electrometer and silver wires, was very desolate and led to unsuccessful results in advanced phases, he was able to mark the start of a revolution. Success was later achieved with the help of Siemens double coil recording galvanometer which made recording of electric signals as small as one ten thousandth of a volt. Berger's preliminary work led to success in elimination of various diseases related to brain cells and the advancement of the Electroencephalography or EEG, in short. The actual birth of the first relative to Brain Computer Interface took place in 1969 at the University Of Washington, School of Medicine (UWSOM), Seattle [2]. A monkey was used as the subject where he successfully deflected a biofeedback meter measuring arm with the intervention of his neural activity. Another subsequent research used the same subject, but in numbers to announce the voluntary control of solo and multiple neurons in the primary cortex. Many research groups have been able to record the neurological stimuli and depict them with the various body movements thus controlled.

A major breakthrough was achieved when Philip Kennedy were successful in implanting electrodes in a monkey's cortex. With the passage of time, a change in the subject was brought when monkeys and rats were replaced by cats for further research issues at Berkeley when visual images were reproduced by decoding the neural shots in the cat's brain. In this process, the electrodes were implanted in the region of the dual lobed grey matter called the thalamus, which records the actions perceived by the retina. The generation and reconstruction of the data was hack done by gradually synthesizing and decoding the signals obtained over a period of time with the visual perceptions involved.

The human intervention as subjects took place in Japan when researchers implanted electrodes to a patient's scalp and similar results were achieved. A further research at the University of Pittsburgh led to the use of even fewer neurons than ever in a working Brain Computer Interface. They brought about a jaw-breaker advancement which displayed a monkey's brain activity to control a robotic arm that fed him with fruits kept beside him. Research groups all around the globe are working towards applying kinematics to the concept of BCIs to retain normal muscle movement of the patient by stimulating the muscles electrically.

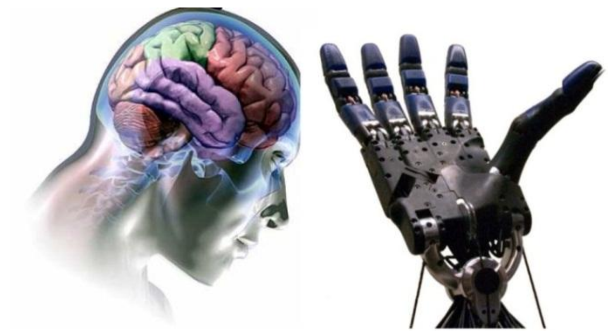


Figure 3. Association of brain signals with movement of hand

III. Operational Traits

The performance of a BCI is completely based on the approach our intellect functions. Our brain is crammed with neurons, entity nerve cells allied to one another by dendrites and axons that bear electronic signals all through the entire body. Every time we think, budge, sense or consider something, our neurons are at labour. That work is carried out by these miniature electric signals that whiz from neuron to neuron as prompt as 250 mph. The signals are generated by differences in electric prospective carried by ions on the membrane flanked by two neurons. Although the routes the signals take are insulated by something called myelin, some of the electric signal escape [3].

Scientists with the exploit of modern technology can perceive those signals, construe what they mean and utilize them to direct a device of some breed. It can also work the other way around. For example, researchers could outline out what signals are sent to the brain by the diverse sensory organs for example, the signals in the aural nerve can be traced and the task of hearing can be mapped. An even more remarkable use of this technology is that the signals can be generated to be sent to the brain and a enthused sense of hearing can be induced in the subject even if one is deaf. One of the biggest challenges in front of brain-computer interface researchers at the moment is the basic technicalities of the interface itself. The sleekest and least detrimental is a set of electrodes, an apparatus known as an electroencephalograph (EEG) which is attached to the scalp.

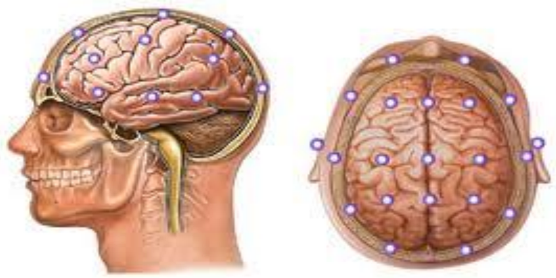


Figure 4. Electroencephalogram (EEG) scan

The electrodes interpret brain signals. However, the cranium restricts a lot of electrical signal, and distorts whatever does get through it.

As a resolution, scientists may implant electrodes directly into the brain, or on the facade of the brain. This aids in higher reception of electric signals and allows electrode positioning in the particular area where those explicit signals are generated. The electrodes assess minuscule differences in the voltage amid neurons. The signals are then augmented and interpreted by a computer program. In the case of an infix BCI, the method is overturned. A mainframe will renovate a signal, like from a tape, into the voltages that would elicit neurons. The signal generated from a peripheral apparatus is sent to an electroencephalograph (EEG) detector installed in the relevant quarter of the brain. The neurons in the head are enthused with these signals and the subject obtains an illustration, reverberation or a related sensory familiarity parallel to whatsoever gear is used.

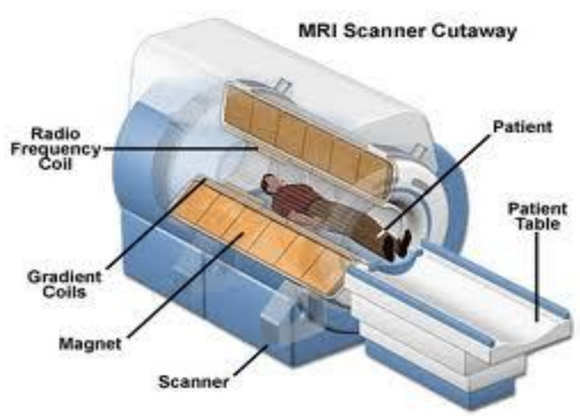


Figure 5. MRI Scanner

A substitute approach to scrutinize and assess brain bustle is using a **Magnetic Resonance Image** (MRI). The MRI setup is a colossal and convoluted apparatus. It produces high decree imagery of the brain, but it can't be used as component of an enduring BCI. Researchers use it to swot up definite brain functions or to plot in the brain the quarter where electrodes should be sited to evaluate an unambiguous task.

IV. FMRI

Functional Magnetic Resonance Imaging or more popularly fMRI is a non-invasive, much similar technique as the MRI or Magnetic Resonance Imaging, the only difference being the basic measure of BOLD i.e. Blood Oxygen Level Dependent in fMRI. It is advancement to MRI and measures brain activity, but by detecting changes in the flow of blood.

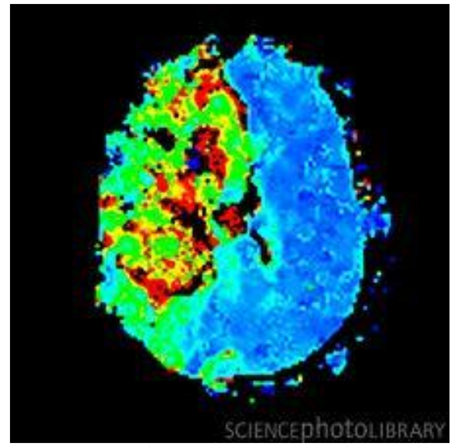


Figure 6. Blood flow imagery obtained from an fMRI

With this advancement, on might just sneak in someone else's brain and derive its thoughts and even detect lies. The basic idea behind this technology originated from the fact that blood rich in oxygen behaves differently to a magnetic field than deoxygenated samples. In other words, both have different magnetic resonance traits. The more active parts of a brain receive higher flow of oxygenated blood. The procedure for the test is similar to the present MRI and uses only magnetic fields and radio waves. Though no radiations are used, it is highly advised not to be performed during pregnancy, and on people with artificial limbs or pacemaker [4].

This technique is medically used rarely but to monitor the growth of tumors and regulate the normal functioning of the brain after a stroke. Brain Mapping is an emerging field for fMRI which will make Brain Computer Interfaces even superior as it determines the areas of brain that control the various body movements and moreover on planning surgery and market survey & research.

Functional Magnetic Resonance Imaging (fMRI) also possesses certain vantages and detriments. The major breaker is that this technology is still in its birth phase and the researchers know very less regarding the overall working of the apparatus. Moreover, the setup is really expensive to be used for research and brain computer interfaces. To continue, it works only when the subject lays still and thus a lot of work needs to be done.

The major breakthrough that gets it going is the absence of any radiation in any phase of the scan as in X-ray scan and PET and the positive feedback and response from the tests carried out. Though scores of tests resulted in a higher level of bewilderment amongst the researchers since fMRI relates to the blood flow only and not on the individual neuron functioning in the brain and the blood flow measure was not enough to critically examine and complement on the results thus obtained.

V. BCI SPEECH

The BCI system can be trained what an individual's brain scrutiny or EEG looks like when one is focused on a fussy article, since the electrical oomph in the brain changes in proportion to what the individual is doing. The rise of Electro-corticography (ECoG) has been a new hint podium for brain-computer interface systems. Consequently, it was anonymous whether further neuro-physiological substrates, such as the vocalization net, might be used to further develop on or harmonize on motor-based power paradigms. For the foremost instance, that ECoG signals allied with diverse, blatant and anticipated phoneme enunciation can facilitate invasively monitored patients to manage a one-dimensional computer pointer precisely.

This stuff was discernible within upper gamma regularity oscillations and enabled clients to realize concluding object accuracies flanked by 68% and 91% in 15 minutes [5]. In addition, one of the patients achieved full-bodied control via recordings from a micro array that consisted of 1 mm spaced micro wires. These results proposed that the cortical system allied with verbal communication could present an added cognitive substrate for BCI maneuver. Subsequently, these signals can be recorded from a cortical assortment that is minute and austerely invasive.

VI. Applications

A. Brain scythe

It's trouble-free to scythe a computer, but nearly everyone always considered that hacking the human intellect was unfeasible. As seen in science fiction movies and novels like the Matrix, Brainstorm & Surrogates, play around with gaining admittance to the human mind. But, researchers have finally confirmed the leeway to hack the human brain [6]. The researchers used a brain mainframe interface, usually used in following a line of investigation that scans brain patterns.

Figure 7. Brain Scythe

The contemporary models consent to users to be in charge of their computers with their thoughts, but the researchers have verified that it goes well both ways. The panel built a part of custom software that can, in essence, interpret the mind. They were proficient enough to effectively use the software pooled with the brain scanner to mine susceptible data including credit card PINs, address, DOB and other.

B. Restores spinal coordination

Researchers at Chicago have lucratively bypassed the spinal flex and restored well motor control to paralyzed limbs by means of a brain-computer interface. The researchers have shaped a neuro-prosthesis that groups a BCI that is wired directly into 100 neurons in the motor cortex of the subject, and a Practical Electrical Stimulation (PES) tool that is wired into the brute muscle of the subject's upper limb. When the subject tries to shift his arm, the huddle of 100 neurons activates, creating a torrent of data which can be examined and analyzed by the BCI to foresee what muscles the subject is trying to budge, and with what intensity. This interpreted statistics is conceded to the PES, which then triggers the correct muscles to execute the preferred movement. The last part outcome is a mainframe network that in actual fact replaces the nervous structure and restores fine motor control to a paralyzed arm.

C. Bionic prosthetic ogle

A bionic prosthetic ogle is the mock eye for the sightless or a being with a hitch in seeing matter due to a defected retina. It speaks the tongue of the brain by transmitting the signals fired by the retina when something is seen. This is the sense in the wake of the prosthetic eye. Electric pulses are measured by attaching electrodes to an animal's optic nerves.

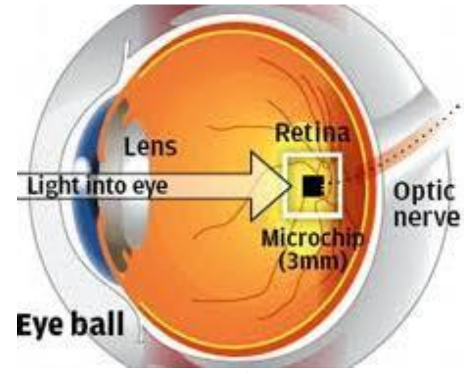


Figure 8. Components of a Bionic Prosthetic Eye

One might not know, but if a retina at all times generates the same electric cipher when looking at a building, and a dissimilar code when a being is in sight, work can be retreated to derive the retina's tangible programming procedure. This revolution would leave mankind startled when a silicon chip would be installed into the eye and then wired to the optic nerves.

D. Restored neurological implementation

Brain computer interface can facilitate the cases where flexibility has been smacked or got narrow by a stroke. Scientists in St. Louis have revealed the detection of the brain signals simply thinking on affecting a partly or wholly paralyzed limb. The damaged half of the brain that is responsible normally for moving the hand can no longer do so, so the hint arrives from the unharmed other half of the brain. To lay bare the prospective to help restore motion, scientists allied electrode-detected brain signals to the movement of a marker on the computer panel. Researchers are now working out a motorized glove which will turn all the imagination into a reality.

VII. Detriments

Every technology suffers severe drawbacks, but it does not halt the research process, although slows down the speed at which advancements appear. The major breaker of this technology is that BCIs are still in crude form and need to be refined for the 'future of thoughts' to come alive. The development process is further slowed by certain ethical issues on the road to success. The complexity of the brain adds up to the massacre being equipped with a staggering 100 billion neurons, and detecting the association of the type of neurons with specific body movements looks arduous.

The structure and placement of electrodes needs a research in itself since the one placed outside the skull are unable to detect appropriate signal voltage to produce optimal results. Moreover, the electrodes implanted inside the skull cause damage to the tissues and may even lead to serious scars resulting in undesired effects

VIII. VANTAGES

But, as BCIs appear to be the gate to an era of thoughtful communication, the research goes on keeping in mind the extra edge its merits showcase despite its failures. Paralysis will never stay a curse as in the history of medical sciences since the patients will be able to communicate with their mates and docs through a graphical BCI. To add to it, they may even help in the movement of prosthetic limbs to sink the effect of any motion related impairment. Visual & hearing impairments may become history as these interfaces act as artificial eyes & ears transmitting data to and fro the human mind successively.

Figure 9. Military “Surrogate” Robots

With gradual improvements, the armed forces may get even better armed with this interface. Communication can occur without any vocal disturbance, and who knows armed surrogates may come in effect very soon [7].

This will lead to less loss of men but money on the battlefield, as they are stationed far away and connected to their subject surrogate machines which operate in the main field and the troops live even if their surrogate is blown away by a grenade. This means a commandment of 1000 soldiers may fight all their life with different bodies. The world of entertainment is about to get vowed with this new technology, as gamers will now control their milieu with the might of their mind.

IX. PROPOSED WORK

(i) Advancements in fMRI must occur so as to scan and process even the moving parts of subject and not the part which lies still.

(ii) Development of robotic limbs to provide co-ordination with the brain computer interface.

(iii) The electrodes to be used in the interface carcass should be made highly sensitive so as to provide exterior signal detection without the installation of the electrodes in the grey matter.

(iv) Combined tests to be conducted based on both MRI & fMRI to determine collectively the association of different hotspots in the brain to diverse body movements by deriving the relationship between blood flow and neuron activity.

(v) Development of high resolution interface displays to record brain activity.

(vi) Provisions and techniques to restrict unauthorized enforced access to someone else’s brain.

X. CONCLUSION

The research goes on and the possibilities of refinement are as vast as the universe, never ending. The maximum a human brain can think of what it can do is the result in the making of the American science scripture Surrogates (2009) directed by Jonathan Mostow [8].

The scripture shows how the world of humans turns into an era of surrogates which are none other than robotic bodies which are controlled by human brains from home. The surrogates are set to work almost everything physically, be it routine tasks or strategic war situations without any human intervention but mind control.

The world changes and divides into 2 parts – humans & machines until the surrogate inventor himself devises a deadly weapon on seeing the far-sighted fate of his invention that starts wiping out the surrogates along with its users. But, this is as we know long way ahead in the future.

As of the birth of the Brain Computer Interface, it is bliss on medical grounds for the patients suffering from spinal cord injuries or impaired hearing or vision. A new era of thought has raised its head in this chaotic world which will help change the way of communication, since humans appeared first on earth.

Partial immobility or paralysis will disappear from the medical books as a curse. This can also be a major breakthrough for the armed forces to communicate with each other in highly stressed situations like war & terrorism, but this may even lead to a major demerit if gone in wrong hands. By wrong hands I mean wrong minds which may use this boon as a bane against the human society as a whole.

With progress in research, one may forget his very own mouse and keyboard, the somewhat most essential input components of a computer system as they are replaced with these interfaces. When applied to robotics, these interfaces may help physically challenged people to gain the same place in the society as any other normal person.

With advancements, BCIs may help such people to control robotic bodies with their brain to cope with their routine tasks. So, let the brain do the talking and send your jaw muscles on a vacation, just saying. Despite of all the detriments, the merits give the BCIs that extra edge which makes it a possible suitor for a better future.

REFERENCES

[1] Emotiv: EEG Systems http://emotiv.com/

[2] About BCIs http://inventors.about.com/od/bstartinventions/a/John-Donoghue.htm

[3] HowStuffWorks: Brain Computer Interfaces http://computer.howstuffworks.com/brain-computer- interface.htm

[4] How fMRI Works http://science.howstuffworks.com/fmri.htm

[5] Using Electrocorticography speech for BCI http://www.ncbi.nlm.nih.gov/pubmed/21471638

[6] Gaining Access To Brain at $300 only http://www.webpronews.com/your-brain-can-now-be-hacked-for-under-200-2012-08

[7] Mind Controlled Robots http://www.extremetech.com/extreme/132389-real-life-avatar-the-first-mind-controlled-robot-surrogate

[8] Surrogates (2009) http://www.imdb.com/title/tt0986263/plotsummary

AUTHORS PROFILE

Sumit Ghulyani is presently a final year student pursuing degree in Bachelor of Technology in the field of Computer Science Engineering from **Dronacharya College of Engineering, Gurgaon, India.** Other than being the campus ambassador for a couple of organisations, he has been one of the soul limbs in successfully coordinating & hosting technical events with Microsoft and HRs of various other technical giants. Authored a couple of research papers, the fields of his research interest include Brain Computer Interface, Windows Operating Systems, Apple MAC OS, Android Operating Systems, Wireless Power Transmission Artificial Intelligence & Virtual Reality. Email: sumit.ghulyani@live.com, GSM: +91-9910550618

Yashasvi Pratap is a final year student pursuing degree in Bachelor of Technology in the field of Computer Science Engineering from **Dronacharya College of Engineering, Gurgaon, India.** Being a fresh seed in the field of research, his major interests include Computer Mediated Reality and Ubiquitous Computing. He is actively engaged in other activities which include interest in world geography, historical timeline & zoology. Email: yashasvi000pratap@gmail.com, GSM: +919968052498

Sumit Bisht is a final year student pursuing degree in Bachelor of Technology in the field of Computer Science Engineering from **Dronacharya College of Engineering, Gurgaon, India.** A soul coordinator for several technical & cultural events at college, he never halts at thriving for the best. Stepping forward to publish his second work of brilliance, his major interests include Programming Paradigms, ATM Networks & Wireless Computing. Email: sumitbisht@live.com, GSM: +919968803069

Ravideep Singh is a final year student pursuing degree in Bachelor of Technology in the field of Computer Science Engineering from **Dronacharya College of Engineering, Gurgaon, India.** Other than his proficiency in learning foreign languages, his major interests include Distributed Operating Systems, Artificial Intelligence & Neural Networks. Email: ravideep_singh@live.com,GSM: +919891912029

Identification Filtering with fuzzy estimations

J. J., Medel J [1]
[1]Computing Research Centre
Venus St., Tel: 56242000 ext. 56570
CP. 07738 Mexico D. F

J. C: Garcia I[2]
[2]Mechanical and Electrical Engineering School
Santa Ana Av., Tel: 56242000 ext. 73258
Mexico D. F

J. C. Sanchez G[3]

***Abstract*— A digital identification filter interacts with an output reference model signal known as a black-box output system. The identification technique commonly needs the transition and gain matrixes. Both estimation cases are based on mean square criterion obtaining of the minimum output error as the best estimation filtering. The evolution system represents adaptive properties that the identification mechanism includes considering the fuzzy logic strategies affecting in probability sense the evolution identification filter. The fuzzy estimation filter allows in two forms describing the transition and the gain matrixes applying actions that affect the identification structure. Basically, the adaptive criterion conforming the inference mechanisms set, the Knowledge and Rule bases, selecting the optimal coefficients in distribution form. This paper describes the fuzzy strategies applied to the Kalman filter transition function, and gain matrixes. The simulation results were developed using Matlab©.**

Keywords- Intelligent Identification; Digital identification filter; Fuzzy estimation; Signal processing; Probability.

I. Introduction

The intelligent inference mechanisms are built and characterized with respect to dynamical system operation uncertainties bounded by a distribution function having different operational levels and needing correct answers in accordance to their dynamic changes. The adaptively is the condition needed in digital systems that interact with the real processes. The identification filter describes the natural evolution based on objective functions adjusting their parametres and gains giving the correct solution for a time interval. With this perspective, one of the best tools used to solve the filter identification is a recursive digital filter with the adaptive parametre and gain estimation giving a limited temporal answer integrated both in the identification frame. The problem observed in the Kalman filter is that it supposes known internal black box parametres and described the gain matrix. Unfortunately, the adaptability of it to natural reference system changes is lost. The unknown parametres and gain are characterized by inference mechanisms in accordance to different dynamical operation levels, corresponding to a fuzzy system description, with different rank operations [1], [2], [3], [4], [5], [6]. The paper integrates the fuzzy model concepts as a intelligence system that describes with time restrictions the Kalman transition and gain adjusting the answers in a natural form to the reference process output answer [7], [8].

Digital fuzzy filter developed as parametre and gain estimations operating automatically into identification improving the identification filter performance changes. The operational fuzzy tools are the inference mechanisms selecting into the Knowledge Bases (KB) the best parametre and gain with respect to the Identification Error Density (IED). These results are applied to the conventional Kalman SISO filter permitting it to be transformed into an expert identification filter in spite of environmental changes suffer by the reference output system [9].

II. Fuzzy Estimation Filtering

A fuzzy estimation filtering operates in a close loop with: *a)* Transition function considering the fuzzy logic identification density error in adaptive parametre estimations and, *b)* The Kalman gain with respect to identification density error, *c)* The density identification error ranks with variable stages considering that the identification error variable through the time. Commonly the estimation filters do not use the operational level processes with time restrictions [3,] [10] satisfying the identification filter requirements. The adaptive inference mechanism regions are built as Identification density error functions [11], being the first stage required in the fuzzy filter in accordance to a knowledge base membership function used in inference rules with adaptive boundaries. Fig. 1 in the diagram block scheme shows the process observing the transition function and gain described as parametres filter estimation $\hat{a}(k)$ and $K(k)$ affecting directly the identification Kalman answer $\hat{y}(k)$ based on the desired signal $y(k)$ (output reference process), generating the identification error $e(k)$ and also affecting the estimation and gain processes. The estimation filtering process dynamically adjusts its parameters in accordance to the membership value selected from the Knowledge Bases appropriates values that allow that the Kalman output signal converges to reference [12]. The estimation fuzzy filter in agreement with [4], [13] have the following elements:

1. *Input inference:* A desired signal and the identification error as the inputs.
2. *Rules base:* The expert membership functions (parametre and gain values) are updated into identification filter [14]. The membership functions metrics are dynamically

adjustable considering the distribution identification error $e(k)$, using the logical connector *if*.

3. *Inference mechanism:* The expert actions select the parametre and gain in accordance to objective functions into specific membership from the Knowledge Bases respectively using the logical connector *then* selecting the section in where the parametre and gain adequate adjust the identification filter [8].
4. *Output inference:* This is the result the estimation fuzzy filter emits to be used into the Kalman identification filter in accordance with the reference signal process.

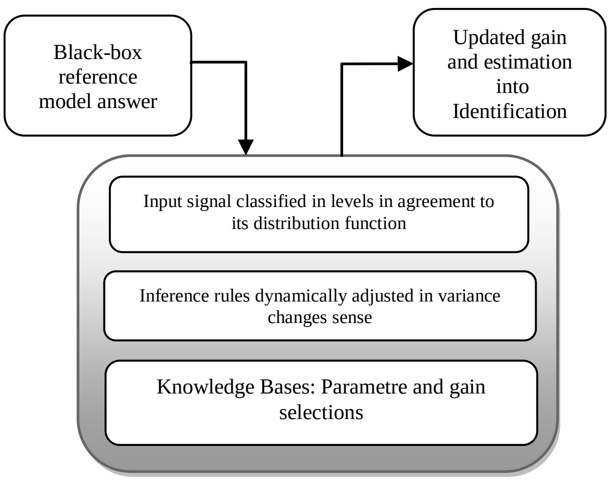


Figure 1. Fuzzy filter general scheme

III. Fuzzy Estimation And Kalman Filtering Properties

A fuzzy estimation adjoins into Kalman filter considers the stochastic digital filter properties and the distribution functional error classified in intervals and associated with intelligence decisions affecting the identification filter [3]. Fig. 2 shows the filter architecture integrating the fuzzy estimation scheme.

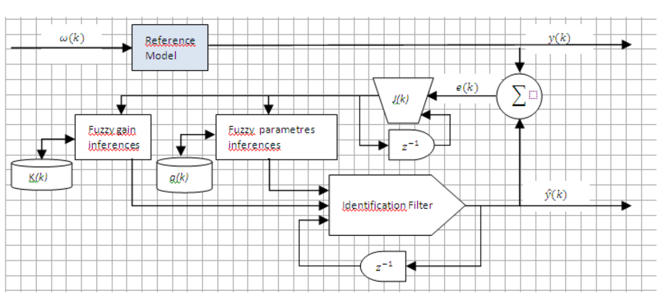


Figure 2. Identification Filter with fuzzy filter estimations

Fig. 2, showed the fuzzy stages needed by the identification filter: $x(k)$ is the reference model input, $y(k)$ is the reference model output and the fuzzy filter input, $\hat{a}(k)$ is the fuzzy parametre value, $K(k)$ is the fuzzy gain, $e(k)$ is the error, $J(k)$ is the functional error and $\hat{y}(k)$ is the identification filtering output.

Rule base: The new information arriving at the estimation filter in accordance to the reference signal changes describing the operational levels [8]. The fuzzy rules base has a set of logical connectors (*if-then*) classifying the reference process conditions in levels, selecting in probability sense the best membership value from the Knowledge Bases in order to update these into the identification filter. This mechanism is limited by the error filter criteria, which previously had all the possible parametre and gain values as a membership function in the Knowledge Bases, in accordance with the reference model interaction [4]. The logical connectors (*if-then*) allow selecting the operational levels membership function indicators (parametre $\hat{a}(k)$ and gain $K(k)$ values) adjusting the identification filtering process [14].

The fuzzy mechanism has the following description [9]:

Inference mechanisms use **if** connector, in order to find the reference model operational levels considering the identification error value in accordance to fuzzy mechanism assigning the membership function **then** connector. This membership value selects from the Knowledge Bases the parametre an gain values affecting the identification filter $\hat{y}(k)$. The fuzzy filter estimation uses second probability moment recursively (1) considering that the identification error has stationary conditions and it is bounded in levels [4].

$$J(k)=\tfrac{1}{k}\left[e(k)^2+(k-1)J(k-1)\right]\in R_{[0,1)} \qquad (1)$$

The fuzzy mechanisms make the minimization error described as J_{min}, having the best neighborhood value to the desired signal $y(k)$. The error must be near to γ within the interval limit [0, 1). The filter membership function answer levels have an affinity between the minimum and maximum intervals limiting the operation description inside the filter distribution function [2], [12]. Fig. 3, shows the filter representation criterion describing its convergence.

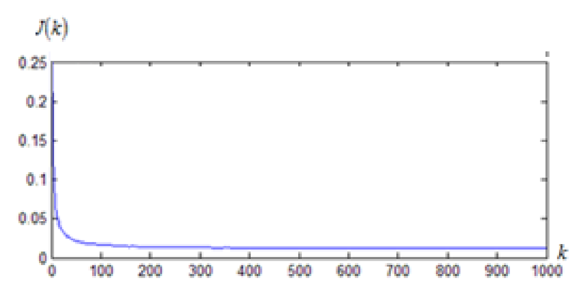


Figure3. Filter convergence functional error (1).

Fig. 3 shown the dynamical error minimization based on the stochastic gradient criterion classified in error levels and in fuzzy stages having the best neighborhood desired values (parametre and gain), in order to update the new conditions required into identification scheme [11].

A. Parametre and gain selection

The membership function selects the parametre $\hat{a}(k)$ and gain values $K(k)$ into the fuzzy stage in accordance with the reference model operational levels and inference rules developed in probability sense where the Knowledge Bases are bounded by a control area T_N as [4]:

$$T_N = \{(y(k), \hat{y}(k))\} \subseteq R^2 \qquad (2)$$

As an example, the membership functions set into the parametre Knowledge Base is bounded by the error criterion [8], [9]. Fig. 4, shows the dynamical filter parametre estimation $\hat{a}(k)$, in accordance with the reference model evolution $y(k)$ and its identification $\hat{y}(k)$.

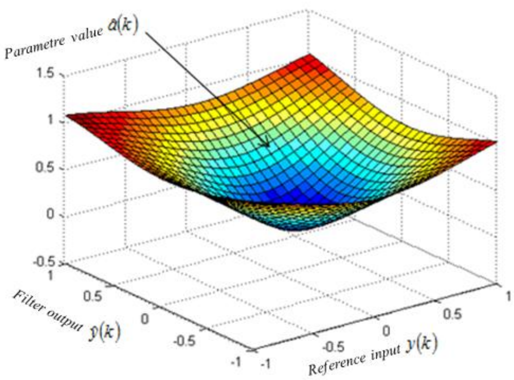


Figure 4. Knowledge Base viewed as a parametric surface.

Fig. 4 shown the parametre selection into operational levels (2) based on changes into the filter input-output conditions [7].

- The parametre and gain selections goal using fuzzy logic strategies [12], [14]:
  - Automatic filter parametre and gain selections base on different levels.
  - Parametre and gain selections in accordance with the desired signal and the identification filter response.
  - Adaptive rules modifying the membership functions limits, renewing and updating its values in accordance to the reference model and the error criteria
  - Each membership value establishes the maximum correspondence between the desired and the reference signals, minimizing the identification filter error [2].

B. *Time constrains*

Identification integrated with fuzzy parametre estimation $\hat{a}(k)$ and gain $K(k)$ strategies is bounded temporally, considering the quality response in accordance with filtering stability and time restrictions [15]. *Global*, are described by functional identification error (see Fig. 4).

If the identification error is stationary then the maximum convergence rate is limited to minimum region established as an identification variance error function. The convergence intervals are $[0, \varepsilon \pm \alpha)$ with functional error *J(k)* upper limit tending to zero. The membership functions are *temporarily bounded by* (3). The global characteristics are specified in probabilistic form where $J(\tau_m) = \inf\{\min\{\tilde{J}_k\}\} \leq \varepsilon$ and $\{\tilde{J}_k\} \subseteq \{J_k\}$ considering $P(\tilde{J}_k \leq \varepsilon \pm \alpha) = 1$ in accordance with the reference model [4].

$$\tau_{\min} = 0.5 f_{\max}^{-1} \qquad (3)$$

The final filter times $f(k)_i$ inside its corresponding absolute limits $\lfloor ld(k)_{i\_\min}, LD(k)_i)$ in the Nyquist and Shannon sense [15], the filter integrate time is limited dynamically considering that $\mu\lfloor ld(k)_{i\_\min}, LD(k)_i) < \tau_{\min}$. Fig. 5, shows the time interval filter process task [4].

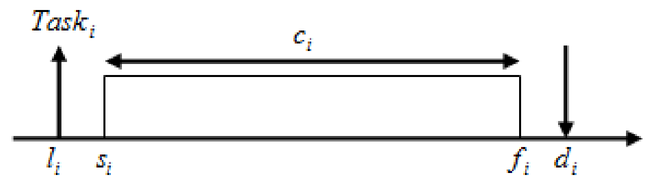


Figure 5. Filtering process task,

The filtering process task is bounded by (3) having the arrival time (l_i), start time (s_i), processing time (c_i), final time (f_i) and the maximum time period (d_i).

IV. RESULTS

Identification technique considers the fuzzy filter estimation describing the known internal black-box parametre and the identification gain (see Fig. 2). Reference model considered an Autoregressive Mobil Average (ARMA) model interacting with fuzzy filter. The discrete SISO states space system as reference model is shown in (4) and (5).

$$x(k+1) = a(k)x(k) + w(k) \qquad (4)$$

The filter output is described in (5) as ARMA (1, 1) model:

$$y(k) = c(k)x(k) + v(k) \qquad (5)$$

With $x(k) \in R$, $|a(k)| \in R_{[0,1)}$, $\{w(k)\} \subseteq N(\mu_w, \sigma_w^2 < \infty)$, $y(k) \in R$, $|c(k)| \in R_{[0,1)}$, $\{v(k)\} \subseteq N(\mu_v, \sigma_v^2 < \infty)$,where: x(k) is the internal reference model state; a(k) is the parametre; w(k) is the reference model noise; y(k) is the reference model signal as a filter input; c(k) is the parametre output system and v(k) is the output vector noise.

The different operation levels must match with the error criteria established in accordance with the desired signal $y(k)$; the parametre $\hat{a}(k)$ and gain as $K(k)$ selections from the Knowledge Bases sand the filter response $\hat{y}(k)$. The first step in accordance with the dynamical selection into the identification ARMA(1,1) model is described in (6) .

$$\hat{y}(k) = \hat{a}(k)\hat{y}(k-1) + \hat{W}(k) \qquad (6)$$

Fig. 6 shows the desired signal $y(k)$ and its approximation to the identification filter response $\hat{y}(k)$.

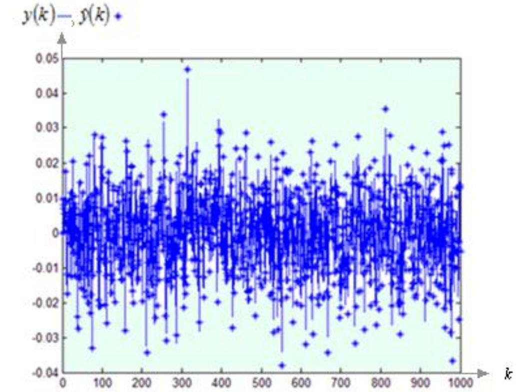


Figure 6. Desired and signal approximations.

In accordance with the reference model operation, the adaptive fuzzy process stage uses the inference mechanisms, selecting the dynamically the parametre value in agreement to (5). Fuzzy inference with respect to the error levels limit the rules set. Fig. 7 shows the error classification in order to select the best answer.

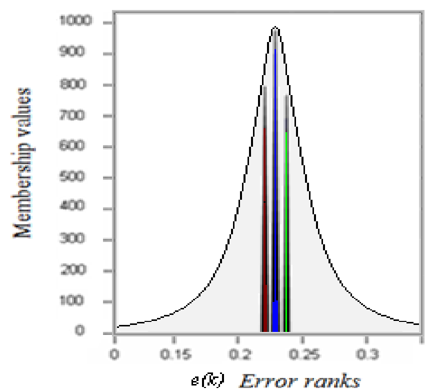


Figure 7. Error levels Classification.

In accordance with the estimation fuzzy filter into the selection process, Fig. 8 shows different filter response operational levels $\hat{y}(k)$.

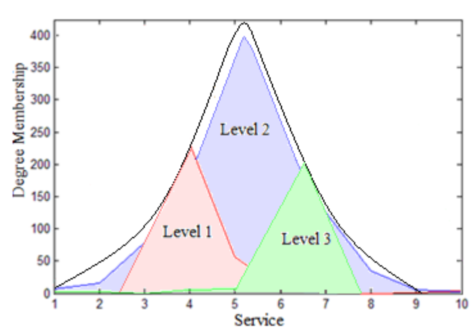


Figure 8. Filter degree operational levels process.

The distribution functions of reference output model and its identification converge in distribution sense viewed in Fig. 9. The identification filter answer describes the reference signal observing in Fig. 9, that in distribution; second converge to the desired answer, selecting into the knowledge base the best parameter, without permits that the filter response breaks the border.

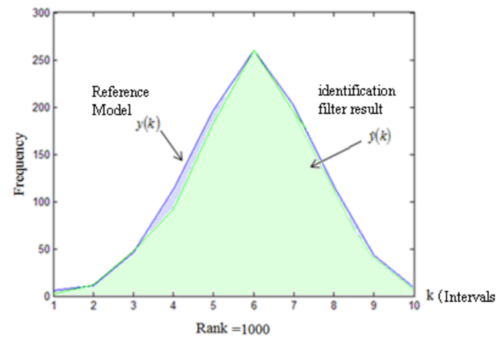


Figure 9. Identification filter bounded by the reference model, in distribution sense.

ACKNOWLEDGMENT

Thanks to Prof. Lorenz Whitehill Waterson (Native English Speaker), National Polytechnic Institute (IPN) and National Council of Science and Technology (Conacyt) for their support, to make this work possible.

REFERENCES

[1] Abdul, A. Fuzzy Logic, Imperial College of London, (2003).

[2] Ash, R. Real Analysis and Probability, E.U.A. Ed. Academic Press, (1970).

[3] García, J., J. Medel and J. Sánchez, Filtrado Neuronal Difuso: Caso MIMO, Revista Ingeniería e Investigación, Vol. 31 (No. 1), (2011).

[4] García, J., J. Medel and P, Guevara, Filtrado Digital Difuso en Tiempo Real, Revista Computación y Sistemas, Vol. 11 (No. 4), (2008).

[5] Haykin, S, Adaptive Filtering, E.U.A., Prentice Hall, (2001).

[6] Leydesdorff, L, The Construction and Globalization of the Knowledge Base in inter-human communication systems, Canadian Journal of Communication, Vol. 28, 267:289, (2003).

[7] Mollov, S., R. Babuska, J. Abonyi, and H. Verbruggen, Effective Optimization for Fuzzy Model Predictive Control, IEEE Transactions on Fuzzy Systems, Vol. 12, 661:675, (2004).

[8] Zadeh, L. The Concept of a Linguistic Variable and its Applications to Approximates Reasoning, Information and control, Vol. 8, 338:353, (1965).

[9] Schneider, M. and A. Kandel. Fuzzy expert systems tools, Wiley, Inglaterra, (1996).

[10] Margaliot M. and G. Langholz, New Approaches to Fuzzy Modeling and Control Design and Analysis, World Scientific, Israel, (2000).

[11] Mamdani, E, Applications of Fuzzy Algorithms for Control of Simple Dynamic Plant, Proc. IEEE, Vol.121, 1585:1588, (1974).

[12] Medel, J., J, García and J, Sánchez, Real-time Fuzzy Digital Filters Properties for SISO Systems, Automatic Control and Computer Sciences AVT, Vol. 42 (No. 1), 26:34, ISSN: 0146-4116, (2008).

[13] Passino, K. M. Fuzzy Control, E.U.A., Addison Wesley, (1998).

AUTHORS PROFILE

Dr. Medel, is a Prof. working from 1998 into Computing Research Centre. He has 3 patents, more that 51 international papers indexed. He consider that the fuzzy logic model descriptions is very important in the real life. Actually the Prof. collaborates with others researchers in Real-time and identification theory.

Dr. Garcia develops Associate Profesor activities with resect to filter theory using fuzzy logic tools. Actually, he has one graduated, and 10 indexed papers.

Dr. Sanchez is a Prof., interested in identification and filter systems with time restrictions. He has more than 74 indexed papers and more that 34 graduated.