17th International Research/Expert Conference
”Trends in the Development of Machinery and Associated Technology”
TMT 2013, Istanbul, Turkey, 10-11 September 2013

CONSTRUCTION OF WORLD MODELS FOR THE IMPLEMENTATION OF INTELLIGENT MANUFACTURING ALGORITHMS IN AGILE ARCHITECTURES FOR PRODUCTION: THE BINARY SYSTEMS CASE

Joaquim Minguella Canela
Fundació Privada Centre CIM – Escola Tècnica Superior d’Enginyeria Industrial de Barcelona, Universitat Politècnica de Catalunya
Llorens i Artigas 12, Barcelona
Spain

Irene Buj Corral
Escola Tècnica Superior d’Enginyeria Industrial de Barcelona, Universitat Politècnica de Catalunya
Av. Diagonal 647, Barcelona
Spain

Joan Ramon Goma Ayats
Fundació Privada Centre CIM
Llorens i Artigas 12, Barcelona
Spain

ABSTRACT
The present paper focuses on the construction of World Models for empowering a control system with intelligent behaviour, for the specific case study of binary systems; namely, systems in which each variable can take one out of two values. The model is built on a logical theory that incorporates the knowledge derived from the results of a set of experiments conducted by the system utilizing a set of different algorithms. In order to materialize the experimentation, it has been built a software capable of undertaking runs of experiments (guided by a human programmer or even by itself standalone) in order to generate a World Model. Based on this initial model, and given the specifications of the final desired state, the program is capable of setting a strategy to achieve the required goal, and to refine it in order to find the best possible strategy undertaking the minimum number of operations.
Keywords: Agile Manufacturing, Intelligent systems, Algorithms

1. INTRODUCTION
As studied in the literature, an automatic control system can perform virtually any process -assuming that the system has enough sensors, speed, degrees of freedom, etc. so to feedback the program information-. However, prior to that, the system will require a comprehensive programming operation in which the operator will have to take into account any eventuality that might occur during the performance. Moving forward, the *World Modelling* and the implementation of *Tasks Execution* from high abstraction levels is a suitable approach for empowering the control system with intelligent behaviour [1].
Some authors have addressed the embodiment of Intelligence in Manufacturing Systems by incorporating advanced sensors [2]. With this starting point, different data processing strategies can be implemented; such as Genetic Algorithms, Neural Networks and other metaheuristic methods [3, 4, 5]. Accordinly, the present work focuses on obtaining a World Model by conducting a set of experiments

utilizing a set of different algorithms. The model obtained is formed by the logical theory that incorporates the knowledge on possible states and world rules derived from the experimentation.

2. DEFINITION OF THE WORK ENVIRONMENT, SYSTEM OBJECTIVES AND CONSTRAINTS

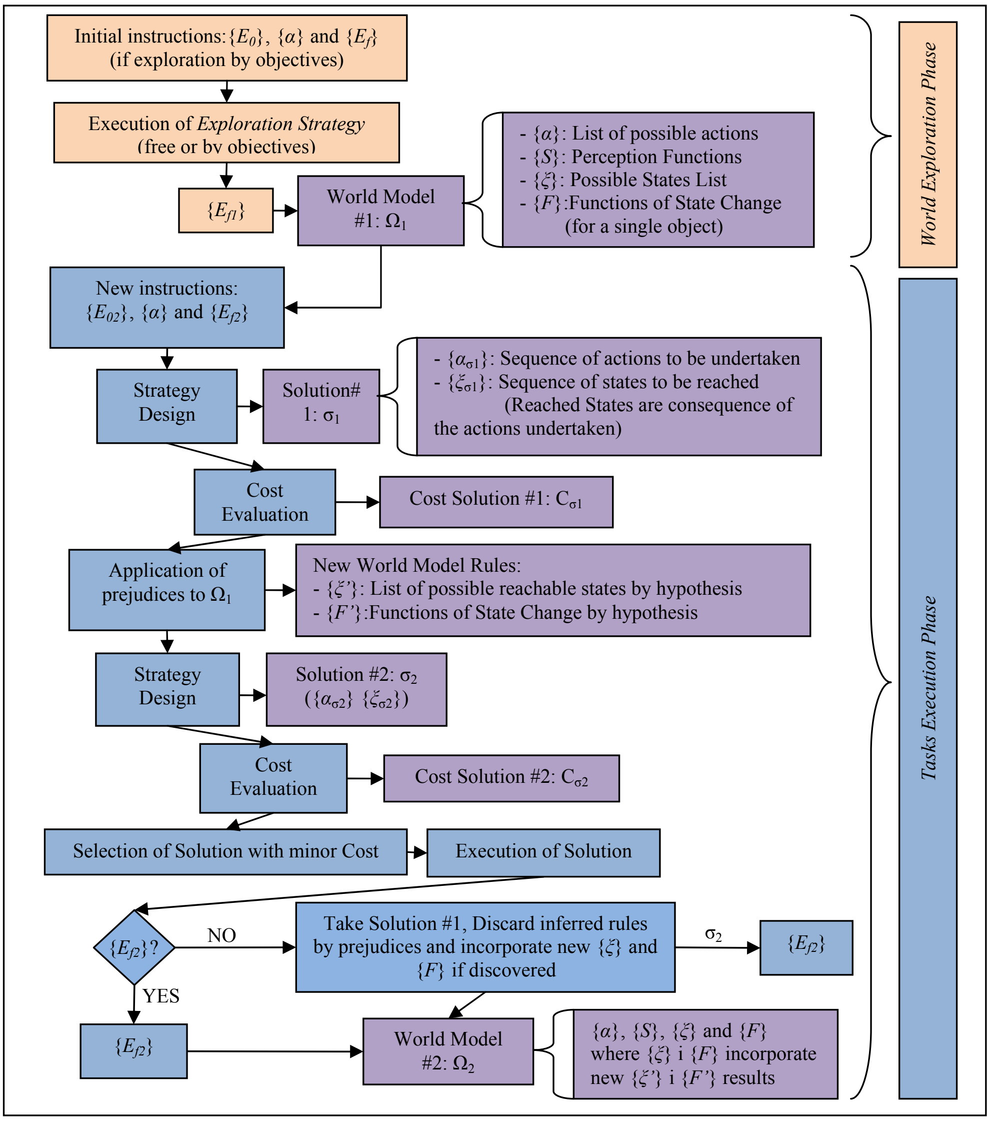
In order to explore new algorithms to construct world models, it has been defined a work environment based on the movements of a working head of a hypothetical machine tool on a 2D board. The variables corresponding to the sensors and actuators of the system can only take binary values (0 or 1). As for the sensors, there is one in each position of the board and its function is to indicate if this position is open (0) or occupied by the head or other obstacle (1). Moreover, there are only two actuators (motion along the 'x' or 'y'axis) so that the head can move into a position if corresponding variables of the actuator take value 1; or to remain motionless if both variables take value 0. This will cause the sensors providing information {S} to take also values 0 or 1.
The objective set for the system is as follows: the user defines an initial position of the head (which involves an initial state of the world $\{E_0\}$ and a final position of the head (which implies a final state of world: $\{E_f\}$). With these premises, the system has to be able to reach the final state without help from the user. In order to do so, the system must perform experiments $\{\alpha\}$ to learn the world rules {F} and the possible reachable states $\{E_i\}$. The results of different runs of experiments provide an information base to formalize a World Model $\{\Omega_I\}$. Based on this model, the system is able to design a strategy to move the head from its initial position to the final desired one.

3. BUILDING MODELS OF THE WORLD AND ACHIEVING STATES

Given the case study constraints, only three possible commands are considered: $\{\alpha1\}=\{0,0\}$, $\{\alpha2\}=\{0,1\}$ and $\{\alpha3\}=\{1,0\}$; i.e., actions caused bytwo actuators moving in both directions 'x'-'y' of the Cartesian plane in growing direction.As commands $\{\alpha_i\}$ are undertaken, each new world state $\{E_i\}$ reached implies a new possible state to be in: $\{\xi_i\}$. Also, the system can store the information of how the action $\{\alpha_i\}$ has undertaken to go from ξ_i to ξ_i +1, thus setting a table of state changes; or in other words, the set of functions of state change {F}.

The proposed algorithm is based on an iterative process that keeps evolving a World Model based on constant exploration of the World possible states. The algorithm starts by setting an initial state $\{E_0\}$ and defining a final state $\{E_f\}$, so to start running actions sequentially until a new state $\{\xi_i\} = \{E_f\}$ is reached. In this first stage –named '*World Exploration Phase'*the aim is to reach $\{E_f\}$, which is not to be exhaustive but effective.Once this phase is completed, the system is ready to proceed to the '*Tasks Execution Phase*'. In this phase, the system takes as initial information $\{\alpha\}$, {S}, $\{\xi\}$, and {F}, and uses it to reach a final state $\{E_{f2}\}$ starting from an initial state $\{E_{02}\}$; given that $\{E_{02}\}$, $\{E_{f2}\} \epsilon \{\xi\}$.
The resolution of the *Tasks Execution Phase* can be carried out easily if the World Model is exhaustive, through the use of a minimum path algorithm. Indeed, the World Model is treated as a graph where each node is anAchievable state, located at a specified distance from the following: to reach a state $\{E_{f2}\}$ from a starting $\{E_{02}\}$ consists on finding a practicable pathensuring a minimum distance track. Otherwise, if the Exploration has not been exhaustive, given the premise that $\{E_{02}\}$, $\{E_{f2}\} \epsilon \{\xi\}$, the system can always find a feasible route to the final state: 'σ' to be built from a sequence of actions and $\{\alpha_{\sigma1}\}$ consist of a route through different states (or nodes) of the graph: $\{\xi_{\sigma1}\}$.
In order to improve the solution by reducing the cost of the route needed, the algorithm incorporates the implementation of prejudices and abstraction rules, inferring cause-effect relationships from observations of similar cases. The introduction of new rules may result in the acquisition of new solutions 'σ_i' a cost '$C_{\sigma i}$' lower cost '$C_{\sigma 1}$' is obtained by solving the problem using only the states $\{\xi\}$ and shift functions State {F} discovered during the initial exploration. Before incorporating new $\{\xi\}$ and {F}, the system performs the chosen solution (σ_2) to make sure that it effectively reaches the desired final state $\{E_{f2}\}$. If the final state is not correct, the system rejects the new solution and keeps σ_1 (although incorporating new $\{\xi\}$ and {F} if discovered). Should the rules inferred for the solution (σ_2) met satisfactorily, the model incorporates all $\{\xi '\}$ and {F'} to the World Model (Ω_2), by extending the number of possible states and including New Functions of Change of State.

The process of World Modelling and Task Execution for achieving states is as presented in Figure 1.



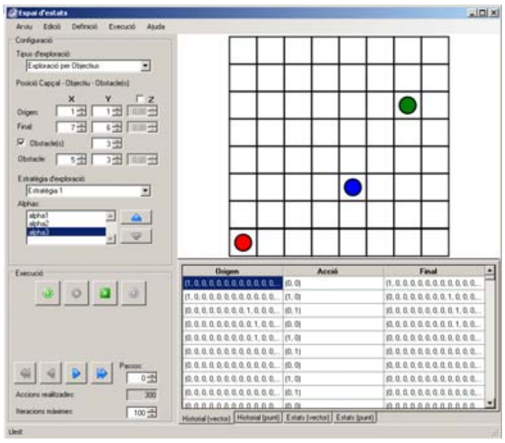
Initial instructions: {E_0}, {α} and {E_f} (if exploration by objectives)
Execution of *Exploration Strategy* (free or by objectives)
{E_{f1}}
World Model #1: Ω_1
- {α}: List of possible actions
- {S}: Perception Functions
- {ξ}: Possible States List
- {F}:Functions of State Change (for a single object)
World Exploration Phase
New instructions: {E_{02}}, {α} and {E_{f2}}
Strategy Design
Solution# 1: σ_1
- {$\alpha_{\sigma 1}$}: Sequence of actions to be undertaken
- {$\xi_{\sigma 1}$}: Sequence of states to be reached (Reached States are consequence of the actions undertaken)
Cost Evaluation
Cost Solution #1: $C_{\sigma 1}$
Application of prejudices to Ω_1
New World Model Rules:
- {ξ'}: List of possible reachable states by hypothesis
- {F'}:Functions of State Change by hypothesis
Strategy Design
Solution #2: σ_2 ({$\alpha_{\sigma 2}$} {$\xi_{\sigma 2}$})
Cost Evaluation
Cost Solution #2: $C_{\sigma 2}$
Selection of Solution with minor Cost
Execution of Solution
{E_{f2}}?
NO
Take Solution #1, Discard inferred rules by prejudices and incorporate new {ξ} and {F} if discovered
σ_2
{E_{f2}}
YES
{E_{f2}}
World Model #2: Ω_2
{α}, {S}, {ξ} and {F} where {ξ} i {F} incorporate new {ξ'} i {F'} results
Tasks Execution Phase

Figure 1. Process for World Modelling and Task Execution for the achievement of States

4. IMPLEMENTATION ACCORDING TO VISUAL STUDIO

To proceed with testing application algorithms it has been implemented a custom-made program runningunder Microsoft Visual Studio®. The software -named "*States Space*"-, implementsthe entire algorithm, including both the stage and the Exploration of the World of Running Tasks. The program is able to perform actions (automatically or by explicit indication the user) and to display the result in real time to actions taken in two ways: (i) graphically -within the window 2D '*World*'-, and (ii) digitally-within the '*History*' window-. Once the Phase corresponding to World Exploration has been successfully completed -resulting in a Model of the World; e.g.: {Ω_1}, the system can start the TasksExecution Phase (which can lead to new World Models: {Ω_2}, etc).

Figure 2 below shows the appearance of the interface for the exploration stage of the World, which includes the main windows: (Configuration, Execution, World Representation and World History).



Tipus d'exploració: Exploració per Objectius

Posició Capçal - Objectiu - Obstacle(s)

	X	Y	Z
Origen	1	1	0.00
Final	7	6	0.00
Obstacle(s)		3	
Obstacle	5	3	0.00

Estratègia d'exploració: Estratègia 1

Alphas: alpha1, alpha2, alpha3

Passos: 0

Accions realitzades: 300

Iteracions màximes: 100

Origen	Acció	Final
(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...	(0, 0)	(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...	(1, 0)	(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0...
(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0...	(0, 1)	(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0...
(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0...	(0, 0)	(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0...
(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0...	(1, 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, 1)	(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, 0, 0...
(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...	(1, 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, 1)	(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, 0, 0...

Figure 2. Screenshot of the program interface for the Exploration and Execution algorithms.

5. DEFINITION OF STRATEGIES AND EXPERIMENTAL RESULTS

According to the binary case, the possible positions of the head are discrete. In the case of the 2D grid of size 8x8, $\{\xi\}$ could grow to up to 64 different states (the initial plus 63 new). Also, a maximum of k=192 Functions of State Change ($\{F\} = \{f1, f2, ...fk\}$) will be possible to be met. However, the size of the $\{\xi\}$ and $\{F\}$ vectors will depend on the completion of the exploration undertaken.

The exploration strategies implemented in the study are (i) limiting the maximum number of actions undertaken or (ii) interrupting the exploration if the aimed $\{E_f\}$ is reached; leading to the construction of many different World Models. The *World Exploration Phase* can be guided by a human programmer or conducted by the programme itself. To facilitate analysis of the results, all the data obtained with the different experiments can be easily exported to text files or plain text type spreadsheet, which can be further processed by other software.

The rules underlying the world can be changed and become more complicated: (i) World without obstacles, (ii) World with immovable obstacles and (iii) World with obstacles that can be pushed by the machine head.To evaluate the solver accuracy of the different exploration methods the cost function definition has been taken in linear relation to the number of moves made.

6. CONCLUSIONS AND FUTURE WORK

The results achieved are aligned with the expected robust programme capable of testing and analysing World Models and undertaking Autonomous Tasks Execution. Also, the extension from *Discrete* to *Continuous* Worlds could be easily undertaken maintaining the algorithms and rationale by increasing the resolution of the possible positions; as well as the incorporation of more moving axes.

Therefore, the presented work for modelling a world (Ω_i) should be extended to scenarios populated by more than a single object as a step towards the effective implementation in real machine tool architecture.

7. REFERENCES

[1] Gomà, J.R. and Vivancos, J.; Construction by a Robot of a Logical Theory of Itself and its Environment through Experiments. Proceedings ETFA '99. Volume: 1, On page(s): 535-543

[2] Vojic S., Karabegovic, I.: Intelligent Systems in Welding Processes.Proceedings TMT2008

[3] Markic B. et al.: Integrating Genetic Algorithm and Rule Based Systems. Proceedings TMT2008

[4] Jun Li et al.: Rapid design and reconfiguration of Petri net models for reconfigurable manufacturing cells with improved net rewriting systems and activity diagrams. Computers & Industrial Engineering 57 (2009) 1431–1451

[5] M. Kazem Sayadi et al.: Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation. J. Manuf. Systems, Vol. 32, Iss.1, Jan 2013, p 78–84.