Structured Language of Decision Network

Soumaya EL MAMOUNE*, Mostafa EZZIYYANI**
* Faculty of Science and Technology/ computer science, Tangier, Morocco
** Faculty of Science and Technology/ computer science, Tangier, Morocco
soumayamgi@gmail.com, ezziyyani@gmail.com

***Abstract*— Based on adaptatif system and decision theory, this paper introduces a new language to manage Decision Network for proposing services to a dynamic event after an adaptation. The new approach presents several solutions to critiques situations but the most problem was how we can manage and structure the decisions in the database. To solve this problem, we propose a Structured Decision Network Language (SDNL) to define all the operations that can be used like add a decision, update, delete and link the decisions. The new language is some instructions to execute for having some operations.**

The main objective of this article is to present the new language and the instruction of this language. We also propose a case study on which we apply the instructions of the language. For that, we suggest a network for the managements of accidents during pilgrimage and we take part of the network to show how to construct a network decision by the language SDNL.

Keywords: SDNL, Decision Network, adaptation, Network Management, structured language, network evolution.

I. INTRODUCTION

A major challenge of adaptive systems is to present a comprehensive approach to adaptation to the environment for the construction of an optimal and effective action plan [1]. The organization of data in the adaptatif system was also a big problem of research [2] [3]. In this context, the Decision Network is a new approach that allows the system to adapt and offer services according to environmental data and the user profile [4]. The approach has been proposed as a graphical model used to structure the services offered according to rules and well-defined constraints. The components of this model must to allow an easy and effective use. The proper implementation of this model will facilitate communication and collaboration to meet the users with quality services.

This paper is structured as follows. In Section 1, we present the basic information of Decision Network. In the section 2, we propose the new language for managing the network decision and the syntax of this language. Finally, to clear up the instructions of this new language, we apply on a real case.

II. DECISION NETWORK

A. Definitions

The decision network is a new approach and adaptive decision support that can be used by adapting systems to provide an adaptive solution to the situation [4]. The idea is to structure the solutions offered by the system as a network. This includes services that are represented by elementary decisions interlinked by rules and constraints that we have defined for the operation of the system.

For the definition of our approach, we proposed several terms as "final decision" which is the set of services to be rendered in a given situation. Each service is called elementary decision and they are connected with each other via connectors used to specify the sequence of use of elementary decisions. Graphically depicts our approach as dependency graph structure where the decision network from the system knowledge base to model the dynamic behavior of the system.

B. Why a new approach

The new approach improves the system's ability to be scalable, adaptive and intelligent. The proposal provides very important features such as the assessment, improvement, and evolution. The degree of adaptability and using a decision used to verify the effectiveness of the decision for a given situation. This level is used to evaluate the efficiency of a decision for an improvement that can be deletion or update decision. The evolution of the system is through the development of the network which is enlarged in progressively use. The network is initially empty and is in operation, gradually creating decisions for a given context. The creation of decisions by themselves is sufficient that's why we offer other components that connect these decisions according to well defined rules.

C. The representation of Decision Network

The decision network (RDD) is an evolutionary graph to model the final decision of the system according to a dependency graph. A node in the graph represents a basic decision that could also be a final decision in a simple situation.

The graph proposed by us uses the standard notion of the directed graphs besides the new notations in order to formalize the new features of the decision graph proposed for the adaptive system. The decision graph is defined by the triplet **GD = { <N^{in}, C, N^{out}> }** whose elements are N^{in} (the node of entry), N^{out} (the node of exit) and C (the connector connecting the nodes realized by an arc) and, graphically, it is presented as shown in Figure 1.

Figure 1. triplet of the graph

After creating several triplets, we represent our decision network as a graph. Each decision graph node (GD) is a separate item (basic decision) and connected these points with connectors and bows.

III. THE NEW LANGUAGE SDNL

A. Why a new langguage?

After the proposal of the new approach, we continue our research and development of decision network by defining a new language that will handle the data of the proposed network. We aim to provide a total manipulation RDD by ordinary operations like creating basic decisions, updating and deleting. The new language will also manage all components of the graph as connectors and arcs. This language helps everyone to create his network and apply all operations whatever his level in computer sciences.

The proposed language is called SDNL (Structured Decision Network Language). SDNL is a computer language used to manipulate and organize all the components of decision network such as the creation of the elementary decisions and making the connection between connectors.

The new language addresses two areas:

- Manipulation of decision (nodes): create, search, add, edit or delete decision making in the network.
- The manipulation of the entire network such as the creation of links between decisions by creating connectors or arcs or generally modify the organization of decision making in the network.

B. SDNL Syntax

SDNL instructions are written in a simple way. They resemble that of ordinary declarative sentences in English according to a precise syntax. This resemblance referred facilitates learning, reading and understanding of the new language. The latter is a declarative language, that is to say, it can describe the transaction or the desired action, without describing how to get it and in parallel we are developing a program that automatically determines the optimal way to perform these operations.

The proposed language will define the main operations performed to manage the organization of network components.

1- Creation of graph

The graph is the element that collects the decision network. Before creating the components of the latter, we must create the graph that will be initially empty but filled as and creating custom other components. To create an empty graph, we use the following statement:

Create graph *'Id graph to create'*

2- Creation of elementary decision

The elementary decision is one of the main components of the graph. It represents the graph node. The elementary decision is not only the base part of the graph, but it is a member composed of a plurality of other elements which are shown below. With this, we guarantee the most important character is the evolution of the system.

TABLE I.
SYNTAX TO CREATE ELEMENTARY DECISION ELEMENTS

Element	Syntax
Elementary decision	**Create decision** *'Id of decision to create'* **in** *'Id of graph where we add the decision'.*
Action	**Create action** *'Id of action to create'* **as** *'defition of action or commande of action* **in** *'Id of decision where we add the action'*
Condition PRE	**Create pre** *'Id of condition pre'* **as** *'the condition to verify'* **in** '*Id of decision where we add the condition'*
Condition POST	**Create post** '*Id of condition post'* **as** *'the condition to verify'* **in** '*Id of decision where we add the condition'*

3- Creation of connector

Create a connector means, the birth of a relationship between two decisions to be determined later. The type of relationship is accurate according to the type of connector.

TABLE II.
SYNTAX TO CREATE THE DIFFERENT CONNECTORS

Type of connector	Syntax
Or	Create connector or 'Id of connector'
And	Create connector and 'Id of connector' in 'graph'
parallel	Create connector parallel 'Id of connector' in 'graph'

4- Linking components

After creating the decision network elements, it is necessary to link the components to construct the network by connecting the basic decisions with connectors and arcs. To construct this network, we suggest these instructions:

- Link decision *'Id of decision'* to connector *'Id of connector'.*

This instruction puts the link between a decision and a connector like Figure 2.

Figure 2. Link the decision to connector

- Link connector *'Id of connector'* to decision *'Id of decision'.*

This instruction puts the link between a connector and a decision like Figure 3.

Figure 3. Link the connector to decision

These instructions automatically create an arc between the two components but the choice of one of these instructions presented makes the difference. The first allows you to create an arc decision to the connector against the second connector creates an arc to the decision.

The arc is a single component but it plays a key role because it is impossible to connect two components without using the bow. In the language we propose, we presented above the arcing due to the relationship between the decision and the elementary connector but this relationship can also be broken following an arc suppression operation we define it according to the syntax follows:

Delete arc *'Id of arc'*

Id of arc is a combination of Id connector and Id of decision. It is created automatically after the connection between the decision and the connector

5- Modification

The decision to offer network (RDD) has a very important property that is improving. This improvement can be made by changing the network elements. It shows the syntax to follow to do. The updating is a modification of one or more components of elementary decision as necessary and useful for it to be effective.

We present the syntax of the update in the following table:

TABLE III.
SYNTAX TO EDIT THE ELEMENTARY DECISION ELEMENTS

Element	Syntax
Elementary decision	**Edit decision** *'Id of decision to update'* **to** *' The new of decision'* *This instruction update the name of decision*
Action	The modification of the action can be done by two things is changing the name of the action or is changing the definition of the action itself. **Modification of action name :** **Edit Action name** *'Id of action to update'* **in** *'Id of decision where it is the action'* **to** *'The new name of action'.* **Modification of action:** **Edit Action** *'Id of action to update'* **in** *' Id of decision where it is the action'* **as** *' the new action to execute it'.*
Condition PRE	Changing the PRE condition can be done by two things is changing the name of the condition or either the change in the definition of the condition itself. **Modification of condition PRE name:** **Edit condition pre name** *'Id of condition pre to update'* **in** 'Id of decision where it is the condition pre' **to** *'the new Id of condition'.* **Modification of condition:** **Edit condition pre** *'Id of condition pre to update'* **in** 'Id of decision where it is the condition pre' **as** *'the definition of condition pre'.*
Condition POST	Changing the POST condition can be done by two things is also changing the name of the condition or either the change in the definition of the condition itself. **Modification of condition Post name:** **Edit condition post name** *'Id of condition post to update'* **in** 'Id of decision where it is the condition post' **to** *'the new Id of condition'.*
	Modification of condition: **Edit condition post** *'Id of condition post to update'* **in** 'Id of decision where it is the condition post' **as** *'the definition of condition post'.*

6- Suppression

If necessary, the administrator can delete an elementary decision is not used or if it is incorrect. We offer several choices for this operation:

• Delete basic decision: it is total suppression that is to say after this operation all actions and all conditions constituting this decision will be deleted.

• Delete action: We can eliminate one or more shares in a decision provided it must keep at least one share in a decision. The decision will be modified but just the number of shares that will be changed following the abolition performed.

• Delete a Condition (Pre and Post): We can remove one or more conditions and this deletion just change the conditions of implementation of the decision.

TABLE IV.
SYNTAX TO DELETE THE ELEMENTARY DECISION ELEMENTS

Element	Syntax
Elementary decision	**Delete decision** *'Id of decision to delete'* This instruction deletes the decision identified by Id and all its components.
Action	**Delete Action** *'Id of action to delete'* **in** 'Id of *decision where the action exists'* The network administrator can remove N actions in a decision but it is forbidden to remove all the actions. The decision must have one action to execute it.
Condition PRE	**Delete pre** *'Id of condition Pre'* **in** 'Id of *decision where the condition exists'*
Condition POST	**Delete post** *'Id of condition Post'* **in** 'Id of *decision where the condition exists'*
Connector	**Delete connector** *'Id of connector to delete'* The suppression of the connector is the suppression of the connection between the decisions.

IV. APPLYING SDNL: MANAGING PILGRIMAGE ACCIDENTS

The year 2015 was marked by a human catastrophe during the pilgrimage. Two accidents were a cause of more than 1,000 dead. This disaster has upset the stability of thousands of families around the world and even pushed us to seek solutions for a more efficient organization.

In this context, we propose a management of problems or accidents encountered during pilgrimage as path loss, loss of contact with family, looking for people in an accident ... Several services can be offered but we chose some of them to present the Decision Network approach. The graph that we offer allows us to apply the instructions of the new language to better understand the utility of each instruction.

In this example, we present all steps to create a graph. For this, we will create the elements of the area surrounded with the linking of these elements.

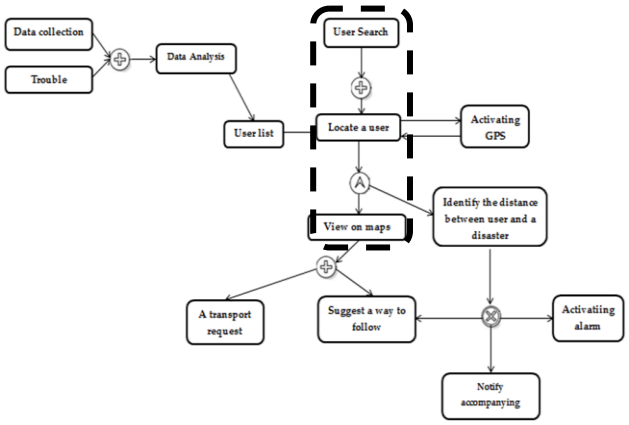

Figure 4. Decision Network to manage the pilgrimage accidents

A. Analysis

Before creating the graph, we must list all the elements of graph: elementary decision, connectors and arc. In this case, we will take a part of graph to apply our language as shown in the following figure:

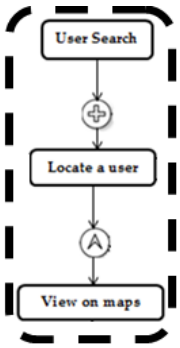

Figure 5. The network to create

If it is necessary like in an accident, the user can look for another user to locate him to see if he is in danger.

For this case, we have one graph which contains:

- Three elementary decisions
- Two connectors
- Four arcs

B. The steps of the Network decision creation

1- Creation of graph

The creation of the graph is the first step. Create a graph is creating an empty container that will be filled by the services. This requires a graph name under which we create. In this example, we create the graph "Hajj".

Create graph "Hajj"

2- Creation of elementary decision

The elementary decision is one of the main components of the graph. It represents the graph node. The elementary decision is not only the base part of the graph, but it is an element composed of several other elements. It is composed by one or more actions that will be performed.

In this example, we have three elementary decisions:

- User search
- Locate a user
- View on maps

First decision: User Search

Create decision 'User Search' **in** 'Hajj'

Create action 'information' **as** 'get information of user to search (Name, Code, Access permission)' **in** 'User Search'

Create action 'verification' **as** 'verification of user's information' **in** 'User Search'

Create action 'reply' **as** 'reply yes or no' **in** 'User Search'

Create pre 'Access permission' **as** 'Verify if user A has permission to search B' in 'User Search'

Create post 'Permission' **as** 'if the altitude is not null' **in** 'Locate a user'

Second decision: Locate a User

Create decision 'Locate a User' **in** 'Hajj'

Create action 'longitude' **as** 'get longitude' **in** 'Locate a user'

Create action 'Id user' **as** 'get Id of user to locate' **in** 'Locate a user'.

Create action 'altitude' **as** 'get altitude' **in** 'Locate a user'

Create action 'latitude' **as** 'get latitude' **in** 'Locate a user'

Create pre 'gps on' **as** 'if the gps is activated' in 'Locate a user'

Create post 'alt' **as** 'if the altitude is not null' **in** 'Locate a user'

Create post 'long' **as** 'if the altitude is not null' **in** 'Locate a user'

Create post 'lati' **as** 'if the latitude is not null' **in** 'Locate a user'

Third decision: View on maps

Create decision 'View on maps' **in** 'Hajj'

Create action 'gps coordinates' **as** 'get gps coordinates' **in** 'View on maps'

Create action 'address' **as** 'get address' **in** 'View on maps'.

Create action 'point address'**as** 'pointing the address on the map' **in** 'View on maps'

Create pre 'gps ok' **as** 'if the gps coordinates are not null **in** 'View on maps'

Create post 'maps ok' **as** 'if the map is correctly downloaded' **in** 'View on maps'

3- Creation of connectors

To link the elementary decisions we need the connectors. In this case, we need parallel connector and or connector so we create two connectors with two different Id.

Create connector parallel 'par1'

Create connector or 'or1'

4- Linking of components

Link decision ‘User Search’ **to connector** ‘or1’
Link connector ‘or1’ **to decision** ‘Locate a user’
Link decision ‘Locate a user’ **to connector** ‘par1’
Link connector ‘par1’ **to decision** ‘View on maps’

The instructions are very simple to use by any user no matter his level in computer science.

CONCLUSION AND PERSPECTIVES

To construct the decision network and not have just a theory, we propose this new language. It has several advantages such as evolution of network and ease of use. This advantage allows any user to build a network of decision and change it without difficulty. The instructions of this language manage the network and to apply this instructions we development also a tool of interpretation with the automatic execution of these instructions.

REFERENCES

[1] Jean-Pierre George, Marie-Pierre Gleizes, Pierre Glize, “Conception de systems adaptatifs à fonctionnalité émergente : la théorie Amas” (2007). Revue d'intelligence artificielle, ISSN 0992-499, vol. 17/4 - 2003, pp. 591-626. http://dx.doi.org/10.3166/ria.17.591-626

[2] Maya Rupert “Organisation du contenu du Web selon la perspective des Systèmes Complexes Adaptatifs”. CORIA'2006(COnférence en Recherche d'Informations et Applications), ARIA de Lyon France. pp. 283-294.

[3] Lotfi Bouzguenda, Manel Turki "Designing an Architectural Style for Dynamic Medical Cross-Organizational Workflow Management System: An Approach Based on Agents and Web Services, Journal of Medical Systems archive, Volume 38 Issue 4, April 2014, Pages 1-14; Plenum Press New York, NY, USA, http://dx.doi.org/10.1007/s10916-014-0032-2

[4] Soumaya El Mamoune, Mostafa Ezziyani and Jaime Lloret « Towards a New Approach for Modelling Interactive Real Time Systems Based on Collaborative Decisions Network ». Network Protocols and Algorithms (NPA), ISSN 1943-3581 Vol 7, No 1 (2015) April 2015.

[5] EL Mamoune S., Ezziyyani M. (2013). Adaptive Learning in the Simulation Of Emergency. In 3rd International Conference on Information Systems and Technologies (ICSIT’13) March 22-24, 2013, Tangier Morocco.

[6] EL MABROUK Marouane, KASSARA Omar, EL MAMOUNE Soumaya, EZZIYYANI Mostafa, ESSAAIDI Mohammad, GAOU Salma, ABAJJA Amal, “An Adaptive Intelligent Decision Support System for Real-Time Flood Forecasting”, International Journal of Research in Computer Applications and Robotics. Vol. 1, Issue 7, pp. 14-23, October 2013.

[7] Manavalan, Subrata Chattopadhyay, Mangala, Yalamanchili S.Rao, “Emerging trends of Computational Grid based Near Real Time/Real Time Flood Assessment and Forecasting Models”, in Proceeding of the Third International Conference on Emerging Trends in Engineering and Technology” November 9- 11, 2012, Kurukshetra, Haryana, India, pp. 471- 475, http://dx.doi.org/10.1109/ICETET.2010.54

[8] Amjad RATTROUT, Taroub ISSA, Hassan BADIR, "Intelligent Web system: Using Intelligent Web Agents to Achieve a Self-Organized System" IEEE Conf SKIMA09 Fes, Morocco, October 21-23, 2009. pp 93-101

[9] Jin Wang, “Development of a Decision Support System for Flood Forecasting and Warning – A Case Studyon The Maribyrnong River”, PhD thesis, Victoria University, Melbourne, Australia, January 2007.

[10] Eirinaki M., Lampos C., Paulakis S., Vazirgiannis M., "Web personalization integrating content semantics and navigational patterns", Proceeding of the sixth ACM CIKM international workshop on web Information and Data Management, November 05 - 11, 2006, Washington DC, USA. pp 72- 79. http://dx.doi.org/10.1145/1031453.1031468

[11] Giovanni Vigna, Fredrik Valeur , Richard A. Kemmere, “Designing and implementing a family of intrusion detection systems’’, CM SIGSOFT Software Engineering, Vol 28 Issue 5, September 2003, Pages 88-97, http://dx.doi.org/10.1145/949952.940084