A Smart Decision Support System for Smart City

Marco Bartolozzi, Pierfrancesco Bellini, Paolo Nesi, Gianni Pantaleo and Luca Santi
Distributed Systems and Internet Technologies Lab, DISIT Lab – http://www.disit.org
Department of Information Engineering (DINFO)
University of Florence
Italy
paolo.nesi@unifi.it, pierfrancesco.bellini@unifi.it, gianni.pantaleo@unifi.it

***Abstract*—ICT and social innovations are playing a decisive role in the development of technologically advanced urban areas. Smart City frameworks are devoted to address new challenges to improve efficiency and sustainability of services for citizens, providing additional features and allowing the city environment to adaptively configure according to collected data and information. To this aim, Decision Support Systems have recently been acquiring increasing importance in such a context. This paper presents a Smart Decision Support System for Smart City, based on the evolution of the Analytical Hierarchical Process model, which has been integrated with the Italian Flag 3-values logic representation. Other original contributes of the proposed work are (i) the capability integrating social and data processes by accessing and querying external repositories, in order to gather Smart City related data to be used to assist decision makers in the decisional process, through the use of properly defined functions and thresholds; (ii) the system is designed as a collaborative framework, allowing multiple users to share, clone and modify models and different instances of a same model. The proposed system has been validated in real use cases by exploiting decision processes and smart city data services of *Km4City* solution on Florence metropolitan area.**

Keywords — Smart City; Decision Support Systems; System Thinking, Anaylitical Hierarchical Process, Italian Flag.

I. INTRODUCTION

The term Smart City refers to an urban system aiming at fulfilling efficiency and sustainability criteria [1] within critical domains and application areas such as mobility, energy and environment management, administrative services etc. This goal can be achieved by exploiting Public Administration (PA) Open and private Data, different kinds of sensors and other data sources, upon which structured information and knowledge can be automatically extracted and infered, in order to make infrastructures and services more accessible and interactive. A study of the Universities of Ljubljana and Delft [2] highlights six key areas acting as indicators for assessing the "smartness degree" of a city: mobility, environment, social life and people, economy, general city governance and living quality. Many other Smart City benchmarks have been identified (http://www.smart-cities.eu, http://www.between.it). There are several examples in the world, such as the city of Barcelona for use of renewable energy. The city of London has developed the Urban OS project (an operating system created to manage computer automated infrastructures dedicated to the improvement of urban services such as water supply and transport). In Italy, cities such as Milan, Rome, Genoa and Florence have been active in the exploitation of ICT technologies, in order to improve traditional services for citizens with smarter and more interactive tools and applications (http://www.between.it).

A city is commonly composed of several different operational environments, infrastructures and networks which can be improved and optimized through the application of advanced solutions. The necessity arises to assess the current status of the City (through data coming from sensor networks placed in the urban area) and make decisions according to specific objectives and goals to be achieved. This implies the development of deeply connected infrastructures, evolving into and together with the Smart City environment. At the basis of such an approach there are computational methods and informative systems, such as Decision Support Systems (DSS), widely applied in many fields and domains for assisting the automation of decisional process, and System Thinking paradigms, consisting in analyzing and understanding the different needs and requirements to be met, taking into account relative benefits and disadvantages of all the constituting elements.

This work presents a new Smart DSS oriented to assist decisional process within a Smart City context (in which for "Smart" we intend capable to keep the decision assessment process always updated, on the basis of data, and to support decision makers in doing that in a more efficient manner). The system is based on Analytical Hierarchical Process (AHP) for automatic decision, properly modified in order to integrate it with the Italian Flag (IF) 3-values logic representation, which allows handling also uncertainty measures. The proposed system is provided with decision models built in a light collaborative manner among decision makers, who can share, reuse/clone and modify models, as well as use different instances of a same model in different context (e.g., geographically located in different locations of the city). The estimation of the Italian Flag probabilities and weights of decisional criteria can be determined by directly accessing Smart City Data. It is possible to pose SPARQL queries to a

RDF store for Linked Data (providing a generic available SPARQL endpoint URL), as well as querying SQL relational databases. The access to data can be adopted to make direct measures on the monitored territory, as well as to assess the citizens' opinions via live polls. Another possibility is the manual insertion of statistical values derived from interviews and workshops getting users' opinions.

The paper is organized as follows: Section II presents an overview of related work and state of the art of current literature. Section III describes the hierarchical model used to build the proposed Smart DSS (http://smartds.disit.org). Section IV is dedicated to explain the most relevant architectural details; while Section V is in charge of analyzing a real case study, in which the *Km4City* Service and model [3] has been exploited for retrieving data for the Florence metropolitan area. Finally, Section VI is left for conclusions and future work perspectives.

II. Related Work

Several approaches and techniques, supporting the process of decision-making, have been recently proposed and investigated. Among them, Goal models, goal state machines [4] integrated with systematic analysis have been proved to be useful in describing a system domain by properly capturing its requirements and allowing the evaluation of objectives achievement [5]. Techniques such as evolutionary algorithms, neural networks, fuzzy systems, and Bayesian networks have been widely used to support financial decision in economics and finance [6], [7], [8]. Recent solutions rely on System Thinking paradigms, oriented to problem solving and decision support in a Smart City environment. According to this approach, a modern city or urban area is seen as a highly interconnected entity, from a social and technological point of view. We find similar definitions in other approaches, such as Cyber Physical systems [9]. System Thinking has been recently adopted in Smart City contexts, as in the STEEP project [10] for energy saving planning and interventions, and also in wider contexts, such as rural environments [11]. Decision Support Systems have been widely studied and used in a large variety of application areas, from healthcare (Clinical DSS) to business and management, including also Smart City. This is due to their flexibility in assisting decision-making processes; they can actually be employed to solve even not well structured problems, combining also complex analytical models and techniques with more traditional data access and data recovery processes. DSS can be divided into five main categories, followed the taxonomy proposed by Power [12]:

- *Model-driven DSS* are focused on extrapolating analytical, mathematical or quantitative models from a general problem-solving task. Users and decision makers can manipulate models parameter to perform custom "what-if" analysis and to assess how this affects the system output [13].
- *Communication-driven DSS* provide coordination and communication among multiple users working on shared tasks and activities, thus enhancing collaborative collaboration and shared decision-making support.
- *Data-driven DSS* support manipulation of data time series (even large data collections, historical data, real-time data, internal or external data etc.), accessible through querying a data warehouse for specific purposes.
- *Document-driven DSS* are represented by computerized frameworks, integrating storage and computational technologies in order to support unstructured document retrieval and analysis.
- *Knowledge-driven DSS* rely on external knowledge in the form of best practices, computational procedures and rules, expert knowledge and problem solving expertise and other source of information which can be stored in logical structures, accessible and readable by machines and software agents [14].

Some software tools are available in the Web, developed for supporting evidence-based reasoning handling also uncertainty, such as Perimeta [15], developed by the Bristol University, which has the limit of being based only on the opinions of involved actors, without the possibility to access to external data and information.

The AHP model is a general evaluation method supporting complex decision-making processes [16]. It is based on values and judgments of individuals and groups, where judgments are determined on the basis of a multilevel hierarchical structure in order to achieve some defined goals. The AHP model allows to decompose the decision problem in a hierarchy of sub-problems, which are easier to understand and can be analyzed independently.

The proposed approach, integrated into the *Km4City* solution and with data accessible on the city, can be regarded as a mix of *Communication-driven, Data-driven, and Knowledge-driven*, approaches for DSS.

III. The Modifed AHP Model

The decision model at the basis of the DSS presented in this paper has been developed according to the System Thinking paradigm, focusing on the AHP model integrated with the IF (Italian Flag) representation structure, which is a confidence-based 3-values logic used to measure uncertainties (often reported in users opinion rates and interviews, or from soundages, questionnaires on the citizens [17]). Decision makers provide decision models, defining criteria and their hierarchy and decomposition in sub-processes. The term "model" addresses only the hierarchical structure without internal data. The term instance is connected directly to a model and contains the data (in terms of IF probability values and criteria priority weights) required to calculate the final decision (as later described). IF values are typically filled by manually inserting statistical values gathered from interviews, user opinions, workshops, etc. In addition, the proposed solution provides the capability of estimating such values through logical functions properly defined by decision makers on the basis of semantic query results on Smart City ontologies and Linked Open Data, or direct queries on other external databases. There is the possibility to share, reuse/clone and modify models as well as creating different instances of the same model, in order to dynamically adapt a general model to diverse and varying contexts, conditions and goals. Besides, cloned instances can be filled with data from scratch, as well as with data imported from another instance. Such a light collaborative workflow significantly shortens the time to activate decision processes, without creating confusion on the responsibility about the single decision model and process.. However, the possibility to perform operations of cloning, sharing and modifying models and instances in such a collaborative context require certain constraints, in order not to generate nor propagate errors and inconsistencies, or creating uncomfortable situations among the decision makers.

The development of the decisional process is carried out through the following steps (which will be described with further details in next subsections): first, the hierarchical model is defined; then, one or more instances can be generated from each model by filling the IF values for decisional criteria (through different modalities, as later outlined in Section III.C). Subsequently, the matrix for pairwise comparison has to be generated and weights for decisional criteria have to be determined. Finally, a bottom-up process performs an overall consistency check of IF probabilities for inner nodes and calculate the final decision, which is represented by the estimated IF values of the Goal (root) node.

A. Implementation of the AHP Model

As a first phase, the decision makers deeply analyzes the problem, organizing it in a hierarchical tree composed by different levels (in the proposed solution, if this work has been already performed or partially performed in the past, he/she can reuse a decision process or some parts). According to the AHP model, at the top of the hierarchy is the Goal, which is the root of the decision tree. The nodes belonging to the first level under the Goal represent the decisional criteria which have been defined to achieve the goal. Lower level nodes can describe sub-criteria, as well as alternatives to reach the Goal, and even properties of corresponding upper level criteria, organized in as many levels as those necessary to have a complete description of the problem.

The next step is the assignment of weights to each node. Such weights are defined as priority values (so that their sum, calculated for all criteria belonging to a same level, yields 1). In order to estimate priority weights, a set of pairwise comparison matrices is built. Each level identifies a different comparison matrix, in which the criteria of the considered level are compared in pairs using the Saaty's scale (shown in Table I). This rating scale assigns integers from 1 to 9 according to the relative importance between the compared elements. The procedure of pairwise comparison matrix generation, oriented to priority weights calculation, is described in more detail in Section III.D.

TABLE I. SAATY'S SCALE FOR PAIRWISE COMPARISON OF DECISIONAL CRITERIA.

Relative Importance	***Definition***	***Explanation***
1	Equal Importance	*The two compared conditions/criteria contribute equally to the objective.*
2	Weak or Slight Importance	*Intermediate state between 1 and 3.*
3	Moderate Importance	*One condition/criterion is slightly favorable over the other.*
4	More than Moderate Importance	*Intermediate state between 3 and 5.*
5	Strong Importance	*One condition/criterion is strongly favorable over the other.*
6	Stronger Importance	*Intermediate state between 5 and 7.*
7	Very Strong or Demonstrated Importance	*One condition/criterion is very strongly favorable over the other.*
8	Very, Very Strong Importance	*Intermediate state between 7 and 9.*
9	Extreme Importance	*The evidence favoring one condition/criterion over the other is of the highest order of affirmation.*
Reciprocals of Above Values	If one of the above non-zero values is assigned to a condition/criterion *i* when compared with another one, *j*, then the reciprocal value will be assigned to condition/criterion *j* when compared with *i*.	

B. Italian Flag

The IF representation is a three-value logic which extends the concept of the traditional two-value logic by providing also a measure of uncertainty. This has been considered to be a suitable representation for the development of the proposed DSS, which has been designed to receive input data from different sources, including citizens and experts opinions and feedbacks (therefore, potentially handling also uncertainty situations, e.g., "*I don't have an opinion yet*"). In such contexts, the belief that an event may occur or not, as well as the reliance that a generic proposition may be true or false, can be only partial, so that some level of belief is assigned to an uncertain state. By this way, given a generic proposition or event E, we can define its probability $P(E)$ as the evidence for E, $P(not(E))$ as the evidence against E and $1 - P(E) - P(not(E))$ as the measure of uncertainty. IF is a graphical representation of the above defined triple form $[P(E), 1 - P(E) - P(not(E)), P(not(E))]$, where $P(E)$ is depicted as a green bar, $1 - P(E) - P(not(E))$ is depicted in white and $P(not(E))$ in red, as illustrated in Fig. 1. A compact way to represent the IF record is to indicate explicitly the interval $[P(E), 1- P(not(E))]$, which reflects the representation of IF examples in Fig. 1. In Smart DSS, the traditional AHP model has been integrated with the IF structure. In the following, we will use the notation $g=P(E)$, $r=P(not(E))$ and, consequently, $w = 1 - (g + r)$ to define the green, red and white probability values, respectively.

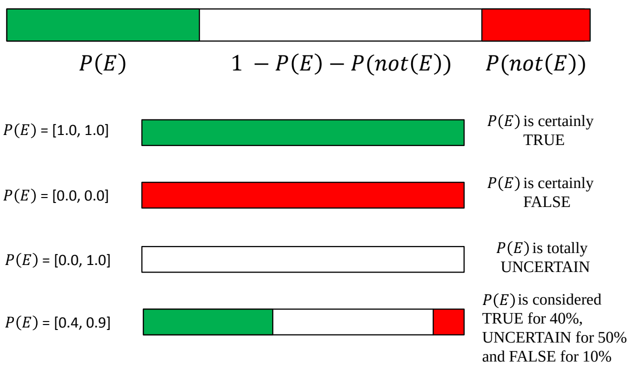

Fig. 1. Three-value logic IF representation for a generic proposition or event *E*, with some examples explained.

A general schema for the modified AHP hierarchy including the IF is shown in Fig. 2, in which the notation that will be used in the following is introduced.

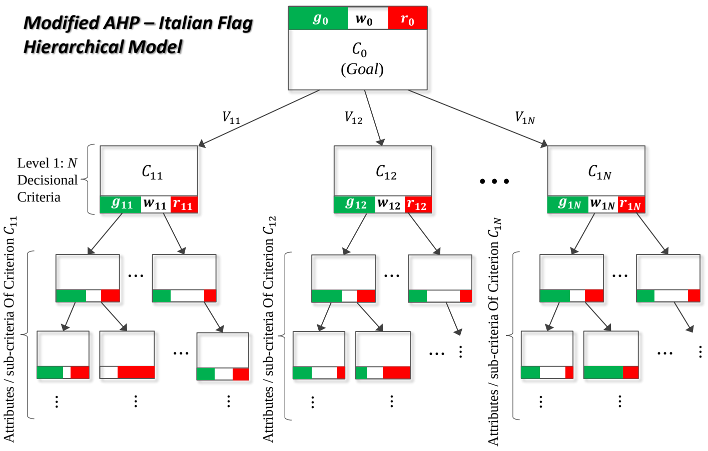

Fig. 2. General schema of the modified AHP hierarchical model.

C. Generation of Model Instances

At this step, once the modified AHP model is created, the decision makers can create an instance of a previously generated model, by filling the nodes of the hierarchical model with IF probabilities. Such values can be gathered from different sources:

1) Data from external semantic or relational database interrogation: in this case, the system provides a functionality which allows the decision maker to pose, for each node, queries to a generic RDF semantic repository (by providing valid SPARQL endpoint URLs in a proper field of the user interface), as well as to a generic SQL relational database. The query to database can be also performed to get resuklts from some online querionnaires. For instance, the PA that would assess the best new position for a bus stop that has to be moved in any way from a former location. A secons example could be: a business stakeholder, who is planning to open a new commercial utility in a certain area of the city, will be interested in posing queries to one or several Smart City repositories, in order to consider, for example, how many commercial services of the same type are located in the neighborhood, how many public transport facilities reach and connect the chosen urban area etc. Such expected query results are supposed to be numerical, actually the system produces non-null outcomes only for queries yielding numerical results. In order to obtain the required statistic values, the decision maker can define piecewise-constant logical functions by logically combining up to two queries whose results are numerically compared to user defined threshold values. An overview of a general logical construct, denoted as *<LCONSTR>* which can be defined for each criterion (in order to estimate the IF probabilities *g*, *r* and *w*), is given in the following, using EBNF notation:

```
<LCONSTR> ::= <Statement> [<LogicCond> <Statement>]
```

where:

```
<LogicCond> ::= “AND” | “OR”
<Statement> ::= g=g1; r=r1; w=1-(g1+r1) | <Function>
g1, g2, r1, r2 ::= prVal
prVal ::= “0" [ “.” digit, {digit}] | “1”
digit ::= “0”|“1”|“2”|“3”|“4”|“5”|“6”|“7”|“8”|“9”
<Function> ::= IF (<Q> <OP> <thres>) g=g1; r=r1; w=1-
     (g1+r1) ( ELSE g=g2; r=r2; w=1-(g2+r2) )?
<Q>::= <user defined numerical query>
thres ::= digit, {digit} [ “.” digit, {digit}]
<OP> ::= “=” | “<=” | “<” | “>=” | “>”
```

This representation explain how the system can manage up to two statements, logically connected through AND/OR logical operators. Each *<Statement>* can be represented by probability values *prVal*, defined as real numbers between 0 and 1. They can be directly assigned to IF values *g*, *r* and *w*, as well as by a logical function *<Function>*, which in turn is a piecewise-defined function assigning different values to *g*, *r* and *w*. Constraints on the definition of *<Function>* are based on a conditional expression defined by the numerical operator <OP>, a predefined threshold real value, *thres*, and a query <Q> generated by the decision maker upon a semantic repository or a SQL relational database, yielding a numerical result.

2) Data coming from opinions and feedbacks gathered by interviewing selected stakeholders or citizens groups. Opinions are directly mapped into IF values, assigning to the green value the percentage of opinions in favor of the addressed decisional condition or criterion, to the white value the percentage of uncertainty opinions (as well as answers not provided), and to the red value the percentage of opinions against the condition. After translating opinions into statistical values, these are used to fill the decision nodes tree as IF records.

3) Manual entry data: this kind of data is represented, for instance, by statistical values coming from the decision maker's experience, existing studies and collaborative workshops. Such entries are ready to be directly inserted as IF probabilities into each node of the hierarchy.

D. Pairwise Comparison Matrix and Priority Weigths

This step is devoted to identify and estimate the weights to be associated with each decisional criterion. As mentioned in Section III.A, this is done by using the evaluation matrix, whose single elements are obtained by pairwise comparisons of the decision criteria: Considering a generic level $\tilde{l}$ of the hierarchy, composed of N criteria $C_{\tilde{l}1}, \cdots, C_{\tilde{l}N}$, the pairwise comparison matrix is defined as:

$$P_{\tilde{l}} = \begin{bmatrix} p_{11}^{\tilde{l}} & \cdots & p_{1N}^{\tilde{l}} \\ \vdots & \ddots & \vdots \\ p_{N1}^{\tilde{l}} & \cdots & p_{NN}^{\tilde{l}} \end{bmatrix}$$

where elements p_{ij} are the Saaty's scale values for comparison between criteria. The pairwise comparison matrix P is composed of finite elements, it is positive-definite (that is, all minors of P are positive), its diagonal elements are equal to 1, and symmetrical elements stand in a reciprocal relationship:

$$p_{ij}^{\tilde{l}} = \frac{1}{p_{ji}^{\tilde{l}}}, \qquad 1 \leq i,j \leq N$$

This last property is in agreement with the Saaty's rating scale.

Once the pairwise comparison matrix $P_{\tilde{l}}$ has been generated for a certain level $\tilde{l}$ of the hierarchy, the priority weights for corresponding criteria are determined through the following procedure: first, a normalization by column is made over P, thus obtaining the $\hat{P}$ matrix. Keeping the assumption to have N nodes at level $\tilde{l}$, the $\hat{P}$ matrix is defined as:

$$\hat{P}_{\tilde{l}} = \begin{bmatrix} \hat{p}_{11}^{\tilde{l}} & \cdots & \hat{p}_{1N}^{\tilde{l}} \\ \vdots & \ddots & \vdots \\ \hat{p}_{N1}^{\tilde{l}} & \cdots & \hat{p}_{NN}^{\tilde{l}} \end{bmatrix} = \begin{bmatrix} \frac{p_{11}^{\tilde{l}}}{\sigma_1} & \cdots & \frac{p_{1N}^{\tilde{l}}}{\sigma_1} \\ \vdots & \ddots & \vdots \\ \frac{p_{N1}^{\tilde{l}}}{\sigma_1} & \cdots & \frac{p_{NN}^{\tilde{l}}}{\sigma_1} \end{bmatrix}$$

where:

$$\sigma_1 = \sum_{k=1}^{N} p_{k1}^{\tilde{l}}, \ldots, \sigma_N = \sum_{k=1}^{N} p_{kN}^{\tilde{l}}$$

Then, priority weighs are obtained by computing the arithmetic mean over the rows of the normalized matrix:

$$V_{\tilde{l}1} = \frac{1}{N}\sum_{k=1}^{N} \hat{p}_{k1}^{\tilde{l}}, \ldots, V_{\tilde{l}N} = \frac{1}{N}\sum_{k=1}^{N} \hat{p}_{kN}^{\tilde{l}}$$

E. Model Consistency Check and Final Decision Computation

Once a creation of a certain instance of a model is completed, before executing the final decision computation, the system is supposed to have in input well defined values for criteria priority weigths, as well as for the IF values of criteria at lowest level (leaf criteria). For inner criteria, IF probabilities can be defined by the decision maker (in one of the ways explained in Section III.C), or they can be left undefined; in this last case, they are calculated through the procedure described in the following. Such procedure is also in charge of validating the consistency of IF values for inner nodes where they are defined, in order to resolve potential inconsistencies between calculated values and existing ones. Following a bottom-up process, consistency for an inner *i*-th criterion at level $\tilde{l}$ composed of N nodes, is calculated as follows:

$$\begin{aligned} g_{\tilde{l}} &= \sum_{k=1}^{N} g_{[\tilde{l}+1]k} \cdot V_{[\tilde{l}+1]k} \\ r_{\tilde{l}} &= \sum_{k=1}^{N} r_{[\tilde{l}+1]k} \cdot V_{[\tilde{l}+1]k} \\ w_{\tilde{l}} &= \sum_{k=1}^{N} w_{[\tilde{l}+1]k} \cdot V_{[\tilde{l}+1]k} \end{aligned} \tag{1}$$

When an inconsistency occurs (that is, when the difference between calculated and existing values exceeds a user defined confidence threshold), the decision maker can choose among three alternatives: he may choose to set new bounds, by replacing existing values with the ones calculated in (1); in this case, the decision maker can select among different alternatives, e.g. setting new values for IF bands upper bound, lower bound or boths. As an alternative, the decision maker may replace existing values with those coming from new interviews and opinions, or he can leave the IF values as they are, without modifications. In any case, the IF values calculated by the systyem will be used for computation of the final decision. By this way, the decision maker is assisted in minimizing errors when filling instance values, due to complex and large model structures, as well as to the fact that models and instance can be shared, cloned and modified as part of a collaborative framework, increasing the risk of propagation of inconsistencies. At end of the whole bottom-up process, the IF values calculated in (1) for the Goal (root) node (for $\tilde{l} = 0$) yields the final decision triple result, providing that to each leaf criterion a valid IF record is assigned, and that each priority weight is defined. The final outcome is defined as:

- *Positive (favorable) outcome*, if $g > th$;
- *Negative (not favorable) outcome*, if $r > th$;
- *Uncertain outcome*, if $g <= th$ and $r <= th$;

where *th* is a threshold imposed by the decision maker.

IV. System Architecture

The Smart DSS framework presented in this paper has been designed and realized as a client-server web application implementing the model described in the previous section. The client side module allows the user to register and to subsequently perform operations on both models and instances definition and management (according to his granted permissions), such as creation, modification, cloning, saving and deletion. Moreover, the client offer the capabilities of defining logical functions (in order to gather data and information related to Smart City), as well as the creation of pairwise comparison matrices and threshold values to estimate IF values, decisional criteria weights and to calculate the finale decision coefficients. The server-side works as an interface for the client-side management and users administration, as well as an interface to a SQL relational database (for the retrieval and storage of data related to DSS models, users management and Smart City related data) and a RDF repository (for accessing Smart City data). The system provides four different user types:

- *Guest User*, who is not required for authentication; he can only view all created models and instances.
- *Registered/Advanced User*: after being registered, the advanced user can (in addition to the actions permitted to guest users) clone and modify models and instances created/owned by other users, and play with them.
- *Decision Maker* is a registered user who has been granted decision-making capabilities by the Administrator. Decision makers can (in addition to the actions permitted to registered/advanced users) create new models and instances, modifying and saving them, saving and deleting existing ones and importing data in cloned instances from other instance of the same model. Moreover, decision makers can also generate and modify pairwise comparison matrices, define logic functions and thresholds, create SPARQL and SQL queries for gathering Smart City related data, manually insert IF probabilities, execute the IF consistency check and eventually compute the final decision for a given instance of their own.
- *Administrator*, who can perform the same actions of a Decision Maker and, additionally, have the complete management of all registered users, being able to assign and change all the user roles.

The block architecture of the proposed Smart DSS is shown in Fig. 3.

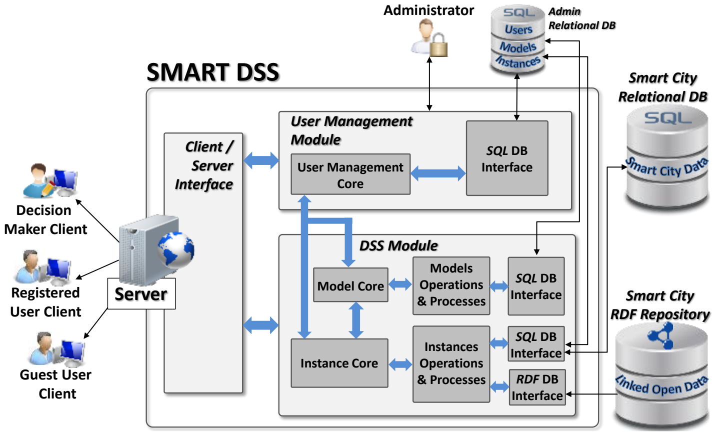

Fig. 3. Smart DSS Block Architecture.

The server-side application has been realized in Java, exploiting the REST paradigm, while XML has been chosen as the exchange format file for the execution of operations upon available resources. The server is functionally divided into two main modules: the *DSS module* and the *User Management module*. The former is in charge of managing DSS modules and instances, as well as the operations performed on them from users, accordingly to their roles and privileges; the latter is used by the administrator to control the different types of registered users and roles. Each of these two models is composed by a client/server interface, the module core (in which the main classes and objects are defined) and an additional interface dedicated to manage information to be gathered or stored from SQL and RDF databases.

The client has been developed in Javascript, exploiting the use of the D3 graphics library with and the jQuery framework. Modified AHP models and instances are represented with the traditional hierarchical tree view, where decisional criteria are represented by nodes (providing a text field for description) and priority weights of each criterion are associated with its incoming edges. The client-side user interface of the web application allows the users to perform the operation according to their privileges and roles, as formerly outlined.

V. A Case Study

In this section, a real world case study is presented, in order to show a complete workflow and processes to create and instantiate a decisional model in our Smart DSS. The addressed problems is to determine whether it is profitable or not to move a bus stop from a certain <*Location1*> to another <*Location2*>.

This is a typical example in which a smart DSS can be useful when dealing with the necessity of diverting a part of the public transportation service, whether temporarily or not, due for instance to modifications to the urban area road map, changes of traffic conditions, temporary works to public infrastructures, or concurrently to the organization of big events etc. IF probability values have been filled by collecting interviews and opinions from citizens. The *Km4City* [3] ontology, designed and developed at our DISIT Lab (using the open source Openlink Virtuoso tool) is used for gathering

Smart City data where needed. The system is capable to query different repository for each process/criterion.

TABLE II. DECISIONAL CRITERIA USED TO BUILD THE HIERARCHICAL MODEL FOR THE PROPOSED USE CASE (BUS STOP MOVING WITHIN THE CITY). ABBREVIATIONS IN THE "DATA TYPE" FIELD: Q=RDF QUERY; M=MANUALLY INSERTED; O=CITIZENS OPINIONS; NO VALUE IF THE FIELD IS LEFT EMPTY;

Goal	***1st Level Criteria***		***2nd Level Criteria***		***3rd Level Criteria***	
	Description	Data Type	Description	Data Type	Description	Data Type
G (= C0): Move a Bus Stop from *<Location1>* to *<Location2>*	***C1***: Modifications to the original Bus line route		***C1.1***: Distance from *<Location1>*	Q		
			C1.2: Keep the new bus stop on the same street of *<Location1>*	O		
	C2: Evaluation of logistic problems of new bus stop location		***C2.1***: Presence of works in the immediate vicinity of *<Location2>*	M		
			C2.2: Evaluation of roadway width at *<Location2>*	Q		
	C3: Evaluation of traffic flow		***C3.1:*** Private vehicles traffic flow in proximity of *<Location2>*		***C3.1.1***: Opinions from citizens	O
					C3.1.2: Reports from Public Administration	O
					C3.1.3: Data from Smart City repository	Q
			C3.2: PA Reports on Public Transport traffic flow in proximity of *<Location2>*	O		
	C4: Points of Interest in proximity (the same street) of *<Location2>*		***C4.1***: Commercial Services (shops & markets)		***C4.1.1***: Opinions from citizens	O
					C4.1.2: Data from Smart City repository	Q
			C4.2: Hospitals and healthcare centers	Q		
			C4.3: Educational Institutions (schools and University)	Q		
	C5: Number of bus lines passing by the old bus stop	Q				

For this use case, a three-level hierarchical model has been designed. A tabular view of the model and the chosen decisional criteria is shown in Table II, where the abbreviations in the "Data Type" field denote the different data sources: *"Q"* indicates that data from which the IF values are derived are gathered from the *Km4City* repository through well-defined logical functions of SPARQL queries; *"O"* stands for opinions and interviews collected among citizens and other actors like business stakeholders and Public Administration; *"M"* means that IF probabilities are manually inserted by the decision maker. The field is left empty whenever IF values are not defined (this case may occur only for inner nodes as stated in the requirements to be met in Section III.E); in this case, they will be calculated during the computation of final decision. Note also that the notation of criteria at different levels is slightly different from the one adopted in the general theoretical exposure in Section III.B and III.D, for a matter of clarity.

In Table III generalized SPARQL queries and probability values used (together with priority weights are reported, whose definition is omitted for brevity) to run the simulation of final decision computation. The following prefixes have been defined:

km4c: <http://www.disit.org/km4city/schema#>

km4cr: <http://www.disit.org/km4city/resource/>

The result of the decision process is shown in Fig. 4; in this simulation, considering a decision threshold of 0.5 (50%), the final decision results to be in favor the defined goal; actually the IF values for the goal (root node) results to be g_0=53.4%, r_0=38.6 and w_0=8.0%.

The solution provided allows keeping trace of the evolving values for the Smart DSS processes set up over time. The data obtained from the day by day activity collected from databases may change the IF of the global decision process. This fact does not mean that one would change decision in real time, while that the trends have to be monitored by the decision makers to detect dysfunctional cases and taking decisions.

TABLE III. SPARQL QUERIES AND PROBABILITY VALUES USED FOR CRITERIA DESIGNED FOR THE STUDIED REAL USE CASE (LISTED IN TABLE II). PREFIXES ARE DEFINED FOR THE FIRST QUERY ONLY.

Criterion	***Data Type***	***Value / Query Function***
C1.1	SPARQL Query	g = 1.0; r = 0.0; w = 0.0 if $Q <= Th_{11}$; g = 0.5; r = 0.25; w = 0.25 if $Q > Th_{11}$; Where Q: `SELECT (bif:st_distance(bif:st_point (<LONG1>,<LAT1>), bif:st_point (<LONG2>,<LAT2>)) as ?dist) WHERE { }` Note: Th_{11} is the defined threshold; <LONG1> and <LAT1> represent longitude and latitude of <Location1> (similarly for <LAT2> and <LONG2>).
C1.2	Citizens Opinion	g = 0.8; r = 0.2; w = 0.0.
C2.1	Manually Inserted	g = 0.0; r = 1.0; w = 0.0.

Criterion	***Data Type***	***Value / Query Function***

C2.2	SPARQL Query	g = 0.4; r = 0.6; w = 0.0 if $Q <= Th_{22}$;
		g = 0.6; r = 0.4; w = 0.0 if $Q > Th_{22}$;
		Where Q:
		
		`SELECT ?roadWidth WHERE {`
		`   ?road km4c:roadName <STREET_TOPONYM>.`
		`   ?road km4c:containsElement ?roadEl.`
		`   ?roadEl km4c:width ?roadWidth.`
		`}`
		
		Note: Th_{22} is the defined threshold; <STREET_TOPONYM> represents the street name of <Location2>.

C3.1.1	Citizens Opinion	g = 0.6; r = 0.1; w = 0.3.
C3.1.2	Reports from PA	g = 0.2; r = 0.65; w = 0.15.

C3.1.3	SPARQL Query	g = 0.4; r = 0.6; w = 0.0 if $Q <= Th_{313}$;
		g = 0.6; r = 0.4; w = 0.0 if $Q > Th_{313}$;
		Where Q:
		
		`SELECT ?TFlow WHERE {`
		`   km4cr:<#SENS> km4c:concentration ?TFlow.`
		`}`
		
		Note: Th_{313} is the defined threshold; <#SENS> is the identifier of a traffic sensor.

C3.2	Reports from PA	g = 0.6; r = 0.3; w = 0.1.
C4.1.1	Opinions from citizens	g = 0.4; r = 0.5; w = 0.1.

C4.1.2	SPARQL Query	g = 0.15; r = 0.75; w = 0.1 if $Q <= Th_{412}$;
		g = 0.75; r = 0.15; w = 0.1 if $Q > Th_{412}$;
		Where Q:
		
		`SELECT (COUNT(?service)) WHERE {`
		`   ?road km4c:roadName <STREET_TOPONYM>.`
		`   ?service a km4c:Shopping.`
		`   ?service km4c:isInRoad ?road.`
		`}`
		Note: Th_{412} is the defined threshold; <STREET_TOPONYM> represents the street name of <Location2>.

C4.2	SPARQL Query	g = 0.35; r = 0.45; w = 0.2 if $Q <= Th_{42}$;
		g = 0.55; r = 0.25; w = 0.2 if $Q > Th_{42}$;
		Where Q:
		
		`SELECT (COUNT(?service)) WHERE {`
		`   ?road km4c:roadName <STREET_TOPONYM>.`
		`   ?service a km4c:HealthCare.`
		`   ?service km4c:isInRoad ?road.`
		`}`
		Note: Th_{42} is the defined threshold; <STREET_TOPONYM> represents the street name of <Location2>.

C4.3	SPARQL Query	g = 0.25; r = 0.35; w = 0.4 if $Q <= Th_{43}$;
		g = 0.65; r = 0.2; w = 0.15 if $Q > Th_{43}$;
		Where Q:
		
		`SELECT (COUNT(?service)) WHERE {`
		`   ?road km4c:roadName <STREET_TOPONYM>.`
		`   ?service a km4c:Education.`
		`   ?service km4c:isInRoad ?road.`
		`}`
		Note: Th_{43} is the defined threshold; <STREET_TOPONYM> represents the street name of <Location2>.

C5	SPARQL Query	g = 0.7; r = 0.3; w = 0.0 if $Q <= Th_{313}$; g = 0.3; r = 0.7; w = 0.0 if $Q > Th_{313}$; Where Q:

Criterion	***Data Type***	***Value / Query Function***

		`SELECT (COUNT(?line)) WHERE {`
		`   km4cr:<BUS_STOP> km4c:hasSection ?x.`
		`   ?line km4c:hasRoute ?x.`
		`}`
		Note: Th_5 is the defined threshold; <BUS_STOP> is the identifier of the bus stop placed at <Location1>.

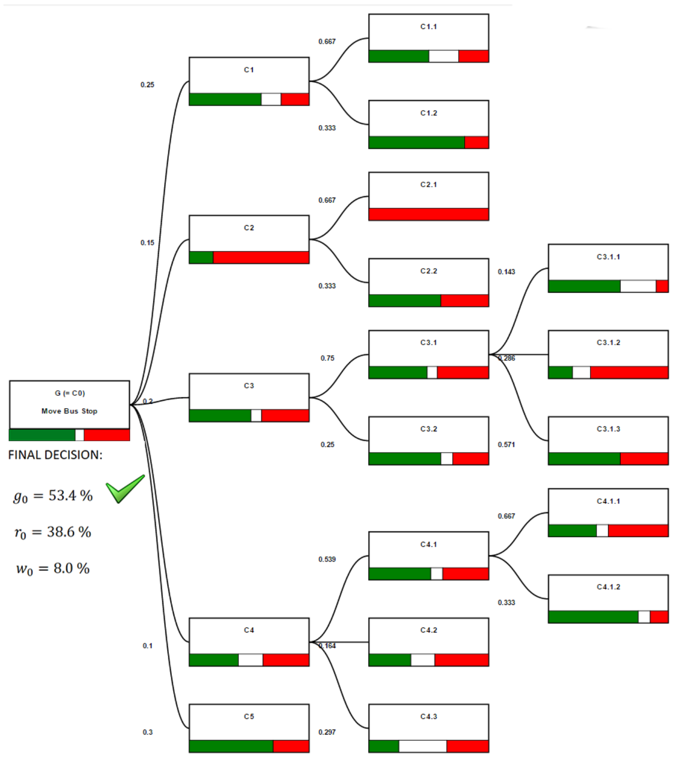

Fig. 4. Simulation results for the considered use case. Considering a 0.5 value for the decision threshold, the final decision results to be in favor of the proposed goal, yielding the following IF of the goal (root node): g_0=53.4%, r_0=38.6, w_0=8.0%.

VI. Conclusions

In this paper, a Smart DSS has been presented, designed as an evolution of the System Thinking model through the integration of AHP model with the IF representation. In addition, the system allows the evaluation of the consistency of IF values, and the definition of a collaborative framework for the creation and management of decision models and instances by multiple users. The proposed system has been designed with a particular focus on supporting decision-making processes in a Smart City environment; actually, it provides the capability of accessing Smart City related data (by querying external semantic repositories or relational databases), and using them to determine IF values leading to the final decision computation. However, the Smart DSS system can be used and applied to any other context in which it is required to automatically assist a decisional process. The model and solution proposed is accessible on http://smartds.disit.org.

A real case study has been studied and developed, in order to assess the effective capabilities and understand the expandability potential of the proposed solution. Future works and developments may be focused in implementing better solutions to check IF consistency (for example, by exploiting non-linear combinations of probability and priority weights). In addition, decision support may be provided in the form of background scheduled processes, integrated with Smart City data access and interrogation, in order to allow the system to adaptively modify its behavior in presence of changing contexts and unexpected events.

ACKNOWLEDGMENT

The proposed work and related activities have been developed within the Sii-Mobility Italian national project[1]. A special thanks to Florence Municipality for the collaboration in the context of Km4City, Sii-Mobility and other projects and activities.

REFERENCES

[1] S. Pellicer, G. Santa, A. L. Bleda, R. Maestre, A. J. Jara and A. G. Skarmeta, "A Global Perspective of Smart Cities: A Survey", in Proc. of the 7th Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 439 - 444 , 2013.

[2] R. Giffinger and N. Pichler-Milanovic, "Smart Cities: Ranking of European Medium Sized Cities", Vienna University of Technology, University of Ljubljana and Delft University of Technology. Available at: http://www.smart-cities.eu/, 2007.

[3] P. Bellini, M. Benigni, R. Billero, P. Nesi and N. Rauch, "Km4City Ontology Bulding vs Data Harvesting and Cleaning for Smart-city Services", International Journal of Visual Language and Computing, Elsevier, Vol. 25, pp. 827-839, 2014.

[4] L. Fu, X. Peng, Y. Yu, J. Mylopoulos, W. Zhao, "Stateful Requirements Monitoring for Self-Repairing Socio-Technical Systems", in Proc. of 20th IEEE Int Conf. on Requirements Engineering Conference (RE), pp. 121-130, 2012.

[5] J. Horkoff and E. Yu, "Analyzing goal models: different approaches and how to choose among them", in Proc. of the 2011 ACM Symposium on Applied Computing, pp. 675-682, 2011.

[6] A. Augello, G. Pilato and S. Gaglio, "A conversational agent to support decisions in SimCity like games", in Proc. of IEEE Int. Conf. on Semantic Computing, pp. 367- 372, 2009.

[7] R. Olsen, "Computational Finance as a Driver of Economics (Developmental tools)", IEEE Computational Intelligence Magazine, Vol. 3(4), pp.35-38, 2008.

[8] H. Ltifi, G. Trabelsi, M. B. Ayed and A. M. Alimi, "Dynamic Decision Support System Based on Bayesian Networks", International Journal of Advanced Research in Artificial Intelligence,Vol. 1(1), pp. 22-29, 2012.

[9] E. Schoitsch and A. Skavhaug, "Introduction to the 2012 ERCIM/EWICS Cyberphysical Systems Workshop", Lecture Notes in Computer Science Volume 7613, pp. 343-346, 2012.

[10] STEEP Project. [Online]. Available: http://www.smartsteep.eu/

[11] Y. Knodratenko, "How Can Systems Thinking Be Used to Support Rural Development Decisions in Latvia", Master Thesis, University of Lund, Sweden, 2003.

[12] D. J. Power, "Web Based and Model Driven Decision Support Systems: Concepts and Issues", in Proc. of American Conf. on Information Systems, 2000.

[13] D. J. Power and R. Sharda, "Model-driven decision support systems: Concepts and research directions", Decision Support Systems, Vol. 43(3), pp. 1044-1061, 2007.

[14] M. M. Hamad and B. A. Qader, "Knowledge-Driven Decision Support System Based on Knowledge Warehouse and Data Mining for Market Management", Int. Journal of Application or Innovation in Engineering & Management (IJAIEM), Vol. 3(1), 2014.

[15] Perimeta tool. [Online]. Available: http://www.bris.ac.uk/civilengineering/research/systems/projects/perimeta.html

[16] T. L. Saaty, "Decision making with the analytic hierarchy process", Int. Journal of Services Sciences, Vol. 1(1), pp. 83-98, 2008.

[17] J. P. Davis, J. W. Hall, "A software-supported process for assembling evidence and handling uncertainty in decision-making", Decision Support Systems, Vol. 35(3), pp. 415-433, 2003.

[1] Sii-Mobility Project: http://www.sii-mobility.org