World Wide Web (2013) 16:379–398
DOI 10.1007/s11280-012-0183-3

# Adaptive support framework for wisdom web of things

**Yang Gao · Mufeng Lin · Ruili Wang**

Received: 8 January 2012 / Revised: 16 April 2012
Accepted: 31 July 2012 / Published online: 14 August 2012


**Abstract** Wisdom Web of Things (W2T) is the next generation of networks, which provides ubiquitous wisdom services in a ubiquitous network in the hyper world. Adaptiveness is the key issue of realizing the harmonious unity of human-information-thing. This paper proposes a self-adaptive support framework for W2T, which has three important components: (i) An adaptive requirement description language, which is to describe the wisdom service models and self-adaptive wisdom service strategies. (ii) Forward reasoning and backward planning ability. We propose that forward reasoning can be implemented based on the Rete algorithm and backward planning can be implemented based on a Hierarchical Task Network (HTN), which enable W2T to achieve complex, rapid, and efficient reasoning and planning to provide active, transparent, safe, and reliable services. (iii) A knowledge base evolution mechanism based on a learning classifier system, which is to realize the evolution of the knowledge base, and to satisfy the dynamic requirements of wisdom services. We take a wisdom traffic system as an example to demonstrate the data conversion mechanism and the functions of the proposed self-adaptive support framework.

**Keywords** wisdom web of things · adaptive suppport framework · adaptive requirement description language · Rete · hierarchical task network · learning classifier system

## 1 Introduction

With the development of Artificial Intelligence (AI) technologies, web technologies and the Internet/Web of Things (I/WoT), the Wisdom Web of Things (W2T) has been proposed recently [20, 31]. W2T is an extension of the wisdom web in the IoT/WoT age, which is to enable each thing in the Web of Things (WoT) be aware of itself and others to provide the

Y. Gao (✉) · M. Lin
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu, China
e-mail: gaoy@nju.edu.cn

M. Lin
e-mail: forest.sky.sea@gmail.com

R. Wang
School of Engineering and Advanced Technology, Massey University, Palmerston North, New Zealand
e-mail: R.Wang@massey.ac.nz

right service for the right object at the right time and in the right context. The hyper world consists of things in the physical world, information/computers in the cyber world and empowering human in the social world [31]. W2T technology is a practical way to realize the harmonious symbiosis of humans, information/computers, and things in the hyper world.

There are, however, still some fundamental issues in W2T, which need to be addressed:

- From the data point of view, it is required to realize an effective data cycle system, which is to achieve the cycle "from things to data, information, knowledge, wisdom, services, humans, and then back to things." [31]
- From the intelligence point of view, it is required to balance the scalability and personalization of the Wisdom Web of Things. Also, the intelligence and wisdom need to be redefined in W2T.
- From the system point of view, it is required to realize the harmonious symbiosis of humans, information/computers, and things in the hyper world.

In the hyper world (i.e. in a particular domain), there are needs to study how to provide active, transparent, safe, and reliable services on the basis of mass data collected from SEA-nets (a number of sensor nets, embedded appliance nets, and actuator nets) and of distributed storages and computing, and then to provide useful information for decision making. In the meantime, such services also require sensing equipment, computing resources, and storage devices to be adaptive. This feature is called adaptiveness, which means a system can change its structure and/or behavior when the internal and/or external context changes. Self-adaptiveness means realizing adaptiveness with closed-loop structure autonomously [15].

Self-adaptation is one of the key techniques of realizing "harmonious intelligence". Self-adaptiveness can satisfy the changes in users' needs, computing resources and services' functions, and data forms. With self-adaptation, W2T can then realize the harmonious symbiosis of humans, information/computers, and things in the hyper world.

The data cycle system in W2T provides the basic structure for self-adaptation:

- In the conversion from data to information and from information to knowledge, and from knowledge to wisdom, the data preprocess, data analysis and data mining methods need to be able to adapt to data scales and types;
- In the conversion from wisdom to services, self-adaptiveness can handle different services and service compositions based on the available computational resources;
- In the conversion from services to human, self-adaptiveness can provide different personalized services according to the changes in users' needs. When there exist multiple choices of services or service compositions that can satisfy users' needs, self-adaptiveness can re-composite or re-select services based on the runtime status of services and the feedback information that reflects the users' evaluations of the quality of the services.

Although we still need to further the research in specific techniques of realizing self-adaptation in W2T, in this paper, we mainly focus on a self-adaptive support framework for the Wisdom Web of Things. Through a W2T application, a Wisdom Traffic System (WTS), we describe a framework, which is based on a double-looped control cycle combined with a reasoning and planning mechanism, a learning mechanism and a self-adaptive requirement description language.

This paper is organizes as follows. Chapter 2 describes the adaptiveness in Wisdom Web of Things. Chapter 3 proposes an adaptive framework for W2T. Chapter 4 illustrates the key technologies for adaptiveness of W2T. Chapter 5 discusses the case study of applying the adaptive framework of W2T to a wisdom traffic system. Finally, conclusions are presented in Chapter 6.

## 2 Adaption in wisdom web of things

Zhong et al. propose that the wisdom in the WoT means to provide the right service for the right object at the right time and in the right context [31]. However, as indicated by Heraclitus that “everything changes and nothing remains still… and… you cannot step twice into the same stream”, W2T is required to provide a right dynamic service for a right dynamic object at a right dynamic time and in a dynamic context to satisfy a dynamic need in the dynamically changing hyper world. In other words, WoT entities need to be able to adapt to internal and external changes.

Similar to a biological/food chain in the nature world, living creatures adapt to each other to accomplish harmonious coexistence. Every individual in the hyper world composed of human in the social world, information/computer in the cyber world, and things in the physical world should be able to adjust itself initiatively to adapt to other individuals in order to reach the harmonious coexistence of all individuals in the hyper world. Different from the traditional information technology, human are neither the master of a thing in the hyper world, nor on the top of the biological chain. Human are just an ordinary component in the hyper world. Hence, human or organizations also need to adapt to changes in the physical world and the cyber world to realize “harmonious intelligence” of the hyper world.

The hyper world is a type of complex self-adaptive system. Compared to the complex adaptive system (CAS) proposed by Holland et al. [7], the hyper world is not only composed of some homogeneous agents, but also composed of heterogeneous agents with relationships in a biological/food chain (“food” in the context of W2T is the data processing results such as information, knowledge, wisdom, or services). Similar to a biological/food chain in nature, the self-adaptiveness of the hyper world can be divided into two levels: (i) At the system level, the hyper world, as an ecosystem, needs to be adaptive to its environment or context; (ii) At the component level, the individual component needs to be able to adapt to the others [15]. We call the former case as “for adaptation”, and the second case as “with adaptation”. In other words, WoT can be seen as W2T, once the WoT is for adaptation and with adaptation.

The hyper world needs to achieve a kind of harmonious intelligence. To achieve such harmonious intelligence requires a comprehensive usage of new technologies such as Web intelligence, brain informatics, ubiquitous intelligence, and empowering individuals. Brain is the material basis for all kind of wisdom. The brain has many intelligent abilities such as vision, hearing, language, memory, learning, emotion, planning, reasoning, abstracting thought and decision making. Currently, the intelligent mechanisms that are verified by brain informatics include neural network, reinforcement learning. [25–27, 32]

Although Web intelligence emphasizes enhanced the applications of intelligent and information technologies to the WWW, we divide the top 10 problems of Web intelligence [31] into two levels, which are demonstrated in Figure 1 from brain informatics to Web intelligence. The lower level emphasizes the individual intelligence including meta-knowledge, language, sensitivity, reproduction, reasoning, self-aggregation, while the high level emphasizes the group intelligence, including coordination, roles, relationships, and personalization. This composes the information basis for all kinds of wisdoms of human or organizations.

Corresponding to the social world and cyber world, the physical world is another significant component of the hyper world. Everything in the hyper world needs to realize Ubiquitous Intelligence (UI) supported by Cyber Individuals (CIs). A smart u-thing (the real physical things are called u-things if they are attached, embedded, or blended with computers, networks, and/or some other devices such as sensors, actors, e-tags and so on [12–

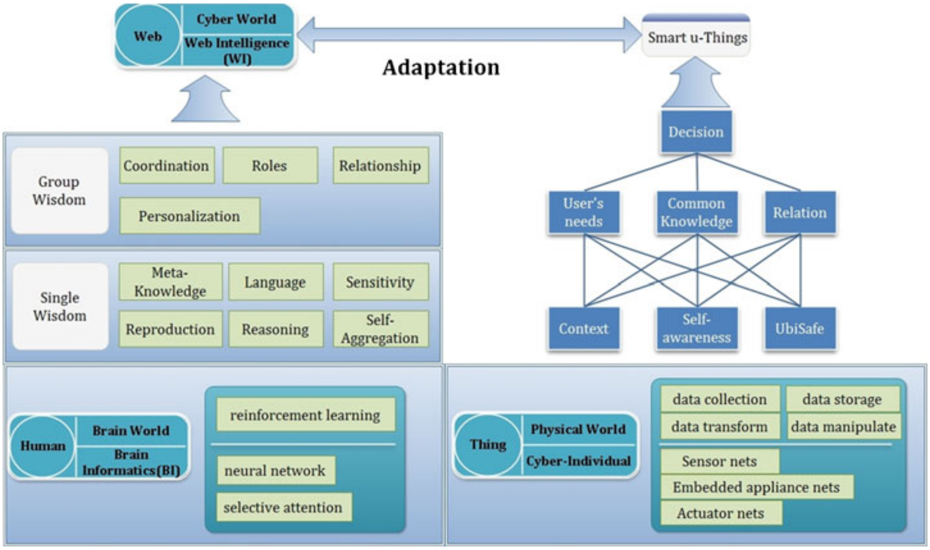


**Figure 1** Harmonious intelligence and self-adaptiveness in the hyper world.

14]) should be able to act adaptively and automatically. Zhong et al. proposes that the construction of the smart u-things has seven challenges [31]. The seven challenges are consistent with self-adaptiveness proposed in this paper. For the wisdom web, the ability of adaptiveness means that when facing changing application requirements, human-information-thing relationships, and the surrounding environment, the wisdom web can use double closed-loop decisions to realize self-awareness with the support of common knowledge and domain knowledge. Obviously, all decisions and the procedures of self-awareness need to satisfy ubiquitous safety.

Here is a simple example to illustrate the relationship between harmonious intelligence in the hyper world and adaptiveness. Suppose every adult in the social word has a cyber-individual counterpart in the cyber world. A cyber-individual (CI) has all information about the individual anytime and anywhere. For example, there is a CI named A, who releases a wedding photo on her Facebook, then the wisdom web can determine a CI named B, who is A's spouse. After that, the wisdom web modifies the relationship between A and B. In the meantime, according to common knowledge and the context, for example, the marriage system in A's country (assume to be monogamy) and the honeymoon custom after marriage (assume they will have an overseas travel), the wisdom web first deletes the information of both A and B on dating websites, then register A and B on a proper travel and hotel website based on the preferences of A and B. Apparently, due to the change of A's marriage status, the wisdom web collects data autonomously, and makes decisions based on the data. Therefore, the decisions in the wisdom web are a kind of self-awareness, which can adapt to its internal changes and requirement changes.

## 3 Adaptive support framework for W2T

The new generation of the WWW will enable users to gain new wisdom of living, working, playing, and learning, in addition to data communications, information searches and knowledge queries [11, 28]. For example, a Web quiz based on Wikipedia, automatically navigation based on Google map, social network analysis [18, 21], recommendation techniques [23] are preliminary applications of W2T. The early Web intelligence emphasizes on handling, organizing and mining Web information (in a specialized way), and the related technologies [19, 24, 29, 30] include Web information retrieval [2, 10], Web information management [1, 8], Web mining [22] and Web agent [16]. Although intelligent human-Web interaction and ubiquitous computing have been mentioned in the early research of Web intelligent, human is the subject of technologies, and Web is the object of research in these technologies. Liu [11] proposed that wisdom is not only referred to be possessed of more knowledge, but also means to take more advantages of knowledge. Also, from the W2T's point of view, human/organizations, information/computers and things are the important components of the hyper world, and all of the three components interact with each other intelligently. Therefore, the three components of the hyper world will eternally be in dynamic harmony and unity.

Compared with an ordinary web that only contains information and knowledge, wisdom web in W2T has more powerful capabilities. Especially, W2T can realize harmony and unity of human-information-thing in a dynamic environment. Figure 2 illustrates the adaptive support framework for W2T. We propose the novel support framework, which has the double-closed-loop structure: the outer loop is designed for realizing "for adaptation" that receives the context and environmental information in order to adjust system behavior; the inner loop realizes "with adaptation" that evaluates the system's behavior and changes an individual component behavior to accomplish better functionality or performance. As shown in Figure 2, when receiving requests, based on the knowledge that W2T has, a wisdom service will perceive the context of the system online, and then update the components and data that the wisdom service requires in order to satisfy the requests of human or organizations. In the meantime, the wisdom service also perceives the appraisals provided by users and the physical system's information, then W2T can update the knowledge base of W2T, and new knowledge will be added to the knowledge base of W2T.

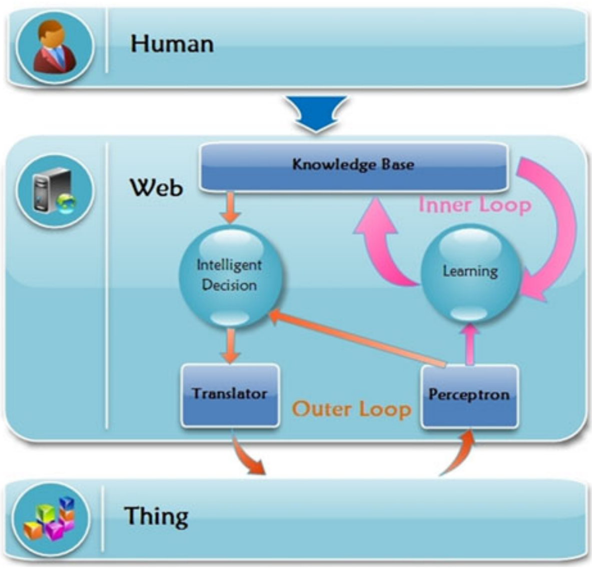


**Figure 2** Self-adaptive support framework.

For realizing the harmonious unity of human-information-thing, the support framework should equip with the following three key capabilities: (i) meta-knowledge, i.e. the knowledge of knowledge; (ii) intelligent decision making. In order to realize autonomy, W2T abstract definite goals from WoT using common knowledge, meta-knowledge, reasoning, and planning techniques to realize goal-oriented decision making; (iii) learning. For the purpose of handling individual users' needs, W2T needs to adapt to the variation of users' requests, and in order to satisfy the requests better, the existing knowledge needs to be improved via learning.

Figure 3 gives a more detailed description of the components in Figure 2, and depicts the particular process of realizing self-adaptiveness in W2T, including the following steps:

(0) An expert user defines domain ontologies and strategies according to the characters of wisdom services.
(1) A general user or an organization defines adaptive requirements based on the requests of functions, performances, security, etc.
(2) W2T monitors the context information of the hyper world at runtime, and preprocess data.
(3) According to the explanation of self-adaptive requirements, WoT calls the policies and strategies in the knowledge base.
(4) WoT makes decisions based on the reasoning and planning of policies and strategies in order to satisfy the dynamic requests of users.
(5) The decisions are conversed to certain actions, which affect every components of the hyper world.
(6) WoT acquires the context information when the decisions take effect.
(7) WoT estimates the decision-effect context information and policies and strategies in the hyper world.
(8) According to the estimated information, W2T refines the weights, parameters, or structure of strategies in order to improve the knowledge base.
(9) W2T adds new knowledge into the knowledge base.

## 4 Key technologies for adaptiveness of W2T

We can see from Figure 3 that in order to achieve self-adaptiveness, W2T requires three key technical components, which are the three basic challenges: (i) to build a reliable knowledge

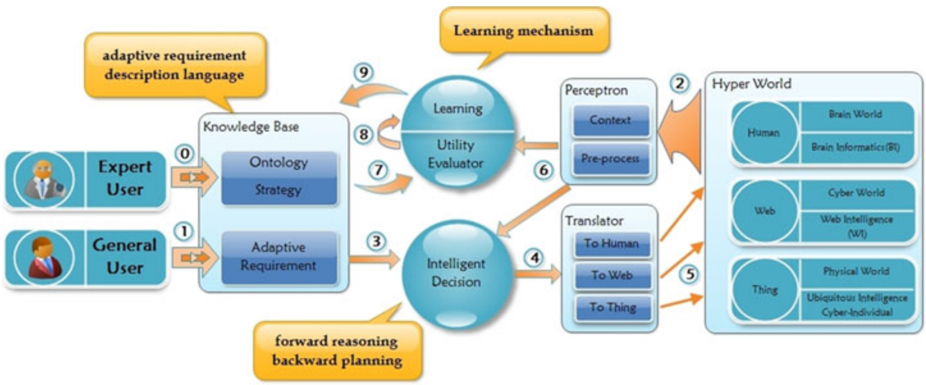


**Figure 3** The self-adaptive procedure in W2T.

base requires an adaptive requirement description language that has a strong ability of expressing knowledge and supporting effective reasoning; (ii) to realize efficient forward reasoning and backward planning techniques. A forward reasoning capability built on a logical language and backward reasoning capability built on the hierarchical networks, with consideration of the effective implementation of the decision process; (iii) to realize an effective learning mechanism that is able to evolve the knowledge base of W2T to adapt to its internal and external changes. In this section, we will discuss these three key technical components further.

## 4.1 Adaptive requirement description language

The requirements of a traditional information system are expressed as definite goals of the system. However, the procedures of realizing the goals are not able to conduct any adaptive operations according to the changes of the physical system. In W2T, in order to realize adaptive wisdom, we need to firstly describe the context of wisdom services, and then define the adaptive strategies which are applied to refining the dynamic requirements into different sub-goals (eventually as a sequence of primitive tasks). In this article, we extend the STITCH language (an adaptive requirement description language), which is used to descript the wisdom services and self-adaptive strategies, i.e. the meta-knowledge and domain knowledge of the wisdom services [4].

The adaptive requirement description language includes two parts, i.e. wisdom-service description files and self-adaptive wisdom-service strategy definition files. All these files are in the XML form.

- The wisdom-service description files include the architecture model of wisdom services (FAMILY), the instance descriptions of wisdom services (INSTANCE), the environment property monitors of wisdom services (PROBES), the property measurement of wisdom services (GAUGES), and operations (OPERATORS) of wisdom-service files.
- The self-adaptive wisdom-service strategy definition files define utilities and self-adaptive hierarchical strategies, including system utility (SYSTEMUTILITY), low level strategies of self-adaptation (TACTICS), and high level strategies of self-adaptation (STRATEGIES) files.

Figure 4 illustrates the relationships and interactions between the components of the adaptive requirement description language for wisdom services. The architecture model of the wisdom-service files (FAMILY), which defines the basic structure of the attributes and components of wisdom services, is a public glossary for all other files. The instance descriptions of the wisdom-service files (INSTANCE) are instantiation of the architecture model of the wisdom services. Therefore, the basic structure of the attributes and components of wisdom services mentioned in the INSTANCE files must be derived from the FAMILY files. The operations of the wisdom-service files (OPERATORS) define the operations which the wisdom services are able to conduct, so that the identifiers in the INSTANCE files will be referred in the OPERATORS files. The identifiers in the INSTANCE files will also be referred in the environment property monitors of the wisdom-service files (PROBES). The OPERATORS in the PROBES files must be defined in the OPERATORS files before, and they can support the environment property monitor operations. The property measurements in the wisdom-service files (GAUGES) define the source of the instance properties of the

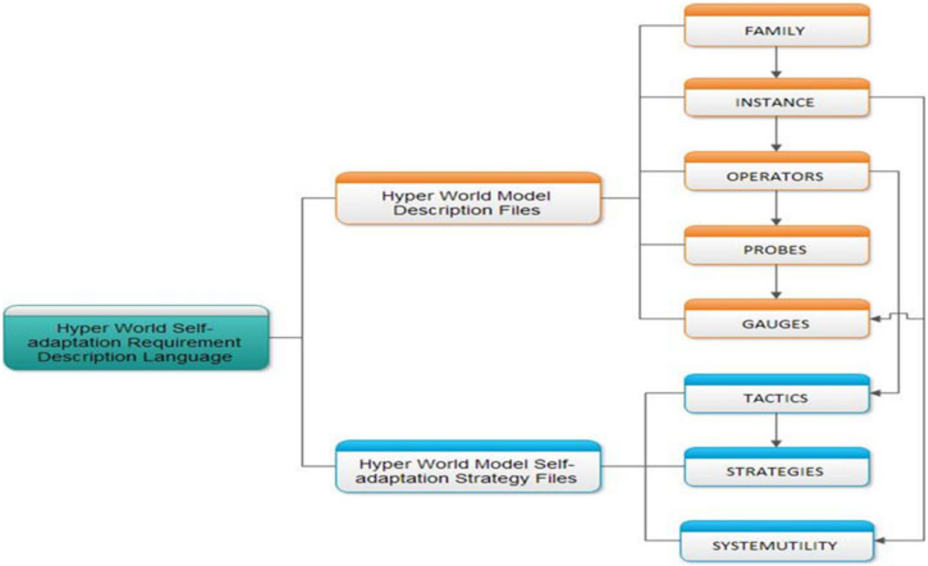


**Figure 4** Adaptive requirement description language for wisdom services.

wisdom services, and the property measurements in the GAUGES files must correspond to the properties defined in the INSTANCE files.

The self-adaptive strategies definition files for wisdom services need to refer to the identifiers defined in the INSTANCE files, and the attributes used in the low level strategies of self-adaptation (TACTICS), and high level strategies of self-adaptation (STRATEGIES) files must correspond to the attributes in the INSTANCE files.

Consider the scenario of the runtime traffic inducement during fastigium. The scenario is described as that various vehicles far and near are heading for one crossroad from west to east. The traffic inducement system undertakes several responsibilities. More specifically, when the vehicle queue is long at the crossroad, the system could notify the vehicle drivers and give them some advice. For the vehicles, which are already approaching to the crossroad, the system could suggest them to change the driving direction and turn out of the crossroad. For the vehicles, which are a block away from the crossroad, the system could notify the drivers to detour to avoid the traffic jam. When a traffic jam happens at the crossroad, the system could handle the traffic jam via human inducement or the interval changes of the traffic lights. Thus, the knowledge base for the example is presented as the following.

Among the seven files, the FAMILY file should be defined firstly. As previously alluded, the FAMILY refers to one specific family of problem and mainly declares necessary attributes of necessary entities, possible structures among them and environmental properties. FAMILY includes three entities (*Crossroad*, *Vehicle, Policemen and their attributes like Crossroad.position, Vehicle.position, Vehicle.direction etc.*) and some properties (*EastSingnalLightInterval* and *EastTrafficLength*). And the INSTANCE file is just an instantiation of FAMILY and all values within the file are filled in the real time as the current snapshot of the target system. Here in the example, every time the INSTANCE has one specific *Crossroad, different numbers of Vehicles and several Policemen.*

Next, The OPETATORS file describes primitive and executable actions provided by the target system. Here in the example, the OPETATORS include Get*SignalLightInterval*, *CalculateEastTrafficLength*, *DecreaseSignalLightInterval, InformDetour, InformChange-Direction* and DispatchPolicemen. The PROBES and GAUGES specify how to fill the values in the INSTANCE file. Here the *GetSignalLightInterval* is used to fill the *East-SingnalLightInterval* and *CalculateEastTrafficLength* to fill the *EastTrafficLength*.

Then, the SYSTEMUTILITY file needs to be defined to express the utility value of the whole target system in terms of some values and/or their compositions in the INSTANCE file. Here in the example, the utility value of the whole system is defined as negative number of the *EastTrafficLength.* The definition indicates the larger the *EastTrafficLength* is, the lower the whole utility value is.

Finally, the users need to define both the low level TACTICS and high level STRATEGIES files. Due to the simplicity of the example, the TACTICS are listed, *DecreaseSignalLightIntervalTactic, InformDetourTactic, InformChangeDirectionTactic and DispatchPolicemanTactic*. And the STRATEGIES include *CrossroadNotificationStrategyOne, CrossroadNotificationStrategyTwo* and *CongestionInducementStrategy.*

## 4.2 Reasoning and planning in W2T

The intelligent decisions in W2T include forward reasoning and backward planning. With the changing of business requirements and the context of the hyper world, W2T needs to provide active, transparent, safe, and reliable services based on the knowledge in the knowledge base. Basically, the requirements of wisdom services include functions, performances, security, costs, etc. We take functions and performances as examples: for performance

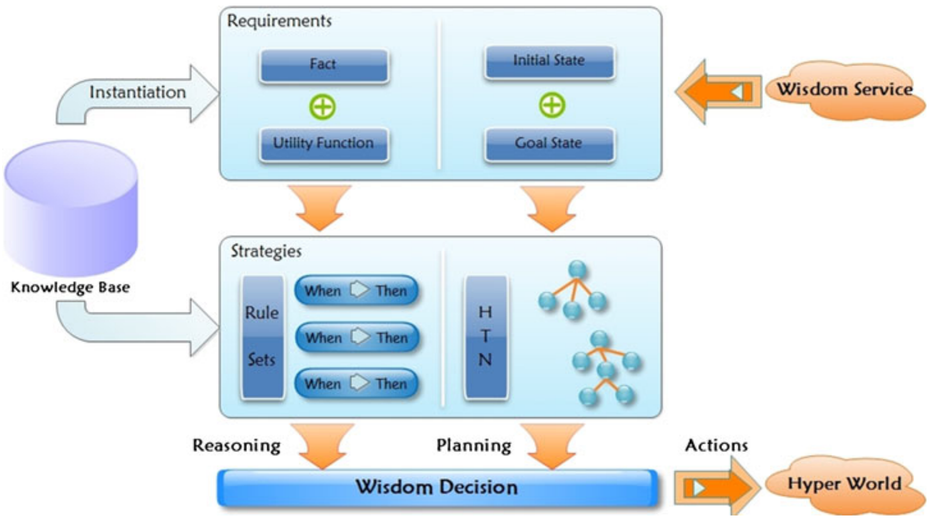


**Figure 5** The procedure of intelligent decision making.

requirements, W2T needs to do forward reasoning based on the context of the hyper world, and then to switch different service components in order to satisfy the users' requirements for performances. For functional requirements, W2T also needs to do backward planning based on the context of hyper world, and then provide proper services and service compositions.

Figure 5 demonstrates the procedure of intelligent decision making. When the requirements of wisdom services change, W2T first instantiates the related knowledge in the knowledge base according to the certain context and requirements. If the requirements are related to performances, then W2T generates a series of proper policies according to the evaluation of utility functions. If the requirements are related to functions, then in the light of the initial state, goal state, and the hierarchical task network in the knowledge base, W2T produces a sequence of primitive tasks. If the results of reasoning and planning have collisions or multiple choices, then W2T generates decision results by means of conflict resolution. Finally, the decision results will be interpreted to primitive actions and take effect in the hyper world.

#### *4.2.1 Forward reasoning based on the extended Rete algorithm*

The forward reasoning of the self-adaptive support framework involves two elements: rules and facts. Rules are in the form of When→Then, where "When" represents a matched condition or conditions, and "Then" means the triggered operation(s) when the condition(s) is matched. Facts express the current context of the hyper world, and are applied to matching the condition parts of rules.

The pattern matching process is often applied into the forward reasoning algorithms that are directly based on the first-order logical language. The first order logical is undecidable, and the pattern matching process can thus be slow when applied to large-scale and complex problems. The Rete algorithm proposed by Forgy is an efficient method for forward reasoning [6, 17]. The algorithm can extract all the antecedents and consequents of rules separately, and constructing a tree like structure, besides, it conserves the temporary matched facts in its networks to achieve the augment matching. Therefore, the Rete algorithm implements an effective pattern matching algorithm. However, for augment matching, the Rete algorithm needs to conserve a huge amount of temporary facts during the matching process. However, when facing with the massive and distributed data in W2T, the classical Rete algorithm cannot be satisfied efficiently. As a result, a heuristic Rete algorithm is proposed as the forward reasoning algorithm when facing to massive data in W2T [31].

The general idea of the heuristic Rete algorithm is as follow. In a specific appliance with massive data, there are similarities between rules or between facts. For instance, in a wisdom traffic system, if we have known a crossroad which has a long period of the signal light intervals is more likely to have congestions from previous data, we can expect that a new fact i.e. a crossroad whose traffic light interval is more than a threshold will tend to have a congestion whatever other properties it has. The heuristic algorithm learns these relevancies during matching and uses the relevancies to heuristically search similar facts from history in the next matching step. In Algorithm 1, the heuristic Rete algorithm is described in detail. The algorithm includes three steps: (i) construct the Rete network; (ii) search a candidate node which is similar with the fact; (iii) speed up the reasoning process.

Table 1 gives an example of the heuristic Rete procedures. Two of the strategies and the instances exemplified in 4.1 are defined in Table 1. The heuristic Rete algorithm firstly builds a Rete network for the two strategies. At the matching state, fact $c_1$ is first asserted in the network

Algorithm 1. The extended heuristic Rete algorithm

```
input rules from TACTICS file and STRATEIGES file, fact F from INSTANCES file, a
heuristic depth h
build root node
for all rules do
        For all patterns do
              build an ObjectTypeNode based on the pattern type
              for all fields do
                      if the field is an 1-input field then build a corresponding alpha node.
              End for
              build an alpha memory
              for all fields do
                      if the field is a 2-input field then build a corresponding beta node.
              End for
              build a beta memory
         end for
   end for
  find a candidate set of terminal nodes which have potential matching object type
  for all candidate terminal nodes do
       choose a candidate terminal node N
       set current memory to the previous beta memory of N
       do while h>0
              find similar facts in current memory
              if   there is a similar fact then
                  terminal the matching fact
              else
                  backtrack to the previous memory of current matching node to find
                  similar facts
               end if
               h--
       end while
       if there is no similar facts in the memory then go to root node.
  end for
  Propagate fact to all type nodes
  If fact match a type node or alpha node then
     propagate it to the sink nodes of this type node
  end if
  for all beta nodes do
       join left input tuples with right input facts
       propagate the join results to the sink beta nodes
  end for
  for all terminal nodes do
       if there are tuples in the node memory then
          activate the rule of this terminal node
  end for
```

**Table 1** Example strategies and instances for the heuristic Rete reasoning algorithm.

*CrossroadNotificationStrategyOne*:
When c:Cross(eastSignalLightInteval>20 and eastTrafficLength>30)
v:Vehicle(position=c.position and direction=east)
Then *InformChangeDirectionTactic*
*CrossroadNotificationStrategyTwo*:
When c:Cross(eastSignalLightInteval>20 and eastTrafficLength>30)
v:Vehicle(position=c.position and direction=east)
Then *DecreaseSignalLightIntervalTactic*
Instances:
$c_1$:Cross(position="streetA",eastSignalLightInteval=30,eastTrafficLengh=50)
$v_1$:Vehicle(position="streetA",direction=east)
$v_2$:Vehicle(position="streetA",direction=east)
……
$v_n$:Vehicle(position="streetA",direction=east)

and is reserved in the alpha memory. When fact $v_1$ is asserted, it matches with $c_1$ in the second beta node, and both *CrossroadNotificationStrategyOne* and *CrossroadNotificationStrategyTwo* are activated. Because of the heuristic mechanism, fact $v_3$- $v_N$ does not need to match as tradition, they just find similar facts in the beta memory before terminal nodes. In this example, fact $v_3$- $v_N$ all find fact $v_2$ as a similar fact, so they directly activate strategy 1 and strategy 2. Figure 6 describes the reasoning process in the heuristic Rete reasoning algorithm.

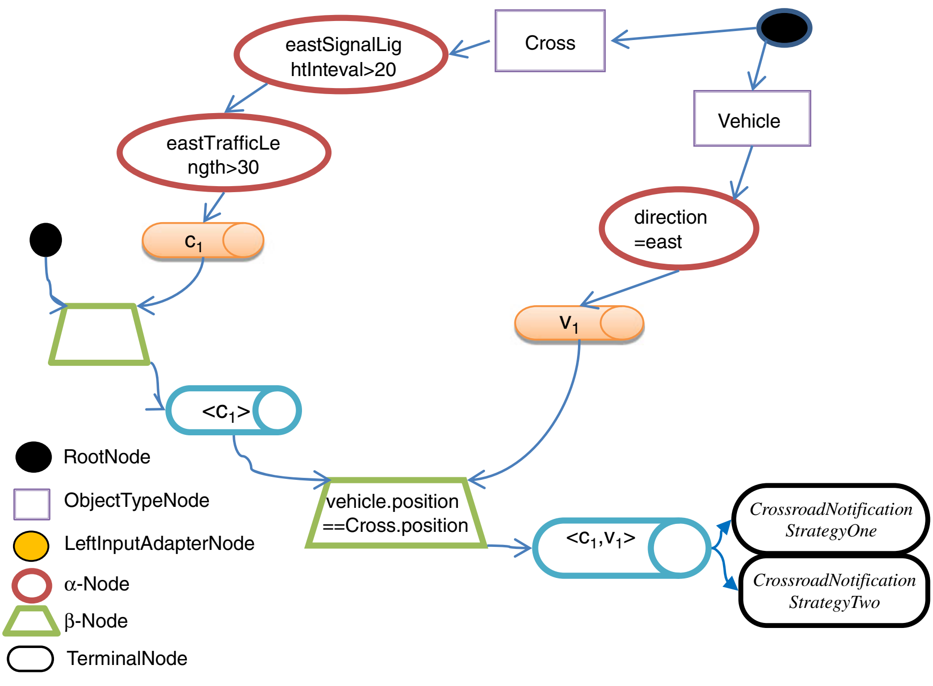


**Figure 6** Example for the heuristic Rete reasoning algorithm.

### *4.2.2 Backward planning based on a hierarchical task network (HTN)*

Traditional AI planning techniques require a huge amount of computational costs in terms of time and space if the knowledge representation is unlimited. The plan algorithms can be very slow when applied to large-scale and complex problems [5]. Hierarchical Task Network Planning (HTN) proposed by Erol specifies the constraints and search paths for task decompositions in order to realize efficient planning [9].

HTN contains three main language factors: Axiom, Primitive Task and Compound Task. The Axiom refers to logical expressions which can be judged true or false; a Primitive Task corresponds to a leaf node of a goal refining tree; a Compound Task is the composition of a primitive task or compound task. Therefore, HTN can build up a complex task network with primitive tasks and compound tasks. In the backward planning, HTN decomposes the goal tasks under certain constraints in the task network, depending on the goal state and initial state, until the goal tasks are decomposed to sequences of primitive tasks. Figure 7 demonstrates the specific process of the HTN-based backward planning.

In our implementations, a compound task is corresponding to a Strategy mentioned in the Adaptive Requirement Description Language, while a primitive task is associated with a Tactic. As sown in Figure 4, every tactic will refer to one or more operators in the hyper world model, which define the operations the wisdom services are able to conduct. As mentioned before, the procedure of solving an HTN planning problem is actually task decomposition, which means recursively decomposing each task in the goal tasks.

The process shown in the Figure 7 can be easily descripted as follows: the planner selects the first task in the goal tasks, and then find it by name in the task networks, If it corresponds to a primitive task, and the precondition of the primitive task satisfies in current state, then the related actions i.e. web services will be conducted. Otherwise, if it is a compound task, then replace it with its subtasks in the task network, as long as the precondition of the compound task is satisfied. However, more than one sequence of subtasks may refer to one compound task, which means there is more than one way to decompose a compound task. A simple method is to conduct the first subtask by the order of its definition. Since the greedy strategy will not always product optimal results, we implement this by assigning a weight value to each subtask, and the planner will choose to conduct the most weighted one. The HTN Planning algorithm is demonstrated as follows:

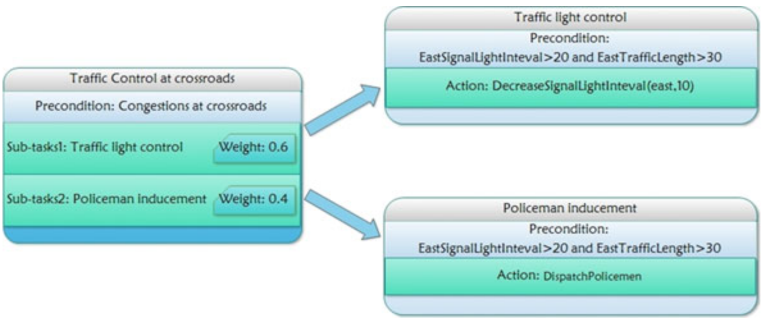


**Figure 7** Task networks for traffic control at crossroads.

Algorithm 2. The HTN planning algorithm

```
input state, goal tasks, task network
for all goal tasks do
        get the first task t in the goal tasks
        find task t' (t' in task network && t'.name == t.name)
        if t' is primitive
              if t'.precondition is satisfied in current state
                      conduct related actions
              endif
        endif
        if t' is compound
              if t'.precondition is satisfied in current state
                        replace task with highest weighted sub-tasks
                        do HTN Planning with new input: state, goal tasks, task network
              endif
        endif
endfor
```

We hope to generate an optimal result by HTN planning, so that the weight value of each sequence will be updated every time after an intelligent decision is made. This part will be presented in the next section of Learning Classifier System based Knowledge-Base Evolution. The decomposition of compound tasks will be recursively carrying out, until every task in the goal tasks is primitive. Then, the sequence of primitive tasks that comes out to be the result of the intelligent decision, will finally take effect on the hyper world.

In our implementation, we call a compound task as strategy and a primitive task as tactic. Thus, the task network is made up of strategies and tactics. Table 2 gives a simple example of the task networks. There are three tasks in the network: the first one is a compound task, and the last two are primitive tasks. The goal of the compound task is traffic control during rush hours, and there are two ways to decompose it, or achieve this goal: traffic light control and policeman inducement, and static weights correspond to the two sub-tasks. Figure 7 gives an explicit representation of the task network as noted above.

**Table 2** Strategies file for the task networks.

*CongestionInducementStrategy*:
 Name: Traffic Control at crossroads
 Precondition: Congestions at crossroads
 *Branch: DecreaseSignalLightIntervalTactic*
  Weight: 0.6
 *Branch: DispatchPolicemanTactic*
  Weight: 0.4
*DecreaseSignalLightIntervalTactic*:
 Name: Traffic light control
 Precondition: EastSignalLightInteval>20 and EastTrafficLength>30
 Action: DecreaseSignalLightInteval(east,10)
*DispatchPolicemanTactic:*
 Name: Policeman inducement
 Precondition: EastSignalLightInteval>20 and EastTrafficLength>30
 Action: DispatchPolicemen

### 4.3 Knowledge base evolution based on learning classifier system

The knowledge base of W2T was defined by domain experts initially. However, because of the limitations of the experts' knowledge and the growth/change of the hyper world, the knowledge base must evolve in order to satisfy the requirements of wisdom services. The self-adaptive support framework for W2T proposed in this paper employs a Learning Classifier System [LCS] as the knowledge base evolution mechanism. LCS integrates intelligence techniques (i.e., genetic algorithms and reinforcement learning) by acquiring the utility evaluations of the context of the hyper world. The reinforcement learning mechanism can be used to adjust the rule strengths of the rules in the knowledge base, while the genetic algorithms mechanism can be used to improve the structure and parameters of the rules using crossover and mutation. Figure 8 illustrates the specific process of knowledge base evolution based on the learning classification system [3, 8].

As mentioned before, we have realized the update of weight values of the subtasks in HTN based planning. The algorithm is presented as follows:

Algorithm 3. The learning classify system algorithm

---

Suppose the strategy *DecreaseSignalLightIntervalTactic* was carried out;
% Weight update process.
Get the updated properties;
Evaluate the utility value of the whole system according the SYSTEMUTILIT file;
Calculate the impact of the current strategy (the delta of the utility value);
Update weight value of the strategy *DecreaseSignalLightIntervalTactic* based on the impact;
% Structure update process
Generate a novel rule based on *DecreaseSignalLightIntervalTactic*, the novel rule random adjust the parameters values in the precondition and action using the genetic algorithms mechanism.

---

## 5 Case study: a wisdom traffic system (WTS) based on W2T

As an indispensable part of human lives, the traffic control is one essential ingredient in modern cities. From the W2T point of view, traditional traffic control systems cannot

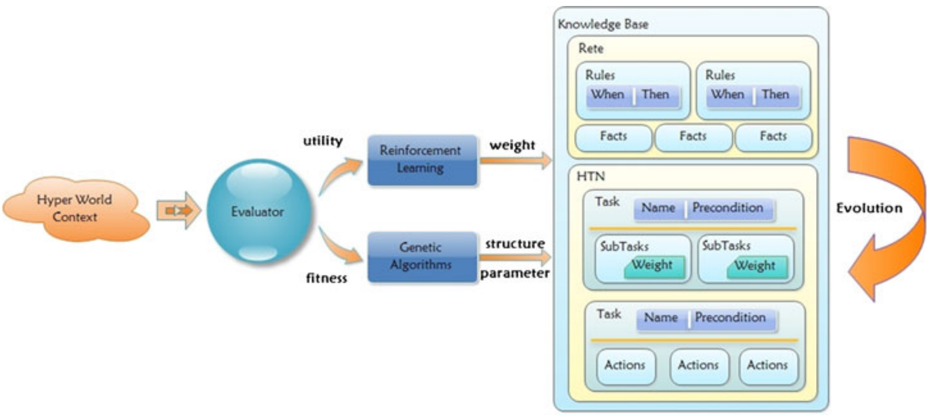


**Figure 8** knowledge base evolution based on a learning classifier system.

maintain the harmonious unity of humans, information and things. However, a W2T-based wisdom traffic system can illustrate the harmonious symbiosis of the physical world, the cyber world, and the social world. The physical world consists of traffic light controllers, traffic sensors, cameras, vehicle equipment, handheld devices, and so on. These physical devices constitute SEA-nets and implement the data conversion of things to data. The cyber world mainly contains knowledge and wisdom, which are used for handling various traffic situations. After traffic information is processed and decisions are made, the social and physical worlds respond correspondingly.

Figure 9 shows (i) the relationships between the related parties in the W2T-based Wisdom Traffic System (WTS), (ii) the dataflow process, and (iii) the self-adaptive support framework. In the leftmost column, the human icon represents the social world, the SEA-nets indicates the physical world and the remaining are main components from the cyber world. The rounded rectangles in the middle column briefly describe the corresponding components in the leftmost column in the WTS. As shown in Figure 9, the proposed self-adaptive support framework resides in the core of WTS. The framework comprises four modules including Knowledge Base, Monitor, Intelligent Decision and Learning. The Knowledge Base integrates traffic related human knowledge and wisdom to represent human expertise. The Monitor acts as the periphery interaction interface between the framework and its surroundings and it periodically acquires and updates real traffic data and information. Within the framework two operating mechanisms are supported. One is the Intelligent Decision, which refers to the framework generates proper decisions to respond to the current runtime context. The other is Learning, which means to update and adjust the strategies in the Knowledge Base according to systematic utility evaluations after requests are served. In short, the support framework is responsible for the whole data cycle which contains data conversion from data to information, information to knowledge, knowledge to wisdom and wisdom to services. In operation the adaptive framework firstly update data in the real time via SEA-nets. Then, it employs the supported mechanisms in terms of services to enable adaptation.

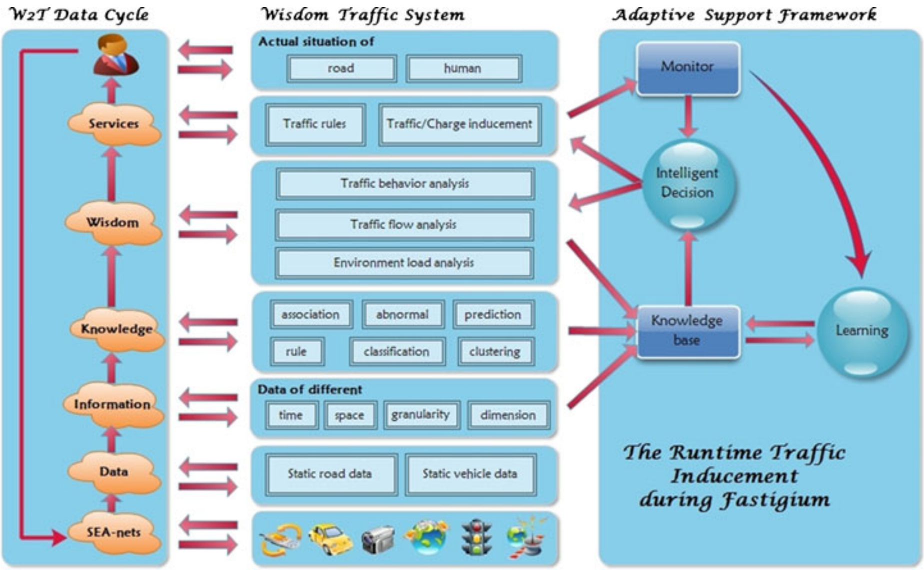


**Figure 9** The data transform process and the self-adaptive support framework in WTS.

After the support framework makes a decision, the SEA-net takes responsive actions and the physical world makes corresponding changes. Thus, humans, information/computers, and things interrelated and mutually interact via the framework.

Consider the scenario of the runtime traffic inducement during fastigium. For simplicity here a crossroad during rush hours is used to exemplify how the support framework works to relieve congestions. The Knowledge Base must be constructed as the first step and it requires users to define the seven XML files mentioned in Section 4.1. Also as shown in Figure 9, FAMILY, entities and properties, is extracted from the Data and the Information. INSTANCE is its dynamic instantiation in the real time. OPERATORS are implemented via SEA-nets like radios and traffic signal lights. The PROBES and GAUGES are used to fill the values in the INSTANCE by relevant operators. SYSTEMUTILITY specifies how to calculate the utility value of the whole system, generally by the defined arithmetic methods. Finally, human expertise is integrated and represented in terms of the low level TACTICS and high level STRATEGIES. The human expertise includes Data, Information, Knowledge (prediction, rule etc.) and Wisdom (several analysis skills).

After the Knowledge Base is defined by users and represents human expertise, the adaptive framework starts to work. Periodically the Monitor obtains the traffic information in the real time via SEA-nets and updates all values and properties in the instance. Then in the Intelligent Decision all conditions from all state-oriented strategies are being detected. Whenever there is any applicable condition, the action block will be executed. When multiple applicable conditions are detected, the strategy with a higher preference value will be chosen to fire. If any explicit goals are stated in the request, the goal-oriented strategy will be triggered. The Intelligent Decision chooses the one with the highest preference value from its applicable alternatives. When one specific strategy is active, it will be carried out via SEA-nets, and consequently human behaviors will be influenced. After every strategy is fired, the Monitor updates the instance again and the Learning evaluates the utility value of the whole system and based on the positive or negative feedback it updates and adjusts the preference value of the chosen strategy. The knowledge base evolves. The whole process keeps working repeatedly. Thus, via the proposed framework, the things, the human and computers are interconnected with each other.

Table 3 shows several simple and simulative cases, which are selected to illustrate how the proposed framework works, including knowledge and some supported functionalities. Knowledge is represented for these cases. Note that the complete XML-version knowledge files are not given here due to the limited space of this paper. The core requisite elements, tactics and strategies are listed in the Table 3 below. All these tactics and strategies and their meanings have been explained in the previous sections. The former two tactics and the former two strategies are from Section 4.2.1. The latter two tactics and the latter one strategy are from Section 4.2.2.

**Table 3** Knowledge and some supported functions in the WTS.

| Core elements in knowledge | Core contents | Initial weight | Modified weight |
|---|---|---|---|
| Tactics | InformChangeDirectionTactic | 0.3 | 0.4 |
| | DecreaseSignalLightIntervalTactic | 0.6 | 0.5 |
| | DispatchPolicemanTactic | 0.4 | 0.5 |
| Strategies | CrossroadNotificationStrategyOne | | |
| | CrossroadNotificationStrategyTwo | | |
| | CongestionInducementStrategy | | |

WTS monitors the traffic information in a crossroad at all times. If the information shows that the number of waiting cars at the crossroad is less than 30, for example, the forward reasoning based on Rete will still be invoked by WTS. However, since the precondition is not matched, the system does not generate any new strategy. If the information shows that the interval of traffic light is greater than 20, and the number of waiting cars is greater than 30, the forward reasoning based on Rete invoked by WTS will generate two appropriate Tactics (as shown in Section 4.2.1). Based on the initial weights of these two Tactics, DecreaseSignalLightIntervalTactic will be trigged.

When a congestion is recognized by WTS (note that the congestion's condition is defined in advance), since CongestionInducementStrategy has been defined in the knowledge base of WTS, the backward planning based on HTN will be invoked by the system. Then, since CongestionInducementStrategy has two branches, according to the weights of these two branches, DecreaseSignalLightIntervalTactic will be trigged first.

In the meantime, WTS continuously collects the traffic information in the crossroad. If the reduction of the number of waiting cars is not detected, the weights of the three Tactics in the knowledge base will be adjusted by the learning mechanism based on LCS. For example, as shown in Table 3, the weight of DecreaseSignalLightIntervalTactic may be reduced, and the weights of other two Tactics may be increased. Then, the intelligent decision mechanism will produce a higher probability for DispatchPolicemanTactic. Similar to this, when DispatchPolicemanTactic cannot bring a positive feedback to the system, the weight of InformChangeDirectionTactic will be increased. Then, after the forward reasoning, WTS will inform the waiting cars to change their directions.

In summary, as the example demonstrates, different types of the components in W2T are organized by the self-adaptive support framework to realize the harmonious symbiosis among humans in the social world, information/computers in the hyper world, and things in the physical world.

## 6 Conclusions

The hyper world consists of the physical world, cyber world and social world. W2T provides ubiquitous and wisdom services in the hyper world based on the ubiquitous networks. The core of W2T is to adapt to the physical world and provide intelligent services. Intelligent services can adapt to the changing requirements of human or organizations, as well as the changing data, computing resources, and security requirements.

In order to achieve the harmonious unity of human-information-thing, we proposed the self-adaptive support framework for W2T, which has three important components: (i) An adaptive requirement description language. The language includes eight language components to describe the wisdom service models and the wisdom service self-adaptive strategies. (ii) Forward reasoning and backward planning capability. We propose that forward reasoning can be based on the Rete algorithm and backward planning can be based on a Hierarchical Task Network (HTN). With the changing of requirements and the context of the hyper world, the self-adaptive support framework selects the right data, right intelligent functions, and right computing resources to provide active, transparent, safe, and reliable services. (iii) A knowledge base evolution mechanism based on the Learning Classifier System (LCS). LCS takes advantages of the reinforcement learning mechanism to adjust the rule strengths of the rules, and takes advantages of the genetic algorithms mechanism to improve the structure and parameters of the rules according to the fitness of the rules. Therefore, the mechanism realizes the evolution of knowledge base, and satisfies the dynamic requirements of wisdom services.

Note that, the new generation of W2T provides us with many challenges with the ambitious goal to achieve harmonious intelligence. In addition to conducting further research in the wisdom service model description language, the reasoning and planning techniques, and the learning algorithms, there are many research issues that are needed to be addressed such as (i) the evaluation model for the components of wisdom services; (ii) intelligent algorithms and parallelization on massive data; (iii) a characteristic description model for massive data; (iv) data management technologies driven by wisdom service requirements; (v) the methodology of designing the wisdom service components driven by data characters.

**Acknowledgments** We would like to acknowledge the support for this work from the National Science Foundation of China (Grant Nos.61035003, 61175042, 61021062), the National 973 Program of China (Grant No. 2009CB320702), the 973 Program of Jiangsu, China (Grant No. BK2011005) and Program for New Century Excellent Talents in University (Grant No. NCET-10-0476).

## References

1. Bertino, E., Ferrari, E., Perego, A.: A general framework for web content filtering. World Wide Web **13** (3), 215–249. Springer, Heidelberg (2010)
2. Bruno, E., Faessel, N., Glotin, H., Le Maitre, J., Scholl, M.: Indexing and querying segmented web pages: the BlockWeb Model. World Wide Web **14**(5–6), 623–649. Springer, Heidelberg (2011)
3. Bull, L., Studley, M., Bagnall, A., Whittley: Learning classifier system ensembles with rule-sharing. IEEE Trans. Evol. Comput. **11**(4), 496–502 (2007)
4. Cheng, S.-W.: Rainbow: Cost-effective software architecture-based self-adaptation. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, May 2008. Technical Report CMU-ISR-08-113
5. Erol, K.: Hierarchical Task Network Planning: Formalization, Analysis, and Implementation. Ph.D. Thesis. University of Maryland, College Park (1995)
6. Forgy, C.: Rete: a fast algorithm for the many pattern/many object pattern match problem. Artif Intell. **19**:17–37 (1982)
7. Holland, J.H.: Studying complex adaptive systems. J. Syst. Sci. Complex. **19**, 1–8 (2006)
8. Huang, H., Liu, C., Zhou, X.: Approximating query answering on RDF databases. World Wide Web **15** (1), 89–114. Springer, Heidelberg (2012)
9. Lekavý, M., Návrat, P.: Expressivity of STRIPS-Like and HTN-Like Planning. In Agent and Multi-Agent Systems: technologies and applications, Proc. of 1st KES Int. Symp. KES-AMSTA 2007, pp. 121–130 (2007)
10. Li, J.-Q., Zhao, Y., Garcia-Molina, H.: A path-based approach for web page retrieval. World Wide Web **15** (3), 257–283. Springer, Heidelberg (2012)
11. Liu, J.: Web Intelligence (WI): What makes wisdom web?. In Proceedings of the 18th International Joint Conference on Artificial Intelligence (Aca-pulco, Mexico, Aug. 9–15). Morgan Kaufmann, San Francisco, 1596–1601 (2003)
12. Ma, J.: Smart u-things-challenging real world complexity. IPSJ Symp. Series **2005**(19), 146–150 (2005)
13. Ma, J.: Active smart u-things and cyber individuals. active media technology. Lect. Notes Comput. Sci. **6335**, 5 (2005)
14. Ma, J.: Smart u-things and ubiquitous intelligence. In: Proc the 2nd international conference on embedded software and systems (ICESS 2005), p. 776
15. Salehie, M., Tahvildari, L.: Self-adaptive software: landscape and research challenges. ACM Trans. Autonom. Adapt. Syst. **4**(2), 1–42 (2009)
16. Sensoy, M., Yolum, P.: Automating user reviews using ontologies: an agent-based approach. World Wide Web **15**(3), 285–323. Springer, Heidelberg (2012)
17. Sottara, D., Mello, P., Proctor, M.: A configurable Rete-OO engine for reasoning with different types of imperfect information. IEEE Trans. Knowl. Data Eng. **22**(11), 1535–1548 (2010)
18. Squicciarini, A.C., Sundareswaran, S.: Web-Traveler policies for images on social networks. World Wide Web **12**(4), 461–484. Springer, Heidelberg (2009)
19. Tao, X., Li, Y., Zhong, N.: A personalized ontology model for web information gathering. IEEE Trans. Knowl. Data Eng. **23**(4), 496–511 (2011)

20. WI-IAT 2011 Panel on Wisdom Web of Things (W2T): Fundamental issues, challenges and potential applications. Available at: http://wi-iat-2011.org/
21. Wu, X., Ngo, C.-W., Zhu, Y.-M., Peng, Q.: Boosting web video categorization with contextual information from social web. World Wide Web **15**(2), 197–212. Springer, Heidelberg (2012)
22. Yang, Y., Huang, Z., Shen, H.T., Zhou, X.: Mining multi-tag association for image tagging. World Wide Web **14**(2), 133–156. Springer, Heidelberg (2011)
23. Yang, J., Pui, G., Fung, C., Lu, W., Zhou, X., Chen, H., Du, X.: Finding superior skyline points for multidimensional recommendation applications. World Wide Web **15**(1), 33–60. Springer, Heidelberg (2012)
24. Yao, Y.Y., Zhong, N., Liu, J., Ohsuga, S.: Web Intelligence (WI) research challenges and trends in the new information age. In: Zhong, N., Yao, Y.Y., Liu, J., Ohsuga, S. (eds.) Web intelligence: research and development. LNAI, 2198, pp. 1–17 Springer, Berlin (2001)
25. Zhong, N.: Impending Brain Informatics research from Web intelligence perspective. Int. J. Inf. Technol. Decis. Mak. **5**(4), 713–727 (2006)
26. Zhong, N.: Ways to develop human-level web intelligence: a brain informatics perspective. In: Franconi, E., Kifer, M., May, W. (eds.) The Semantic Web: Research and Applications. LNCS, 4519, pp. 27–36. Springer, Heidelberg (2007)
27. Zhong, N., Liu, J., Yao, Y., Wu, J., Lu, S., Qin, Y., Li, K. and Wah, B.: Web intelligence meets brain informatics. State-of-the-art-survey, LNCS 4845, pp. 223–243. Springer, Berlin (2007)
28. Zhong, N., Liu, J., Yao, Y.: In search of the wisdom web. IEEE Comput. **35**(11), 27–31 (2002)
29. Zhong, N., Liu, J., Yao, Y.Y.: Envisioning intelligent information technologies through the prism of web intelligence. Commun. ACM **50**(3), 89–94 (2007)
30. Zhong, N., Liu, J., Yao, Y.Y., Ohsuga, S.: Web Intelligence (WI). In: Proc the 24th IEEE computer society international computer software and applications conference (COMPSAC 2000), pp 469–470
31. Zhong, N., Ma, J.H., Huang, R.H., Liu, J.M., Yao, Y.Y., Zhang, Y.X. and Chen, J.H.: Research challenges and perspectives on Wisdom Web of Things (W2T). J. Supercomput., 1–21 (2010)
32. Zhong, N.: Building a Brain-informatics portal on the Wisdom Web with a multi-layer grid: a new challenge for web intelligence research. In: Torra, V. et al. (eds.) Modeling decisions for artificial intelligence. LNAI 3558, pp. 24–35. Springer, Berlin (2005)