

A Method of Public Policy Refinement Based on OWL and Linear Temporal Logic

D.P. Lang, S.B. Huang, L.S. Shen, T. Zhang, H. Chen
 Computer Science and Technology
 Harbin Engineering University
 Harbin City, Heilongjiang Province, China

Abstract—It's the consistency, safety and sustainability of public policy that to some extent determines the effectiveness of policy implementation. This paper focuses on whether we can use formal verification methods to verify the policy implementation consistent with their objectives. Based on a number of policy refinement methods, this paper uses ontology language OWL to break the objectives of public policy into several objects which can be represented by linear temporal logic in the second step for their logical reasons. At last model checking technology is applied to an example to verify the entire process. Then we give the performance analysis of the method and prospects.

Keywords-public policy; refinement; OWL; LTL; model checking

I. INTRODUCTION

A. Public Policy and Policy Analysis

Policy science is an interdisciplinary and comprehensive field of research after World War II in western countries, which was taken as a science revolution in the development of western social science. Late 1970s, along with the historical process of reform and opening up, policy analysis attracted scholars and policy researchers' attention in China.

Policy analysis, also known as policy science [3] is a policy research, formulation, analysis, screening, implementation and evaluation of the whole process of research methods. Back in the 1940s, the American political scientist Harold Lasswell has proposed a *policy science* concept [4]. Then a Israeli scholar Yehezkel Dror wrote the famous trilogy of science policy: The re-examination of public policy, Policy Sciences Vision, Policy Sciences exploration, which was considered the second milestone of policy science.

Policy analysis being developed into an independent social science, deductive reasoning, experience, analysis and other artificial means to determine the principles are still the main methods of policy analysis. As the rapid development of computer and information technology, formalization and formal verification came in to the map.

B. Model Checking Technology

Model checking was proposed by Clarke Emerson [5], Quielle and Sifaki [6] for automatic verification of finite state concurrent systems. Model checking has been successfully applied to computer hardware, communication protocols,

control systems, and other aspects of security authentication protocol analysis and verification. Current research shifts to the need for verifying more complex software system. Purposes of software model checking is to extend the field of application of model checking techniques, which applies its reasoning to the program to maximum the automation of verification process.

II. MOTIVATIONS

A. General Idea of Policy Refinement

Policies can be defined on different levels, ranging from high-level business objectives to a single resource policy. Hopefully we can directly or indirectly gain the system policy that is lower policy from the overall goals that are high-level policy. There are also many progresses in this area. With the increasing complexity of the network, the management of related systems becomes increasingly difficult. Managing such a complex and distributed systems faces a series of challenges. Among these challenges, one is how to ensure the operation of the system is consistent to a given overall objectives. In order to solve this problem have been proposed policy-based management system [7].

In a policy-based management system, the expected behaviors are defined in the policy which can be defined as a clear goal or guide of current or future action method [8]. In other words a policy is a set of rules that are used to describe how to achieve an expected behavior. This can be accomplished by policy refinement. The process is to map a higher-level policy to the lower-level policy goals.

B. Use of Model Checking

The essence of policy refinement is to give an accurate format description of the policy through mathematical methods, procedures and methods, formal methods, What we do is to facilitate following verification and implementation. This paper borrows the definitions of policy refinement in the references [9] by Moffett and Sloman. Then we clarify the process of refinement and objectives:

(1) Clearly state all the resources needed in public policy analysis

(2) During the process of transmitting the high-level policy into the low-level policies, the meta policy that composes of low-level policy and can be used to verify properties should be explicitly stated.

(3) Verify whether the transformed low-rise policies consistent with high-level policy, in other words we have to verify the properties and specification that the low-level policies satisfy consistent with the ones that high-level policy satisfies.

The concept ontology language OWL mentioned in the second step will be used to describe the meta policy and given in the following section; this paper uses linear temporal logic (LTL, given in Section III) to show the logical relationship among meta policies, namely the lower logical derivation relations in the policies.

III. PRELIMINARIES

A. Policy Refinement

Reference [1] uses formal techniques from KAOS to achieve target refinement, each of which represents a temporal logic rules. And the paper used the refined model to transform the original goals into logical included sub-goals. A set of refined goals is generated after the process. Key step in policy refinement process is to describe the high-level policy objectives using low-level goals which can be achieved by systems. In the reference [5], goal elaboration is achieved by using temporal logic and related evidence to detect the problems that may arise during the process of refinement.

Javier in the paper [9] proposed a policy refinement method based on goal-oriented requirements engineering and module testing technology. The paper derived low-level policy containing logically high-level management objectives through KAOS goal refinement methods.

B. Semantics of LTL

Linear temporal logic combines propositional logic with timing operations. The analysis method based on LTL is an important formal analysis method that performs validation and description of software architecture characteristics. It takes a path (sequence of states) as an assertion object, whose true value will be confirmed in the state sequence. Linear temporal logic can be easily and accurately used to describe important properties of concurrent systems, such as safety and liveness. Safety is about bad things will never happen; Liveness is about good things will eventually happen. There are syntax and semantics of linear temporal logic from Pnueli's work [8] below.

Definition 1: Grammar of LTL can be recursively defined as follows:

1) Atomic propositional variables P, Q, R, \dots are linear temporal logic formulas.

2) If P, Q are linear temporal logic formulas, then the formulas built by Boolean operators such as \vee (or), \wedge (and), \neg (not) or timing operator \cup (until), \diamond (finally), \square (always), X (next) are also temporal logic formulas like $P \vee Q, P \wedge Q, \neg Q, P \cup Q, P \diamond Q, P \square Q$ and PXQ .

IV. AN EXAMPLE OF POLICY REFINEMENT

A. Building the Prototype

Policy refinement has long been considered to be a daunting task. One of the reasons is that the process requires a lot of manual intervention. With the popularity of ontology representation language OWL, people consider the use of the ontology to describe policies and regulations, such as the use OWL to formalize the policy.

In this section we are going to give an example to illustrate the policy refinement method. The example only refine the constraint conditions of the policy and similar to the rest of the policy refinement. We want to define such a policy: When employees retire, if the payment period is longer than 15 years he has the right to enjoy the pension; otherwise he has to fill the difference amounts. After finish the process of policy refinement, the relevant information rises (payment and certain conditions of enjoying the treatment).

There are 10 concepts in this simple ontology and 1 data structure that are shown in the figure 1.

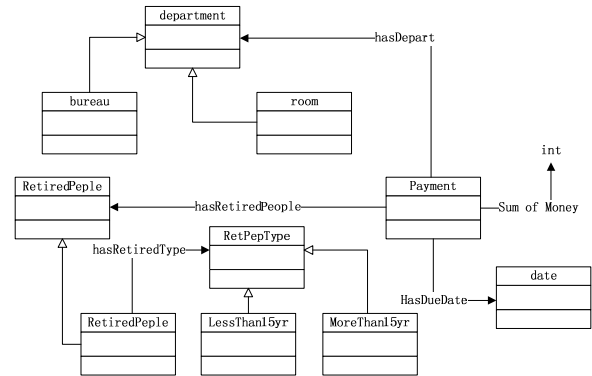


FIGURE I. PAYMENT ONTOLOGY DIAGRAM.

Meanwhile, in this public policy, the following information is also included: the policy subject, the policy object, the policy trigger conditions, the policy constrains and actions. Figure 2 shows the policy ontology diagram.

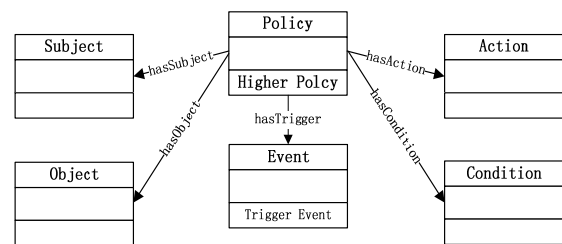


FIGURE II. POLICY ONTOLOGY DIAGRAM.

B. Establishing LTL Formulas

Based on the analysis above, this paper will break down the policy concepts of atomic propositions expressed as follows:

Let P be the meaning that the payment period is greater than or equal to 15 years; then the payment period is less than 15 years would be $\neg P$;

R represents paying the pensions and enjoying pensions is expressed by Q .

So, we want to establish an assertion that if a retired person wants to enjoy the pension, his payment period must be longer than 15 years or he must make a supplementary payment before he enjoys the pension.

$$P \Box Q \vee \neg P \wedge R \Diamond Q \vee \neg P \wedge \neg R \Box \neg Q \quad (1)$$

On the last step, we run SPIN as a model checking tool to verify the results in line with expectations. To sum up the policy refinement steps are as follows:

- Goal refinement

The original target should be decomposed into OWL ontology language to describe policy.

- Assign responsibilities to the managed entity

This will assign the refined sub-goals to the appropriate execution module.

- Target implementation

Make sure that the managed entity completes the relevant action.

- Policy encoding

Encode the policy formation expressed in OWL logical language with LTL language

- Model check the LTL formula

Run the SPIN model checker to verify the consistency of formulae with specifications[10].

C. Performance Analysis

Assuming the number of concepts involved in the ontology is m , the concept of number involved in the policy refinement process is n and apparently $m \geq n$. And furthermore the greater m is, the more complex the ontology is, which means we need more time to analyze the ontology. In the policy refinement process, we set up a concept decomposition tree synchronously and the number of nodes of the tree is multiple times of n so the time complexity is $O(n)$.

At the meantime from the aspects of semantic correlation between the concepts, if there are too many levels the semantic relevancy between the low-level constraint set and the original high-level constraint may go relatively lower. This may affect the consistency between refined results and the original target.

V. PROTOTYPE OF THE METHOD

To simulate the whole process of refinement, we need to borrow the definition of agents to represent both sides of policy control. We run the simulation on the platform called JADE which is developed by Java. And based on Protégé OWL API, which we take as semantic reasoning tools to control and

analyze the object and subject and the relations between them. The software is demonstrated on Figure 3. And we can insert a new object by insert the necessary agent parameters shown on Figure 4.

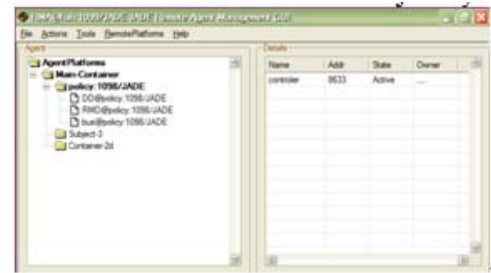


FIGURE III. REMOTE AGENT MANAGEMENT GUI.



FIGURE IV. INSERT AGENT DIAGRAM.

After we build the policy ontology diagram like what Figure 1 and Figure 2 show, what we have is several group of terms composed of OWL. Next part of this chapter is to demonstrate how to use SPIN verify the model based on OWL.

The basic idea of model checking is to input the specification (1) in the last chapter into the model checker, which is SPIN in this paper.

VI. CONCLUSIONS AND FUTURE WORK

Policy refinement plays a vital role in policy-based management system. This paper borrows the idea of policy refinement and applies the methods to the public policy refinement. Though there exists some policy refinement methods, on the one hand, most of them are confined to a specific application; on the other hand these methods have not been used in the area of public policy analysis. Besides the implementation process is increasingly complex, and the automation is still far from reaching. Whereas we intend to make some progress in the following aspects of work: During the process of policy analysis, we want to set the analysis model and domain knowledge apart, i.e. decoupling. 3. Explore a new language based on OWL-LITE and OWL-DL as a better and more powerful expression tool of public policy.

ACKNOWLEDGMENT

This work is sponsored by National Key Project of Scientific and Technical Supporting Programs under grant number 2012BAH08B02, the Fundamental Research Funds for the Central Universities under grant number HEUCFZ1212. Heilongjiang province degree and postgraduate research project

of education and teaching reform under grant number JGXM_HLJ_2012036.

REFERENCES

- [1] SLOMAN, M.S 1994. Policy Driven Management for Distributed Systems[J]. *Journal of Network and Systems Management* 2(4):333-360
- [2] Arosha K Bandara, A Formal Approach to Analysis and Refinement of Policies. Department of Computing Imperial College London University of London, London, July 2005,pp:14-55
- [3] Chen Zhenming. Policy Scienc[M]. Beijing: China Renmin University Press, 1998.
- [4] D.Lerner and H.D.LassWell, *The Policy Science: Recent Development in Scape and Method*, Stanford, CA:Stanford University Press, 1951.
- [5] Clarke E, Emerson E. Design and synthesis of synchronization skeletons using branching time temporal logic[J]. *Logics of Programs*, 1982,pp.52-71.
- [6] Queille J, Sifakis J. Specification and verification of concurrent systems in CESAR[A]. *International Symposium on Programming[C]*. 1982,pp.337-351.
- [7] Wies, R., Policies in Network and Systems Management - Formal Definition and Architecture, *Journal of Networks and Systems Management*, vol. 2, no. 1, pp.63-83, Mar.1994.
- [8] Alcantara, O. D. and Sloman, M. QoS Policy Specification A mapping from Ponder to the IETF. 2003.
- [9] J. Moffett and M. S. Sloman, Policy Hierarchies for Distributed Systems Management, *IEEE JSAC*, vol. 11, pp. 1404-14, 1993.
- [10] Dapeng Lang, Shaobin Huang, Yuan Cheng, Xinxin Yang and Ya Li, A State Space Abstract Algorithm of Incremental Data Recognition Based on Model Checking, *Journal of Computational Information Systems*, 10: 4 (2014) 1731-1742