Property Specification Patterns for intelligence building software

Seungsu Chun

1Research Fellow, Business Planning & Budget Division, Korea Institute of S&T Evaluation and Planning

Abstract. In this paper, through the property specification pattern research for Modal MU(μ) logical aspects present a single framework based on the pattern of intelligence building software. In this study, broken down by state property specification pattern classification of Dwyer (S) and action (A) and was subdivided into it again strong (A) and weaknesses (E). Through these means based on a hierarchical pattern classification of the property specification pattern analysis of logical aspects Mu(μ) was applied to the pattern classification of the examples used in the actual model checker. As a result, not only can a more accurate classification than the existing classification systems were easy to create and understand the attributes specified.

1. Introduction
Model checking is a method for logically verified whether the intelligence building software meets the desired properties [1]. To test the model, the user must express the property to be examined. This property is called the specification. There were tense used in the specification means a logical, automata, various expressions, including visual language, it is the most widely used among these temporal logic [2]. Because, temporal logic is because it is excellent in expressive power can specify a variety of attributes. However, the property has a temporal logic specification created by the questions that could be easily understood by users. Conversely, there is the question of exactly what we can attribute to specify a given temporal logic. For example consider the following attribute specification written in a logical aspects Mu(modal μ-calculus) [2].

\[ \forall Z. \text{[emergency]}(\mu X. \text{[alarm]}(\mu Y. \text{[alarm]}(\forall Z. \text{[alarm]}ff \land \neg Z) \leftrightarrow tt \land \neg \text{[alarm]}Y) \leftrightarrow tt \land \neg \text{[alarm]}X) \land \neg Z) \]

In order to interpret the above specifications, it is required to understand the meaning and symbols used in logic, and background knowledge. For a particularly aspect mu logic it is essential to understand the maximum and minimum fixed point fixed point. The meaning of the above specification is "continuously when an emergency situation occurs two times and the alarm must be sounded."

If you called a temporal logic CTL (Computation Tree Logic) and LTL (Linear Temporal Logic) advanced Specification Language, MU logical aspect is the same as assembly language. Therefore, it is not easy to write and interpretation of the specification. This study was research to help the creation of specifications and logical aspects Mu understand.

In the past, study or the like dedicated language specification [3], using the pattern [4] was to solve the problem related to the attribute specification. Survey of existing research results specified in the...
Dwyer pattern property was found to be the best in our research direction. However, the pattern of Dwyer applies only CTL, LTL, GIL (Graphical Interval Logic). In this study, the mapping to the aspect targeted μ logic did not address. In addition, CTL and LTL covers only path to each state and GIL, on the other hand because it is a modal logic to accommodate both the concept of the state and behavior not accept the logic that his classification system deals only with aspects Mu(μ) events.

So, this study was subdivided according to the classification system of modal logic Dwyer, was applied to the classification of the sample which was used in the model checker. The results were classified more accurately than existing classification systems. This suggests that it is helpful to understand the specification creation and use of the property specification pattern.

2. Pattern based property specification

In this study, it chose the following aspects μ logic for the same reason. First, it is the most highly expressive. Second, it is possible to express both the state and behavior. Third, CTL, LTL, CTL * can be converted into a logical aspects μ. That aspect μ logic is that to provide a single framework for the temporal logic.

Classification of property specification pattern for pattern μ logic is shown in figure 1. A large category are the Occurrence and Order, there are eight patterns of classification, as shown below.

Absence, Universality,
Bonded Existence, Existence, Response,
Precedence, Chain Response, Chain Precedence.

![Figure 1. Pattern’s classification.](image)

Modal μ logic is broken down into these state (S) and the action (A), because all of the possible representations of the state and behavior. If the property is satisfied in all the paths is referred to as a 'strong', and as, if there is a path that satisfies the attribute 'weak'. Thus, the state and action expression is subdivided into strong (A) and weak (E).

For example, let's look at the pattern indicating the safety Universality property. The meaning of this pattern is an "attribute is always satisfied." It is broken down according to the state and the behavior as follows:

The pattern is further divided as follows depending on whether satisfactory in all the properties of the path state or behavior-oriented in the satisfaction of one or more paths:

- State oriented(S): $\forall Z.\Phi \land [\neg] Z$
- Behavior oriented (A): $\forall Z.[K] ff \land [\neg] Z$

- State oriented strong(SA): $\forall Z.\Phi \land [\neg] Z$
- State oriented weakness(SE): $\forall Z.\Phi \land [\neg] Z$
- Behavior oriented Strong(AA): $\forall Z. [K] ff \land [-] Z$
- Behavior oriented Weakness(AE): $\forall Z. [K] tt \land \leftrightarrow Z$

The detail of such means based pattern classification enables the expression of the correct meaning. On the other hand, Universality pattern is expressed by a fixed point of maximum $= f Z (Z)$. Its meaning is 'always complete set has a value of true (True) $S$ is unfolded from the top down (Top-down) method from the first state. In releasing strong (AA) in the center of the path, repeat calculation as follows:

It means the following expression is ‘on all paths, an infinite series of movement by the event $K$ is available’. It corresponds to the stability properties of the system. In the case of the minimum fixed point, to represent the ultimate property for the system, and which is classified as Existence pattern. All attributes with the system is represented by logical aspects $Mu$ as shown in Table 1.

$$\forall Z. [K] ff \land [-] Z = true$$

$$\land [K] ff \land [-] true$$

$$\land [K] ff \land [-] ([K] ff \land [-] true)$$

...  

| Pattern type          | Num. |
|-----------------------|------|
| Absence               | 6    |
| Universally           | 16   |
| Existence             | 20   |
| Bounded Existence     | 1    |
| Response              | 6    |
| Precedence            | 0    |
| Response Chain        | 0    |
| Precedence Chain      | 0    |
| UNKNOWN               | 0    |
| Total                 | 43   |

### Table 1. Property specification pattern for Modal Mu Logic.

3 Pattern based specification system

When you check the properties of Fairness using the specification pattern is a required attribute ‘when you approach the car go across the car’. And the ‘crossing the car or train it should occur indefinitely’. If you specify the attributes Bounded Existence pattern can be expressed as follows:

$$\forall X. [car] (\mu Y. <-> tt \land [-cCross] Y) \land [-] X)$$

And, "If the car is gone (Q) should always be followed by (R) occurs passing train.” When you apply the pattern Precedence it can be specified as follows:

$$\mu Z. ([Q] ff \lor ([R] ff \land [-] Z \land \leftrightarrow tt))$$

Therefore, the specification equity properties with infinite recursion property of causality can be specified as follows:

$$\mu X. v Y 1. ([Q] ff \lor ([cCross] (v Y 2. (R \land X) \land [-cCross] Y 2)) \land [-cCross] Y 1)$$

The more complex attribute specifications, pattern-based approach is more efficient.

For example on CWB-NC (The concurrency Work Bench of the New Century), Let us apply the substantive requirements of the property specifications ‘railway ringer system’ [5]. The meaning of the
attributes ‘occurs sometime (Eventually) silent.’, If the structure is a complex system specifications are as follows:

\[
\text{prop} \quad \text{eventually\_silent} = \text{not ( min A1 = ( not ( max D = ( min E1 = ( not (} \text{not ( min F(not([comm\_out:2,stat\_out:2]ff)\(\text{\textless}\text{recovered:0}\text{F)})\(\text{\textless}\text{tick:4}\text{D} \lor \text{\textless}\text{tick:4}\text{E1 } ) ) ) ) ) ) ) ) ) ) ) ) } \lor <\text{A1 })
\]

If the specification of the above complicated expression patterns using the specification can be simply represented as a set relationship of the pattern is as follows.

**Existence.** ~Universally(~Response(~BoundedExistence~(~Existence ‘recovered:0 or (comm\_out:2 or ~statout:2)and tick:4-ff or tick:4))))

This study was conducted to analyze the physical property requirements of the specification and the CWB-NC ECW (Edinburgh Concurrency Workbench) The results are shown below.

1) Safety = { Absence, Universality }
   (1) Absence = { AG(!Φ), EG(!Φ), AG(!K), EG(!K) }
   - AG(!Φ) = SA : vZ.¬Φ∧¬[-Z]
   - EG(!Φ) = SE : vZ.¬Φ∧<-Z
   - AG(!K) = AA : vZ,[K]ff∧¬[-Z]
   - EG(!K) = AE : vZ,[K]ff∧<-Z

   (2) Universality = { AG(Φ), EG(Φ), AG(K), EG(K) }
   - AG(Φ) = SA : vZ.Φ∧[-Z]
   - EG(Φ) = SE : vZ.Φ∧<-Z
   - AG(K) = AA : vZ,[K]tt∧[-Z]
   - EG(K) = AE : vZ,[K]tt∧<-Z

2) Cyclic = { Universality(Response) }
   (1) Universality = { AG[Φ→AX(AF(ψ))], EG[Φ→EX(EF (ψ))]}, AG[K→AX(AF(L))], EG[K→EX(EF(L))]
   - AG[Φ→AX(AF(ψ))] = SA : vZ.Φ[¬](ψ ∨ <-tt ∧ [],Z) ∧ [-]Z
   - EG[Φ→EX(EF (ψ))] = SE : vZ.Φ(<¬tt ∧ <-Z)) ∧ <-Z
   - AG[K→AX(AF(L))] = AA : vZ,[K][¬](<¬tt ∧ [-L]Z) ∧ [-]Z
   - EG[K→EX(EF(L))] = AE : vZ,[K](<¬tt ∧ <-L> Z) ∧ <-Z

3) Linveness = { Existence }
   (1) Existence = { A F(Φ), E F(Φ), A F(K), E F(K) }
   - A F(Φ) = SA : µZ.Φ(¬)<tt∧[-]Z)
   - E F(Φ) = SE : µZ.Φ(<¬>Z
   - A F(K) AA : µZ.(<¬>tt∧<¬K)Z]
   - E F(K) AE : µZ.(<¬tt∧<¬K>)Z]

4) Counting = { Bounded Existence }
   (1) Bounded Existence = { AG F(Φ), EG F(Φ), AG F(K), EG F(K) }
   - AG F(Φ) = SA : vX.µY.¬Y∨(¬X∧Φ)
   - EG F(Φ) = SE : vX.µY.¬Yψ(¬X∧Φ)
   - AG F(K) = AA : vX.µY.[¬]<¬tt∧<¬K]Y)∨[-]X
   - EG F(K) = AE : vX.µY.¬<¬tt∧<¬K>Y) ∨ <-X

5) Until = { Until, Precedence, Chain Precedence }
(1) Until = \{ A[Φ U ψ], A[Φ W ψ], E[Φ U ψ], E[Φ W ψ], A[K U L], A[K W L], E[K U L], E[K W L] \}
- A[Φ U ψ] = SA : μZ.ψ∀(Φ ∧ <−−> tt ∧ −−> Z)
- A[Φ W ψ] = SE : μZ.ψ∀(Φ ∧ <−−> tt ∧ <−−> Z)
- E[Φ U ψ] = SE : νZ.ψ∀(Φ ∧ <−−> tt ∧ <−−> Z)
- E[Φ W ψ] = SE : νZ.ψ∀(Φ ∧ <−−> Z)
- A[K U L] = AA : μZ.¬(K∧L)[ff ∧ <−−> tt ∧ −−> L]Z
- A[K W L] = AE : νZ.¬(K∧L)[ff ∧ −−> L]Z
- E[K U L] = AA : μZ.¬(K∧L)[ff ∧ <−−> tt ∧ <−−> L]Z
- E[K W L] = AE : νZ.¬(K ∧ L)[ff ∧ <−−> L]Z

(2) Precedence = \{ A[!Φ W ψ], E[!Φ U ψ], A[!K W L], E[!K U L] \}
- A[!Φ W ψ] = SA : μZ.ψ∀(!Φ ∧ −−> Z)
- E[!Φ U ψ] = SE : νZ.ψ∀(!Φ ∧ <−−> tt ∧ −−> Z)
- A[!K W L] = AA : νZ.(−(K∧L))[ff ∧ tt ∧ −−> L] > Z
- E[!K U L] = AE : μZ.(−(K∧L))[ff ∧ tt ∧ <−−> L] > Z
- Chain Precedence = \{ !E[!Φ U(Ψ∧!Φ∧EX(Ε(Γ))), !E[!Φ W(Ψ∧!Φ∧EX(E F(Γ))), !E[!K U (L∧!K & EX (Ε(Γ))), !E[!K W (L∧!K∧EX(E F(Γ))) }
- !E[!Φ U(Ψ∧!Φ∧EX(Ε(Γ))] = SA : μZ.(ψ∧!P∧tt <−−> (Y∧tt <−−> Z)∧ ( ! Φ ∧ <−−> −−>) −−> Z)
- !E[!Φ W(Ψ∧!Φ∧EX(E F(Γ))] = SE : μZ.(ψ∧!P∧tt <−−> (Y∧tt <−−> Z) v ( ! Φ ∧ −−>) −−> Z)
- !E[!K U (L∧!K & EX (Ε(Γ))] = AA : μZ.(L∧!K)[tt <−−> tt∧tt <−−> J <−−> Z]v (<−−> tt∧−−>) −−> J Z)
- !E[!K W (L∧!K & EX(E F(Γ))] = AE : μZ.(L∧!K)[tt <−−> tt∧tt <−−> J <−−> Z]v (<−−> tt∧−−> K Z)

6) Fairness = \{ Response, Chain Response \}
(1) Response = \{ AG(Φ→AF(ψ)), EG(Φ→EF(ψ)), AG(K→ AF(L)), EG(K→EF(L)) \}
- AG(Φ→AF(ψ)) = SA : νX.Φ(μY.ψ∀(tt ∧ −−> Y)∧ −−> X)
- EG(Φ→EF(ψ)) = SE : νX.Φ(μY.ψ∀(tt ∧ −−> Y)∧ −−> X)
- AG(K→ AF(L)) = AA : νX.[K](μY. (tt ∧ −−> Y)∧ −−> X)
- EG(K→EF(L)) = AE : νX.[K](μY. (tt ∧ −−> Y)∧ −−> X)
(2) Chain Response = \{ AG(Φ→AF(Ψ∧AX(Ε(Γ))), EG(Φ→EF(Ψ∧EX(Ε(Γ))), AG(K→AF(L∧AX(AF(J))), EG(K→EF(L∧EX(EF(J)) \}
- AG(Φ→AF(Ψ∧AX(Ε(Γ))] = SA : νX.Φ(μY. (Ψ∧tt ∧ −−> Y)∧ −−> −−>) −−> X)
- EG(Φ→EF(Ψ∧EX(E(Γ))] = SE : νX.Φ(μY. (Ψ∧tt ∧ −−> Y)∧ −−> −−>) −−> X)
- AG(K→AF(L∧AX(AF(J))] = AA : νX.[K](μY. (−−> J)[ff ∧ −−> tt ∧ −−> Y)∧ −−> X)
- EG(K→EF(L∧EX(EF(J))] = AE : νX.[K](μY. (−−> J)[ff ∧ −−> Y)∧ −−> X)

Lamport is largely classified as safety and finality of property [6]. However, this classification is too comprehensive and may need to be broken down more. Mana has offered a little more detailed classification structure for linear temporal logic [7]. However, this classification is mainly on the syntax of temporal logic formulas. The syntax is more preferable than the classification means as the size of the expression go large. So Dewyzer is complementary to the two above classification presents a detailed and even semantic-based taxonomy. His classification CTL, LTL has been well applied aspects Mu inadequate in logic that we have to target. The Stirling was classified as greater safety and finality, fairness, circulation, recovery of property emergence of Modal Mu logic [8]. But we should classify the structure is covers all five properties. So it would help the creation and understanding aspects of Mu if you are using a logical classification structure presented.

4. Result and future works
This paper provides a single approach that encompasses all of the logical specification through the property specification pattern research for Modal MU logic. As a result, the Modal Mu made it possible to analyze and understand complex modal logical expressions and facilitate state and behavior were the center of the property and the analysis and representation of the more complex Modal Mu logic.
through the extension of the detailed specifications of the actual classification of the path to the center of. Also it enables the study of new and advanced specification language approach. Future research is as follows. First, we will proceed with the collection, classification, and statistical work of the actual specifications for ensuring the safety of the specification pattern. Second, prove the regularity between attributes, patterns, meaning, and expression patterns for the expansion of inventory and will apply it to a component based methodology. Third, to implement a pattern based specification of the property inventory system for the automatic Modal Mu logic.

Reference
[1] Clarke E M et al. 1999 Model Checking MIT Press.
[2] Emerson E A 1997 Model Checking and the Mu-Calculus Proceedings of the DIMACS Symposium on Descriptive Complexity and Finite Models, pp. 185—214.
[3] Corbett J C et al. 2000 A Language Framework For Expressing Checkable Properties of Dynamic Software Proceedings of the SPIN Software Model Checking Workshop, LNCS 1885.
[4] Dwyer M B et al. 1998 Property Specification Patterns for Finite-State Verification Proceedings of the Workshop on Formal Methods in Software Practice.
[5] Cleaveland R The Concurrency Workbench of the New Century http://www.cs.sunysb.edu/~cwb/
[6] Lamport L 1977 Proving the correctness of multiprocess Programs IEEE Transactions on Software Engineering, SE-3(2) 125-143.
[7] Manna Z et al. 1992 The Temporal Logic of Reactive and Concurrent Systems SpringerVerlag.
[8] Stirling C 1996 Modal and Temporal Logics for Processes Lecture Notes in Computer Science 1043 149-237.

Acknowledgment
This work was supported by the Korea Institute of S&T Evaluation and Planning grant funded by the Korean Government (No. KISTEP-2016-BB16010)