information

Article

Operational Rule Extraction and Construction Based on Task Scenario Analysis

Xinye Zhao *, Chao Wang, Peng Cui and Guangming Sun

Operational Software and Simulation Institute, Dalian Naval Academy, Dalian 116018, China; wangchaosuper@sina.com (C.W.); cuipeng1978@sina.com (P.C.); Sgm_nudt@163.com (G.S.)
* Correspondence: zhaoxinye@nudt.edu.cn

Abstract: Changes in the information age have induced the necessity for a more efficient and effective self-decision-making requirement. A method of extracting and constructing naval operations decision-making rules based on scenario analysis is proposed. The template specifications of Event Condition Action (ECA) rules are defined, and a consistency detection method of ECA rules based on SWRL is proposed. The logical relationships and state transitions of the naval operational process is analyzed in detail, and the association of objects, events, and behaviors is realized. Finally, the operation of the proposed methods is illustrated through an example process, showing the method can effectively solve the problems of self-decision-making rule extraction and construction among naval battlefield decision environment, and avoid relying on artificial intelligence, which may have brought some uncertain factors.

Keywords: rule extraction; rule construction; self-decision-making; intelligent decision-making

check for updates

Citation: Zhao, X.; Wang, C.; Cui, P.; Sun, G. Operational Rule Extraction and Construction Based on Task Scenario Analysis. *Information* **2022**, *13*, 144. https://doi.org/10.3390/info13030144

Academic Editor: João Manuel R. S. Tavares

Received: 6 January 2022
Accepted: 15 February 2022
Published: 9 March 2022

1. Introduction

The intelligent self-decision-making of naval operations is of great significance to the research of auxiliary decision-making for naval operations. Under the conditions of information-based joint operations, the naval battlefield situation is changing rapidly, from early warning and detection, command and control, information transmission to the implementation of confrontation, and the pace of confrontation is accelerating. In view of the characteristics of high requirements for decision timeliness in operational planning, how to effectively alleviate the time faced by operational decision generation and dynamic pressure has become a technical difficulty that the operational command system urgently needs to solve. In the aspect of decision making, rule-based decision-making methods were generally utilized in earlier research on operational support. Among them, the most typical rule-based decision-making method is finite state machine (Finite State Machine, FSM). Rule-based methods were widely applied in solving decision-making problems because of the clear logic and strong interpretability. Besides, rule-based decision-making methods rarely have the ability to deal with unseen scenarios [1].

Intelligent autonomous decision-making is mainly through mathematical optimization, artificial intelligence and other methods to build a mapping from naval operational situation to operational commands. According to the different ideas of solving the mapping, there are mainly: influence diagram [2–4], genetic algorithm [5,6], fuzzy logic [7,8], neural network [9] and other methods. In response to different scenarios, these methods have been used to solve the problem of autonomous decision-making, and many results have been achieved [10–12]. However, these methods are still plagued by problems such as "dimensionality disaster", "human subjective influence", and "rule loopholes". When fighters face unpredictable complex conditions, the uncertainty of decision-making increases. A large number of decision-making reasoning rules need to be extracted from data and processes with certain mutually independent attributes. Each scene of the naval surface vessel represents a series of scenarios constructed by the environment, objectives and naval surface vessel activities. The construction of the scene is the process of plot design based on the existing materials and domain knowledge based on the operational mission. For this reason, on the basis of mission scenario analysis, a method of extracting decision rules for naval surface vessel based on scenario analysis is proposed [13,14].

There are two things that must be discussed before analyzing the operational planning rules. The first is the extraction of domain terms in the operational planning field. The precise extraction of terms in the domain can assist analysts in understanding and analyzing operational planning sentences more accurately; on the other hand, it is the work of synonym extraction between terms. Synonym extraction can identify different descriptions of the same entity. Term extraction is a basic and important research direction in the field of text processing, which can be applied to many fields such as ontology construction, machine translation, semantic retrieval, and planning sentence engineering. However, terminology extraction is a complex and difficult task, and requires certain linguistic knowledge and a related field background. In this method, the data to be processed comes from a operational planning document with a strong domain. The planning sentences for word segmentation are different from the commonly used word segmentation database, and it is necessary to manually establish a term database to assist word segmentation. Therefore, it is of great significance to develop a set of automatic, efficient and highly portable term extraction and construction methods.

The research on automatic extraction of domain terminology has lasted for more than 20 years. In the 1990s, there were a number of terminology extraction systems abroad. Domestic research is concentrated in the past 10 years. The research on automatic extraction of Chinese terms also has its own particularities. The English corpus has natural advantages over the Chinese corpus. English words are separated by spaces, while in the Chinese corpus, there is no separator between words. Therefore, in the process of automatic extraction of Chinese terms, the corpus first needs to be segmented. However, it is difficult to carry out relevant research when the effect of early word segmentation is not very satisfactory. At present, the most commonly used word segmentation tools are CoreNLP [15], Jieba [16] and HanLP [17] of Stanford University, and support part-of-speech tagging. In the early work of automatic extraction of domain terms, most of them were based on linguistic knowledge. Later, with the rapid development of statistical natural language processing technology, one or more statistical strategies were gradually introduced into the term extraction system. With the use of machine learning algorithms such as Hidden Markov Model (HIMM) [18] and Conditional Random Field (CRF) [19] in the areas of part-of-speech tagging and named entity recognition, the method of combining machine learning algorithms also introduced into terminology extraction research.

Rule-based method mainly uses linguistic knowledge such as the word part of speech and morphological pattern of the term, which can be used to automatically extract terms from the corpus. Foreign research on automatic term extraction based on linguistic knowledge is mainly concentrated in the 1990s. The FASTER system developed by Jacquemin in 1994 first summarized the term composition rules from the known terminology database, and combined with 73 extended operating rules to extract terms from the corpus of the field of grammar medicine. The final accuracy rate was 86.7%, and the recall rate was 86.7%. 74.9%. The NODALIDA-95 launched by Linsoft in 1995 uses a structured method to organize language knowledge, with NPtool [20] as the core. It first judges the boundary of the sentence, the idioms and compound forms in the sentence, and then disambiguates it through part-of-speech analysis.

Based on statistical theory, the statistical-based method uses the statistical attributes of the distribution of terms in the corpus to identify terms. Frequently used statistical methods can be divided into two categories: one category measures the domain of words or phrases, such as word frequency (Frequency) [21,22], IF-IDF [23] value, domain relevance and domain consensus [24], etc.; another type of measurement of the unit of phrases, such as mutual information [25], Log-Likchood Ratio [26], etc. The ECA rule has a sound theoretical basis. Once a set of ECA rules required for process execution is prepared, we can take advantage of the theoretical basis. However, it is not easy to automate extracting and constructing the operational rules, thus, it is very difficult for users to manage and utilize the rules.

In this paper, aiming at the complexity of decision-making issues such as coordinated operations of naval formations or integrated defenses of the ship, combined with ECA rules, a method for extracting and constructing naval operational decision-making rules based on scenario analysis is proposed.

2. ECA Rules

An ECA model is originally used in active database systems. If an event occurs and a condition turns out to be true, then the active database executes a corresponding action. The event is a change of database contents, the condition is associated with a database query to check it, and the action corresponds to a set of statements that may trigger other changes in the database. All these are carried out automatically without any intervention from users or external applications [27].

2.1. Rule Design Principles and Requirements

There are three interpretations of the rules in "The Chinese Word Dictionary": one is standard; the other is neat and in a certain way; and the third is a written document formulated on one or some matters. Therefore, when designing the rules of marine surface vessel operations, it should also meet the requirements of its format specification, clear purpose, flexibility and compatibility:

(1) The state should be consistent before and after the rule is applied. When compiling the operational rules of the naval surface vessel, if there is any adjustment to the actions of the surface vessels, the status of the force should be saved in advance before the implementation of the rules. And after the implementation of the rules, it will resume the previously interrupted mission from the original status;

(2) The format and parameters of the command action must be clear. Different from military theory, all combat instructions that require staff officers to understand and execute in actual combat must be transformed into one or more simple actions in the operational rules. Although the rules themselves can be flexible and diverse, these actions must have a limited number, clear format, and unique parameters;

(3) The triggering conditions of the rules must be set reasonably. Naval surface vessel operations are no longer a linear structure formed by a series of operations such as crossing, search, tracking, or attack, but a network structure composed of multiple forces and multiple combat operations. The design rule trigger conditions should start from the change of the surface vessel status and find the perfect and unique representation method as possible;

(4) The structure of the rules must be discrete and concise. The naval surface vessel operational rules are not only applied to the command-and-control system, but also applicable to the operational simulation system. Therefore, the operational rules are bound to the execution object through the dynamic parameters input by the data interface after the simulation system is running, and there is no mutual interaction between the rules;

(5) The proper optimization and perfection of the set of operational rules must be adhered to. According to the development of military needs and the improvement of experimental experience, various rules will also be continuously expanded and improved. However, when adding simulation rules, correcting inappropriate rules, and merging duplicate or similar rules, the downward compatibility of the rule set should also be fully considered.

2.2. Event

Events in ECA rules can trigger a rule. In different applications, the classification of events is different. In order to achieve standardization, on the basis of analyzing the possible situations of general events, events are defined as two categories: (1) External events introduced from outside, or caused by external devices, entities, etc. (2) Time events that are timed or not. It is important to note that events can also be constructed by another rule.

2.3. Condition Template

In ECA rules, conditions are the states that must be met to trigger a rule. The expression of the condition can be simple true or false; it can also be the limited part of the condition clause in the database query language. These two condition expressions are called relationship conditions, and the template specification is defined as follows:

Definition 1. *Condition::=<Relation Condition>[AND<Relation Condition>]. Relation Condition::=<Expression><Relation Operator><Expression>. Expression::=<Variable>i<Attribute Value>. Relation Operator::="=" | "≠" | ">" | "<".*

In most scenarios, the conditions will not be simple "true" and "false", which requires analysis of the specific situation to find out the judgment conditions for the execution of the action.

2.4. Action Template Specification

Actions in ECA rules are actions triggered when the event that triggers the rule occurs and the conditions are met. Sometimes it confuses the concept of actions. It should be emphasized here that actions are for the system and not executed by external entities. Here, the action is defined as a method call, and its template specification is defined as follows [28]:

Definition 2. *Method Calls::=<Method Names>([Parameter]).*

Substitute a generated parameter into the template to execute the corresponding action. In practical applications, it is hoped to exchange information with sensors, mobile interrupts and other devices or external information sources through existing ECA rules. At the same time, the actual system needs to execute and control the generated ECA rules. Therefore, all events and conditions are the results of the feedback on external devices such as sensors or external information sources, and the action is implemented by the developer. When generating ECA rules, it is necessary to find the encapsulated events, conditions and actions in advance to meet the standardization of ECA rules.

3. Task Scenario

Task scenario analysis is to decompose the tactical task execution scenario into multiple combinable atomic tasks based on combat tasks, domain knowledge, and materials, and determine the logical structure between these atomic tasks. Among them, the experience formed in actual combat and training, the handling plan for emergencies, and the data of operational simulation can all be used as the material for scenario construction and analysis; domain knowledge is the facts and rules that military experts have been able to explain clearly and refine into rules. Rules, etc., are the basic constraints of scene construction and analysis. The parent–child relationship between tasks is formed through the decomposition mapping of high-level tasks and atomic tasks, which can be described by the task hierarchy diagram shown in Figure 1. Each complex task corresponds to a scene, and the atomic task describes more specific action details, so the execution of the atomic task is regarded as an atomic scene constructed by the scene. The composability of subtasks indicates whether a group of subtasks can be combined into a more complex task. Only subtasks that belong to the same parent task can be combined. For example, in Figure 1, the actions {AtomTask*4*, AtomTask*5*, AtomTask*6*} are combinable, and {AtomTask*1*, AtomTask*5*} are not combinable.

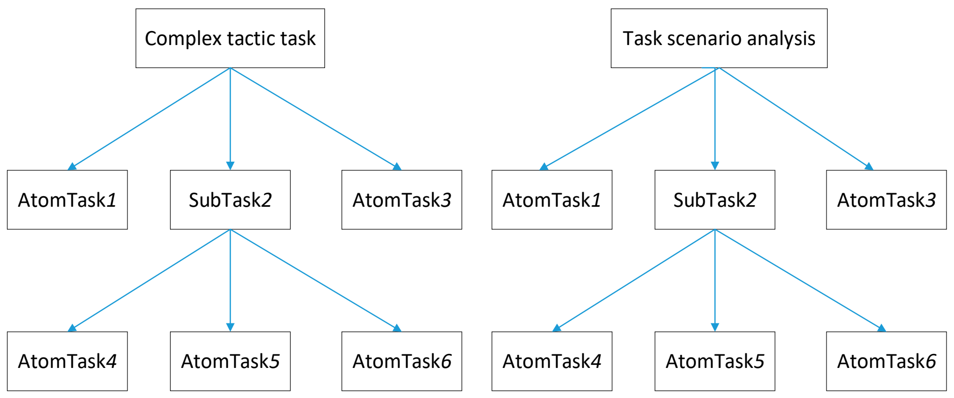

Figure 1. The hierarchical structure and corresponding relationship of tasks and scenarios.

A complex task can be expressed as:

$$\text{Task} = \{\text{TaskGroup, TaskStucture, TaskDependency}\}$$

where: TaskGroup = {SubTask*1*,..., SubTask*n*} represents the sub-task group obtained by task decomposition; TaskStucture represents the execution structure between sub-tasks; and TaskDependency $\in \{\&, |\}$ represents the dependency of the parent task on the sub-task.

The execution structure of the task group is mainly aimed at the execution relationship between the subtasks decomposed by the same parent task. Its basic structure mainly includes four types: sequence structure, concurrent structure, selection structure, and loop structure. Input and Output, respectively, represent the input and output of the task. Output, Cpre and Cpost, respectively, represent the pre-condition and post-condition of task execution. Among them, the sequence structure means that the execution of the next subtask can only be started after the execution of the previous subtask is completed. The selection structure means that the execution process of the subtasks is mutually exclusive, that is, there is one and only one subtask to be executed. Therefore, the preconditions of any two atomic tasks cannot be exactly the same, and they must be mutually exclusive. Concurrent structure refers to a group of atomic tasks that need to be executed at the same time. Therefore, the preconditions for the subtask group are exactly the same. The cyclic structure refers to a group of subtasks that need to be executed cyclically, and the order of execution of the subtasks in the cycle is fixed. In the process of task decomposition and scene design refinement, in order to further describe the dependency relationship between parent and child tasks, the relational operators '&' and '|' are introduced. The relational operator '&' connects the parent task to all subtasks, and the completion of the parent task depends on the execution of all subtasks. The relational operator '|' means that the realization of the parent task only needs to be satisfied by any subtask.

4. ECA Rule Extraction Process for Decision-Making Scenarios of Marine Formations

Different plans need to be conceived according to the occurrence of the scene, and different actions and the results of the actions must be evaluated according to these plans. According to the goals to be completed and the actual situation that may occur, list all the plans in the scene, and draw the processing process according to the execution steps of the plan. A task scene is composed of multiple atomic scenes, and each atomic scene implements a specific atomic task through the execution of the behavior of task participants. By traversing the sub-task groups and atomic tasks of each level of the task, a complete task scene can be constructed to fully reflect the details of task execution. According to the composability of atomic tasks, the process of extracting decision-making rules for naval formations only needs to decompose complex tasks hierarchically into a series of atomic tasks, conceive the execution scenarios of atomic tasks with reference to existing materials and domain knowledge, analyze the events that may be generated by the system, evaluate the execution conditions and possible results of different behaviors, and finally realize the system's event, condition, and behavior association mapping.

Therefore, the steps for extracting and constructing naval operational decision-making rules based on task scenario analysis are as follows:

Step 1: Task analysis. According to the operational tasks to be completed by the naval surface vessel system, combined with the scene design materials and domain knowledge constraints, the task is decomposed and classified, all possible situations are listed, complex tasks are decomposed into atomic tasks, and the execution sequence structure between these atomic tasks is determined by dependency relationships.

Step 2: Situation analysis. According to the actual situation that may occur, list the possible states of the system during the execution of each atomic task and the actions that may be executed in each state. The execution of each action will bring the system to a new specific state, as shown in Figure 2.

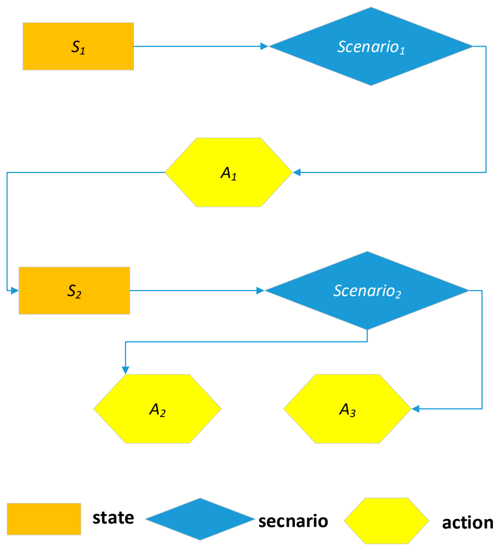

Figure 2. Scenario analysis diagram.

Step 3: Trigger event analysis. Comprehensive task analysis and scene analysis extract the corresponding events and trigger conditions. An event refers to a time-space state change or attribute change of a certain military significance that occurs in the system, target, and environment under certain conditions. The triggering event is the "switch" that activates the corresponding rules. The processing process of the integrated scenario analysis diagram and scenario plan is described as follows:

(a) The state in the scene analysis diagram is transformed into an event (represented by a rectangle) or a condition (represented by a diamond) of the ECA rules, which needs to be artificially distinguished according to the actual situation.
(b) Transform the conditions in the process into the conditions of the ECA rules.
(c) The actions that should be performed in the scene analysis graph are transformed into the actions of the ECA rules (represented by hexagons).

Step 4: State transition condition analysis. The event in the rule is actually a change of the situation, and the condition is a purposeful judgment on the past, present, and future scenarios to determine whether the action in the rule is executed. The "purpose" here includes the current state in the scenario analysis, the state to be achieved, and the constraints provided by domain knowledge. The conditions of judgment can be either the quantified values of the systems, goals, and environmental attributes that make up the scenario, or the qualitative complex relationships of events.

Step 5: Rule expression. According to the grammatical structure of the defined ECA rules, the corresponding rules can be obtained by substituting the trigger events, conditions, and actions obtained in each state into the grammatical structure of the rules. The essence of rule extraction is to express the state transition process in the form of rules, as shown in Figure 3. Illustrated as follows: S_1 is the initial state of $Scenario_1$, and A_1 occurs the $Scenario_1$ to S_2, which means the S_1 transfer to S_2 by A_1.

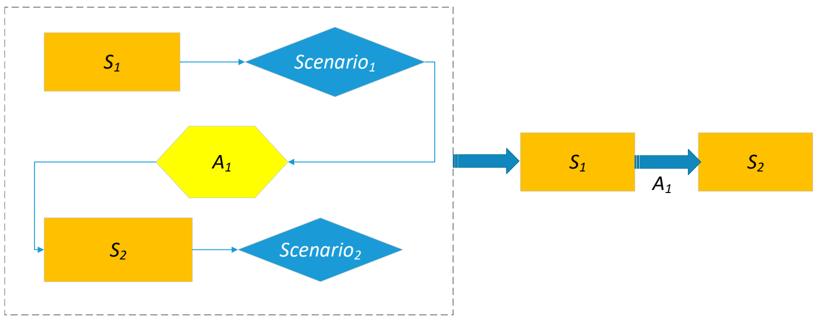

Figure 3. Correspondence between scenario analysis and rules.

According to the conditions in the action template specification, the action template specification, and the grammatical structure of the ECA rules, the process information obtained is finally mapped to the corresponding rules.

5. Consistency Detection of ECA Rules

5.1. Condition Template

In actual applications, in order to avoid redundancy of rules and ensure the accuracy of rule execution, it is necessary to check the consistency of the extracted ECA rules to obtain a complete ECA rule.

Regulation 1. *Given action a and action b, exclusive (a, b) means that action a and action b are mutually exclusive, that is, action a and action b cannot occur at the same time.*

Exclusive (a, b) means that action a is compatible with action b, that is, action a and action b can occur at the same time without affecting each other.

Regulation 2. *For event (or condition) c and event d, preceding (c, d) indicates that event c is the predecessor of event d, that is, the occurrence of event c will lead to the occurrence of event d.*

Consistency check: Let E, C and A correspond to the set of events, conditions and actions, respectively, Rule 1:{On e_1 If c_1 Do a_1}, Rule 2:{On e_2 If c_2 Do a_2}, and $\exists e_1,e_2 \in$ E, $\exists c_1,c_2 \in$ C, $\exists a_1,a_2 \in$ A.

Conclusion 1. If $e_1 = e_2$, $c_1 \subset c_2$, and $a_1 = a_2$, it is easy to get that either Rule1 or Rule2 is inaccurate. Then, if the occurrence of c_1 can fully trigger the action, it means that c_2 is redundant, and it can be further concluded that Rule2 is redundant. On the contrary, if c_2 is necessary, then c1 cannot be used as a condition to trigger an action. At this time, Rule1 is wrong.

Conclusion 2: When satisfied:

(1) preceding (e_1, e_2) (or $e_1 = e_2$), $c_1 = c_2$.
(2) preceding (c_1, e_2) (or $c_1 = e_2$), and the condition c_2 is established.

It can be seen that Rule 1 will be triggered at the same time Rule 2 will be triggered. If exclusive (a_1, a_2), then at least one of the rules Rule 1 and Rule 2 is wrong; if exclusive (a_1, a_2) (or $a_1 = a_2$), then Rule 1 and Rule 2 are compatible with each other. At this time, you need Integrate rules Rule 1 and Rule 2.

5.2. Consistency Detection of ECA Rules Based on SWRL

After extracting and constructing naval operational decision-making rules based on mission scenario analysis, it needs to be verified to ensure the consistency between the conceptual domains of the operational decision-making rule set and the elements within the conceptual domain due to complexity. SWRL is a language that uses semantic forms to present rules. The concept of its rule part is derived from RuleML and combined with OWL ontology to finally get this language. Use SWRL (Semantic Web Rule Language, Semantic Web Rule Language) rules to verify the operational decision-making rule set for a certain mission scenario. It is necessary to ensure the correctness of combat operations and the consistency of combat operations with other conceptual domains.

First of all, according to the services provided and the actual situation that may occur, the scenario plans of all services should be listed, and the logical model of each service should be described by ECA rules, and the ECA rules should be stored in the rule library; at the same time, in accordance with the provided services and ECA, the actions performed in the rules use certain ontology modeling tools such as protégé to define the contents of the top-level ontology, domain ontology, task ontology, and application ontology, and analyze and describe the classes and attributes of the ontology based on actual needs. Then, check the key factor action (Action), respectively, check the relationship between combat actions and actions, combat actions and resources (Resource), combat actions and combat strength (Strength), combat actions and combat objectives (Objection), Through the time (Time) description and space (Space) description to express the time and space description of each factor, and the time and space relationship between the factors, and to achieve the verification of the time and space relationship.

- ➢ Combat operations–actions mainly check the dependence and constraints of actions. The current action can only be carried out when its prerequisite actions are completed. The action–action check mainly realizes the pre-set check of each action in the joint combat plan. Whether the time of action is consistent with the sequence of actions.
- ➢ Combat operations–resources check whether the resource allocation in the plan meets the resource type and quantity required by the current action. First, check the resource type's constraints on the action, and then check the usable time of the reused resource and the resource of the consumable resource. Whether the quantity meets the needs of the action.
- ➢ Between combat action and force, it mainly realizes the verification of the constraints of the type and quantity of combat force on the action, which is similar to the verification of combat action-resources. First, verify whether the type of force meets the action, and then verify the satisfaction of the number of troops.
- ➢ The content of the verification between combat operations and targets is that combat operations must include more than one target and combat targets must be executed by more than one action.

Conflicts exist in any service system. There are many types of conflicts between services, and their manifestations are also different. The possible conflicts caused by the occurrence of two services include: purpose conflict, effect conflict, resource conflict, and redundant service; the time period of occurrence may be inclusive, overlapping, or sequential. In terms of performance, conflicts may appear in one system state (for example, when two conflicting services occur when the time period intersects), they may also appear in the comparison of several states, or when the results differ from expectations. Generally speaking, conflicts between rules mainly refer to that in certain states, multiple rules in a rule set may be triggered at the same time. If there are conflicts between multiple actions triggered by these rules, these rules conflict with each other. The action in the rule corresponds to a behavior in the service, and the action conflict refers to the resource conflict (used by the service) and effect conflict (affecting the service) in the service behavior. Therefore, in the conflict of operational rules, conflict detection refers to judging conflicting service behaviors and then detecting conflicting rules.

SWRL is a language that uses semantic forms to present rules. The concept of its rule part is derived from RuleML and combined with OWL ontology to finally get this language. In RuleML, the result of reasoning is represented by hand, and the basic form of the reasoning premise will be retained in SWRL and represented by body. Therefore, SWRL can be regarded as a combination of rules and ontology. Through the combination of the two, the relations and vocabulary described in the ontology can be directly used when writing the rules, because in SWRL, the relations between these categories no longer need other methods and rule descriptions, and the ontology can be directly used. To describe. The main idea of SWRL is to expand the OWL interpretation, and it can map variables to the domain. Only when the antecedent and subsequent parts of this mapping are met at the same time, a rule is established. According to the way the SWRL expresses the rules, the following grammatical format is defined:

Definition 3. *Predicatel (?Variablel)ˆPredicate2 (?Variable,?Variable3)→Predicate3 (?Variable4, ?Variable5).*

Among them, the implication is represented by the arrow symbol "→", which logically associates the premise and the conclusion, that is, the premise before the arrow and the conclusion after the arrow; the predicate uses "∧" for conjunctive connection; variables start with a question mark (?).

There are two aspects to service conflict discovery and detection. On the one hand, it is the discovery and detection of conflicting service behaviors, which are mainly resolved through semantic analysis, such as the context-aware conflict framework, which uses the method of constructing contextual conflict management middleware to detect and resolve conflicts. On the other hand, it is to discover conflicting service logic. Most of these methods assume that the service is known, and predefine conflicting service behaviors. On this basis, discover the service logic that will cause conflicts, such as that based on conflicts. The time for detection and resolution, the conflicts in the pervasive computing environment are divided into static conflicts and dynamic conflicts, and a method of adding post-conditions to ECA to detect conflicts is proposed.

According to the above prerequisites, this project uses SWRL to construct meta-rules for resource consumption and resource state transition. The required predicates are as Table 1.

Table 1. List of air and sea coordinated assaults on the sea.

Event Name	Remark
Rule (?r)	r is a rule
FamilyService (?fs)	fs is a subclass of the FamilyService class
Property (?p)	p is an object attribute (operation on resources)
Resource (?res)	res is a subclass of Resource
hasService (?r, ?Is)	Rule r has service fs
hasProperty (?fs, ?p)	Service fs has object attribute p
hasConstraint (?p, ?res)	Object attribute p has attribute constraints res
resConsumption (?fs, ?res)	Service fs consumes resources res
Changestate (?res, ?s)	The status of resource res becomes s

(Firstly, construct meta-rules for resource consumption.If the execution of the service in the rule needs to consume a certain resource, it can be transformed into the following meta-rule:

metaR-1:

Rule (?r)∧FamilyService (?fs)ˆhasService (?r, ?fs)ˆProperty (?p)ˆhasProperty (?fs, ?p) ˆResource (?res)ˆhasResource (?p, ?res)→resConsumption(?fs, ?res)

Afterwards, construct meta-rules for resource state transition.Suppose the resource has three states: on, off, and occupied, that is, the values of the variable s in the above table changeState (?res, ?s) are on, off, and busy, respectively. After the ECA rule is executed, the condition for the resource state to change is:

① If the Action (action or service) part of an ECA rule has object attribute turn-on, then after this rule is executed, the resource status will become on;

② If the Action (action or service) part of an ECA rule has an object attribute turn-off, then after this rule is executed, the resource status will become off;

③ If the Action (action or service) part of an ECA rule has object attribute occupation, then after this rule is executed, the resource status will become busy.

According to the above natural language description, it is transformed into a meta-rule, as shown below:

① metaR-2:

Rule (?r)^FamilyService (?fs)^hasService (?r,?fS)^Property (tum-on)^hasProperty (?fs,tum-on)^Resource (?res)^hasResource (tum-on, ?res)→changeState (?res, on)

② metaR-3:

Rule (?r)^FamilyService (?fs)^hasService (?r, 7fs)^Property (turn-off)^hasProperty (?fs, turnoff)^Resource (?res)"hasResource (tum-off, ?res)→changeState(?res, off)

③ metaR-4:

Rule (?r)^FamilyService (?fs)^hasService (?r, ?fs)^Property (occupy)^hasProperty (?fs,occupy)^Resource (?res)^hasResource (occupy, ?res)→changeState (?res, busy)

Another definition is as follows:

Definition 4. *If ∃resl | sl (changeState (?res, ?s)), ∃res2 | s2 (changeState (?res, ?s)), and resl = res2, s1 ≠ s2, then changeState (resl, s1) = changeState (res2, s2) (or changeState (res2, s2) = changeState (resl, s1)).*

Based on the above meta-rules, to determine and detect ECA rule conflicts according to the conflict detection algorithm, first define the following:

Definition 5. *FPR (service-property-resource collection) = {(fs, p, res) | fs is a subclass of Family-Service, p is an object property, res is a subclass of Resource}.*

Definition 6. *SC (Resource Conflict Set) = {(Fsl, Fs2, Res) | Fsl and Fs2 resource conflict, and occupy resource Res at the same time}.*

Definition 7. *EC (effect conflict set) = {(Fsl, Fs2) l Fsl and Fs2 effect conflict}.*

The specific detection algorithm is as follows:

Definition 8. *RC (Rule Conflict Set) = {(Evt, Fs, conEvt, conFs, type) | Evt is the action (the Action of the ECA rule) is the triggering event (ECA rule) of the ECA rule of sci.Fs1 (or eci.Fs1) Event), Fs is the corresponding sci.Fs1 (or eci.Fs1), conEVt is the triggering event of the ECA rule whose action is sci.Fs2 (or eci.Fs2), and conFs is the corresponding sci.Fs2 (or eci.Fs2)), type is SC (or EC); i = 1,2...}.*

The rule conflict set marks the triggering event of the ECA rule that may cause conflict (Event in the ECA rule). When the rule is executed, when one of the events is triggered and the corresponding conditions are met, the triggering event of the conflicting rule is detected in real time Whether it happens, once the conflict triggering event occurs, you need to take corresponding solutions.

6. Verification Example

In order to verify the feasibility of the technology, take the naval surface vessel air–sea coordinated assault operation as an example, and conduct rule extraction according to the ECA-driven decision-making mechanism. In response to unexpected situations that may occur in a certain sea area, against the enemy's naval fleet, the use of aviation, surface ships, submarines and shore guidance forces to implement multi-wave contract attacks against the enemy with the support of shore-based electronic warfare forces. The command post has direct command of its troops. The shore-based command post and various combat platforms jointly complete various combat phase tasks such as comprehensive situation analysis, determination of strike determination, formulation of combat plans, adjustment of combat plans, and execution of combat plans.

The process of extracting and constructing operational rules is as follows:

(1) Task analysis.

Based on the analysis of the initial situation of the battlefield, the task is decomposed, and the air and sea coordinated assault mission is decomposed (as shown in Figure 3).

Decompose task dependencies. Based on the "task decomposition diagram", further analyze the sub-task execution structure and dependencies of the air–sea jointed anti-sea assault mission, and draw a "task execution structure diagram" (as shown in Figure 4).

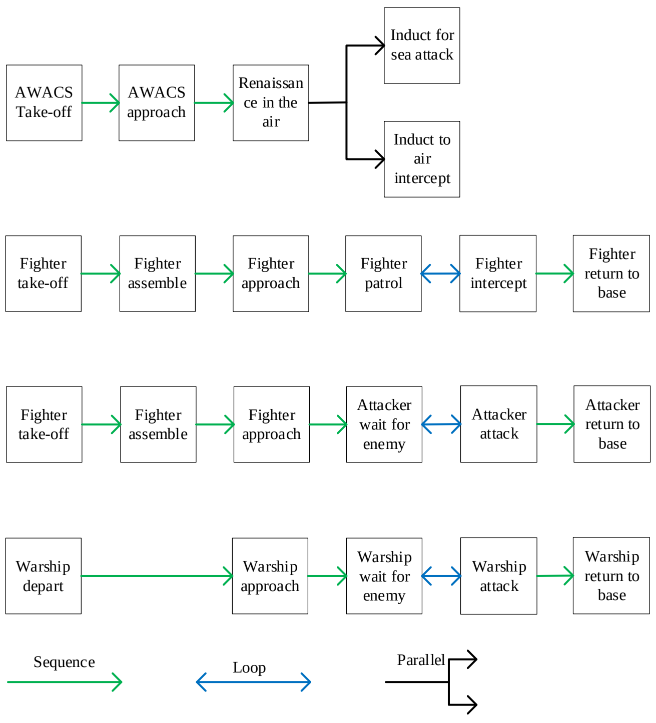

Figure 4. The execution structure of air–sea jointed anti-sea assault missions.

(2) Scene analysis.

Based on task analysis, combined with domain knowledge, analyze and list various possible states, events, actions, and new states after the actions during combat operations, and obtain "air–sea jointed anti-sea assault scenario analysis diagram", as shown in Figure 5.

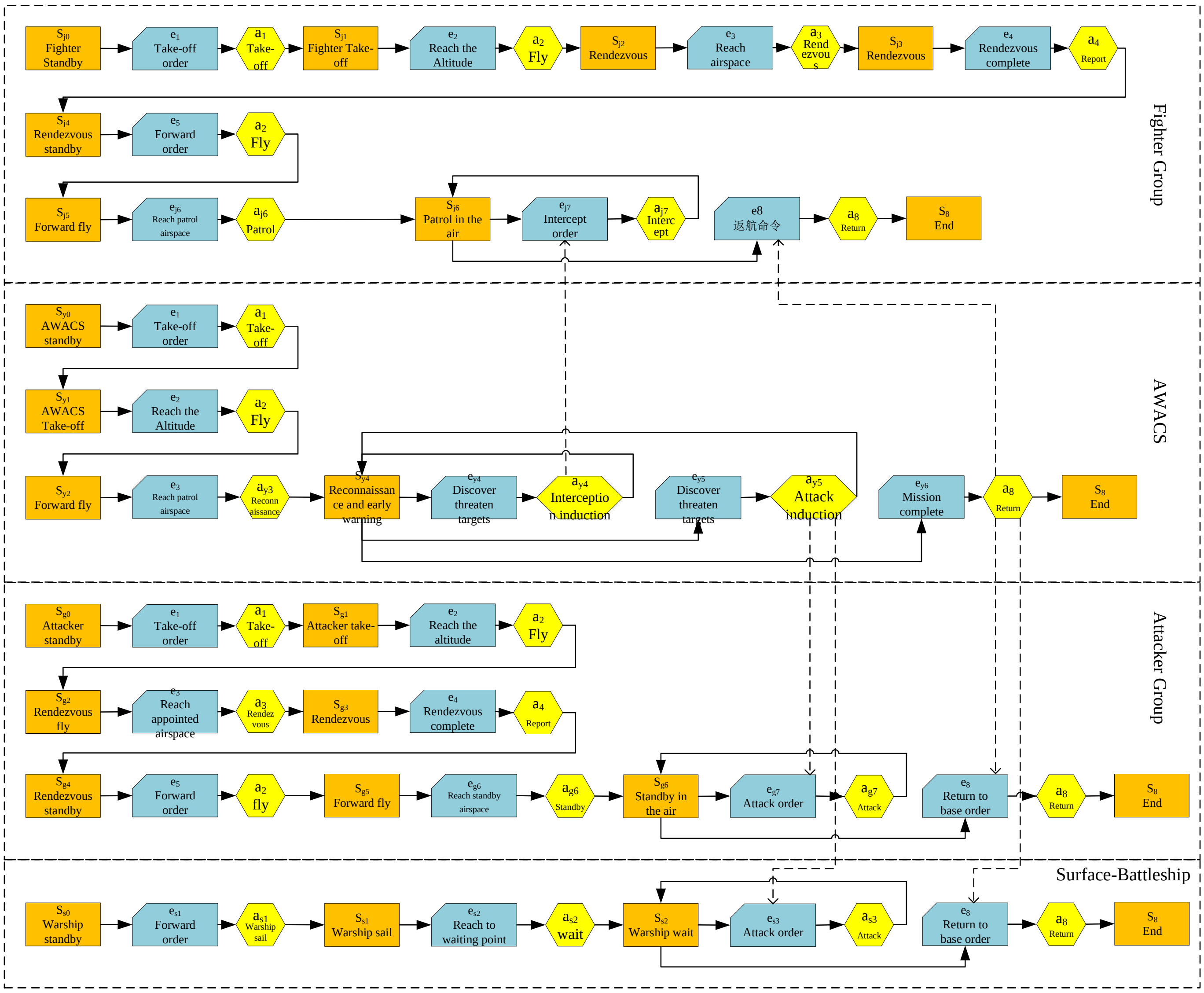

Figure 5. Analysis of air–sea jointed anti-sea assault scenarios.

(3) Trigger event analysis.

Based on the "scene analysis graph, analyze and sort out the events that trigger new actions, and "air–sea jointed anti-sea assault event list" is proposed (shown in Table 2).

Table 2. List of air and sea jointed anti-sea assaults in naval battlefield.

Event	Event Name	Remark
e_1	Takeoff order received	The plane received a takeoff order
e_2	Reach height	The plane reaches the specified altitude
e_3	Arriving in the airspace	Plane arrives in designated airspace
e_4	Assembled	The aircraft is assembled
e_5	Forward order received	The aircraft received a forward order
e_{j6}	Fighter arrives in patrol airspace	The plane arrives at the designated patrol airspace
e_{j7}	Receipt of interception order	Fighter received intercept order
e_8	Received return order	Received return order
e_{y4}	Aerial threat found	Early warning detection of aerial threats
e_{y5}	Discover and identify the target of the attack	The early warning aircraft found and identified the target of the attack
e_{y6}	mission completed	The attack target reaches the specified damage level, and the attack mission is completed
e_{g6}	Attack aircraft arrives in standby airspace	The attack aircraft arrives in the designated standby airspace
e_{g7}	Attack order received	The attack aircraft received the command to attack the sea
e_{s1}	Receipt of sailing order	Warship received an order to sail
e_{s2}	Arrived in the standby area	Warship arrives in the standby area
e_{s3}	Attack order received	Warship receives an order to attack the sea

(1) State transition condition analysis.

Based on the scene analysis graph and event table, the trigger conditions for triggering the state transition were analyzed and sorted out, and the “air–sea coordinated assault state transition table” was obtained (as shown in Table 3).

Table 3. Changes in the state of air and sea jointed anti-sea assaults in naval battlefield.

State Transfer	Event	Transfer Condition	Execution Action	Etraction Rule
$S_{j0}\rightarrow S_{j1}$	e_1	c_{j1}: Fighter received takeoff order	a_1: take off	R_{j1}
$S_{j1}\rightarrow S_{j2}$	e_2	c_{j2}: The fighter plane reaches the predetermined height	a_2: flight	R_{j2}
$S_{j2}\rightarrow S_{j3}$	e_3	c_{j3}: Fighter arrives in the assembly airspace	a_3: Assemble	R_{j3}
$S_{j3}\rightarrow S_{j4}$	e_4	c_{j4}: Fighter assembly is complete	a_4: The report is assembled	R_{j4}
$S_{j4}\rightarrow S_{j5}$	e_5	c_{j5}: Fighter receives forward order	a_2: flight	R_{j5}
$S_{j5}\rightarrow S_{j6}$	e_{j5}	c_{j6}: Fighter arrives in patrol airspace	a_{j6}: patrol	R_{j6}
$S_{j6}\rightarrow S_{j6}$	e_{j6}	c_{j7}: Fighter received intercept order	a_{j7}: Air intercept	R_{j7}
$S_{j6}\rightarrow S_8$	e_8	c_{j8}: Fighter received a return order	a_8: Return home	R_{j8}
$S_{y0}\rightarrow S_{y1}$	e_1	c_{y1}: AWACS received takeoff order	a_1: take off	R_{y1}
$S_{y1}\rightarrow S_{y2}$	e_2	c_{y2}: AWACS reached a predetermined height	a_2: flight	R_{y2}

Table 3. *Cont.*

State Transfer	Event	Transfer Condition	Execution Action	Etraction Rule
$S_{y2}{\rightarrow}S_{y3}$	e_3	c_{y3}: AWACS arrives at the scheduled airspace	a_{y3}: Reconnaissance search	R_{y3}
$S_{y3}{\rightarrow}S_{y3}$	e_{y4}	c_{y4}: An early warning aircraft detects an air threat^The fighter plane arrives in the patrolled airspace	a_{y4}: Guide fighter interception	R_{y4}
$S_{y3}{\rightarrow}S_{y3}$	e_{y5}	c_{y5}: The early warning aircraft finds and recognizes the attack target^The attack aircraft arrives in the attack airspace^Warship arrives in the standby area	a_{y5}: Guide attack aircraft and surface ships to attack the sea	R_{y5}
$S_{y3}{\rightarrow}S_8$	e_{y6}	c_{y6}: All assault targets have reached the level of damage	a_8: Return home	R_{y8}
$S_{g0}{\rightarrow}S_{g1}$	e_1	c_{g1}: Attack aircraft received takeoff order	a_1: take off	R_{g1}
$S_{g1}{\rightarrow}S_{g2}$	e_2	c_{g2}: The attack aircraft reaches the specified altitude	a_2: flight	R_{g2}
$S_{g2}{\rightarrow}S_{g3}$	e_3	c_{g3}: The attack aircraft arrives in the assembly airspace	a_3: Assemble	R_{g3}
$S_{g3}{\rightarrow}S_{g4}$	e_4	c_{g4}: Attack aircraft assembled	a_4: The report is assembled	R_{g4}
$S_{g4}{\rightarrow}S_{g5}$	e_5	c_{g5}: The attacker received the forward order	a_2: flight	R_{g5}
$S_{g5}{\rightarrow}S_{g6}$	e_{g6}	c_{g6}: Attack aircraft arrives in standby airspace	a_{g6}: Air standby	R_{g6}
$S_{g6}{\rightarrow}S_{g6}$	e_{g7}	c_{g7}: Attacking aircraft received an attack command	a_{g7}: Attack aircraft to sea attack	R_{g7}
$S_{g6}{\rightarrow}S_8$	e_8	c_{j8}: Attack aircraft received a return order	a_8: Return home	R_{g8}
$S_{s0}{\rightarrow}S_{s1}$	e_{s1}	c_{s1}: Warship received an order to sail	a_{s1}: Surface ships set sail	R_{s1}
$S_{s1}{\rightarrow}S_{s2}$	e_{s2}	c_{s2}: Warship arrives in the standby area	a_{s2}: Surface ship standby	R_{s2}
$S_{s2}{\rightarrow}S_{s2}$	e_{s3}	c_{s3}: Warship receives an attack order	a_{s3}: Surface ship to sea attack	R_{s3}
$S_{s2}{\rightarrow}S_8$	e_7	c_{s4}: Surface ship receives order to return	a_8: Return home	R_{s4}

(2) Rule expression.

According to the grammatical structure of ECA rules, each change in the state of air and sea jointed anti-sea assaults in the state transition table is mapped to a corresponding rule.

7. Conclusions

In this work, the design of a rule-based extraction and construction architecture has been proposed with the aim to help commanders in defining their decision rules. Decision-making is the central task of command. Improving the timeliness and accuracy of operational decision-making in naval formations is an important method to improve combat command capabilities [29]. In view of the high requirements of current operational planning on decision timeliness, the situation analysis and plan formulation in the decision-making process should be assigned to the intelligent autonomous decision-making model to minimize the workload of the commander's decision-making. Improve the efficiency and rationalization of operational decision-making, and realize the rapid formulation and optimization of decision-making plans to adapt to the sensitivity requirements of wartime decision-making on time, data, and changing battlefield environments. However, the naval surface vessel jointed operational decision-making is a complex issue. Different operational missions and different jointed operational platforms have different operational rules. How to effectively obtain, organize, and describe operational rules requires a lot of experimentation and continuous improvement. It also needs to further research on the perception and learning behaviors of naval operational decision rules based on task scenario analysis.

Author Contributions: Conceptualization, X.Z.; formal analysis, C.W.; investigation, X.Z.; resources, X.Z.; data curation, G.S.; writing—original draft preparation, X.Z.; writing—review and editing, P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, Y.; Xu, X.; Zhang, X.; Qian, L.; Zhou, X. Hierarchical Reinforcement Learning for Autonomous Decision Making and Motion Planning of Intelligent Vehicles. *IEEE Access* **2020**, *8*, 209776–209789. [CrossRef]
2. Su, X.; Zhong, M. Supply Chain Risk Prevention and Control Based on Fuzzy Influence Diagram and Discrete Hopfield Neural Network. *Discret. Dyn. Nat. Soc.* **2021**, *2021*, 1–15. [CrossRef]
3. Khakzad, N. Optimal firefighting to prevent domino effects: Methodologies based on dynamic influence diagram and mathematical programming. *Reliab. Eng. Syst. Saf.* **2021**, *212*, 107577. [CrossRef]
4. Weflen, E.; MacKenzie, C.A.; Rivero, I.V. An Influence Diagram Approach to Automating Lead Time Estimation in Agile Kanban Project Management. *Expert Syst. Appl.* **2022**, *187*, 115866. [CrossRef]
5. Tkatek, S.; Bahti, S.; Abdoun, O.; Abouchabaka, J. Intelligent system for recruitment decision making using an alternative parallel-sequential genetic algorithm. *Indones. J. Electr. Eng. Comput. Sci.* **2022**, *22*, 385. [CrossRef]
6. Smitha, R.E.; Dikeb, B.A.; Mehrac, R.K.; Ravichandranc, B.; El-Fallahc, A. Classifier systems in combat: Two-sided learning of maneuvers for advanced fighter aircraft. *Comput. Methods Appl. Mech. Eng.* **2000**, *186*, 421–437. [CrossRef]
7. Chen, T. Decision-making support for transportation and logistics combining rough set fuzzy logic algorithm. *J. Intell. Fuzzy Syst.* **2021**, *41*, 1–10. [CrossRef]
8. Dumitrescu, C.; Ciotirnae, P.; Vizitiu, C. Fuzzy Logic for Intelligent Control System Using Soft Computing Applications. *Sensors* **2021**, *21*, 2617. [CrossRef]
9. Makarov, M.; Kuryshov, A. Researcing the Fault Tolerance of Robotic System Designed via Use of Neural Network Decision Making Component of Image Processing. In Proceedings of the 2018 Engineering and Telecommunication (EnT-MIPT), Moscow, Russia, 15–16 November 2018; pp. 197–200.
10. Winkel, D.J.; Breit, H.C.; Weikert, T.J.; Stieltjes, B. Building Large-Scale Quantitative Imaging Databases with Multi-Scale Deep Reinforcement Learning: Initial Experience with Whole-Body Organ Volumetric Analyses. *J. Digit. Imaging* **2021**, *34*, 124–133. [CrossRef]
11. Cheng, Y.; Song, Y. Autonomous Decision-Making Generation of UAV based on Soft Actor-Critic Algorithm. In Proceedings of the 2020 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020; pp. 7350–7355.
12. Bouguettaya, A.; Hafed, H.; Ahmed, Z.; Amine, M.T. Vehicle Detection From UAV Imagery With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–21. [CrossRef]
13. Hu, G. Research on the Combat Forms of Future Naval Battlefields. *Mod. Radar* **2019**, *41*, 16–20.

14. Hu, L.; Liang, X.; He, L. Construction method of aviation cluster decision rule base based on scenario analysis. *J. Aeronaut.* **2020**, *41*, 723–737.
15. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.J.; McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 22–27 June 2014.
16. Gu, F.; Jiang, D. Prediction of Political Leanings of Chinese Speaking Twitter Users. In *Center for Complex Networks and Systems Research*; School of Informatics and Computing: Bloomington, IN, USA, 2021.
17. Wu, Q.; Li, Q.; Zhou, J.J.; Long, X.Y.; Lin, H.P. Film and TV News Digest Generation method Based on HanLP. In Proceedings of the 2020 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking IEEE, Exeter, UK, 17 December 2020.
18. Rina, D.; Fauziah, F.; Hayati, N. Aplikasi Spoxtech Untuk Penyandang Tuna Rungu Wicara Menggunakan Algoritma Hidden Markov Model dan Metode Finite State Automata (FSA). *STRING* **2021**, *5*, 236. [CrossRef]
19. Paripremkul, K.; Sornil, O. *Segmenting Words in Thai Language Using Minimum Text Units and Conditional Random Field*; Graduate School of Applied Statistics, National Institute of Development Administration (NIDA): Bangkok, Thailand, 2021; Volume 12.
20. Morfouace, P.; Séréville, N.; Flavigny, F.; Labiche, M.; Shearman, R. NPTool: A simulation and analysis framework for low-energy nuclear physics experiments. *J. Phys. G Nucl. Part. Phys.* **2016**, *43*, 045113.
21. Huang, B.; Bai, Y.; Zhou, X. Hub at SemEval-2021 Task 1: Fusion of Sentence and Word Frequency to Predict Lexical Complexity. In Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), Bangkok, Thailand, 5–6 August 2021.
22. Wan, I.P.; Allassonnière-Tang, M. *The Effect of Word Frequency and Position-in-Utterance in Mandarin Speech Errors: A Connectionist Model of Speech Production*; Cambridge University Press: Cambridge, UK, 2021.
23. Bolshakova, E.; Loukachevitch, N.; Nokel, M. Topic models can improve domain term extraction. In *European Conference on Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 684–687.
24. Sawczyn, A.; Binkowski, J.; Janiak, D.; Augustyniak, Ł.; Kajdanowicz, T. Fact-checking: Relevance assessment of references in the Polish political domain. *Procedia Comput. Sci.* **2021**, *192*, 1285–1293. [CrossRef]
25. Haque, R.; Penkale, S.; Way, A. TermFinder: Log-likelihood comparison and phrase-based statistical machine translation models for bilingual terminology extraction. *Lang Resour. Eval.* **2018**, *52*, 365–400. [CrossRef]
26. Cohen, J.D. Highlights: Language-and domain-independent automatic indexing terms for abstracting. *J. Am. Soc. Inf. Sci.* **1995**, *45*, 162–174. [CrossRef]
27. Bae, J.; Bae, H.; Kang, S.; Kim, Y. Automatic control of workflow processes using ECA rules. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1010–1023.
28. Bak, N.; Chang, B.-M.; Choi, K. Smart Block: A visual block language and its programming environment for IoT. *J. Comput. Lang.* **2020**, *60*, 100999. [CrossRef]
29. Liu, M.; Zhang, H.; Hao, W. Intelligent decision-making method for combat operations of tactical-level war chess entities. *Control Decis.* **2020**, *35*, 2977–2985.