

# Supplementary Materials: Rule Sets and Class Files

Kazuto Sasai, Ryota Fukutani, Gen Kitagata and Tetsuo Kinoshita

As part of the data availability, this document provides rulesets for the Java classes and distributed agent system-based hybrid architecture (DASH) framework [1] that are commonly used in the programs created in this research, and is distributed for the purpose of promoting reuse by readers.

## 1. Implementation of Rule Sets

The DASH agent framework provides OAV (Object-Attribute-Value) format for rule set description. The rule sets provide the common action rules for each agent.

### 1.1. Repository

Listing S1 is the rule set for data processing agents (DPAs) in the repository. The rules on the 12th and 37th lines are a group of rules executed by a data processing agent that converts the database into an agent. This rule activates the agent when it receives a message whose performative attribute is “request notification” and the device attribute of the search object in the request content matches the data it holds. The rule on line 62 is for the data processing agent that holds network monitoring information. When a message with the performative attribute of “request notification” is received, it is generated in the operating environment regardless of its contents. The rule on line 135 is a rule that fires when a message whose performative attribute is “task notification” is received. When this rule fires, the `judgeProcess` method commonly implemented in the base process is executed to determine whether the agent is activated. If the judgment result is “TRUE”, the rule on the 162nd line is executed, and if it is “FALSE”, the rule on the 189th line is executed. The rules on lines 202 and 213 are rules for deleting messages that do not correspond to the above.

Listing S1: Rule set file of Repository

```

1 (rule-set RepositoryRulesVer2
2
3   (property)
4   (initial_facts)
5
6   (rule load_BP
7     (baseprocess :name ?name)
8     -->
9     (loadBP ?name)
10  )
11
12  (rule receive_request_notification_to_DBAg_1
13    (Msg
14      :performative "request notification"
15      :from ?agName
16      :to _broadcast
17      :departure ?dep
18      :content (search :device ?dev :target ?tar :action ?action)
19    ) = ?msg
20    (data :hostname ?dev)
21    (process :type ?action)
22    (startup :number ?num) = ?startup
23    (workplace :name ?wp)
24    -->
25    (instantiate :into ?wp :facts (
26      (files :number ?num)
27      (notification :from ?agName :departure ?dep)
28      (search :device ?dev :target ?tar :action ?action :relevance 1.0)
29      (Members :manager ?agName :contractor ())
30      (process :type ?action :flag TRUE))

```

```

31     )
32     (bind ?number (+ ?startup:number 1))
33     (modify ?startup:number ?number)
34     (remove ?msg)
35 )
36
37 (rule receive_request_notification_to_DBAg_2
38   (Msg
39     :performative "request notification"
40     :from ?agName
41     :to _broadcast
42     :departure ?dep
43     :content (search :device ?dev :target ?tar :action ?action)
44   ) = ?msg
45   (data :hostname ?dev)
46   (process :type ?action)
47   (startup :number ?num) = ?startup
48   (workplace :name ?wp)
49   -->
50   (instantiate :into ?wp :facts (
51     (files :number ?num)
52     (notification :from ?agName :departure ?dep)
53     (search :device ?dev :target ?tar :action ?action :relevance 1.0)
54     (Members :manager ?agName :contractor ())
55     (process :type ?action :flag FALSE))
56   )
57   (bind ?number (+ ?startup:number 1))
58   (modify ?startup:number ?number)
59   (remove ?msg)
60 )
61
62 (rule receive_request_notification_to_DBAg_3
63   (Msg
64     :performative "request notification"
65     :from ?agName
66     :to _broadcast
67     :departure ?dep
68     :content (search :device ?dev :target ?tar :action ?action)
69   ) = ?msg
70   (accept :target ?dev)
71   (data :hostname ?dev)
72   (process :type ?action)
73   (startup :number ?num) = ?startup
74   (workplace :name ?wp)
75   -->
76   (instantiate :into ?wp :facts (
77     (files :number ?num)
78     (notification :from ?agName :departure ?dep)
79     (search :device ?dev :target ?tar :action ?action :relevance 1.0)
80     (Members :manager ?agName :contractor ())
81     (process :type ?action :flag TRUE))
82   )
83   (bind ?number (+ ?startup:number 1))
84   (modify ?startup:number ?number)
85   (remove ?msg)
86 )
87
88 (rule receive_request_notification_to_DBAg_4
89   (Msg
90     :performative "request notification"
91     :from ?agName
92     :to _broadcast
93     :departure ?dep
94     :content (search :device ?dev :target ?tar :action ?action)
95   ) = ?msg
96   (accept :target ?dev)
97   (data :hostname ?dev)
98   (process :type ?action)
99   (startup :number ?num) = ?startup

```

```

100     (workplace :name ?wp)
101     -->
102     (instantiate :into ?wp :facts (
103         (files :number ?num)
104         (notification :from ?agName :departure ?dep)
105         (search :device ?dev :target ?tar :action ?action :relevance 1.0)
106         (Members :manager ?agName :contractor ()))
107         (process :type ?action :flag FALSE))
108     )
109     (bind ?number (+ ?startup:number 1))
110     (modify ?startup:number ?number)
111     (remove ?msg)
112 )
113
114 (rule receive_request_notification_to_InfoAg
115     (Msg
116         :performative "request notification"
117         :to _broadcast
118         :content (search :device ?dev :target ?tar :action ?action :from User)
119     ) = ?msg
120     (data :index NONE)
121     (startup :number ?num) = ?startup
122     (workplace :name ?wp)
123     -->
124     (instantiate :into ?wp :facts (
125         (files :number ?startup:number)
126         (notification :from ?msg:from :departure ?msg:departure)
127         (search :device ?dev :target ?tar :action ?action :relevance 1.0)
128         (Members :manager ?msg:from :contractor ()))
129     )
130     (bind ?number (+ ?startup:number 1))
131     (modify ?startup:number ?number)
132     (remove ?msg)
133 )
134
135 (rule receive_task_notification_to_ProcAg_1
136     (Msg
137         :performative "task notification"
138         :content (search :action ?action)
139     )
140     (data)
141     (process :type ?proc :action ?action :degree ?deg)
142     //(process :type ?proc :action ?action :degree ?degree)
143     (check)
144     -->
145     (make (check :flag TRUE))
146     (control judgeProcess(?proc ?deg))
147 )
148
149 (rule receive_task_notification_to_ProcAg_2
150     (Msg
151         :performative "task notification"
152         :content (search :action ?action)
153     )
154     (data)
155     (process :type ?proc :action "ALL" :degree ?deg)
156     (process :type ?proc :action ?action)
157     (check)
158     -->
159     (make (check :flag TRUE))
160     (control judgeProcess(?proc ?deg))
161 )
162
163 (rule check_task_TRUE
164     (Msg
165         :performative __Event
166         :content (process :type ?type :flag TRUE)
167     ) = ?msg1
168     (Msg

```

```

169         :performative "task notification"
170         :content (search :device ?dev :target ?tar :action ?action)
171     ) = ?msg2
172     (startup) = ?startup
173     (workplace :name ?wp)
174     (check) = ?check
175     -->
176     (instantiate :into ?wp :facts (
177         (files :number ?startup:number)
178         (notification :from ?msg2:from :departure ?msg2:departure)
179         (search :device ?dev :target ?tar :action ?action :relevance 1.0)
180         (Members :manager ?msg2:from :contractor ())
181         (process :type ?type :flag TRUE))
182     )
183     (bind ?number (+ ?startup:number 1))
184     (modify ?startup:number ?number)
185     (remove ?msg1)
186     (remove ?msg2)
187     (remove ?check)
188 )
189
190 (rule check_task_FALSE
191     (Msg
192         :performative __Event
193         :content (process :type ?type :flag FALSE)
194     ) = ?msg1
195     (Msg :performative "task notification") = ?msg2
196     (check) = ?check
197     -->
198     (remove ?msg1)
199     (remove ?msg2)
200     (remove ?check)
201 )
202
203 (rule ignore_irrelevant_request
204     (Msg
205         :performative "request notification"
206         :content ((search :relevance ?rel) ?noti)
207     ) = ?msg
208     (relevance :border ?border)
209     (> ?border ?rel)
210     -->
211     (remove ?msg)
212 )
213
214 (rule ignore_request_notification_1
215     (Msg
216         :performative "request notification"
217         :content (search :device ?word)
218     ) = ?msg
219     (data)
220     (data :hostname ?word)
221     -->
222     (remove ?msg)
223 )
224
225 (rule ignore_request_notification_2
226     (Msg
227         :performative "request notification"
228         :content (search :device ?word)
229     ) = ?msg
230     (data)
231     -->
232     (remove ?msg)
233 )
234
235 )

```

## 1.2. Workplace

Listing S2 is the rule set for the DPAs activated to the workplace. In this rule set, in addition to calling the base process, the rule sets for F1, F2, and F3 are switched.

Listing S2: Rule set for workplace

```

1 (rule-set WorkplaceRulesVer2
2
3   (property)
4   (initial_facts)
5
6   (rule load_BP
7     (baseprocess :name ?name)
8     (target :url ?url)
9     (files :number ?num) = ?files
10    (base :path ?path)
11    -->
12    (loadBP ?name)
13    (control setURL(?url))
14    (control makePathName(?path ?num))
15    (make (state :name "F1"))
16    (remove ?files)
17    (activate (F1Rules WorkplaceRulesVer2))
18  )
19
20  (rule receive_file_path_name
21    (Msg
22      :performative __Event
23      :content (file :path ?path)
24    ) = ?msg
25    (Status :name ?myName)
26    -->
27    (make (file :path ?path :by ?myName))
28    (remove ?msg)
29  )
30
31  (rule receive_output_info
32    (Msg
33      :performative __Event
34      :content (output :type ?type)
35    ) = ?msg
36    -->
37    (make ?type)
38    (remove ?msg)
39  )
40
41  (rule activate_f2_rules
42    (state :name "F2") = ?state
43    -->
44    (modify ?state:name "F2_1")
45    (activate (F2Rules WorkplaceRulesVer2))
46  )
47
48  (rule activate_f3_rules
49    (state :name "F3")
50    -->
51    (activate (F3Rules WorkplaceRulesVer2))
52  )
53
54  (rule activate_default_rules
55    (state :name "processing")
56    -->
57    (activate (WorkplaceRulesVer2 _default))
58  )
59
60  (rule activate_post_processing_rules
61    (state :name "post processing")
62    -->

```

```

63     (activate (PostProcessingRules WorkplaceRulesVer2))
64   )
65 )

```

### 1.3. F1: Request matching

Listing S3 is the rule set of F1 which corresponds to the request matching property defined in Section 2.1.1 of the main manuscript. In this ruleset, it reports to the request issuer that it has been generated (lines 12 and 21), and from lines 37 to 77, each time-series data it holds is sent to the administrator. It is determined whether to provide to If there is more than one data to provide, he moves on to F2, and if there is none, it disappears from the workplace (line 120).

Listing S3: Rule set file of F1

```

1  (rule-set F1Rules
2
3    (property)
4    (initial_facts)
5
6    (rule send_request_acceptance
7      (notification :from ?agName :departure ?dep)
8      (data)
9      (Status :name ?myName)
10     (state :name "F1") = ?state
11     -->
12     (send
13       :performative "request acceptance"
14       :to ?agName
15       :arrival ?dep
16       :content (accepted :by ?myName)
17     )
18     (modify ?state:name "F1-check")
19   )
20
21   (rule send_task_acceptance
22     (notification :from ?agName :departure ?dep)
23     (data)
24     (process :flag TRUE)
25     (Status :name ?myName)
26     (state :name "F1") = ?state
27     -->
28     (send
29       :performative "task acceptance"
30       :to ?agName
31       :arrival ?dep
32       :content ()
33     )
34     (modify ?state:name "F2")
35   )
36
37   (rule match_attribute_hostname_and_description
38     (state :name "F1-check")
39     (search :device ?dev :target ?tar)
40     (data :description ?tar :hostname ?dev :flag FALSE :judge FALSE) = ?data
41     -->
42     (modify ?data:flag TRUE)
43     (modify ?data:judge TRUE)
44   )
45
46   (rule match_attribute_hostname_and_field
47     (state :name "F1-check")
48     (search :device ?dev :target ?tar)
49     (data :field ?tar :hostname ?dev :flag FALSE :judge FALSE) = ?data
50     -->
51     (modify ?data:flag TRUE)
52     (modify ?data:judge TRUE)

```

```

53     )
54
55     (rule match_attribute_hostname
56         (state :name "F1-check") = ?state
57         (search :device ?dev :target ?tar1)
58         (data :description ?tar2 :hostname ?dev :flag FALSE :judge FALSE) = ?data
59         (!= ?tar1 ?tar2)
60         -->
61         (control judge(?tar1 ?tar2))
62         (modify ?data:judge JUDGING)
63         (modify ?state:name "F1-judge")
64     )
65
66     (rule match_accept_target
67         (state :name "F1-check") = ?state
68         (accept :target ?dev1)
69         (search :device ?dev1 :target ?tar1)
70         (data :description ?tar2 :hostname ?dev2 :flag FALSE :judge FALSE) = ?
71         data
72         (!= ?tar1 ?tar2)
73         (!= ?dev1 ?dev2)
74         -->
75         (control judge(?tar1 ?tar2))
76         (modify ?data:judge JUDGING)
77         (modify ?state:name "F1-judge")
78     )
79
80     (rule check_all_data_1
81         (state :name "F1-check") = ?state
82         (search :device ?dev)
83         (data :hostname ?dev :flag TRUE) = ?data
84         (data :hostname ?dev :flag FALSE :judge FALSE)
85         -->
86         (modify ?state:name "F2")
87     )
88
89     (rule check_all_data_2
90         (state :name "F1-check") = ?state
91         (search :device ?dev)
92         (accept :target ?dev)
93         (data :flag TRUE) = ?data
94         (data :flag FALSE :judge FALSE)
95         -->
96         (modify ?state:name "F2")
97     )
98
99     (rule find_NONE_data
100         (state :name "F1-check") = ?state
101         (search)
102         (data :index NONE)
103         -->
104         (modify ?state:name "F2")
105     )
106
107     (rule do_not_search
108         (state :name "F1-check")
109         (search :device ?dev)
110         (data :hostname ?dev :flag FALSE :judge TRUE) = ?data
111         (data :hostname ?dev :flag FALSE :judge FALSE)
112         (notification :from ?agName :departure ?dep)
113         (Status :name ?myName)
114         -->
115         (send
116             :performative "processing failure"
117             :to ?agName
118             :arrival ?dep
119             :content (failed :by ?myName)
120         )
121     )
122     (terminate :at now)

```

```

121 )
122
123 (rule receive_judgement_result
124   (state :name "F1-judge") = ?state
125   (data :flag FALSE :judge JUDGING) = ?data
126   (Msg :performative __Event :content (?cont)) = ?msg
127   -->
128   (modify ?data:flag ?cont)
129   (modify ?data:judge TRUE)
130   (modify ?state:name "F1-check")
131   (remove ?msg)
132 )
133 )

```

#### 1.4. F2: Secondary Request Generation

Listing S4 is the rule set for F2 of secondary request generation is a part of request expansion property mentioned in Section 2.1.2 in the main text. The 6th or 74th line is the rule for calling the next process in the processing sequence of the secondary request generation function. When this rule fires, it sends a message whose performative attribute is “task notification” to the repository and waits for 10 seconds. Then, when the data processing agent wants to provide additional information to the user, the rule on line 117 is executed. In this rule, if there is a `relatedword` object, it rewrites the information of the target device of the request, sends a message whose performative attribute is “request notification” in the repository, and waits for 10 seconds. After that, we move to the rule that executes the function of F3.

Listing S4: Rule set file of F2

```

1 (rule-set F2Rules
2
3   (property)
4   (initial_facts)
5
6   (rule send_task_notification_1
7     (data :unit ?unit :flag TRUE) = ?data
8     (next :process ?proc :data ?unit :flag FALSE) = ?next
9     (next :process ?proc :flag TRUE)
10    (search) = ?search
11    (process :flag TRUE)
12    (repository :name ?name)
13    (state :name "F2_1") = ?state
14    -->
15    (modify ?search:action ?proc)
16    (send
17      :performative "task notification"
18      :to _broadcast
19      :arrival ?name
20      :content ?search
21    )
22    (modify ?state:name "F2_1-wait")
23    (modify ?next:flag TRUE)
24    (alarm :after 10000 :content ())
25  )
26
27  (rule send_task_notification_2
28    (data)
29    (next :process ?proc :flag FALSE) = ?next
30    (search) = ?search
31    (process :flag TRUE)
32    (repository :name ?name)
33    (state :name "F2_1") = ?state
34    -->
35    (modify ?search:action ?next:process)
36    (send
37      :performative "task notification"

```



```

38         :to _broadcast
39         :arrival ?name
40         :content ?search
41     )
42     (modify ?state:name "F2_1-wait")
43     (modify ?next:flag TRUE)
44     (alarm :after 10000 :content ())
45 )
46
47 (rule send_task_notification_3
48     (data :unit ?unit :flag TRUE) = ?data
49     (search) = ?search
50     (process :flag FALSE)
51     (repository :name ?name)
52     (state :name "F2_1") = ?state
53     -->
54     (send
55         :performative "task notification"
56         :to _broadcast
57         :arrival ?name
58         :content ?search
59     )
60     (modify ?state:name "F2_1-wait")
61     (remove ?search)
62     (alarm :after 10000 :content ())
63 )
64
65 (rule send_task_notification_4
66     (data :unit ?unit :flag TRUE)
67     (next :process ?proc :data ?unit :flag FALSE) = ?next
68     (next :process ?proc :flag TRUE)
69     (process :flag TRUE)
70     (repository :name ?name)
71     (state :name "F2_1")
72     -->
73     (modify ?next:flag TRUE)
74 )
75
76 (rule receive_alarm_1
77     (Msg :performative __Alarm) = ?alarm
78     (state :name "F2_1-wait") = ?state
79     -->
80     (modify ?state:name "F2_1")
81     (remove ?alarm)
82 )
83
84 (rule no_task_1
85     (data :unit ?unit :flag TRUE) = ?data
86     (next :data ?unit :flag TRUE)
87     (next :data ?unit :flag FALSE)
88     (process :flag TRUE)
89     (state :name "F2_1") = ?state
90     -->
91     (modify ?state:name "F2_2")
92 )
93
94
95 (rule no_task_2
96     (data)
97     (state :name "F2_1") = ?state
98     -->
99     (modify ?state:name "F2_2")
100 )
101
102 (rule no_task_3
103     (data)
104     (data :index NONE)
105     (process :flag FALSE)
106     (state :name "F2_1") = ?state

```

```

107     -->
108     (modify ?state:name "F2_2")
109 )
110
111 (rule no_task_4
112   (next)
113   (state :name "F2_1") = ?state
114   -->
115   (modify ?state:name "F2_2")
116 )
117
118 (rule send_related_word
119   (search :device ?dev :relevance ?rel) = ?req
120   (relatedword :wordA ?dev :wordB ?wordB :relevance ?rel2) = ?relword
121   (repository :name ?name)
122   (state :name "F2_2") = ?state
123   -->
124   (bind ?nextRel (* ?rel ?rel2))
125   (send
126     :performative "request notification"
127     :to _broadcast
128     :arrival ?name
129     :content (search :device ?wordB :target ?req:target :action ?req:action
130               :relevance ?nextRel :from DSAgent)
131   )
132   (alarm :after 10000 :content ())
133   (remove ?relword)
134   (modify ?state:name "F2_2-wait")
135 )
136
137 (rule receive_alarm_2
138   (Msg :performative __Alarm) = ?alarm
139   (state :name "F2_2-wait") = ?state
140   -->
141   (modify ?state:name "F2_2")
142   (remove ?alarm)
143 )
144
145 (rule send_all_related_words_1
146   (search :device ?dev)
147   (relatedword :wordA ?dev :wordB ?wordB :relevance ?rel2)
148   (state :name "F2_2") = ?state
149   -->
150   (modify ?state:name "F3")
151 )
152
153 (rule send_all_related_words_2
154   (search :device ?dev)
155   (relatedword :wordA ?dev :wordB ?wordB :relevance ?rel2)
156   (state :name "F2_2") = ?state
157   -->
158   (modify ?state:name "F3")
159 )

```

### 1.5. F3: Data Processing Sequence Organization

Listing S5 is the rule set for F3: data processing sequence organization. In this rule set, the rule on line 77 is a rule that fires when there is a message addressed to itself with the performative attribute of “task notification”. When this rule fires, the procedures necessary for cooperating with other agents, described from lines 105 to 169, are executed. If there is no message addressed to itself with a performative attribute of “task notification,” the rule on line 94 fires and this data processing agent disappears from the operating environment.

Listing S5: Rule set file of F3

```

1 (rule-set F3Rules
2
3   (property)
4   (initial_facts)
5
6   (rule receive_request_acceptance
7     (Msg
8       :performative "request acceptance"
9       :to ?myName
10      :content (accepted :by ?agName)
11    ) = ?msg
12    (Status :name ?myName)
13    (notification :from ?notiName :departure ?dep)
14    -->
15    (make (receiver :name ?agName))
16    (send
17      :performative "request acceptance"
18      :to ?notiName
19      :arrival ?dep
20      :content ?msg:content
21    )
22    (remove ?msg)
23  )
24
25  (rule receive_file_path_1
26    (Msg
27      :performative "sending file path"
28      :to ?myName
29      :content (file :path ?path :by ?agName)
30    ) = ?msg
31    (Status :name ?myName)
32    (receiver :name ?agName) = ?reciever
33    (notification :from ?notiName :departure ?dep)
34    -->
35    (send
36      :performative "sending file path"
37      :to ?notiName
38      :arrival ?dep
39      :content ?msg:content
40    )
41    (remove ?reciever)
42    (remove ?msg)
43  )
44
45  (rule receive_processing_failure
46    (Msg
47      :performative "processing failure"
48      :to ?myName
49      :content (failed :by ?agName)
50    ) = ?msg
51    (Status :name ?myName)
52    (receiver :name ?agName) = ?reciever
53    (notification :from ?notiName :departure ?dep)
54    -->
55    (send
56      :performative "processing failure"
57      :to ?notiName
58      :arrival ?dep
59      :content ?msg:content
60    )
61    (remove ?reciever)
62    (remove ?msg)
63  )
64
65  (rule receive_all_file
66    (Msg
67      :performative "request acceptance"
68      :to ?myName

```

```

69     )
70     (receiver :name ?agName)
71     (data :index NONE)
72     (state :name F3) = ?state
73     -->
74     (modify ?state:name "processing")
75 )
76
77 (rule receive_task_acceptance
78     (Msg
79         :performative "task acceptance"
80         :to ?myName
81     ) = ?msg
82     (Status :name ?myName)
83     (type) = ?type
84     -->
85     (make (accepter :name ?msg:from :flag FALSE))
86     (reply
87         :performative "setting information"
88         :to ?msg
89         :content ?type
90     )
91     (remove ?msg)
92 )
93
94 (rule no_task_acceptance_1
95     (Msg
96         :performative "task acceptance"
97         :to ?myName
98     )
99     (Status :name ?myName)
100     (Msg :performative __Alarm)
101     -->
102     (alarm :after 3000 :content ())
103 )
104
105 (rule receive_setting_information_1
106     (Msg
107         :performative "setting information"
108         :to ?myName
109         :content ?cont
110     ) = ?msg
111     (Status :name ?myName)
112     (accepter)
113     (state :name "F3") = ?state
114     -->
115     (print ?cont)
116     (control setup(?cont:host ?cont:port ?cont:topic))
117     (reply
118         :performative "ready"
119         :to ?msg
120         :content ()
121     )
122     (modify ?state:name "processing")
123     (remove ?msg)
124 )
125
126 (rule receive_setting_information_2
127     (Msg
128         :performative "setting information"
129         :to ?myName
130         :content ?cont
131     ) = ?msg
132     (Status :name ?myName)
133     (accepter :flag TRUE)
134     (accepter :flag FALSE)
135     (Msg :performative "request acceptance")
136     (state :name "F3") = ?state
137     -->

```

```

138     (control setup(?cont:host ?cont:port ?cont:topic))
139     (reply
140      :performative "ready"
141      :to ?msg
142      :content ()
143     )
144     (modify ?state:name "processing")
145     (remove ?msg)
146 )
147
148 (rule receive_ready
149   (Msg
150    :performative "ready"
151    :from ?agName
152    :to ?myName
153   ) = ?msg
154   (Status :name ?myName)
155   (accepter :name ?agName :flag FALSE) = ?accepter
156   -->
157   (modify ?accepter:flag TRUE)
158   (remove ?msg)
159 )
160
161 (rule receive_all_ready
162   (accepter :flag TRUE)
163   (accepter :flag FALSE)
164   (Msg :performative "request acceptance")
165   (data)
166   (state :name "F3") = ?state
167   -->
168   (modify ?state:name "processing")
169 )
170 )

```

### 1.6. Post processing rules

Listing S6 is the rule set for post processing. Post processing defines the processing to be executed after the DPA processing task is completed. Line 6 is a rule that fires when the agent receives the data processing result from the agent that received the data output by itself. Here, the path to the file containing the processed results is passed to the base process. Line 21 is a rule that fires when processing results are passed from all agents that have received data output by itself. This rule aggregates the processing results and passes them to the other data processing agent that called it or his UI agent. When all post-processing is completed, the data processing agent disappears from the operating environment.

Listing S6: Rule set file of post processing

```

1 (rule-set PostProcessingRules
2
3   (property)
4   (initial_facts)
5
6   (rule receive_file_path_2
7     (Msg
8      :performative "sending file path"
9      :from ?agName
10     :to ?myName
11     :content (file :path ?filePath)
12    ) = ?msg
13    (accepter :name ?agName :flag TRUE) = ?accepter
14    (Status :name ?myName)
15    -->
16    (control receiveFilePath(?filePath))
17    (modify ?accepter:flag END)
18    (remove ?msg)
19  )
20 )

```

```

21 (rule send_result_1
22     (accepter :flag END)
23     (accepter :flag TRUE)
24     (notification)= ?noti
25     (file) = ?file
26     -->
27     (control makeResultFile())
28     (send
29         :performative "sending file path"
30         :to ?noti:from
31         :arrival ?noti:departure
32         :content ?file
33     )
34     (remove ?file)
35     //(terminate :at now)
36 )
37
38 (rule send_result_2
39     (receiver)
40     (Msg :performative __Alarm :content (request :acceptance END))
41     (RWAlarm)
42     (data :index NONE)
43     (notification) = ?noti
44     (file) = ?file
45     -->
46     (control makeResult())
47     (send
48         :performative "sending file path"
49         :to ?noti:from
50         :arrival ?noti:departure
51         :content ?file
52     )
53     //(terminate :at now)
54 )
55
56 )

```

## 2. Commonly Used Class Files in Base Process

### 2.1. Data Proces Module

Listing S7 is a class file to define data process module which is commonly used in the base processes.

Listing S7: Class file for base process

```

1 public class DataProcessingAgent implements DashBP {
2     protected DashAgent dashAgent = null;
3     protected Worker worker = null;
4     protected ZmqSub input = null;
5     protected ZmqPub output = null;
6
7     protected Thread th = null;
8     protected List<String> pathList;
9     protected String url, path = "";
10
11     @Override
12     public void setAgent(DashAgent dashAgent) {
13         this.dashAgent = dashAgent;
14         output = OutputSetting.outputSetting(dashAgent);
15         pathList = new ArrayList<>();
16     }
17
18     @Override
19     public void finalizeBP() {
20         if (worker != null) {
21             worker.stop();
22             worker.dispose();
23         }
24     }
25 }

```

```

24     }
25
26     @Override
27     public void run() {
28         if (worker != null) {
29             worker.start();
30         }
31     }
32
33     public void start() {
34         th = new Thread(this);
35         th.start();
36     }
37
38     public void stop() {
39         finalizeBP();
40     }
41
42     public void makePathName(Object[] obj) {
43         makePathName(String.valueOf(obj[0]), String.valueOf(obj[1]));
44     }
45
46     private void makePathName(String basePath, String num) {
47         String resultPath = basePath + num + ".html";
48         setPath(resultPath);
49         dashAgent.raiseEvent("file :path\\" + resultPath + "\\");
50     }
51
52     public void setURL(Object[] obj) {
53         setURL(String.valueOf(obj[0]));
54     }
55
56     public void setURL(String url) {
57         this.url = url;
58     }
59
60     public void setPath(Object[] obj) {
61         setPath(String.valueOf(obj[0]));
62     }
63
64     public void setPath(String path) {
65         this.path = path;
66     }
67
68     public void setup(Object[] obj) {
69         try {
70             String address = InetAddress.getByName(String.valueOf(obj[0])).
71                 getHostAddress();
72             int port = Integer.parseInt(String.valueOf(obj[1]));
73             String agentName = String.valueOf(obj[2]);
74             setup(address, port, agentName);
75         } catch (UnknownHostException e) {
76             e.printStackTrace();
77         }
78     }
79
80     private void setup(String address, int port, String agentName) {
81         input = new ZmqSub(new ZmqSetting(ZmqProtocols.TCP, address, port,
82             false), agentName);
83     }
84
85     public void receiveFilePath(Object[] obj) {
86         receiveFilePath(String.valueOf(obj[0]));
87     }
88
89     private void receiveFilePath(String filePath) {
90         pathList.add(filePath);
91     }

```

```

91     public void judge(Object[] obj) {
92         judge(String.valueOf(obj[0]), String.valueOf(obj[1]));
93     }
94
95     public void judge(String word1, String word2) {
96         Tokenizer tokenizer = new Tokenizer();
97         List<Token> tokens = tokenizer.tokenize(word1);
98         int counter = 0;
99         int match = 0;
100         for (Token token : tokens) {
101             String surface = token.getSurface();
102             String speechLevel = token.getPartOfSpeechLevel1();
103             if (speechLevel.equals("noun")) {
104                 if (word2.contains(surface)) {
105                     match++;
106                 }
107                 counter++;
108             }
109         }
110
111         if (match == counter) {
112             dashAgent.raiseEvent("TRUE");
113         } else {
114             double border = (1 - (match / counter)) / 2.0;
115             Random rand = new Random();
116             double value = rand.nextDouble();
117             if (border <= value) {
118                 dashAgent.raiseEvent("TRUE");
119             } else {
120                 dashAgent.raiseEvent("FALSE");
121             }
122         }
123     }
124
125     public void judgeProcess(Object[] obj) {
126         judgeProcess(String.valueOf(obj[0]), Double.valueOf(String.valueOf(
127             obj[1])));
128     }
129
130     public void judgeProcess(String word, Double score) {
131         double border = 1.0 - score;
132         Random rand = new Random();
133         double value = rand.nextDouble();
134         String content = "process :type \"" + word + "\" :flag ";
135         if (border <= value) {
136             dashAgent.raiseEvent(content + "TRUE");
137         } else {
138             dashAgent.raiseEvent(content + "FALSE");
139         }
140     }
141
142     public void makeResultFile() {
143         try {
144             PrintWriter result_pr = new PrintWriter(new FileWriter(path));
145             String content_line;
146             for (String url : pathList) {
147                 BufferedReader content_br = new BufferedReader(new FileReader
148                     (url));
149                 while ((content_line = content_br.readLine()) != null) {
150                     result_pr.println(content_line);
151                 }
152                 content_br.close();
153             }
154             pathList.clear();
155             result_pr.close();
156         } catch (IOException e) {
157             e.printStackTrace();
158         }
159     }

```



158 | }

## 2.2. Data Proces Module

Listing S8 is a class file to define data-process module which is commonly used in the base processes. This class file uses the PostEvent method, which takes character strings or byte array data as arguments, to pass data from the receiving module to the data processing module and from the data processing module to the sending module.

Listing S8: Data process module

```

1 public class PostEvent extends EventObject {
2     private byte[] data;
3
4     public PostEvent(Object source, String message) {
5         super(source);
6         this.data = message.getBytes(Charset.forName("UTF-8"));
7     }
8
9     public PostEvent(Object source, byte[] data) {
10        super(source);
11        this.data = data;
12    }
13
14    public String getMessage(){
15        return new String(data,Charset.forName("UTF-8"));
16    }
17
18    public byte[] getData(){
19        return data;
20    }
21 }
```

## Reference

1. Uchiya, T.; Kinoshita, T. Design support functions for developing multiagent system on repository-based multiagent framework. *2011 10th IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI\*CC)* **2011**, 240–246.