

EXPORTING NATURAL LANGUAGE

GENERATING NL SENTENCES OUT OF S-BPM PROCESS MODELS



 Stephan Sneed, Consultant Metasonic AG

THE AUDIENCE FOR NATURAL LANGUAGE

	Internal	External
Human Interface	NL behavior description	SLA
System Interface	Requirement spec / Test plan	SLA

REQUIREMENTS TO AN NL EXPORT

SLAs

- The name representing the functionality
- The definition of the call or the requesting message, precondition
- The time constraint of that functionality to be completed
- The result generated by that functionality, postcondition
- The possible return status of the functionality or the replying message

TEST CASES

- Test case id
- Specification of the input data
- Specification of the output data
- Procedural requirements
- Dependencies on other test cases

USECASES

- The name of the use case or functionality
- The description of that use case or functionality
- The actor triggering the use case
- The system that provides the use case or functionality
- The relation of the actor to that use case or functionality
- The result of the Use Case
- The path that the use case should follow

NL export will be changed, enhanced, extended to generate other NL languages or, to say it with one word: it will be maintained.

Communication View

The subject “WFC” receives the message “GetWeatherByRegion” from the subject “TravelAgent01”.

The subject “WFC” receives the message “GetWeatherByCity” from the subject “TravelAgent01”.

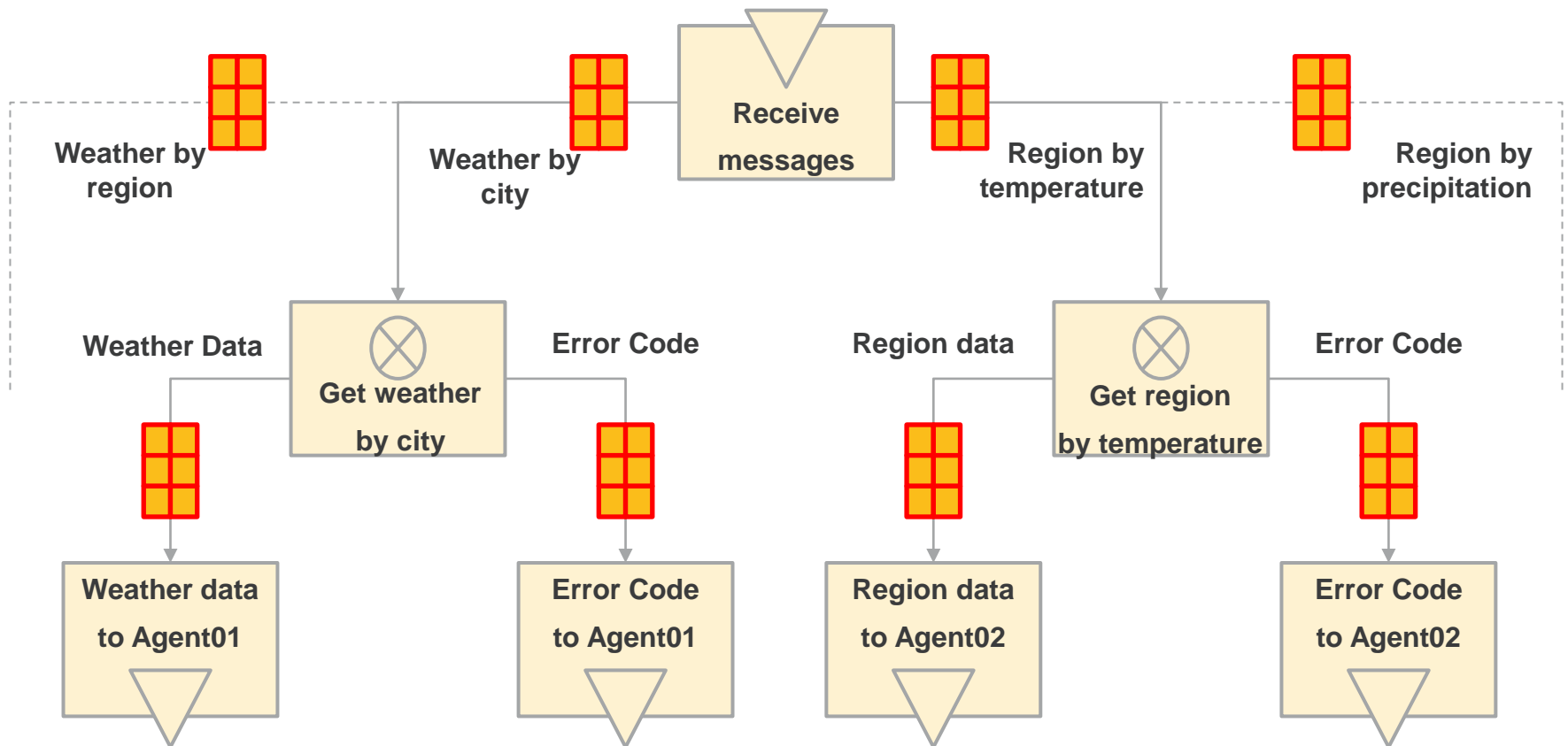
The subject “WFC” receives the message “GetRegionByTemp” from the subject “TravelAgent02”.

The subject “WFC” receives the message “GetRegionByPrecip” from the subject “TravelAgent02”.

Internal Behavior

ID	State Description
1-1	Subject “WFC” is waiting for incoming messages. The next state is 1.1-1
1.1-1	If the subject “WFC” receives the message “GetWeatherByRegion” from the subject “Agent01”, the subject “WFC” executes the action “GetWeatherByRegion”. The action “GetWeatherByRegion” demands the business object “ParamStructure01” as parameter and returns the business objects “WeatherByRegion” or “ErrorCode”. The next states are either 1.1.1-1 or 1.1.2-1.
1.1.1-1	If the action “GetWeatherByRegion” returns the business object “WeatherData”, the subject “WFC “ sends the message “WeatherData” to the subject “Agent01” along with the business object “WeatherData”. This path has ended.
1.1.2-1	If the action “GetWeatherByRegion” returns the business object “ErrorCode”, the subject “WFC” sends the message “ErrorCode” to the subject “Agent01” along with the business object “ErrorCode”. This path has ended.
1.2-1	If the subject “WeatherForeCast” receives the message “GetWeatherByCity”

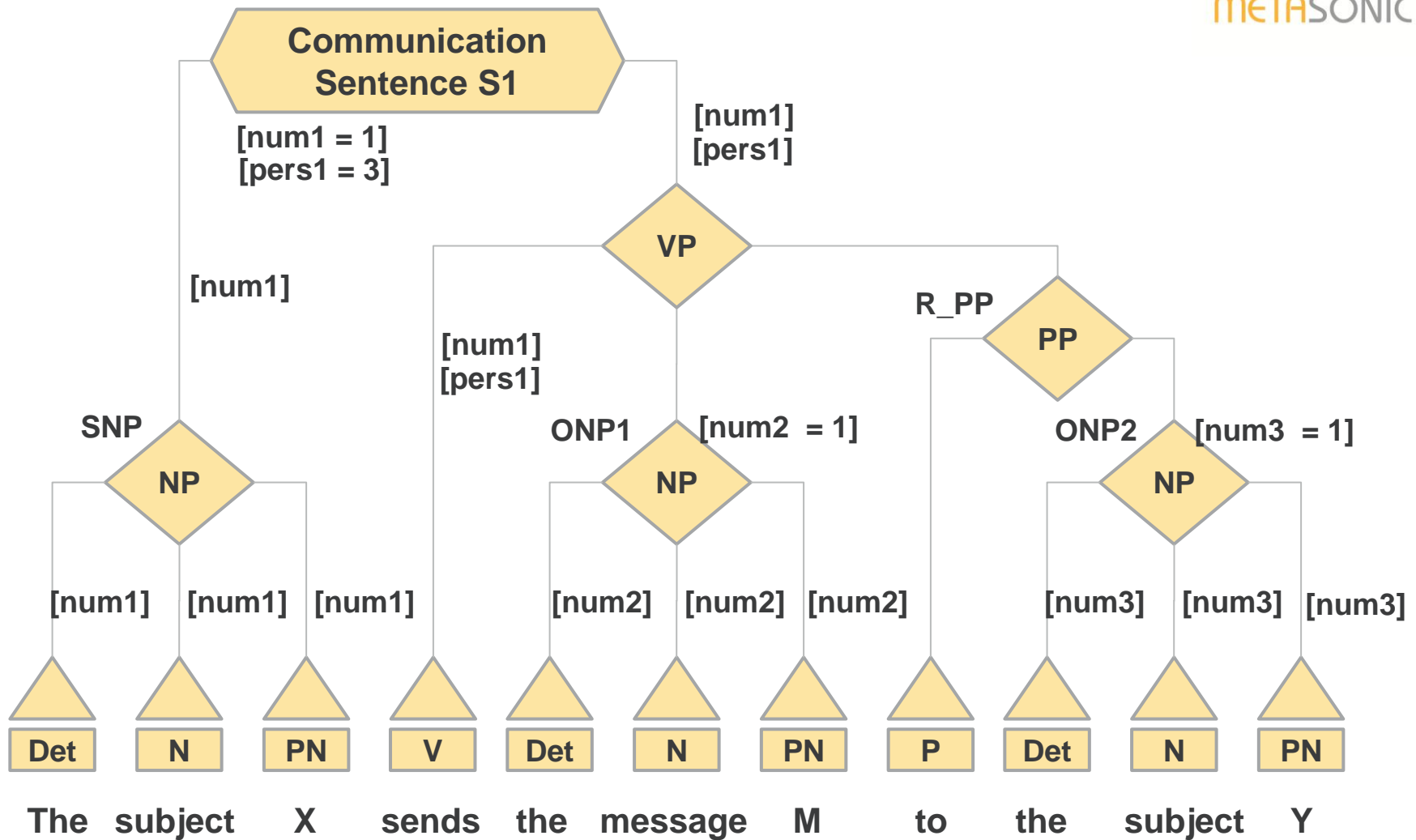
SERVICE SUBJECT EXAMPLE



STRUCTURE OF A NATURAL LANGUAGE SENTENCE



METASONIC



COMPUTER LINGUISTIC PHRASE DEFINITIONS

NP → Det, N, PN.

PP → P, NP (Det, N, PN).

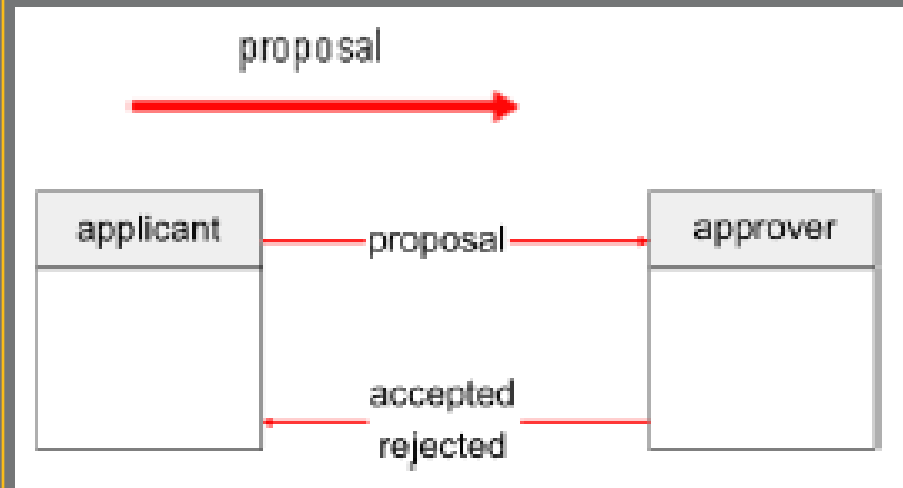
VP → V, NP (Det, N, NP), PP (P, NP (Det, N, PN))

S1 → NP (Det, N, PN), VP (V, NP (Det, N, PP), PP (P, NP (N, PN))).

MAPPING RULES (CV)

Outgoing messages:

noun1 → subject
properName1 → applicant
verb → send
noun2 → message
properName2 → proposal
preposition → to
noun3 → subject
properName3 → approver
num1 = num2 = num3 → sing



MAPPING ALGORITHM (CV)

```
for (all subjects) {
    currentSubject.getOutgoingMessages;
    for (all outgoingMessages) {
        noun1 = "subject";
        properName1 = currentSubject.getName();
        verb = "sends";
        noun2 = "message";
        properName2 = currentMessage.getName();
        preposition = "to";
        noun3 = noun1;
        properName3 = currentMessage().getRecipient().getName();
        CS1 = new CS1(noun1, properName1, verb, noun2, properName2,
            preposition, noun3, properName3, language);
        CS1.print();
    }
}
```

Action state after receive (Applicant):

noun1 → subject

properName1 → applicant

verb1 → receive

noun2 → message

properName2 → rejected

preposition → from

noun3 → subject

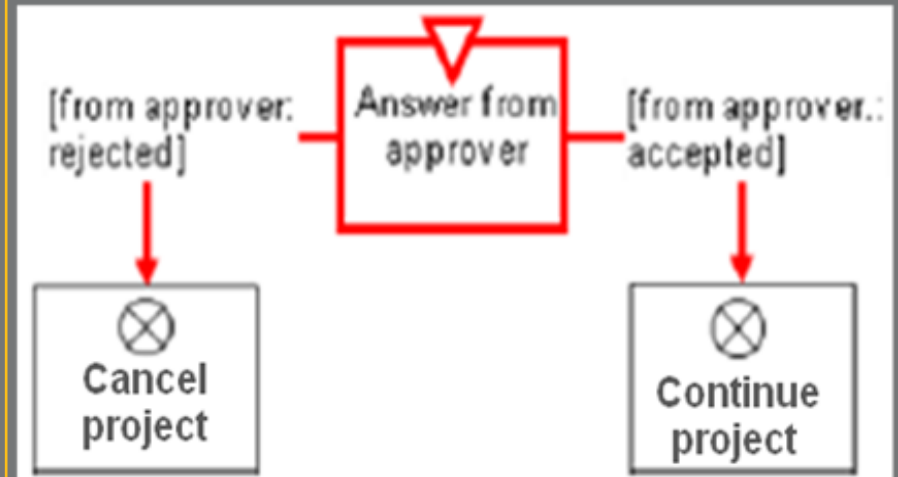
properName3 → approver

verb2 → executes

noun4 → action

properName4 → cancel project

num1 = num2 = num3 = num4 → sing



MAPPING ALGORITHM (CV)



METASONIC

```
public void recFunc (node, currentSubject) {
    stack = node.getOutgoingEdges();
    for (stack>0) {
        edge = stack.pop();
        noun1 = "subject";
        properName1 = currentSubject.getName();
        verb1 = "receive";
        noun2 = "message";
        properName2 = edge.getMessage().getName();
        preposition = "from";
        noun3 = subject;
        properName3 = edge.getMessage().getSender().getName();
        nextNode = edge.getDestination();
        noun4 = nextNode.getType()
        properName4 = nextNode.getName();
        recFunc (nextNode, currentSubject);
    }
}
```

Thank you for your attention.

© copyright Metasonic AG, Pfaffenhofen, 2010

All rights reserved.

The content of this document is subject to copyright.

Any changes, modifications, additions or amendments require prior written consent from Metasonic AG, Pfaffenhofen. Reproduction in any form is only permitted on the condition that the copyright notice remains on the actual document.

Publication or translation in any form requires prior written consent from Metasonic AG, Pfaffenhofen.

„Metasonic“, „Metasonic Suite“, „Metasonic Base“, „Metasonic Build“, „Metasonic Proof“, „Metasonic Flow“, „Metasonic Academy“ and „S-BPM“ are trademarks or registered trademarks of Metasonic AG in Germany and in many other countries worldwide.

All other trademarks are the property of their respective owners.

Some software products marketed by Metasonic AG and its distributors can contain proprietary software components of other software vendors. These materials are subject to change without notice. The materials are provided by Metasonic AG for informational purposes only, without representation or warranty of any kind, and Metasonic AG shall not be liable for errors or omissions with respect to the materials. The only warranties for Metasonic products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

