# XML based Extended Super-function Schema in Knowledge Representation

Qiong Liu, Xin Lu, Fuji Ren and Shingo Kuroiwa

The University of Tokushima,
2-1 Minami Josanjima, Tokushima, Japan 770-8506
{liuqiong,luxin,ren,kuroiwa}@is.tokushima-u.ac.jp

**Abstract.** In recent years, the usual knowledge representation (KR) problem in artificial intelligence is how to automatically represent and transform different kinds of knowledge using one kind of schema. Especially this problem focuses on representing formal knowledge in natural language for human understanding. For this purpose, this paper proposes an extended super-function (ESF) schema to build a novel KR system. This system can translate the data of stock market or other fields into the corresponding natural language expression automatically. Moreover, this system benefits from XML techniques which formalize and construct all information using the common Web rules to realize the ESF schema.

## 1 Introduction

In artificial intelligence (AI), knowledge representation (KR) includes two basic kinds of knowledge objects (formal objects and natural objects) in its fundamental conception. Formal objects like mathematical entities can be captured exactly and precisely by machine because of their formality. Natural objects like natural language entities can be understood easily and commonly by human being through their flexibility. Then KR provides the representation function to deal with the correspondences between the formal objects and the natural objects, acting as surrogates in the real world as well as in the machine space. Given the relationship with human and machine is made closer, the last role of KR will become more significant and necessary.

In recent years, the KR technique has shown its superiority in knowledge collection and organization. The natural objects in knowledge base have been organized in highly structured form to satisfy the requirements that people wish to understand and master various kinds of knowledge easily by natural objects. The KR system based on natural objects is most sophisticated, and its construction is depended on some kinds of logic. Different formal methods (such as predicate logic, fuzzy logic, semantic networks, frames and related techniques) have been developed to represent natural objects. They also have been used by expert systems frequently in decision making and reasoning. The KR system [1–5] based on natural objects has been implemented in almost every aspect such as weather forecast, letters response, network analysis, disease diagnosis and so on. All of these systems have been recognized as the considerable enhancement of KR technique for natural objects management. Because natural objects like human natural language are complex, irregular, and diverse, the previous KR systems based on natural objects just act as interfaces to knowledge base, which perform formal tasks

*© A. Gelbukh, R. Monroy. (Eds.)*
*Advances in Artificial Intelligence Theory*
*Research on Computing Science 16, 2005, pp. 3-12*

separated from nature objects processing. Nevertheless the computational characteristic of representation and inference in natural objects can improve the efficiencies of processing all tasks in the KR system. Therefore, the extended super-function (ESF) schema is proposed to build a novel KR system which processes vast amounts of knowledge systematically like machine and logically, deeply like human being. It is capable to incorporate both the formal objects and the natural objects. We think the ESF schema is a new direction in KR technique.

The paper is structured as follow. Section 2 describes the grammar of ESF schema for KR. In Section 3, the ESF based KR system is realized as example, where the ESF schema is utilized to produce technical report of stock market from data for non-expert user. Finally, Section 4 presents a discussion and conclusion of this paper.

## 2 Extended Super-function Schema

In ESF schema definition, symbol set includes formal objects and natural objects. Natural language is a symbol set, mathematic expression is a symbol set, music is a symbol set, and so on. The ESF schema is applied to the translation from one symbol set to another more variously than from one kind of natural language to another as SF.

An ESF is a symbol set that denotes the correspondence between source symbol patterns and target symbol patterns. The definition of symbol pattern is most necessary for ESF, and it will be described firstly. Then we give ESF a definition.

**Definition 1**: A team of a token $n$, some attributes $A$ and corresponding values $v$ can form a symbol pattern $p_n ::= p[n,\ A : v]$.

**Definition 2**: A symbol pattern $p$ can be sorted into source pattern $ps$ and target pattern $pt$ according to its origin. A set of source pattern $ps$ and a set of target pattern $pt$ can aggregate a source pattern set $Ps$ and a target pattern set $Pt$ respectively.

**Definition 3**: A function from $Ps_1 \times \cdots \times Ps_{n-1}$ to $Ps_n$ is a subset $fs$ of the Cartesian product $Ps_1 \times \cdots \times Ps_n$, such that for each pair $(ps_1, \ldots, ps_{n-1})$ in $Ps_1 \times \cdots \times Ps_{n-1}$, there is a unique $ps_n$ in $Ps_n$ such that the ordered pair $(ps_1, \ldots, ps_n)$ is in $fs$. The source pattern $ps_n$ can be described as $ps_n ::= ps[n,\ A : fs]$

**Definition 4**: A function from $Pt_1 \times \cdots \times Pt_{n-1}$ to $Pt_n$ is a subset $ft$ of the Cartesian product $Pt_1 \times \cdots \times Pt_n$, such that for each pair $(pt_1, \ldots, pt_{n-1})$ in $Pt_1 \times \cdots \times Pt_{n-1}$, there is a unique $pt_n$ in $Pt_n$ such that the ordered pair $(pt_1, \ldots, pt_n)$ is in $ft$. The source pattern $pt_n$ can be described as $pt_n ::= pt[n,\ A : ft]$

**Definition 5**: A function from $Ps$ to $Pt$ is a subset $r$ of the Cartesian product $Ps \times \cdots \times Pt$, such that for each $ps$ in $Ps$, there is a unique $pt$ in $Pt$ such that the ordered pair $(ps, pt)$ is in $r$.

**Definition 6**: A set of function $fs$ can aggregate a source function set $Fs$. A set of function $fs$ can aggregate a target function set $Ft$ in the same way. A set of function $r$ can be considered as relation set $R$. $R$ means the translations between source patterns and target patterns.

In ESF, the based element is atomic pattern. Any symbol pattern, whose value is obtained from function, can be defined as a complex pattern. Its value is composed of some atomic patterns or other complex patterns ordered in function structures. Therefore, these symbol patterns are not ordered in one layer, they are ordered like net by functions. The grammar of ESF can be set as a five-tuple.

$$< Ps,\ Pt,\ Fs,\ Ft,\ R >$$

## 3 KR System Construction using ESF Schema

We can master the ESF based KR system from three layers (i.e., abstracting layer, describing layer and implementing layer). The three layers is used for understanding, detailing and coding the KR system respectively. In this section we specify KR system in these three layers for understanding.

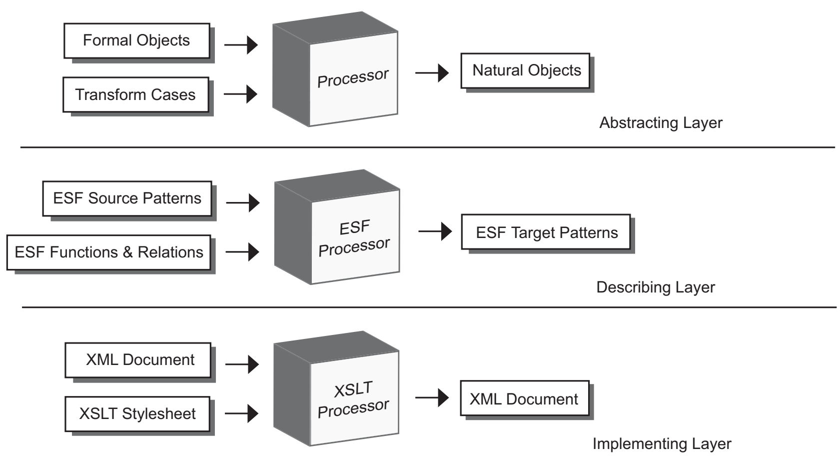


**Fig. 1.** Fundamental structure of the KR system based on ESF schema

As shown in the top layer of Figure 1, the processor receives the formal objects with the transform cases, and generates the natural objects. This is a pipeline mechanism. Here, the transform cases are regarded as principles for guiding processor what to do and how to do. They are the crucial parts of KR system.

### 3.1 Transform Case Extraction in Abstracting Layer

There are hundreds s of indicators in use today. Every technical indicator can be regarded as one transform case in the abstracting layer of our KR system. The technical indicator data and its perspective can be considered as the formal objects and the natural objects of transform case respectively. In this paper, Moving average convergence

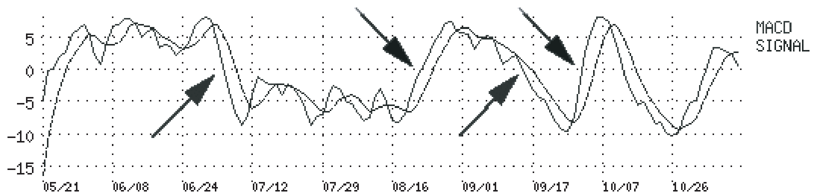


**Fig. 2.** Moving Average Convergence Divergence i)Upper-oriented arrows marks bearish centerline crossover. ii) Lower-oriented arrows marks bullish centerline crossover.

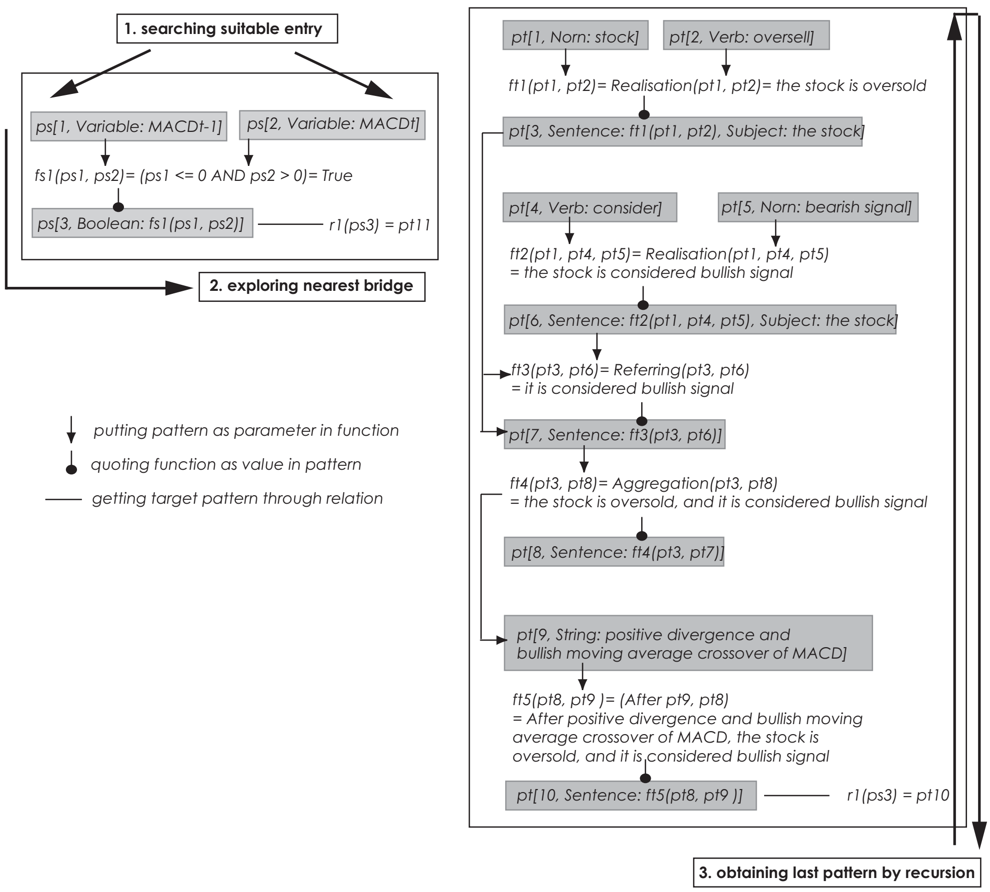


**Fig. 3.** ESF structures of MACD indicator (source patterns and source functions (left), target patterns and target functions (right))

divergence (MACD) is extracted as a transform cases from our KR system for clear specification.

A bullish centerline crossover occurs when MACD moves above the zero line and into positive territory. This is a clear indication that momentum has changed from negative to positive or from bearish to bullish. After a positive divergence and bullish moving average crossover, the bullish centerline crossover can act as a confirmation signal. Conversely, after a negative divergence and bearish moving average crossover, the bearish centerline crossover can act as a confirmation signal. (The MACD chart is illustrated in Figure 2). Then we can get a transform description of MACD:

```
IF MACD(Day)≥0 AND MACD(Day-1)<0 THEN "after a positive divergence
and bullish moving average crossover of MACD, the bullish centerline
crossover can act as a confirmation signal”
IF MACD(Day)≤0 AND MACD(Day-1)>0 THEN "after a negative divergence
and bearish moving average crossover of MACD, the bearish centerline
crossover can act as a confirmation signal”
```

### 3.2 ESF Structure Design in Describing Layer

From abstracting layer to describing layer, we utilize the ESF schema to design the ESF structure of our KR system detailedly. All non-linguistic inputs are defined as ESF source pattern; the transform cases are generalized as ESF functions and relations for reuse. The ESF processor can accelerate the map between the source patterns and the target patterns through the functions and relations as illustrated in the middle layer of Figure 1.

Here a transform case (i.e., MACD indicator mentioned above) has been generalized as corresponding functions and relations by ESF schema in Figure 3 as an example. In this example every block denotes a pattern, such as atomic or complex one. Every relation can be considered as a translation bridge between numerable source patterns and single target pattern. Every function acts as telephone line which is connecting every user – pattern. We need to emphasize that the process from pattern to function means putting pattern into function as parameter. On the other hand, the process from function to pattern is quoting function as value in pattern. These two processes will be coded in computer language for the actual experiment in following section. The ESF process consists of three steps. They are marked in Figure 3.

1. **Searching suitable entry:** We put some source patterns, which will be translated into natural language, into processor. Then the processor searches source functions' parameters as the suitable entries for these source patterns. Here the ”suitable” denotes the pattern's domain is consistent with a parameter's domain or contained by it. In this example, if the source patterns include other technical indicator information except the $MACD$, it is difficult for processor to find the suitable entries.

2. **Exploring nearest bridge:** After searching suitable entry, the processor explores downwards, passes numerable functions and finds a nearest relation. Then the processor can obtain a corresponding target pattern through the relation bridge. There is a necessary condition in this process. It is all of the numerable functions' parameters (from suitable entry to nearest relation) should be filled by source patterns. In this example, if there is only the $MACD(Day)$ pattern $ps_1$ and not the $MACD(Day-1)$

pattern $ps_2$, the processor cannot find the relation $r_1$.

3. **Obtaining last pattern by recursion:** If the processor gets the corresponding target pattern, it will use the correlative target functions structured recursively to get the corresponding target pattern's value.

ESF based KR system is a new KR system that has the characters between template-based and standard KR. This is not only because the ESF based KR system combines standard KR with templates, but also because it tends to use syntactically structured templates (here the "template" in ESF schema is a function. It has more changes than the common template), and allows the gaps as parameters in them to be filled recursively (i.e., by filling a gap, a new gap may result). The ESF based KR system can use grammars to aid linguistic realization. For example, in Figure 5, it includes lexical items (e.g., referring expression function, aggregation function and so on) which always exist in standard KR. Therefore, it is difficult to give a definition of "template based" for our KR system. We think the word – "function based" is more suited than "template based".

### 3.3 ESF Structure Realization in Implementing Layer

It is necessary to utilize a ready-made and convenient technique to realize ESF schema in KR systems. We regard the extensible markup language (XML) is the best choice to define the ESF pattern. We also select extensible stylesheet language transformation (XSLT) to describe the ESF function and relation. This section helps the reader master and apply these ideas to KR problems. We utilize the XML and XSL techniques to realize ESF schema for building a KR system. As shown in the bottom layer of Figure 1 , in a KR system the ESF patterns in XML is fed into the XSLT processor as one input, and the functions and relations in XSL is provided as a second input. The output is then sent directly to user as a stream of HTML, XML or other formats. The ESF functions and relations in XSL generate the transformation instructions about ESF patterns, and the patterns in XML provide raw data. It is the implementing layer of the KR system based on ESF schema.

**Pattern construction using XML:**

Code 1 – ESF source patterns

```
<stock symbol="TOPIX" name="Tokyo Stock Exchange Prices Index">
...
<date="2004-08-20" open="1105.08" close="1109.84" volume="1072250000"
MACD="-1.8351"/>
<date="2004-08-23" open="1115.93" close="1114.24" volume="1047230000"
MACD="0.1509"/>
<date="2004-08-24" open="1116.74" close="1116.60" volume="1065260000"
MACD="1.5825"/>
...
</stock>
```

We begin with an XML document that represents a portion of quotations about MACD in stock market, which is shown above. The XML elements include their attributes with their respective values within the element's start tag. Because the ESF pattern's attribute has the same name-value form, it can map the corresponding XML element's attribute with value. Therefore, all XML elements can be considered as the ESF patterns.

**Function and relation generation using XSLT:**

Here the centric problems are how to build the XSLT stylesheet for realizing ESF functions and relations, then how to process XML documents including ESF patterns by the XSLT stylesheet. In this paper we utilize XSLT templates to do this work. In Code 2, 3 and 4, we write some templates for realizing the ESF functions and relations of the MACD transform case (shown in Figure 3), and process the XML document mentioned above by any XSLT processor.

The XSLT templates are always written in one XSLT file together. We separate it into three pieces in Code 2, 3 and 4 as their corresponding ESF roles for understanding easily. Code 2, 3 and 4 respectively show source functions set, relations set, and target functions set. The process of ESF is from "searching suitable entry in source functions set, to "exploring nearest bridge" in relation set, then to "obtaining last pattern by recursion" in target function set. For this example we describe detailed technological process as following:

**1** Source patterns extraction in Code 2 – (Searching suitable entry)

Code 2 – ESF source function

```
<xsl:template name="fs1" match="MACD">
<xsl:variable name="ps1"/>
<xsl:value-of select="//ps1@value"/>
</xsl:variable>
<xsl:variable name="ps2">
<xsl:value-of select="//ps2@value"/>
</xsl:variable>
<xsl:choose>
<xsl:when test="$ps1 &lt;= 0 and $ps2 &gt; 0">
<xsl:value-of select="1"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="0"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

Put two patterns $ps_1,\ ps_2$ into source function $fs_1\ (ps_1,\ ps_2) = (ps_1 \leq 0, ps_2 > 0)$, and quote the function value as new pattern $ps_3$. The obtained patterns $ps_3$ are regarded as source patterns.

**2** Transformation from source patterns to target patterns in Code 3 – (Exploring nearest bridge)

Transform source pattern $ps_3$ into target pattern $pt_{10}$ through relation $r_1(ps_3) = pt_{10}$. In the transformation, the target patterns $pt_{10}$, is needed.

Code 3 – ESF relation

```
<xsl:template name="r1">
<xsl:variable name="macd">
<xsl:value-of select="//ps3@value"/>
</xsl:variable>
<xsl:choose>
<xsl:when test="$macd = 1">
</xsl:call-template name="fs5">
</xsl:when>
</xsl:choose>
</xsl:template>
```

**3** Target patterns extraction in Code 4 – (Obtaining last pattern by recursion)

Code 4 – ESF target functions

```
<xsl:template name="ft3">
<xsl:choose>
<xsl:when test="//pt3@subject = //pt6@subject">
<xsl:call-template name="search-and-replace"/>
<xsl:with-param name=" input " select="//pt6@sentence"/>
<xsl:with-param name="search-string" select="//pt6@subject"/>
<xsl:with-param name="replace-string" select="it"/>
</xsl:call-template >
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="//pt6@sentence"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name="ft4">
<xsl:value-of select="//pt3@sentence"/>
<xsl:text>, and </xsl:text>
<xsl:value-of select="//pt7@sentence"/>
</xsl:template>

<xsl:template name="ft5">
<xsl:text>After </xsl:text>
<xsl:value-of select="//pt9@sentence"/>
<xsl:text>, </xsl:text>
<xsl:value-of select="//pt8@sentence"/>
</xsl:template>
```

Put pattern $pt_3$, $pt_6$ into target function $ft_3\ (pt_3,\ pt_6) = Referring(pt_3,\ pt_6)$, and quote the function value as new pattern $pt_7$.

Then put pattern $pt_3$, $pt_7$ into target function $ft_4\ (pt_3,\ pt_7) = Aggregation(pt_3,\ pt_7)$, and quote the function value as new pattern $pt_8$.

Then put pattern $pt_8$, $pt_9$ into target function
$ft_5\ (pt_8,\ pt_9) = (After\ pt_9,\ pt_8)$, and quote the function value as new pattern $pt_{10}$. The obtained patterns $pt_{10}$ is regarded as target patterns.

These XSLT codes mentioned above are just a portion of all. If the XML document constructed as Code 1, then all of XSLT codes are implemented, the result text can be obtained as shown in follows:

Result

```
"In Tokyo Stock Exchange market, the Tokyo Stock Exchange Prices Index
(TOPIX) is analyzed now. ...
It opened at 1105.08, and closed at 1109.84 on 2004-08-20. Its total
turnover was 1072250000. It opened at 1115.93, and closed at 1114.24 on
2004-08-23. Its total turnover was 1047230000. After positive divergence
and bullish moving average crossover of MACD, the stock is oversold,
and it is considered bullish signal. It opened at 1116.74, and closed at
1116.60 on 2004-08-24. Its total turnover was 1065260000. ..."
```

## 4 Discussion and Conclusion

Our hypotheses are that texts which contain technical indicators as described above will help non-experts to retain more information and perform better than charts, and that non-experts will rate these texts as more interesting and pleasant to read. For this point, the evaluation experiment is carried out in which learning outcomes of texts will compare with charts'. The MACD indicator is chosen for the evaluation. For each indicator two evaluation suites are prepared. The only difference between them is that one uses the texts of technical indicators and the other uses the charts.

The tested are 40 students who do not have expert knowledge of stock market. They are separated into two equally sized groups. Group A read the charts where the technical indicators are marked and group B read the analysis texts which our KR system generates. After reading, two groups will finish the evaluation suite including "comprehension", "accuracy", "time", "interest", "remembrance", and "usefulness" items. The evaluation results are shown in Figure. Obviously, the "comprehension", "interest", "remembrance" items' scores of group B are much higher than group A. Because the tested think not only charts but also texts are necessary for the technical indicators to be described, the "usefulness" item's score between group A and group B is about same. Otherwise, the "accuracy", "time" items' scores of group B approach group A. The reason is that the ESF schema has linguistic completeness and the computer's speed is much faster than before. The ESF based KR system can generate sufficiently quality analysis texts for non-experts.

In the traditional view, KR can be separated into two kinds: *Template based* KR and *standard* KR. Template based KR system maps its non-linguistic input directly (i.e., without intermediate representations) to the linguistic surface structure. Crucially, this linguistic structure may contain gaps. Well-formed output can be obtained when the gaps of linguistic structure are filled until the linguistic structure does not contain gaps. By contrast, standard KR systems use a less direct mapping between input and surface

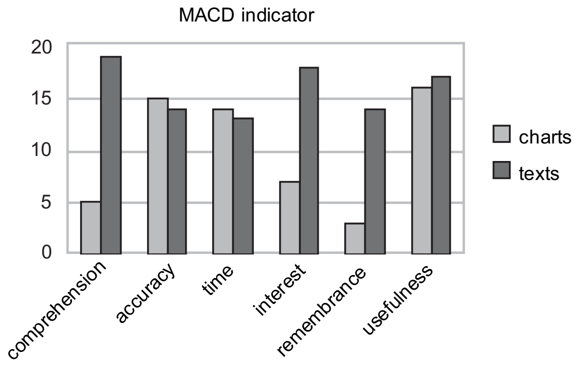


**Fig. 4.** evaluation experiment

form. Such systems could begin with inputting semantic representation, subjecting it to a number of consecutive transformations until a surface structure results.

Within this paper, the ESF based KR system can do the accurate and convenient transformation between all kinds of knowledge objects. Because not only the ESF schema combines standard KR with templates, but also it tends to use syntactically structured "templates" – function (here the function has more changes than the common template), and allows the gaps as function's parameters to be filled recursively for realizing linguistic expression like standard KR system. Therefore, we name our KR system – *function based* KR system.

Furthermore, we utilize the XML and XSTL techniques to describe the ESF schema in this paper. Because the XML and XSTL have become the main techniques of information formalizing on Web, they supply the common rules for developers to format all information under one standard. Therefore, this schema using XML and XSTL techniques can be realized on Web for information's integration, distribution and transformation.

## References

[1] V. Mittal, J. Moore, G. Carenini and S.F. Roth, Describing Complex Charts in Natural Language: A Caption Generation System *Computational Linguistics* (1998, 24(3), pp. 431-467).

[2] M. Theune, E. Klabbers, J. Pijper, E. Krahmer and J. Odijk, From data to speech: a general approach, *Natural Language Engineering* (2001, 7(1), pp. 47-86).

[3] Susan W. McRoy, S. Channarukul and Syed S. Ali, An augmented template-based approach to text realization, *Natural Language Engineering* (2003, 9(4), pp. 381-420).

[4] M. White, Reining in CCG Chart Realization, in *Proceedings of the 3rd International Conference on Natural Language Generation* (2004).

[5] Mary E. Foster and M. White, Assessing the Impact of Adaptive Generation in the COMIC Multimodal Dialogue System. in *Proceedings of the Workshop on Knowledge and Reasoning in Practical Dialogue Systems* (2005).