BASIC SEMANTIC ELEMENT EXTRACTION :

THE RULE WRITING EXPERIENCE

This is a dissertation submitted to
The University of Manchester Institute of Science and Technology
for the degree of MSc in Machine Translation

written by

EKATERINI PASTRA

Supervised by : Mr. J. McNaught

Department of Language Engineering

Manchester, September 2000

DECLARATION

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

To my beloved,

« Κάμε το χρέος σου στη γη.
Τράβα και βρες πούθ' αναβλύζει αυτό το φως μες στην Ελλάδα...»

A K N O W L E D G E M E N T S

There are many people one can thank for their direct or indirect help in writing a dissertation; I would like to express my gratitude and my deep respect to my supervisor Mr J. McNaught, for his being so co-operative and considerate. I would also like to thank Mr B. Black and Mr P. Thompson for their contribution to the rule writing experience for the Concerto Project. Last, it is my family I would like to thank, for all their support and love…

CONTENTS

ABSTRACT			6
OVERVIEW			7
PART ONE		**INFORMATION EXTRACTION AND THE CONCERTO PROJECT**	9
CHAPTER	1	INFORMATION EXTRACTION : ITS INTENTION AND EXTENSION	10
	1.1	LOCATING INFORMATION EXTRACTION	11
	1.2	HISTORICAL BACKGROUND	18
	1.3	NAMED ENTITY RECOGNITION TASK	21
CHAPTER	2	THE SYSTEM USED	24
	2.1	CONCERTO : HOW IT WORKS	25
	2.2	BSEE MODULE	27
	2.3	RULE SYNTAX	30
PART TWO		**THE RULE WRITING EXPERIENCE**	32
CHAPTER	1	METHODOLOGY	33
CHAPTER	2	TIMEX RULES	35
	2.1	DATE RULES	37
	2.2	TIME RULES	50
CHAPTER	3	NUMEX RULES	54
	3.1	MONEY RULES	55
	3.2	PERCENTAGE RULES	57
CHAPTER	4	ENAMEX RULES	58
	4.1	PERSON RULES	59
	4.2	LOCATION RULES	66
	4.3	ARTIFACT RULES	73
	4.4	ORGANISATION RULES	78
CHAPTER	5	RULE ORDER	94
CHAPTER	6	REMARKS - CONCLUSIONS	97
APPENDIX	**A**	**RULE NOTATION**	99
APPENDIX	**B**	**RULE SET**	102
APPENDIX	**C**	**SAMPLE RESULTS**	118
BIBLIOGRAPHY			123

A B S T R A C T

Nowadays, knowledge is considered the greatest asset of all. Managing this asset requires – among other things –efficient information systems that will support accessing, analysis, classification, storing of data and its transformation into useful information, leading therefore to knowledgeable decision making. Information Extraction is a core Language Engineering Technology with crucial role in Knowledge Management systems. Its first phase, the Named Entity Recognition Task, 'forces' researchers to face Proper Names, analyse them and benefit from all information revealed by them. NLP tools such as POS taggers, tokenisers, chunkers and databases assist in the whole procedure of extracting basic semantic elements. Rule based approaches dominate in the field; all modules mentioned provide essential feedback to the rules, which are in addition, context-sensitive. The input of Information Extraction systems is raw text and the output is a series of edges carrying valuable information. This output can be used in more advanced tasks, such as Template Filling or be stored / displayed. Automatic evaluation can be performed and the results are usually quite close to human performance.

O V E R V I E W

Rule – writing for Named Entity Recognition is the topic of this dissertation. We have worked on the Basic Semantic Element Extraction (BSEE) module of the Concerto system, a system still under development. UMIST is one of the main partners of the Concerto Project and therefore software and technical documentation was at our disposal.

The dissertation is divided in two parts. The first part gives background information on both Information Extraction and the Concerto System. In particular, the first chapter attempts to locate Information Extraction in the field of Knowledge Management. To define and understand what Information Extraction is, we need to differentiate it from tasks that seem similar, such as Text Data Mining and Information Retrieval. To explain its crucial role and contribution to Language Engineering Technologies, we explore its relation to Knowledge Management and Knowledge Discovery fields. Message Understanding Conferences' contribution to the field is briefly presented showing us the history of Information Extraction. After having a clear view of the field, we refer to its prototypical subtasks, emphasising the Named Entity Recognition task. The nature and characteristics of Named Entities are mentioned and they are contrasted with those of common nouns.

In chapter two, systems for the particular task are mentioned and details on the Concerto system are provided. The module used is presented in more detail; one must be familiar with the way it works, so as to understand the rule writing procedure. Additionally, the structure of rules used in the module is explained.

The second part of the dissertation is the technical one; it is here where the rule file created is presented and commented. Chapter one deals with the methodology followed for developing rules. Rule categories are named after the SGML tag assigned to the respective expressions they identify, in Message Understanding Conferences, and though the marking does not appear in the Concerto interface as an SGML tag, we have preserved these names. Timex rules are explained first, that is rules for expressions denoting Time / Date. Next, Numex rules (Money and Percentages) are commented upon and last come the Enamex rules (Persons, Locations, Artifacts, Organisations). All sources used, internal and external evidence, are reported and

example – cases from the rule refinement phase are provided so as to justify all choices made during rule writing. Chapter five discusses the rule order suggested for the rule file and gives evidence in support of this order. Then, we conclude with some remarks on the whole rule writing procedure, points we would like to make based on the experience obtained from the whole task.

Rule notation and the actual rule file are available in Appendices A and B respectively. Appendix C contains sample results of our test runs; rule performance on actual texts / text fragments could be a whole separate task for the evaluation of the results; MUC has established appropriate metrics. This is further work that has to be done by Concerto partners.

PART ONE

INFORMATION EXTRACTION AND THE CONCERTO PROJECT

CHAPTER 1

INFORMATION EXTRACTION: ITS INTENTION AND EXTENSION

The more Informatics advances, the more data becomes available; digital documents are easily accessible through the Internet and a plethora databases and data warehousing technologies assist in storing this bulk of data. In the business world, the largest database created handles 20m transactions a day; Mobil Oil Corporation is developing a data warehouse capable of storing over 100 terabytes of oil exploration related documents and these are just a few cases. However, all these large textual collections contain just raw data; it is its conversion into useful information that becomes, nowadays, a "competitive advantage" and knowledge inferred from this information is indisputably the most valuable asset of all. So, efficient access and manipulation of data is on demand; Language Technologies do have an answer for that and Information Extraction has a significant place among them.

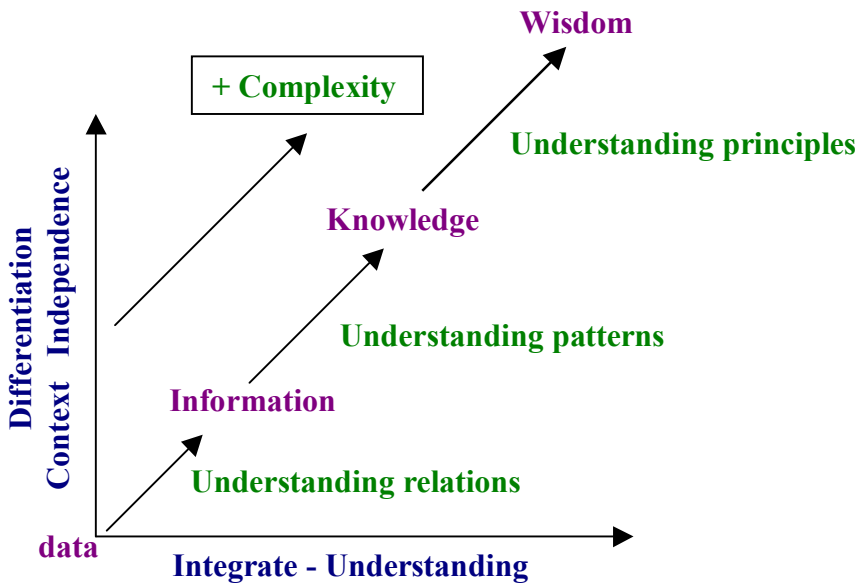
Information Extraction is a relatively new field in Natural Language Processing. As such, its relations with existing processes and tasks is not quite clear. Its boundaries with Information Retrieval and Text Data Mining are not clear cut, but they should not also be blurred. Sometimes in the literature, Information Extraction is presented as a component of Information Retrieval Systems and the latter has been said to form a core process in Knowledge Management (EAGLES 1996). Things get more confused, when Knowledge Management is related to Knowledge discovery, the latter being used interchangeably with Data Mining (Chios at al 1998).

Reviewing literature on all the above fields, one can see what relates them, but also what differentiates them. This chapter attempts to locate Information Extraction among all other Language Engineering and in general Knowledge related fields. In this way, what Information Extraction is and why it is useful will be presented; exploring the 'semantic net' in which it is involved, one sees both its 'context' and its characteristics. The subtasks of Information Extraction are also presented; they are the

prototypical ones specified by the Message Understanding Conferences. Named Entity Recognition, which is our interest, will be presented in more detail.

1.1 LOCATING INFORMATION EXTRACTION

Storing data has become quite easy during the last decade. Large databases and data warehouses are being constantly created, as mentioned earlier. Nevertheless, there is a gap between data collection and data comprehension. Raw data has to be analysed and processed, if one is to get the most out of it. Data has no inherent meaning on its own; when it is endowed with substance and purpose then it is not still data, it has been converted to information. Bellinger (2000) illustrates the relation between data – information and knowledge successfully :



Raw data out of context, isolated, is not useful at all. One needs to understand the relations between pieces of data in order to be able to use it. Usually, one understands something new by integrating it with knowledge that one already has. In this way differentiation is smoothed and things seem clearer. It is only then that one talks about information. To go from information to knowledge, one has to understand patterns that relate data and information; patterns are repeated, fixed, consistent and predictable types of relations. So, one would be right if one said that we need information systems

that will "understand" documents/data, if we are to handle data efficiently, if we are to manage knowledge.

Knowledge can be seen either as a thing–with structure and content- to be stored and manipulated or as a process (Zack 1999). Knowledge as a process is the application of expertise. The Davenportist notion of knowledge[1] is supported here, which sees knowledge as a level of information in a value chain. Value adding processes can lead therefore to knowledge and it is a series of such processes that Knowledge Management Systems consist mainly of.

Knowledge Management is considered with all phases of the conversion of data to knowledge. Acquisition, processing and manipulation of data, extraction of information from it, all are of interest in Knowledge Management. Managing of knowledge assets and processes that act upon these assets (Godbout 1996) is the core task in this field. The means to accomplish that, the methods and tools that support the whole procedure, form the Knowledge Engineering environment. The main goal is knowledgeable decision – making. This is where competition takes place nowadays; chasing data that others may not have is not an asset anymore. The 'espionage' era is already past; it is the retrieval of information from data with precision, in real time and with the minimum labour possible that makes the difference. So, the business world is looking for integrated, multifunctional systems, that can support all Knowledge Management and Knowledge processing activities such as : capturing, organising, classifying, understanding, debugging, editing, retrieving, disseminating, transferring and storing knowledge.

One understands that Knowledge Management is (or at least should be) document based and computer-aided. It is endless, but most of all it is not necessarily profound; this means that full text understanding is not essential, a partial understanding of texts is enough to manipulate data. This is important in terms of computational efficiency in the field, as Language Technologies required for building Knowledge Management systems do not have to analyse texts in full.

It would be quite biased a view to say that Knowledge Management starts and ends in building or acquiring efficient Information Systems. The human parameter is always important. Managing and organisation roles are equally important in the whole

[1] This is opposed to the Polyanist view that assimilates knowledge to the process of knowing (Zack 1999).

procedure and interfaces with computational output are considered essential. The typical Knowledge Management architecture (Zack 1999) consists of four basic parts :

a) Repositories of explicit knowledge / data (concepts, terms, categories, indexes…)
b) Refineries (accumulation, analysis, distribution of knowledge)
c) Organisation roles (for execution and managing of the refining process)
d) Information Technology support (technical support, training, compatibility with existing systems…)

Repositories of raw data are easily obtained. Sometimes this data is manually classified, value is added. Mainly, a by product of the whole Knowledge Management procedure is the creation of repositories with valuable information. The refinery part is the main part of processing the data and Information Technology is the means for doing it. Organisation roles are assigned for the surveillance of the flow of the procedure. So, this second part is the one that interests language engineers most. The refinery section has itself five stages:

- acquisition of data
- refining (before added to repositories, captured data is subjected to value adding processes)
- storage and retrieval (it "bridges upstream repository creation to downstream knowledge distribution" - Zack 1999)
- distribution mechanism (make repository contents accessible)
- presentation

Information Extraction is a value adding process, it can be located at the second stage of the refinery part. It deals with raw data and looks for knowledge in the document, it has access to the documents themselves. Herein lies its main difference to Information Retrieval. Retrieval of data has to do with identification and ranking of documents relative to a query made. The main search units in Information Retrieval are only documents (EAGLES 1996). It can be either an individual document one wants to find from a large collection, or any document of a specified content (needed for improving one's knowledge on a subject), or any document containing a specific text string (needed for studying the context of such a string). On the contrary,

Information Extraction's search units involve data itself. Named Entities and patterns are identified. The documents are not selected; they are processed so as to "identify pre-specified entities and the relations between them and fill in a structured record / template with found information (Gaizauskas and Robertson, 1997).

The key notion in Information Retrieval is in fact, the template. It has been defined (Wilks and Catizone 1999) as "a linguistic pattern, usually a set of attribute – value pairs. The values are text strings created by experts to capture the structure of the facts sought in a given domain and which Information Extraction must apply to a text corpus with a set of extraction rules that seek those fillers in the corpus…". It is evident that Information Extraction looks for facts buried in texts, whereas Information Retrieval looks for whole texts in which key – terms appear. The latter is domain independent, since no linguistic analysis of the semantics of stored texts or inquiries is required. On the contrary, Information Extraction is domain dependent, it is a skimming mechanism for routing (filtering)[2] of texts and template filling, and to achieve these it uses all syntactic, semantic and pragmatic constraints needed…

One can see now, how different these two tasks are. They both belong to the refinery part of the Knowledge Management procedure, but they are quite distinctive tasks. However, their boundaries are not that clear. One seems to be a prerequisite for the realisation of the other; in other words, these two processes are of "complementary nature"(Gaizauskas and Robertson, 1997). Information Retrieval may act as a filter of texts to be input to the Information Extraction module and Information Extraction may be used for the optimisation of the retrieval process. The latter can help in encountering successfully the well known keyword barrier in Information Retrieval. Instead of relying on the absence or presence of text strings given in a query in the text corpus, templates may be used and a matching of concepts may take place instead[3].

To the extend at which Information Extraction deals with patterns, it is often confused with Text Data Mining. To say though that these two are the same is to equate a procedure with its means. Data Mining is "the application of an algorithm for pattern extraction from data…" (Fayyad et al. 1996), it is the method used for converting information to knowledge. These methods may vary : decision trees, rules,

[2] It divides input text into relevant and irrelevant sections…

[3] The notion of conceptual annotation and Information Retrieval was described in literature by Mauldin (1991) and has been applied quite successfully (FERRET System and CRISTAL). In Concerto System conceptual annotation is done via an Information Extraction module (McNaught et al. 2000).

statistics, neural networks, fuzzy sets, clustering, regression models or even example based techniques may be used. The Text Data Mining methods are mainly rule based.

So, it is a matter of point of view, to name the field Text Data Mining or Information Extraction. The former focuses on the procedure (the mining of data with certain methods), whereas the latter focuses on the result of the procedure (the information drawn from the original data). According to the method followed, Data Mining may be pure unrestricted, that is no indication of what kind of discovery would be of interest is given; unexpected patterns may be found or well known ones may result, that will give no new insights into data. Directed Data Mining is exactly the opposite; there is a specified focal point and this is most commonly followed in Information Extraction. Hypothesis testing and refinement may also take place with Data Mining techniques.

Last, the notion of Knowledge Discovery is quite often used interchangeably with Data Mining. Again, the process is confused with the methods used; Knowledge Discovery has been defined as the process of "identifying valid, novel, potentially useful and ultimately understandable patterns in data, using Data Mining methods to extract what is deemed knowledge according to specific measures and thresholds, using the database feature along with any necessary pre-processing…" (Fayyad et al. 1996). Is Knowledge Discovery then, the same procedure that Information Extraction is? The definition may lead to this conclusion; Information Extraction aims at extracting patterns from data, does use Data Mining techniques, it is domain dependent and takes advantage of databases and results of pre-processors[4].

Nevertheless, Knowledge Discovery is a superordinate concept when compared to Information Extraction and it is the term itself that may intuitively lead one to this conclusion. Apart from intuition, a detailed description of the field leads to the same conclusion; Knowledge Discovery is described (Fayyad et al. 1996) as a procedure consisting of the following stages :

- Acquisition of raw data from databases
- Target Data selection
- Pre-processing

[4] The extraction rules that are usually used (mentioned earlier) do use syntactic, semantic and sometimes pragmatic constraints; pre-processing (Tokenisation, tagging, morphological analysis etc) does provide evidence for these constraints…

- Pattern Extraction with Data Mining techniques resulting in data transformation
- Interpretation and Evaluation

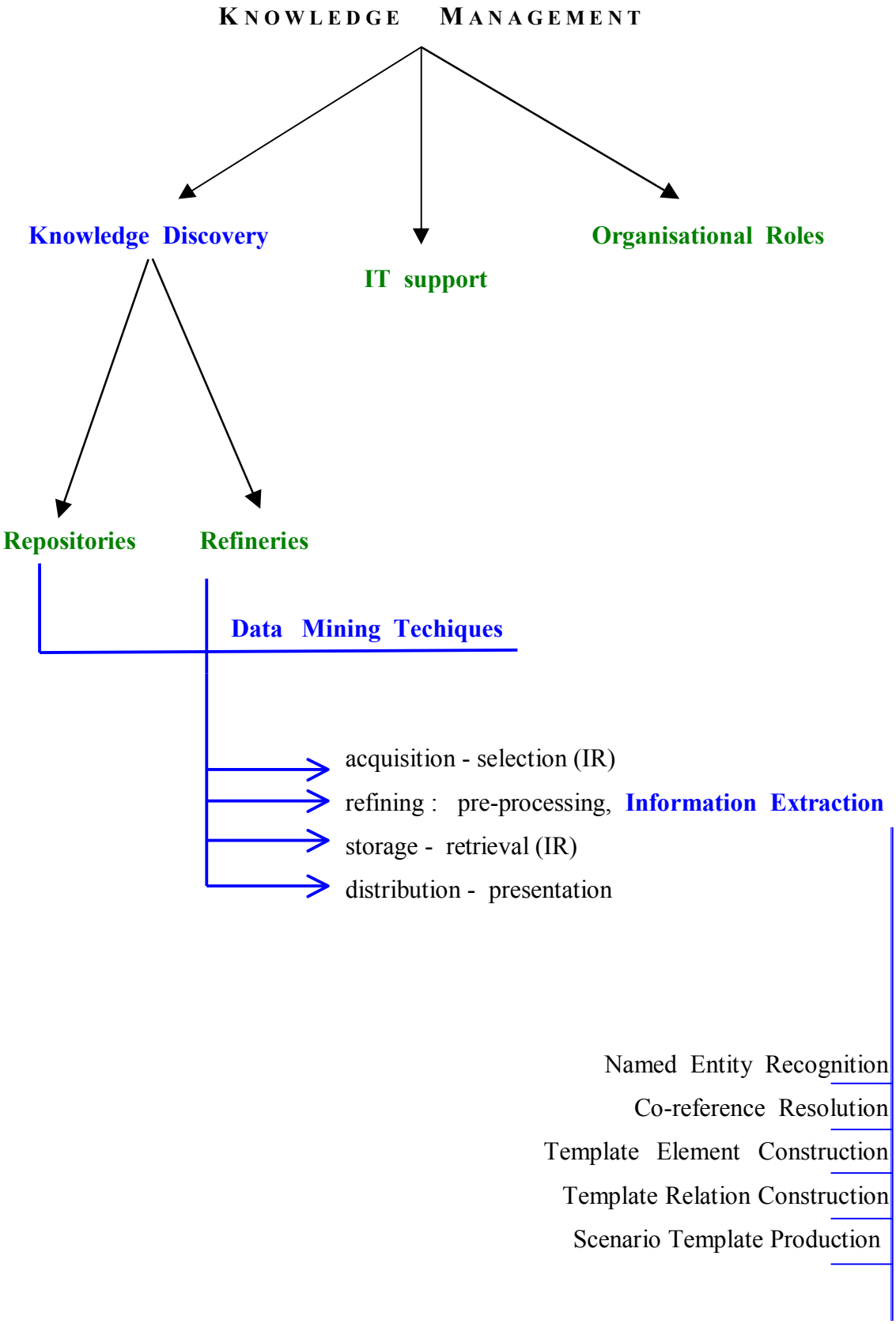
The second stage describes –more or less- an Information Retrieval task. Pre-processing involves tokenisation, tagging and morphological analysis of the contents of the text. The fourth stage describes the transformation of data into knowledge, through pattern extraction; that is what Information Extraction does. So, Information Extraction is a stage of Knowledge Discovery. One may say though, that Information Extraction does not involve only pattern extraction; it also extracts basic semantic elements. Nevertheless, basic semantic element extraction is a pre-requisite stage, a basis for the more advanced task of template construction and filling. As we will also see later on, the first task of Information Extraction not only extracts valuable information for immediate display to the user or storage in a repository, but it also helps significantly in the pattern extraction procedure, in the conversion of information into knowledge.

We have concluded that Information Extraction is a stage of Knowledge Management and Knowledge Discovery as well. Does this mean that Knowledge Management and Knowledge Discovery are the same? As far as we know, the literature gives no insight into the relation between the two. Independently studied, they both seem to have the same objectives (knowledgeable decision making) and as mentioned above, they seem to have the same relation to certain language technologies. Nevertheless, Knowledge Discovery lacks certain, significant parts of Knowledge Management architecture, such as Organisation Roles. So, we believe that it is a subordinate concept that focuses just on the language processing part of Knowledge Management and has no organisational / managerial aspect[5], it does not address the implementation problems that arise when knowledge discovery is used for the needs of the business world…

So, Knowledge Discovery covers half of the parts of Knowledge Management architecture. It involves the repositories and refineries; the latter consists of several stages, one of which is Information Extraction with its own subtasks. Information Retrieval may precede extraction for filtering text input and / or is also used for dealing

[5] This 'managerial' aspect involves strategies for successful implementation of Knowledge Management in corporations and industries. The needs of the company have to be considered, the employees have to be trained in using the systems that will be introduced, a whole adaptation stage has to be organised…

with queries. Data Mining techniques may be used for Knowledge Discovery. The following scheme illustrates all concepts mentioned above :



1.2 HISTORICAL BACKGROUND

We saw earlier that Information Extraction itself is divided into five subtasks. These are the tasks defined in Message Understanding Conferences and especially in the last Conference – MUC 7 - held in 1999. Before explaining the objectives of these tasks one has to refer to these conferences around which a progressively augmenting research and development community arises.

Text Understanding is not a new idea in the field of Natural Language Processing. In Cowie and Wilks (2000), we read that "early works by DeJong at Yale University focused on searching texts with a computer to fill predetermined slots in structures called 'scripts' " and this was back in the late 70s. Carnegie Group developed JASPER in the mid 80s; this commercially available system depended on very complex handcrafted templates and a very specific extraction task was performed. At the same time ARPA, the U.S defence agency, funded the first Message Understanding Conference (1987). Competing research groups were funded to pursue Information Extraction and this was a great motivation for the field to grow and to grow rapidly. Naval ship to shore messages were processed originally and later on, terrorism information from newspapers, joint ventures and microelectronics were the fields of interest. These tasks, with obvious applicability, plus the development of an uncontroversial evaluation method, attracted more and more interest in the Conferences; U.S participants are joined, nowadays, by several groups from Canada, Europe and Japan[6] ; industry and academia are brought together.

According to the MUC procedure, participants agree to evaluate their systems on a common task and compare the results against human performance, that is with manually extracted information. So, MUCs define the objectives and tasks of Information Extraction, the metrics for evaluation of the results and the nature of data for training and testing. ARPA has also initiated TIPSTER, an object oriented data model designed to support a broad range of document processing tasks; it attempts to standardise Natural Language Processing around a common architecture for document annotation. In particular, Information Retrieval and Information Extraction Tasks are addressed by this model. It has already proved very useful for building multi-

[6] A detailed discussion on the contribution of MUC, the drawbacks for the European scene etc can be found in Wilks (1997).

component NLP systems[7]; reusability of the components, so as to reduce duplication of effort, is of significant importance.

In MUC – 7 (held in 1998), five tasks have been defined for Information Extraction:

- Named Entity Recognition (NE)
- Co-reference Resolution (CO)
- Template Element construction (TE)
- Template Relation construction and (TR)
- Scenario Template production (ST)

Each task is dependent on the ones that precede; a data to information and then to knowledge conversion takes place. The first task finds and classifies named entities such as person names, organisations, locations etc. Raw data becomes then information, ready either for storage or display or to form the basis for the subsequent advanced tasks. The second task deals with identity relations between entities in texts: an organisation name, reported originally in full, is later on mentioned in the text partially: "MedLex GlycoSciences Corporation….Medlex…". This task is sometimes integrated with the first, since it complements the recognition of Named Entities. The results of the stages so far are used for adding descriptive information to Named Entities, creating this way database - like records. This Template Element construction phase enhances information already available from the previous stages. It is only when relations between these enriched pieces of information are identified that information is converted to knowledge. Patterns are found during the Template Relation construction stage. Last, Scenario Templates 'tie' template elements together into event and relation descriptions and these filled templates are typically the output of an Information Extraction System[8].

The above tasks follow a less - to - more domain dependent order; generally speaking, they are weakly domain dependent. This excludes the case of Co-reference resolution and of the Scenario Template extraction task, that are by definition dependent not only on the domain of interest, but also on specific scenarios of interest, within the same domain.

[7] Cf. GATE developed at the University of Sheffield (Wilks and Gaizauskas, 1999).
[8] Detailed presentation of the tasks with examples is available in Cunningham (1999).

Performance evaluation that took place in MUC-7 provides a clear view of the current level of technology in the field. Precision and Recall, well-known measures from Information Retrieval tasks, are used for the evaluation (plus the f-measure, the differential weighting of precision and recall)[9]. The more demanding / advanced a task is, the lower it performs. Named Entity Recognition is quite close to human performance (a combined measure of precision and recall of 95% has been attained). Co-reference results vary widely depending on the domain: "perhaps only 50-60% may be relied upon …with human scores only around 80%" (Cunningham 1999), having though in mind that both proper noun co-reference identification and anaphora resolution are hiding behind this score. Template Element Construction scores around 80% (that is for the best system in MUC-7), with humans achieving 93%. Last, Scenario Template extraction reaches 60% of precision and recall with humans achieving 80+%[10].

One should not forget that the type of texts used (articles, e-mail messages, HTML documents etc), their style (formal – informal), the domain in which they belong (subject field) and the scenarios sought for (event types) do affect performance. What is important is that evaluation metrics established by MUCs do provide a basis for finding out what developed technologies may achieve and what can be pursued in the near future.

[9] For more details and a report on an enriched set of evaluation metrics (a slot error rate has been suggested as a performance measure) cf. Makoul (1998). A technical report from NIST (1999) refers also to Entity recognition evaluation, based on SAIC score software, and suggests three new more error measures (entity error rate, type error rate and true content error rate).

[10] The percentages are drawn from Cunningham (1999), where one may find more details.

1.3 THE NAMED ENTITY RECOGNITION TASK

The Named Entity Recognition task is the first step in Information Extraction. For other tasks to be performed, Named Entities must have been identified and classified, that is, converting raw data into information. The success of this task greatly affects the performance achieved in the following tasks. So, the fact that current Information Extraction systems have been said to function at human performance levels for the specific task is important for further achievements in the field, in even more advanced tasks.

Multilingual Named Entity Recognition is a rapidly growing task as well. Multilingual Entity Task Evaluation conferences (MET) have been held, sponsored by DARPA . The first, held in 1996, evaluated systems that marked Named Entities in Japanese, Chinese and Spanish newspaper articles. One more conference has taken place; it is MET-2 which was run in conjunction with MUC-7. The focus of these multilingual Information Extraction conferences is only the Named Entity task, but there are future plans for higher level tasks as well. The languages of MET-2 were Japanese and Chinese with an additional, experimental track using Thai (Marsh, 1998). Established test methodology and multilingual task descriptions are available for all concerned and this is very important for the development of the field.

This new branch of Named Entity Extraction bridges the field with the more traditional field of Machine Translation. Partial understanding of texts is enough for the needs of the Named Entity Recognition task; on the contrary, Machine Translation deals with deeper understanding of texts and here is where most problems reside. In cases when only specific information is needed, Multilingual Information Extraction may be the answer to translational problems. The input text may be in one language and the output information may be in another, having in between a translation phase where no full translation is required, just translation of certain text strings.

Data of interest in the Named Entity Recognition task are Proper Names. The difference between Proper Names and Proper Nouns is not quite clear in the literature. We believe that the former term avoids specifying the part of speech of the linguistic unit of interest. We will use this term to refer to all single and multiword Named Entities, since these entities are usually but not exclusively nouns.

Proper names have been quite neglected by linguists because of their nature:

(a) They are used for denoting individual entities in a unique way: This characteristic seems disputable when considering cases of coinciding Proper Names (Lopez – Trigueros, 1998), nevertheless it is a fact when comparing with common nouns that refer to classes of entities.
(b) They seem 'de-lexical'[11], as if they do not carry any semantic information, they are just 'deictic'.
(c) Their syntactic behaviour varies widely: "They typically do not allow quantifiers, demonstratives, possessives, specifiers or modifiers" (Allerton 1987), but one may find for instance Trade-names that are preceded by possessives: "Avirons's Multikine™…", "Our Clarity 2000 database". Sometimes, they themselves are used as modifiers: "the Clinton case". They are used in metaphors carrying then a particular connotation: "a new Shakespeare". At other times, they form metonymies, behaving thus as common nouns: " a Ford speeded ". One can see how creativity in language leads to unpredictable uses of Proper Names…
(d) Their structure is unpredictable, especially in cases of multiword Proper Names:

- A single Proper name may consists of all capital letters (IBM), a mixture of capital and lower case letters (GlycoSciences), letters and numbers (M16).
- Common nouns may form part of the Proper name (New York City), or even more, a Proper Name may consist entirely of common nouns (The House of Representatives).
- Prepositions, articles, conjunctions and other particles may form part of the Name (University of Manchester Institute of Science and Technology).
- Trigger words may precede or follow Proper Names. A trigger word indicates that the tokens surrounding it are probably Proper Names and may reliably permit the classification or even the subtype of the Name to be determined (MedLex Inc., Clarity database, Mr Johnston…). Unfortunately, their presence is not obligatory whenever a Proper Name appears in a text.

All these characteristics of Proper Names are the cause of being neglected by researchers. Nevertheless, Information Extraction, and its implementation for real

[11] They do not point to any concept, they are of empty content. The term is drawn from Collins Co-build English Grammar, where it is used for specific verbs (i.e. make, do – in forming verbal phrases).

texts, has changed things. The Named Entity Recognition task encounters exactly these difficulties, since Proper names are its main search unit. Whoever works on such a task can observe, in the corpus, the characteristics mentioned above. Local context, internal and external evidence can all be used for the objectives of the task to be attained. For example, Trigger words are of great importance in Entity Recognition; the fact that Proper names are capitalised can also help. The second part of the Dissertation that discusses the rules built for the Named Entity Recognition task refers to the matter in detail.

Last, one should mention that the task, as defined in Message Understanding Conferences is not only interested in typical Proper Names and the expressions they form (person names, organisations, locations), but also in time and date expressions, money expressions and percentages. These four categories cannot be characterised as Proper Names; however, they share many of the characteristics mentioned above. They are Basic Semantic Elements and this term may characterise better the nature of entities that are sought …

Terms can also be considered 'Basic Semantic Elements'; however, they refer to specific / special entities within a discipline, sharing thus characteristics more with common nouns rather than with Proper Names (Lopez - Trigueros 1998). Term extraction goes beyond the scope of prototypical Entity Recognition tasks, but the latter may help in performing the former…

CHAPTER 2

THE SYSTEM USED

Most Natural Language Techniques have been applied for Information Extraction purposes. Currently, the "most successful systems use a finite – state, automata – based approach, with pattern either being derived from training data and corpora, or being specified by linguists" (Cowie and Wilks, 2000). Systems with such a design may test patterns rapidly using feedback from the scoring software. Linguistic experience has a significant place in these systems as well.

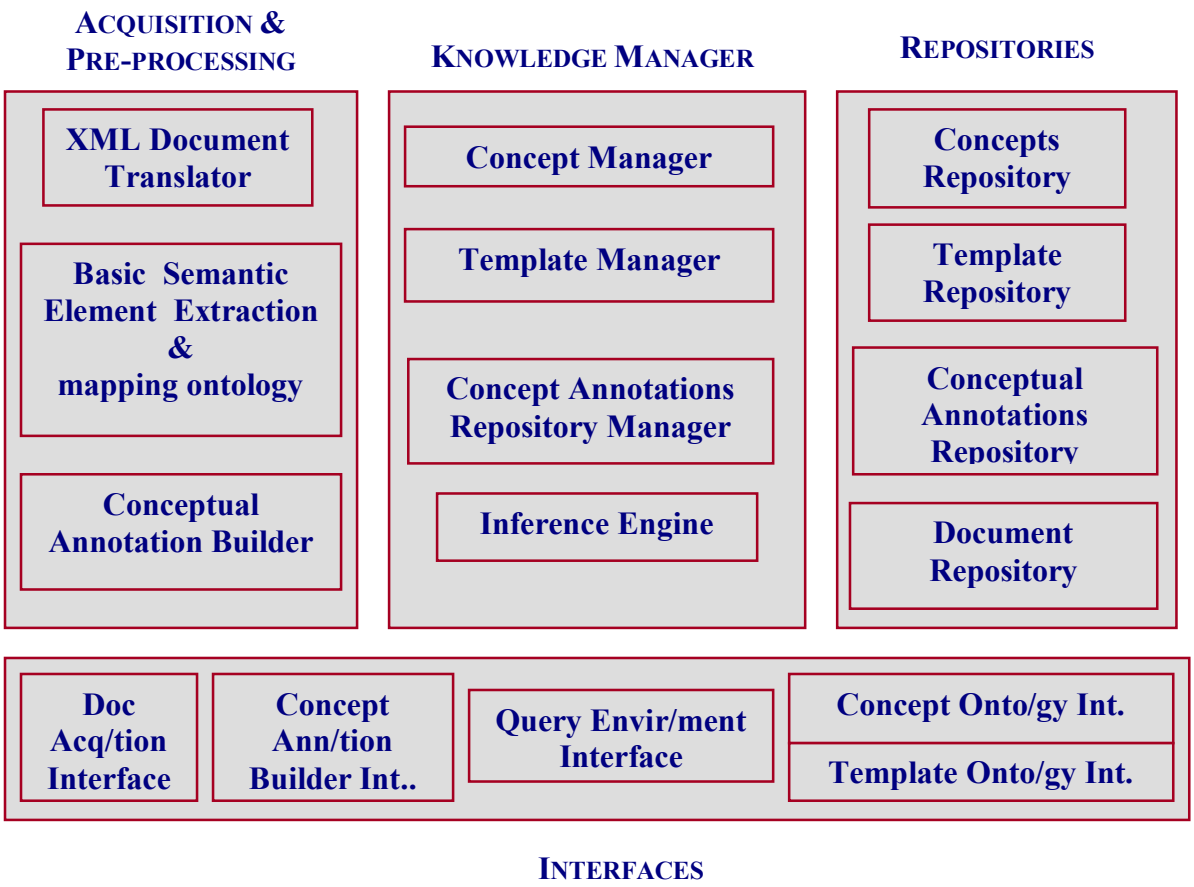
Using training data to find patterns in the texts and extract lists of Named Entities that would help if stored in a database has been attempted in Message Understanding Conferences. Nevertheless, there is a bulk of training keys available for one to use for MUC evaluations and therefore, this attempt though successful, cannot be the sole basis for an Information Extraction system. Learning and statistical methods have been applied to the field and they have led to the broad use of independent components in extraction tasks; POS taggers, tokenisers, morphological analysers, machine readable dictionaries and databases all may facilitate extraction. They assist in pre-processing data and they usually form an essential phase preceding the main Information Extraction Tasks.

A commonly used approach for the Named Entity Recognition task is the rule – based one. Rules are built for identifying and classifying basic semantic elements; they rely on results of the pre-processing phase and they are context – sensitive. One may not talk about syntactic – driven or semantic driven systems. All levels of linguistic information may be used. There are many Named Entity extraction systems, most of which have participated in MUC-7. One may have a very clear view of their abilities and performance both in proceedings of the conferences and in technical reports created by the system developers. IdentiFinder, Proteus /PET, LaSIE, NetOwl (with the highest score in the Named Entity task – 96.42), LTG and FACILE , all participated in MUC-7 and with the exception of the first, they all use rules, sometimes in combination with probabilistic methods (Lopez - Trigueros 1998).

2.1 CONCERTO : HOW IT WORKS

The FACILE (Fast and Accurate Categorisation of Information by Language Engineering) system forms the core of the Basic Semantic Element Extraction module of CONCERTO, the system we have used for the Named Entity Recognition task. CONCERTO is a project funded by the European Union (ESPRIT) and it is still in progress. Partners from U.K, Italy, France and Greece have been involved in developing several parts of the CONCERTO system architecture (http://concerto.ccl.umist.ac.uk).

The goal of the project is to create and manage knowledge repositories by conceptual indexing, querying and retrieval of documents. Its core activity is to set up a full Knowledge Engineering Software Environment to enable computer aided conceptual annotation and further intelligent information retrieval. It does not attempt a full conceptual analysis of documents, since a partial –selective- analysis of them can satisfy all the requirements of the user. The main domains of interest in the project are biotechnology and publishing. In order to locate the module of interest, the following figure may help by exposing the architecture of CONCERTO (Black et al. 1999) :



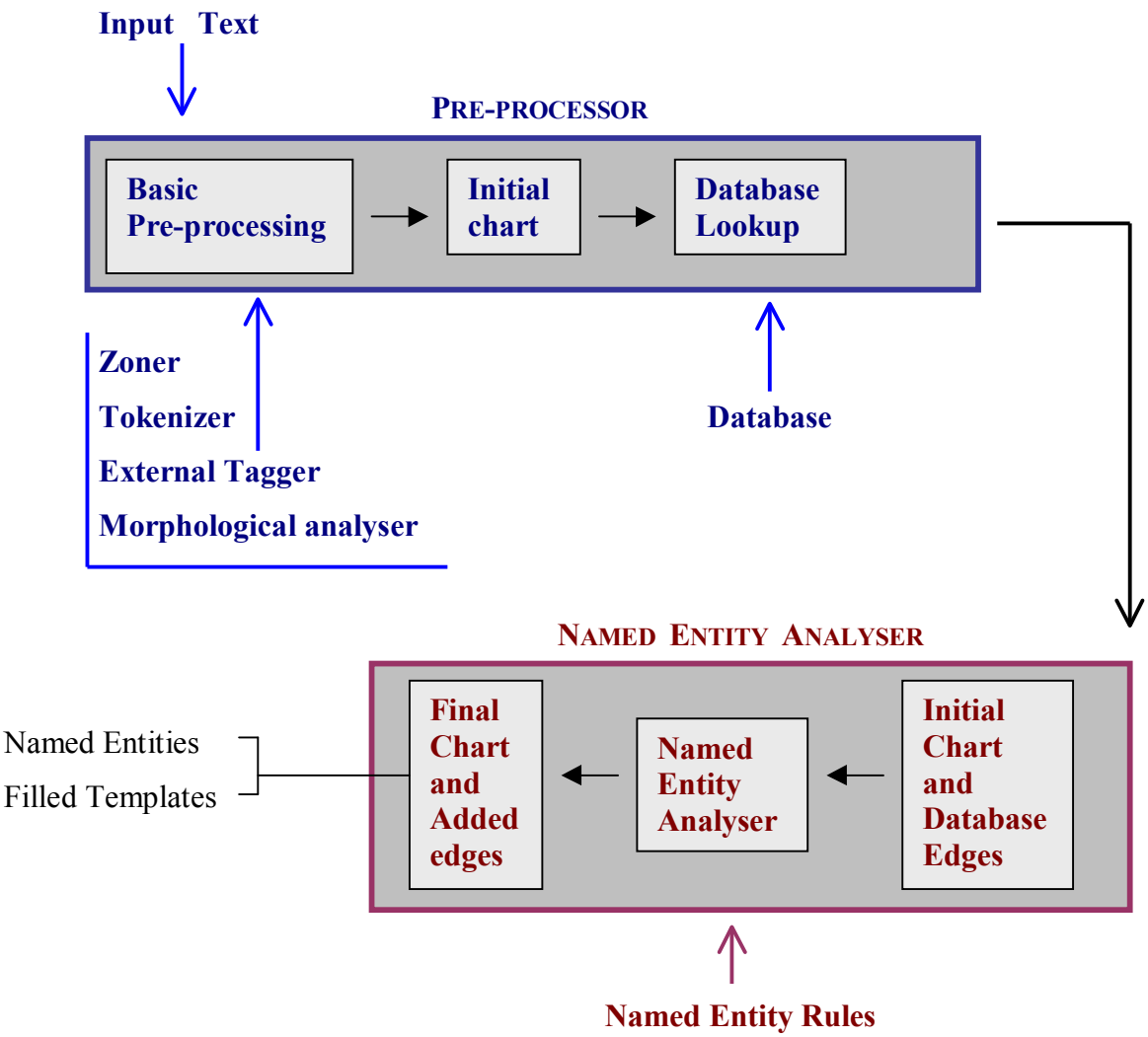
As one can see, the system consists of four main parts; Acquisition and pre-processing, Knowledge Manager, Repositories and Interfaces. It is a typical Knowledge Management system, in which Information Extraction technologies have been integrated along with other components for storing, accessing and analysing data. First, document capture and normalisation to XML mark-up takes place. An automatic low level linguistic processing follows, that leads to the identification of Named Entities, their relations and the actions undertaken by them. Conceptual Annotation is performed interactively; a trained user corrects and augments the system's proposed annotations. The interface helps in building representations (in a knowledge representation language – 'NKRL') which map Natural language terms, names and phrases used in texts to object and template instances.

The Knowledge Manager modules provide access to the repositories and to advanced manipulating operations. The Concept Manager stores and manipulates mainly concepts. The Template Manager creates templates and uses them to construct conceptual annotations. These annotations are then sent to the Knowledge Manager to update the respective repository. One can easily understand that the Knowledge Manager is a module that interacts with the majority of all other components of the system; it mainly provides basic functions for storing, modifying, deleting and querying templates, conceptual annotations and documents. Along with the Concept Manager, it allows other modules to request specific services and provides them with corrected results; for that reason, they are both designed and implemented as software server modules.

The Inference Engine answers search patterns specified by the users through the use of a Query Environment Interface. Last, a wide variety of repositories exists; these repositories store documents, concepts and annotations, as well as information essential for their construction (e.g. templates). In the Concerto system, all types of textual information are represented in XML format, whereas templates are expressed in Resource Description Format (RDF), a general toolbox for implementing specific semantic vocabulary (e.g. NKRL); it makes use of directed labelled graphs and it can easily be implemented in XML…

2.2 The Basic Semantic Element Extraction Module

The Named Entity Recognition task takes place in the BSEE module of the first part (Aquisition and processing) of the Concerto architecture. A modular language Engineering solution is used for the task, extended to extract other basic semantic elements of relevance to the application domain. The module was basically developed in FACILE (Black and Rinaldi 2000), but it has been adapted to the needs and purposes of the Concerto project. The main aim of the module is to automatically extract and tag basic semantic elements in a text. It collaborates with the Ontology Mapper, so as to aggregate the information extracted into templates. The Ontology Mapper associates names and partially filled templates with classes and templates in the main repositories of the Knowledge Engineering Software Environment. The following figure illustrates how the module works :



The module consists of two main parts, the Pre-processor and the Named Entity Analyser which relies on the results of the Pre-procesor. The input text is in XML format. A Text Zoning component selects the portions of text that are of interest to the system and these portions are tokenised, morphologically analysed and tagged. The input text is split into individual words, numbers, punctuation marks and other symbols; all tokens are morphologically analysed, so that their root is identified and other grammatical information is assigned to them. A Part of Speech tagger deals with disambiguation and assigns a syntactic label to each token that filters in fact the preceding morphological analysis. In Concerto, the Xerox InXight tools are used, but the user is always allowed to use the external resources of his/her preference, given they have similar functionality.

An initial chart is created then, exactly after the basic pre-processing has been performed. This is a tabular structure that contains all information derived so far for the tokens. Words and phrases are looked-up in a database of known names, trigger words and phrases designating domain items. This subsequent database look up stage is performed in two phases: first all single tokens are found in the text and then, multi-word look up takes place; when identified in the text, individual words (the constituents of the multi-word) are replaced with a spanning token that is considered atomic. Database entries are not case-sensitive; they are compared to the normalised (lower case letters) form of the tokens of the text.

All this information on the tokens is stored in a structure, called 'edge' (or feature vector); this structure consists of named slots with the following information :

- Where it begins and ends (through a pair of offsets).
- What separates it from its predecessor (white space, hyphen…).
- What text zone it comes from (title, main body of the text, trailers…).
- Its various orthographic versions (as it appears and normalised).
- Its syntax (category and features).
- Its semantic class (obtained from the database or the morphological analyser).
- Morphological analysis (partitioned into the consistent and non-consistent one with the tagger's output).

All follow up modules derive information about a token exclusively from its corresponding feature vector. So, the output of the Pre-processor is a chart comprising sets of active and inactive edges and a vertex index, and database

edges consist of tokens that are 'known', they have been identified, classified and stored in the database.

It is at this point that the Named Entity Analyser is invoked. Along with the chart and the database edges, a set of context sensitive rules is applied to tokens in order to identify words and phrases that denote expressions of interest (e.g. time expressions, money expressions e.t.c.). A final chart is produced then, a chart with all active and inactive edges, plus all added edges; the latter are edges that resulted from rule invocation, they have a list of constituents and their edge number is greater than the initial edges[12].

The output of the analyser is the starting point of the annotation process; it is presented to the shallow analyser, the first stage of the Ontology Mapper. It is "a shared internal data structure via an API (application programme interface) defined by a set of LISP function calls (for accessing the chart data structure)" (Black et al. 1998)...

[12] All edges are grouped in an array, following the sequence of the respective tokens as they appeared in the text. This array has an index that enables added edges to be accessed in place of their constituents (cf. Black 2000).

2.3 RULE SYNTAX

In this section we will briefly discuss the structure of the rules written for the Named Entity Recognition task in Concerto system. In Appendix A, a full list of symbols and abbreviated attribute names is given, that helps in understanding the rules that will be presented in the second part of the dissertation. We avoided giving any example – rules that would illustrate the rule syntax in this section, because a very detailed report on the rules we have created follows. The aim of the present section is just to provide a general description of the rule format.

All rules have the following form :

A => B \ C / D or

A => B \ C / D >> E

The form denotes a production rule and separates the head from the conditions. Both head and conditions are attribute – value expressions[13]; values may be atomic, disjunctions or negated atomic or disjunctions (they have the form of a one level attribute value matrix). The head specifies the properties of the edge to be built. Syntactic and semantic properties are always specified. Of the standard attributes found in the edge-structure, only these two plus the normalisation and the antecedent slots can be referred to. All the rest are added automatically, since they depend on how the edge is derived. The zoning attribute is also necessary so as to test for accidental matches of patterns across zone boundaries of the text. Additional properties that are important for the user may be added[14]. These may enrich information for the specific edges found, or may be useful for co-reference. They can be written using any letters, digits, dashes, underscores etc not in quotes.

The condition – part of the rules may consist of one or more structures (AVMs) that precede (e.g. 'B') or/and follow (e.g. 'D') the slashes. These are the left and right hand side context respectively. Their presence is optional. Most standard properties may be used, but no additional ones, except if they have been assigned to edges found by previous rules. The attribute value matrix (or the sequence there of) that exists between the slashes is the pattern to be labelled.

[13] The developers of FACILE had attempted to use a pattern matching language, such as PERL for the rule interpreter. They preferred though the attribute – value notation for its readability and its effectiveness in handling co-reference fast… (cf Black et al. 1998)

[14] The working data structure available is not only an active chart as mentioned earlier, but also a property value table which stores the values of any attributes not in the standard chart.

The ‘ >> ’ symbol is a co-reference operator; whatever follows this operator (e.g. ‘E’) is an antecedent of the pattern of interest. This means that it has to precede the edge that the rule will identify. The antecedent has to be already processed and to share at least one value with the pattern.

Each AVM is optionally followed by an iteration specification. The iterators used may be quite general (e.g. the star (*) denotes that something may be present zero or more times) or very specific (a pair of integers denote minimum and maximum iteration e.g. {1,2}). AVMs in parenthesis form a group of edges strictly in sequence. Last, variables may be used with an underscore preceding to denote their being such; Prolog style unification of variables is allowed and it is in fact very functional.

P A R T B

THE RULE WRITING EXPERIENCE

CHAPTER 1

METHODOLOGY

Before presenting the rules we have written, we shall briefly discuss the procedure followed for writing these rules. We have used both internal and external evidence; that is, evidence from the expressions that are to be recognised themselves and from the context that appears to accompany them most of the time. The corpus examined in order to find all evidence needed was selected at random from the 'Today's news in Pharmaceutical / Biotechnology and Health field' section of the PRNewswire web site[15] (www.prnewswire.com). The specific subject area was chosen because Concerto is interested in Biotechnology texts and the rules will be used for information extraction from such texts. So, the corpus was the main guide in rule writing, since it provided contextual clues for the recognition of named entities and evidence of their properties.

The output of the tokeniser provided us with information on the orthographic features of the entities. The POS tagger supplied all syntactic information needed and the morphological analyser either confirmed and complemented this information or suggested something else. Semantic information was obtained from the database. All these assisted rule writing; attribute-value pairs, that pertained to both the tokens of interest and their context, were used. Nevertheless, relying on such information from the pre-processing stage is quite risky; sometimes, tokenising restrictions or mistaggings are encountered and inconsistencies in the database are an unavoidable problem. One has to take into consideration that these natural language processing tools do make mistakes and the rules must use their help without relying absolutely on them.

Apart from these ways of finding evidence, MUC specifications (MUC-7, Chinchor et al., August 1999) were also at our disposal. No matter whether followed strictly or not, they guided us in decisions that had to be taken and they pointed out cases of Named entities that require careful and special treatment.

Having all these in mind, we started working with one rule category at a time.

Timex and Numex rules are more easily identified from their internal constituents and that is where we started from, leaving Enamex rules last. The way we present the rules in the following sections reflects more or less the order in which they were built, an order that goes from relatively 'easier' rules to more complicated ones. However, this order is not the one suggested for the final rule set[16].

For each category, the rules that already exist in the 'Current rule file' of Concerto and the ones suggested by the UMIST Concerto team were studied; then, we attempted to build rules that would be more concise and would cover more cases Whenever writing a rule, its results were tested in a text created especially for this reason. This text contained example phrases, either from actual newswire texts or made up sentences that resembled the 'real ones', so that all cases covered by a specific rule category were gathered together. After a rule category was built, example phrases for the next category were added. Potential interaction of the rules is an important consideration; having all cases covered by rule categories together shows how a simple change or addition / deletion in/of a rule would affect the performance of the rules as a set. This text – collage is given in Appendix C; it contains examples for all rule categories.

After building the whole rule set and making sure that they work perfectly well on the sample text, a refinement phase took place. Fifty texts from the PRNewswire site were selected at random; these were the first fifty news stories of August 9th, 2000, in the field of Biotechnology. The rule set was invoked on each text separately. Mistakes or missed expressions led to changes in the rules; after revising rules on the basis of processing the first 25 texts, performance improved. In the remaining 25 texts, all mistakes or missed expressions were mainly due to inconsistencies in the database. The final, refined rule set was again tested against the text – collage, so as to see whether all cases worked after the changes.

One can see that the whole procedure is a test and refine one, a constant 'discussion' with the corpus. The rules presented in the following sections are the refined ones, that is the final version of the rule set…

---

[15] Several texts were used from this site while writing the rules. PRNewswire delivers breaking news and multimedia content.

[16] The 'Rule order' section explains the matter in detail…

CHAPTER 2

TIMEX RULES

Temporal expressions (TIMEX tag element) are of three types : DATE, TIME and DURATION. Concerto is interested only in the first two types. One may note that the boundaries between what is considered a Date expression and what a Time expression are vague. It is worth, then, presenting the MUC specifications (Chinchor et al., 1999) on the matter, which are our guides for the rule-writing: "Date is a temporal unit of a full day or longer", whereas Time is defined as "a temporal unit shorter than a full day, such as evening, minute, hour".

Before commenting on each type of Timex rules, we will discuss in detail all MUC specifications for the specific type. This will reveal our guidelines in rule writing, as already mentioned, but also cases where we prefer not to follow the guidelines because of the needs of the texts used in Concerto.

Also, in our attempt to comment on the Timex rules we propose, it has been considered helpful to give some information concerning the semantic categories in the database, that are related to them directly or indirectly :

Category	Instances
Date	Today, tomorrow, yesterday…
Date_pre	End of, all of, past, mid- …
Dateunit	Day, week, month, season, year, century..
Time	Tonight, evening, midday, dawn…
Timeunit	Afternoon, evening, night, minute, hour…
Timex_pre	Last, previous, during, later, next,**this, ago**
Time_zone	EST (eastern standard time) , **p.m, a.m** …
Event	World War II, birthday… **anniversary**
Festival	Christmas, Thanksgiving, Easter…
Fraction	Quarter, half, third, fourth, fifth…

The bold instances are words that we suggest that should be included in the database. The reasons will be explained when explaining the rules. One may observe that some words e.g. evening, belong to two different categories, their meaning and behaviour is expressed in full only when they are assigned to more than one semantic category. We will explain such cases further on. The database contains also the word "**quarter**". One of its categories is "Fraction", but it has also been assigned to the "Dateunit" category. We believe that the latter is not only redundant, but also incompatible with the meaning of the specific category and we suggest that it should be excluded from that category.

2.1 DATE RULES

The date-rules that are currently used in the BSEE analyser ("current rules") are three in number. One can easily see that they cover a very restricted number of cases; in fact the only dates that can be recognised have the format of the following examples:

- *27 October, 1999* (comma can be omitted)
- in *1999* (followed by punctuation or conjunction – "in" is not tagged conforming to MUC specifications, but its presence is essential according to the rule)
- *last October* (followed by punctuation or conjunction – "last" and other words that behave similarly are tagged as well according to MUC specifications)

Work on the date-rules has also been done by Attar (2000). She wrote about 40 rules, that deal with many cases of dates, but they are too case-specific. In fact, there is one rule for each case; considering that one needs the fewest rules possible with the best possible performance in terms of recall and precision, one cannot include all these rules in the system. Nevertheless, her project illustrates the needs that the rules must cover, the cases that have to be recognised and draws attention to problems that one encounters when writing rules. While explaining our rules we will refer to some of these rules in detail…

For the Date rules, one must have in mind the following MUC guidelines (Chinchor et al., 1999) :

- Absolute date expressions are to be tagged. Expressions of days, months, years have to indicate particular days, months, years e.g. '1999', 'October' and not relative expressions such as e.g. the 'last 10 years', 'next season', 'this year'.

As far as we understand it, this guideline has to do with the fact that the task of Named Entity recognition is just a part of the whole procedure of Information Extraction; it is the basic, first step in identifying elements of interest, categorising them and exploring the relationships between them, so as to prepare the ground for the template writing task. So, with reference to the semantic categories of Concerto's database, "Dateunits" should never be tagged as dates, because of their "relative"

denotation. We believe nevertheless, that at least cases when a "dateunit" is preceded by a word of the "timex-pre" category might be of importance. Consider the case : 'Trinity Ltd signed the…in August, last year'. If our rules tag just the name of the month, then valuable information is lost and cannot be used in a template that would be interested in dates. We suggest that the following rules should exist in the rule file[17], for use in templates :

```
  [SYN=NP, SEM=REL_DATE, ZONE=_Z] =>
\ [SEM=DATE_PRE|TIMEX_PRE]+, [SYN=NUM]?, [SEM=NUM]?, [SEM=DATEUNIT]/;

  [SYN=NP, SEM=REL_DATE, ZONE=_Z] =>
\ [SYN=NUM]?, [SEM=DATEUNIT], [SEM=TIMEX_PRE]+ /;
```

This pair of rules complements the database in a way, since it creates a new semantic category for the above expressions. We have used this semantic category in the rules that follow, in order to tag such expressions as Dates, but this is something that can easily be omitted if thought unnecessary by Concerto partners. In particular, the first rule defines an expression as 'relative date' if and only if the presence of a 'Dateunit' is preceded by one or more 'Date_pre'/ 'Timex_pre' tokens or a combination thereof. Consider for example the following cases :

'early this week', 'next four years', 'last 16 days', 'end of next month', 'mid- summer'.

One may notice that the presence of numbers in these expressions is optional. According to the tagger, all numbers both in numeric and alphabetic form belong to a syntactic category called NUM. Their difference is that numerals have a NIL value in the semantic attribute, whereas the rest have a NUM value in the semantic feature too. This common syntactic value could be well exploited, so as to write a concise rule, as one may see in the second rule. However, the behaviour of the tagger is not always predictable, even when one thinks it is. When testing the rules, the expression 'next four years' was not marked up, because the token 'four' was considered a noun and not a number. The nature of the preceding word affected the results of the POS Tagger. The semantic value of the token did not alter, so we had to add it explicitly in the rule…

[17] In the rule file provided in the appendix, an 'optional' Status is assigned to these rules to denote their character.

The second rule presented for relative dates deals with the case of a 'Dateunit' followed by a 'Timex_pre' token: *'a year earlier', 'two weeks later', 'three months ago'*
(for such cases 'ago' is essential in the database as a 'Timex_pre' token). One should note that in both cases of relative expressions, a 'Dateunit' may never stand alone not even as a 'Relative date' expression, because of the wide usage of these tokens.

- Determiners, prepositions, adverbs or other words / phrases that modify the date expressions should not be tagged (e.g. "about", "around", "in the").
- Expressions combining numerals or designators are to be tagged as a single token even in the case of possessive or partitive constructions (e.g. '*…first half of fiscal 1999'*).
- In the case of date expressions containing adjacent absolute and relative strings "only the absolute expression is to be tagged".

We object to such a case and suggest that the whole phrase should be tagged: 'last July', 'next Tuesday'. The "timex_pre" words that precede do change the meaning of the date drastically. We believe that if one had access only to the absolute part of the expression, the information extracted would be partial if not imprecise.

- Subparts of date expressions and number range expressions should be marked up separately, even if a portion of a subpart is elided e.g. 'from *1999* through *2000'* In such a case only the years will be tagged and not the whole expression. In cases such as : '*6 – 7 April'*, number 6 must be identified as a date expression and tagged separately from the expression 'April 7'. Elision must not hinder the right recognition of the element.

For these cases of elision, cases that may appear in any rule category (date, time, money , percentages, locations), we have written two general rules:

```
 [SYN=_X, SEM=_S, ZONE=_Z] =>
\ [SYN=NUM|PLACE, SYN=_X, ZONE=_Z, SOURCE!=RULE]+ /
 [NORM="-"|"and"|"or"|"to"], [SEM=_S, SOURCE=RULE, ZONE=_Z];
```

```
  [SYN=_X, SEM=_S, ZONE=_Z] =>
  [SEM=_S, SOURCE=RULE, ZONE=_Z], [NORM="-"|"and"|"or"|"to"]
\ [SYN=NUM|PLACE, SYN=_X, ZONE=_Z, SOURCE!=RULE]+ /;
```

Both these rules deal with the case of having a number (or more than one in sequence e.g. 'twenty two') that cannot be categorised semantically by a rule. There is the case of this number being followed by a hyphen, a conjunction or a preposition and by an expression that has been recognised by a rule. Then, the number inherits the semantic and the syntactic category of the expression that follows. The same case may happen when the recognised expression precedes. Consider the cases :
'April 5 and *7'*, *'50* to *60 percent'* of… *'twelve twenty* to *three p.m.'* , *'50 – 60 million dollars'* etc.

One must note that the number we are interested in must be one that cannot be recognised by an other rule. Consider the case: 'from 1999 to 2000'. The years mentioned may be recognised as dates from a rule we have written. When we examined the properties of the edge created when identifying 1999 as a date, the rule that was said to be responsible for the result was not the one that we have written for dates. Surprisingly, it was said that the above general rule was the one that found the edge. One might think that this happened because 2000 had been identified as a date, so the number that preceded inherited its semantic features. However, by looking at the properties of the edge created for "2000", we found out that this general rule was again responsible for it and not the specific one for the dates! One gets into the problem of the chicken and the egg in such a case, but one thing is certain: Because of the interaction of the rules, one must be very precise and careful when writing general rules.

We had first preferred to write in these rules that the 'connection' between the number and the identified expression could be either a conjunction, a preposition or a hyphen. Unfortunately, this may lead to undesirable markings of phrases such as: '*…50%* with *500* investors…', where the percentage expression seems connected to the following number through a preposition, but in fact it is not. So, we explicitly provide the norm of the tokens that may connect the number and the identified expression. Hyphen and conjunctions do lead to the right results and the preposition 'to' does the same as far as a thorough observation of our text corpus has proved. Except for a number token, a token denoting place may behave in the same way. The

POS Tagger recognizes as 'PROPER PLACEs' tokens such as 'south', 'west' and some known locations. We have used just the 'Place' value[18] for such tokens in order to include them in the general rules and deal with cases of elision in locations: '*North and South America'*. So, our general rules cover location cases too. If thought necessary, these rules can be expanded to deal with more cases of elision (e.g. for persons: 'Bill and Maria Jones'), if the needs of the texts demand so…

- "Special days, such as holidays, that are referenced by name should be tagged ". In fact, there are many holidays and not all of them can be included in a database. Some of them are named after the name of a Saint (e.g. St. Patrick's Day), some others just contain the word "day" capitalised, always in singular, preceded by a token known to be a 'Festival' in the database (e.g. Boxing Day). We have written a rule that deals with such 'predictable' cases of festivals, leaving the rest to the database. It is a rule that identifies some expressions as festivals. We take advantage of the presence of the word 'day' and the words considered by the tagger TITLEs (that is for words such as 'Saint' etc) :

```
  [SYN=PROP, SEM=FESTIVAL, ZONE=_Z] =>
\ [SYN=TITLE]?, [SEM=FESTIVAL]?, ([SYN=NAME], [SYN=POSS])?,
  [NORM="day", ORTH=C|A] /;
```

Now, we can see in detail one of the most basic Date rules, a rule that covers many cases of date expressions :

```
  [SYN=NP, SEM=DATE, ZONE=_Z] =>
\ [SEM=DATE_PRE|TIMEX_PRE]*,
  [SYN=NUM, GOOD-MORPH=((“Card”))]?, ([SYN=ORD], [NORM="of"])?,
  [SEM=MONTH|WEEKDAY| SEASON|FESTIVAL| DATE |REL_DATE, ZONE=_Z],
  [SYN=NUM]?,
  [SEM=TIME|TIMEUNIT]?,
  ([SYN=DET], [SYN=ORD])? /
  [SYN=PUNCT|CONJ| POSS |PREP |PROP|NN];
```

[18] The 'Proper' value may not be used instead, because capitalised tokens at the beginning of a sentence are sometimes mistaken by the tagger as 'proper' names.

To understand how the rule works, one should look at the longest line, the fourth. The core of the rule is the element described at this line. This is the only element in the rule that has no iterator, so its presence is obligatory for the expression to be marked as a date. This entails that it can be alone in a text, with none of the other tokens preceding or following it in the rule, and still be marked up as a date. This element may be defined in the database as a 'Month' or 'Weekday' or 'Season'. In that case, expressions such as: 'in *October'*, 'on *Monday'*, 'in *spring'* will be recognised as dates. This item can also be a 'Festival' semantically. It might be word present in the database and categorised as 'Festival', or a word not present in the database, but identified as a 'Festival' by the rule we have written and explained above. It may also be a token that belongs to the 'Date' semantic category of the Database: e.g. *Today*. The first four instances (month-weekday-season-festival) are proper names and indisputably form a date expression on their own. The first three, in particular, are finite sets of words, all contained in the database. The fifth category elements ('Date') are not proper names, but behave like proper names, since they are 'deictic' and do express an absolute date on their own. Lastly, we have included the case of 'Rel_Date' tokens as defined in the respective rule, since we believe that such expressions do form useful date expressions.

To start now from the beginning, the first line of the rule describes the properties of the expression to be recognised. In the second line, one deals with the case when a 'Timex_pre' or 'Date_Pre' word precedes the core element: e.g. '*next June', 'end of spring'* or when even words of both these categories precede one after another : 'The project is due *end of next June'*. At this point, we would like to explain why we suggest that the word 'this' should be included in the database in the 'Timex_pre' category. 'This' is used in date expressions, along with 'last' and 'next': last June – this June – next June. Since the other two words are included in the category mentioned, it is suggested this word be included too. If so, then phrases such as '*this June*', '*this Monday morning'*, '*end of this spring*' will be recognised by our rule. We had first considered including in the rule an optional element with the following properties [SYN=DET, SYN!=DEF|INDEF]. In this way, the word 'this' does not have to be included in the database as a timex_pre and it is also distinguished from the articles, both definite and indefinite, which also are determiners. On the other hand, this would complicate things and 'every' would not be excluded from the mark up of phrases such as 'every June'. We should not mark up this, but it would be

unavoidable since it is also tagged syntactically as a determiner. Things are much simpler with just an insertion of the token in question in the database…

In the next line, we deal with cases when a number precedes the core element e.g. *30 June*. The format (numeric or alphabetic) is of no concern since in both cases the tokens have a NUM syntactic label. However, the number must be a cardinal one, so as to avoid by chance the presence of e.g. a decimal in front of a date expression; when refining the rules we came across the following case:' …increased 4.3 compared to 1.8 last year'. There is also the case of an ordinal number plus the preceding preposition 'of ': '…the *third of July*' or even in the abbreviated format: '*3rd of July'*. Our rule deals with this case since both forms of ordinals are recognised syntactically as such. The tagger used in Concerto, recognises the first case as an ordinal number. The second, the abbreviated one, is dealt by a small rule we have written :

```
 [SYN=ORD, SEM=ORDABBR, ZONE=_Z] =>
\[SYN=NUM], [TOKEN="th"|"st"|"rd"]/;
```

The tokens appear grouped in the rule (they appear as a sequence in parenthesis) and this is not by chance, it is meaningful and should not be changed. One may think of the following case: '3rd July' with no preposition between the ordinal and the month. This is possible of course and it cannot be covered by the rule when the relative tokens are grouped. One might suggest a change in the rule then:

[SYN=NUM|ORD]?, [NORM="of"]?…….

This change, though apparently simple and innocuous it may seem, has in fact many undesirable effects; 'of' is 'free' to be marked up in a case such as 'of July' where there is no ordinal preceding. So one has to think of all possible combinations that may be allowed, before making any changes. The case of omitting the preposition after the ordinal has not been encountered in our texts: it is not a formal, standard way of expressing a date, so we will not cover it with the rule.

The fifth line of the rule deals with the case of a number following the main element of the expression : '*spring 1999', 'June 2000', 'June 30'* (the American style of expressing dates: Month – day). Combining the previous elements in the rule as well, we deal with cases such as: '*17 July 1999'*, '*end of spring 2000', '3rd of July 2002', 'Christmas 2000'* etc. Cases when a full American style date is given e.g. 'June 30, 2000' are not covered by this rule, because we do no want to recognise such an

expression altogether, according to MUC specifications. The comma is typically and essentially used to separate days from years and it is this element that would not help being part of an expression tagged as date. In such cases the two parts of the date will be tagged separately (the year will be recognised by another rule, not the present one). The sixth line covers the case of a 'Time' or 'Timeunit' word following the main element of the rule: '…on *Monday morning', '...yesterday evening',* on '*New Year's Eve'*. The next line deals with cases of the form: Month-determiner-ordinal e.g. '*June the 3rd 'Monday the fifth…'*

Lastly, we would like to explain the reasons for restricting our rule using a Right Hand Side Context. The last line of the rule denotes that the token following the date expression may be, syntactically, either punctuation, conjunction, preposition, possessive, proper name or noun. First of all, this restriction does not affect other rules we have written for dates. Secondly, the restriction is necessary for defining the end of the date expression, avoiding thus the following problems that Attar (2000) has also illustrated:

- Sometimes the name of a month/weekday/season may be the name of a female person or a name in general (e.g. of a company/product). In this case, it would be a mistake to mark up the word as a date expression.
- The word 'may' is not only the name of a month, but also an auxiliary verb. So, it should not always be tagged as a date expression. Attar solved this problem by saying in the rule that the name of the month must be capitalised. We cannot do the same, because in our rule we deal not only with 'Months', but also with 'Seasons', 'Dates' and 'Rel-dates' that are not normally capitalised.

Our answer to such problems is the right hand context. We believe that in order to surpass difficult cases, such as the above, we need to take advantage of a fundamental characteristic of date expressions. The text corpus has given the answer: punctuation. According to English Grammar (Collins, 1990), adverbial groups, prepositional phrases and noun groups that talk about the circumstances of an event / situation are normally either in the end of a clause or in the beginning for emphasis. "The adjunct is often separated by a *comma* from the rest of the clause". Even more, Collins Cobuild Grammar, a corpus-based descriptive grammar, notes: " if there are more than one adjunct in the sentence, their usual order is manner-place-time and comma usually separates them", which brings the date expressions to the end of the sentence again (thus a full-stop follows).

This is the reason why we believe that saying punctuation must follow a day expression can lead to successful recognition of these expressions. Names and auxiliary verbs are hardly ever followed by punctuation. The exceptions have to do, according to our test runs with the newswire texts, with cases when, after a date expression, one finds a preposition (e.g.…in *June* **for** the conference…), a conjunction (..in *June 6* **and** 7…), a possessive (e.g. in *today*'**s** issue…), a proper name (e.g. The *July* **Pirax** Index…), or noun (for cases such as 'the *autumn* **report**').

At this point, we should mention a dysfunction of the above rule. The problem is that not all words that belong to the 'Festival' semantic category are found in the texts only in their capitalised tokens. This means that a Proper name denoting a Festival, when capitalised, may also be a common noun, when in lower case, e.g. 'Assumption' – 'assumption'. The database entries are not case-sensitive, so help from the database for avoiding the confusion cannot be obtained. In our rule, whenever a 'Festival' word is found in the text, it is marked up as a 'Date'. So, the word 'assumption' will be wrongly marked up as a date expression. We cannot use the ORTH attribute in the rule and ask only for capitalised instances of Festivals, because the rule covers also cases of 'Date' elements and 'Seasons' that are not always capitalised. We have just preferred to delete the 'Assumption'-'Festival ' pair from the database, because it is really hard to find such a festival mentioned in our texts. If one would like to avoid this deletion, one may just write a separate rule for 'Festival' saying that they have to be capitalised in order to denote a date expression.

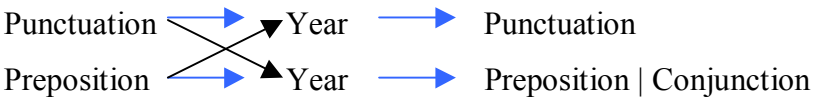
Our next rule deals with date expressions, containing mainly a number that denotes a specific year :

```
 [SYN=NUM, SEM=DATE, ZONE=_Z] =>
 [SEM=DATE_PRE|TIMEX_PRE]*
\ [ORTH=1, GOOD-MORPH=(("Card")), NORM<2100, TOKEN="????", ZONE=_Z],
 [NORM="s"]? /
 [SYN!=PL|NUM|ADJ];
```

This rule attempts to recognise years that are reported in a text on their own, without any other sign that one talks about a date; no other sign except real world knowledge. Whatever the previous rule cannot recognise as a year (because of the obligatory

presence of the name of a month/season/festival, etc.) is identified by this one. We have to deal with a four-integer number, whose right context may be anything else other than an Adjective, another Number or something in Plural. If we did not use the right hand context as a 'guard' for our rule, then the rule would recognise as a date any number consisting of four digits. Consider the following mistakes that can now be avoided : '*1550* million people' 'earned *2089* points' 'out of *1000* votes…'. With these restrictions on what follows a date expression of this kind, we do not exclude numbers used as modifiers of a noun or adjectival phrase. If a four digit number precedes such phrases, then there must be no agreement between the number and the number of the phrase, if we are to talk about a date expression e.g.: 'The new Pirax *2000* Index', 'the company's new *1999* report …'. In these cases, the number is a date. If the following nouns were in plural, then one would not talk about a date expression… Punctuation does follow this case of date expression too and in fact quite often:
'due *1999*', '(Smith A., Investments, *1992*)', 'in *1999* for the conference..'
Our rule can also cover the following cases : '*end of 2002' 'early 1999' etc.*

While trying to form a rule that would be asaccurate as the one described above, we made some interesting observations. The corpus we were looking at gave us evidence for creating the following scheme:



More often than not, a year was between punctuation symbols or prepositions, or a combination of them (preposition-year-punctuation or punctuation-year-preposition|conjunction). We then thought of denoting in the rule both the Left and the Right hand contexts. However, then we found phrases where the year-date did precede a noun to which it gave a temporal specification e.g. '…the 1998 US National Medal…'. So, we arrived at the above rule, which denotes only the right hand context. Attar (2000) had another rule for the recognition of years: saying that the number must be >1900 and <2020, since years that belong to this range are found in our texts. Nevertheless, this is too general and it is not always the case. The right hand context is essential, so as to avoid mistakes. Nevertheless, we use the restriction <2100, for cases of 'Artifact names': 'Clarity 7000 system'. In such cases the number does not denote a

date expression. It is not often the case, but we came across this during the refinement of the rules. One more thing that only test runs could have pointed out is that the number in question must be a cardinal number. We came across the following case: '£0.15' where the decimal number was marked up as a date from our rule. Though a rule for money expressions covers this case, the date rule precedes and identifies the four digits numeral as a date. So, the morphology is our solution to the problem… Our rule covers also cases where one refers to a decade using a numerical expression e.g 'in the *1960s*', 'in the *late 1960s*'. One may note that people usually refer to decades by abbreviating the year: 'in the '60s', but this is not the case in our texts, which follow the conventions of formal writing.

The next rule is one that deals with abbreviated dates:

```
 [SYN=NUM, SEM=DATE, ZONE=_Z] =>
\ [ORTH=1, ZONE=_Z, TOKEN="??"],
 ([TOKEN="-"|"/"|"."], [ORTH=1, TOKEN="??"]){1,2} /;
```

We have chosen to cover only the abbreviations of the standard format that are most commonly found in texts: two digits, hyphen or slash or full stop, two digits and again possibly a separator of the above kinds and two digits e.g. 03–05-99. Sometimes, one may also find something like'1-3-1999' or '1-30-99', but in formal texts the standard format is preferred. Some other times telephone numbers may be included in the text or numerical names of products may contain slashes and hyphens; to avoid confusion, we have restricted the number of digits contained in the date abbreviation to two for each value (day, month, year). The main problem with such cases has to do with the tokenizer used in Concerto. If there is no white space before and after the hyphen/slash /full stop, then the abbreviated date is considered a single token by the tokenizer, having the syntactical label NUM. Then, the above rule does not work, because we explicitly say that the hyphen is a separate token. However, these cases are rare, rarely does anybody leave a space in between. So, we have modified our rule, so as to recognise the most frequent cases of abbreviated date expressions:

```
[SYN=NUM, SEM=DATE, ZONE=_Z] =>
\ [SYN=NUM, ZONE=_Z, TOKEN="??-??-??"]/;
```

By modifying the rule we cannot deal with cases when someone uses something else instead of a hyphen, or when one does not refer to the year as well e.g. 08-09.
Nevertheless, we propose that only this rule should be in use in the rule file, because the abbreviated dates do not occur that often in our texts and when they do they have the format already mentioned, the one the modified rule can recognise. Exceptions to this format are not frequent enough, to justify separate rules for their recognition.
Attar (2000) has suggested a token of the form " *-* " in the rule, which we find too general. Such a rule would mistake the number '0161-2254733', which is a telephone number, for a date.

The following two rules are very simple and may optionally be included in the date rules category, if we are to cover a really large variety of cases.

```
  [SYN=NP, SEM=DATE, ZONE=_Z] =>                    (instances: …'end of World War II')
\ [SEM=DATE_PRE], [SEM=EVENT, ORTH=C|A] /;
```

```
  [SYN=NP, SEM=DATE, ZONE=_Z] =>       (instances: '21st century', 'eighteenth century')
\ [SYN=ORD], [NORM="century"] /;
```

Well-known events may form a date expression only when preceded by a 'Date_pre' word. We note the fact that the event must be capitalised because there are words in the database such as 'birthday' that are included in the 'Event' semantic category.
'Century' is the only 'Dateunit' that when preceded by an ordinal, forms a date expression and in particular an absolute date expression. So it has to be treated in a separate rule…

Lastly, we have written a rule mainly for financial quarters or halves of a specific year. The following examples are characteristic: '*last quarter of 1999*', '*first half of the fiscal 2000*' ' *second half of eighteenth century*' etc. By the way, cases such as: *50th anniversary of the end of World war II*, '*4th of July*', '*first of June*', etc. can also be recognised ;

```
  [SYN=NP, SEM=DATE, ZONE=_Z] =>
  [SYN=PREP|DET]+
\ [SEM=NUM|ORDABBR|EVENT]+, [SEM=FRAC]?,
```

```
[SYN=PREP|DET]*,[NORM="fiscal"]?,
[SEM=DATE, SOURCE=RULE, ZONE=_Z] /;
```

Because of the fact that such cases are usually preceded by a determiner or preposition and sometimes by both, we have defined a Left hand side context, to prevent any chance collocations that will not form a date expression (e.g. 'he came *third last year')*. The core of the rule is the last line, the expression that has already been recognised as a date. This date expression may be preceded by a number or ordinal or event whose presence is obligatory. We have to explain at this point that by 'event' we mainly imply the word 'anniversary' that is present in date rules quite often. One may find more of these elements in sequence e.g. 50^{th} anniversary = ORDABBR and EVENT (this is why the '+' iterator has been used). A fraction may be adjacent to Numbers / Ordinals, a preposition and /or a determiner, or the word 'fiscal' optionally.

2.2 TIME RULES

The current rule file of the Concerto BSEE analyser does not contain any rules for time expressions at the moment. Attar (2000) has written 9 time rules. We propose the following four rules that cover a wide variety of cases. The MUC specifications (Chinchor et al., 1999) discussed above are valid for this category as well. In addition, we note that :

- Both numerals and time–designators must be tagged as single tokens (e.g.'*19:30 p.m.*').
- "Expressions of minutes must indicate a particular minute and hour" (e.g. '*20 minutes after ten'* not 'a few minutes after the hour').
- "Expressions of hours must indicate a particular hour" (e.g. '*midnight*', '*twelve o'clock noon*', not 'morning' or 'mid-day' by themselves).

We believe that this guidance is very strict. Expressions such as 'in the morning' occur quite often in texts and reveal useful information. Taking into consideration that texts in newswires do come out every minute on a 24 hour basis and that people interested in such texts do read them the same day they come out, one understands why such relative expressions of time should be of interest to us.

- Locative Entity strings embedded in expressions of time must be marked up together with the numerals and the same is true for the word 'time' that usually follows (e.g. *'1:30 p.m., Chicago time'*). In the rules that follow, we suggest the locative entity strings to be marked up separately, since they may be embedded in all cases of time expressions and it is more useful not to include punctuation that usually exists in between the time and the location indication.

Our first rule deals with numeric time expressions : '2:50 p.m.' '10 a.m.', '13:30', '1:30 p.m. EST'...

```
  [SYN=NUM, SEM=TIME, ZONE=_Z] =>
\ [SYN=TIME], [SEM=TIMEZONE]* /;
```

The tokeniser recognises as TIME numerical combinations of the form:
Digit {1,2} : Digit {2} e.g. '12:30'. These tokens may be found by themselves with no other time designators, so they can form a time expression on their own: 'the conference will be held at *13:00*'. We have suggested that the 'p.m.' and 'a.m.' abbreviations should be included in the database because they are used very often and they are precise designators of time expressions. We have chosen the 'Timezone' semantic category because they are abbreviations; their meaning has nothing to do with time zones (that are geographically defined), but they do divide the day in zones. The way they are used is closer to the 'Timezone' instances.

Our next rule deals with the case of expressions whose core element is a word of the 'time', 'Timeunit' or 'Timezone' category. The presence of such an element is essential for the expression to be recognised. All other elements in the rule are optional. This means that the core element may form a time expression on its own. Indeed, words such as 'tonight', 'morning', 'night', do express time: '…in the *morning*'. However, these specific categories in the database contain some other words as well e.g 'hours', 'minutes' that are not always used in time expressions and even when they are, they do not stand alone, without a numeral. To prevent such cases from being marked up as time expressions, we define that the token in question must not be in the plural.

The context also helps. A determiner must not precede (consider the case 'the/every hour' which is not a date expression) and a noun or adjective may not follow ('the one hour lecture' denotes duration, not time; 'second line' –second is a 'Time' token, but here it does not express time). In fact, the case is similar to the date expressions; hardly ever is a time expression followed by something other than punctuation… One may note that the tokens that belong to the 'Timezone' category do not form time expressions by themselves; this is true, but what is also true is that such tokens are never alone in the text. There is no point in seeing a 'Timezone' unit without a numeral preceding…One may also find a case where the time expression has both a 'Timeunit' and a 'Timezone' token: 'at *eight hours EST'* so the iterator '+' is essential…

```
 [SYN=NP, SEM=TIME, ZONE=_Z] =>
 [SYN!=DET]
\ [SEM=TIMEX_PRE]*,
```

```
([SYN=NUM], [SYN=PREP], [SYN=DET])?,
[SYN=NUM]*,
[SEM=TIME|TIMEUNIT|TIMEZONE, SYN!=PL, ZONE=_Z]+ /
[SYN!=NN|ADJ];
```

The rule covers cases when a timex_pre word precedes the time unit (e.g. '*this morning*', '*last evening'*) and when a numeral followed by a preposition and a determiner precedes (e.g. '*8 in the morning'*, '*nine in the evening'*). In the latter case, we have a group of tokens in sequence. All are defined by their syntactic value. We have not preferred to use a disjunction of the form: [SYN=NUM|PREP|DET]* which could result in a sequence of all three because of the iterator, since by doing so their sequence is not obligatory and cases when just e.g. a preposition only would precede the core element would be marked up altogether: '*at night'* (but the preposition should not be tagged too).

Our last time rule recognises cases when time is not expressed in an electronic format. Instances: 'a *quarter to nine*', '*half past seven'*, '*five minutes to eight*, '*8 o'clock*', '*nine o'clock*'…This format is not often found in texts, so its inclusion in the rule file is optional, depending on how exhaustive one wants to be.

```
  [SYN=NP, SEM=TIME, ZONE=_Z] =>
\ [SYN=NUM]?, [SEM=FRAC|TIMEUNIT]?,
  [NORM="past"|"to"|"o'clock"], [SEM=TIMEUNIT]?, [SYN=NUM, SEM!=DATE]? /
  [NORM="."|","|"and"|"or"];
```

The core is the token "past | to | o'clock" . The last token is never found on its own, so it will never lead to a wrong recognition of a time expression. For the other two, the right hand context guards them from being marked up whenever they occur in a sentence. The number that follows them must not have a semantic label 'Date'. This restriction is necessary for cases such as: 'from *1999* to *2000*'. In this case, the years are marked up as dates by our date rules. However, the present rule imposes a time label that will include the two years with the preposition in between, if and only if the semantic features of the numbers involved have not been defined by another rule… The status of this rule is optional, since such time expressions are not often found in formal writing.

The last rule deals with cases when the timezone is given indirectly with the word ‘time’ preceded by an adjective e.g. ‘local, ‘international’, or a proper noun e.g. ‘New York City’: 'local - time', 'Greek time', ‘Crooks Valley time'….

```
  [SYN=NP, SEM=TIME, LOC_INDICATION=_L, SPECIFIC_TIME=_S, ZONE=_Z] =>
  [SEM=TIME, SOURCE=RULE, TOKEN=_S,ZONE=_Z], [NORM=","]?
\ [SEM=LOC|COUNTRY_ADJ, TOKEN=_L]?, [SYN=ADJ, TOKEN=_L]?,
  [NORM="-"]?, [NORM="time", ZONE=_Z] /;
```

The left hand side context helps first of all, in marking up this time expression no matter what type of time indication precedes; this way we do not repeat the above pattern in all time rules. We avoid marking up COMMAs between the actual time and the locative string and last, we avoid having the word ‘time’ marked up whenever it occurs in a text on its own. The rule makes use of elements that belong to either the ‘Location’ semantic category or to the ‘Country_adj’ one. The latter is in the database whereas the former is defined by location rules[19].

[19] This entails that the location rules must precede the time rules in the actual rule file, but the order of the rules is a matter that will be discussed later on.

CHAPTER 3

NUMEX RULES

According to MUC specifications (Chinchor et al., 1999), the NUMEX rules are of four types: Money, Measure, Percent and Cardinal. Concerto is interested only in Money rules, that is rules for monetary expressions and Percent rules, rules for fractions expressed in terms of hundredths. The guidelines discussed for previous rule types are valid for this type as well, when applicable. Additionally :

- " The word 'minus' or the minus sign should be included in the tagged numeric expression, if it is a negative value".
- Juxtaposed strings expressing values in two different currencies are to be tagged separately (e.g *'£29 million* (*$43.6 million*)').
- "Modifiers that indicate the multiplied value of a number unit should be included in the tagged string", but no other modifier or approximator may be included.

The semantic categories that appear in our rules, together with some instances and some words that we suggest be included in the database are the following :

Category	Instances
Currency_abbr	Bef, ffr, euro…
Currency_unit	Dollars, francs, pesetas.. **pounds**
Money	Percent, per cent , per-cent…
Percent	Million, thousand, hundred, **billion**

3.1 MONEY RULES

In the current rule file, there are eight rules for money expressions. Attar (2000) has written 6 rules, which do not cover all the cases needed. We have arrived at just two rules that cover a wide variety of money expressions. In our rules, we use two rules from the current rule file which define a CARD-NUM semantic category. Since this category of expressions is used in the rules we would like to present and explain these rules briefly :

```
# Big cardinal numbers
  [SYN=NUM, SEM=CARD-NUM, NORM=(* _N1 _N2), ZONE=_Z] =>
 \ [SYN=NUM,NORM=_N1,ZONE=_Z],
  [SYN=NUM,NORM=_N2,ZONE=_Z] / ;

# Big cardinal numbers
  [SYN=NUM, SEM=CARD-NUM, NORM=(* _N1 _N2), ZONE=_Z] =>
 \ [SYN=NUM,NORM=_N1,ZONE=_Z],
  [NORM=_N2,NORM=1000|1000000|1000000000, ZONE=_Z] / ;
```

These two rules define which tokens may be assigned to a Cardinal number semantic category. The first one deals with numbers of numeric or alphabetic form that come one after another. An additional property 'norm' is given to them, whose value is the product of the multiplication of the two numbers e.g. 'two hundred' = 2 * 100 = 200.
The second rule deals with bigger numbers that it also normalises. The second must be either 'thousand' or 'million' or 'billion' e.g. '120 millions = 120*1000000=120000000. We have to note that the multiplication of two variables (lisp notation) is restricted only to numerals…

Our first rule deals with cases when the money expression consists of a currency sign (usually pounds or dollars in our texts) followed by a cardinal numeral of any format (normalised or not). The 'norm' of the currency sign instantiates the variable that specifies the name of the currency (e.g. £ = pound sterling) and the 'norm' of the number that follows specifies the amount in question. These tokens may be preceded by a minus sign or the word 'minus' or they may be followed by an abbreviation for millions or billions:

```
  [SYN=NP, SEM=MNY, nkrl=(spec. money_ Curr (spec. amount_AMT)), ZONE=_Z] =>
\ [TOKEN="-" | "minus"]?,
  [NORM=_Curr, TOKEN="£"|"$", ZONE=_Z],
  [SYN=NUM,  NORM=_AMT, ZONE=_Z],
  [TOKEN="m"|"bn"]? /;
```

Instances: ‘ *$ 2.6 million* ‘, *‘£400 bn*’, *‘£175,000*’, *‘- £50.3 m’.* We define the currency signs that may occur, because there is no syntactic or semantic category to describe them with. The dollar sign, is considered syntactically ‘punctuation’ by the preprocessors, whereas the pound sign is labelled as ‘Proper name’. The latter is of course quite peculiar. We have found out that the dollar sign is defined as punctuation denoting money, whereas the same does not happen for the pound symbol; the rules of the POS tagger have not covered the case of the ‘Pound currency unit’. While refining the rules we came across the following cases : ‘$.09’, ‘$(0.06)’. These two cases cannot be dealt with by our rule because the tokeniser considers the sequence ‘currency unit – decimal or parenthesis decimal’ as a single token. To deal with these one should write specific separate rules, but since the cases are extremely rare, it is not worth doing something like that, at least for the moment.

The next rule marks up expressions that have a currency unit or currency abbreviation token (alphabetic form). This may be preceded by the word ‘several’ that modifies the expression by multiplying its value, or by a minus sign, a number of any form (normalised or not, in alphabetic or numeric form), or even a country adjective that may probably specify the ‘nationality’ of the currency (e.g. Cypriot pound). Consider the expressions: ‘*200 million pounds’*, ‘*several thousand dollars’*, ‘*three hundred billion drachmas minus’*, *‘-200,000 Cypriot pounds’*…

```
  [SYN=NP, SEM=MNY, ZONE=_Z] =>
\ [NORM="several"]?, [TOKEN="-" | "minus"]?,
  [SYN=NUM]?,[SEM=CARD-NUM|NUM, ZONE=_Z]*,[TOKEN="m"|"bn"]?,
  [SEM=COUNTRY_ADJ]?,
  [SEM=CURRENCY_UNIT|CURRENCY_ABBR, ZONE=_Z], [TOKEN="-" | "minus"]? /;
```

3.2 PERCENTAGE RULES

There is only one rule for percentages in the current rule file that marks up cases such as : ‘30%’, where the sign for percentages is used. In these cases, both the number and the sign are considered one single token, a ‘percent syntactically’ whose root is the number on its own :

```
  [SYN=NUM, SEM=PCT, VALUE=_R, ZONE=_Z] =>
\ [SYN=PERCENT, ZONE=_Z, ROOT=_R] /;
```

We have also written a rule for the expressions that consist of the word ‘percent’ itself e.g. ’30 percent’, ’30 per cent’, ’30 per-cent’…

```
  [SYN=NUM, SEM=PCT, ZONE=_Z] =>
\ [SYN=NUM], [SEM=PERCENT] /;
```

CHAPTER 4

ENAMEX RULES

The Enamex tag element is assigned according to MUC Specifications (Chinchor et al., August 1999) to main named entities, in particular proper names, acronyms and other unique identifiers, categorised in the following three types:

- Location: "name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.)"
- Person: named person
- Organisation: "named corporate, governmental, or other organisational entity".

Concerto is interested in all three categories and, in addition to them, an 'Artifact' category is also of interest. This last category deals with names of products, with 'Trade-names'. The texts analysed in Concerto contain such proper names and their being marked up is of great importance. We have to note that Organisations and Artifacts are the most domain-dependent named entities in terms of structure and behaviour. They are quite unpredictable and the rules have therefore to be very specific. Some guidelines that pertain to all Enamex types are given for MUC (Chinchor et al., 1999) and one has to note that they help a lot in deciding what will be marked up as a Named Entity and what not:

- "Taggable multi-word strings will contain entity name sub strings: such multi-word strings are not decomposable; therefore, the strings are not to be tagged": e.g. 'Arthur Anderson Consulting' – no mark-up for the name alone.
- "In a possessive construction, the possessor and possessed Enamex sub strings should be tagged separately". e.g. *>California< 's >Silicon Valley< (Both Locations)…>Canada< 's >Parliament< (Location and organisation respectively)*

4.1 PERSON RULES

The current rule file contains three active rules for the mark up of person names. These rules cover the following cases:

- A person with a known first name.
- A person with a title or preceding occupation.
- A person with a title and preceding occupation and surname only.

These rules have been refined by Concerto partners so as to cover possible presence of initials or appositives (e.g. 'Jr., Sr.'). Rules have also been suggested for the following cases / patterns :

- People with unknown first names may be identified by a following title or company position E.g. '*E. Kiro Arington,* PHD', '*Dolan Geaney*, Elan's Chairman'…(Thompson, 2000)
- Company's company position, person E.g. 'Duramed President and Chief Executive Officer, *Dolan Geaney*'…(Thompson, 2000)
- People join companies E.g. '*Tamar Howson* joins NPS Pharmaceutical's Board of Directors'… Thompson, 2000)
- Companies / Boards announce the appointment of people E.g. 'Redo Pharmaceutical Corporation announces the appointment of *Jorgen Borg* as director of Marketing'…(Thompson, 2000)
- Companies name people / people are named E.g. 'PRO-West names *Traudel Altmann* President and CEO'…(Thompson, 2000)

We have a total of ten rules that cover some interesting cases of person identification. They all give significant evidence on collocations found in newswire texts, that involve person names. However, they are quite complicated and case specific. They take into consideration not only the immediate context of the tokens to be marked up, but also a wider context either preceding or following. They presuppose rules for distinguishing company positions from companies and rules for organisations that have to precede in the rule set…Therefore their performance in actual texts is questioned. Based on clues that these rules have provided us with, we suggest eight rules that cover most of the above cases and some more, without burdening the rule file with presupposed – complementary rules.

Before explaining these rules, we prefer to present the MUC specifications that guided our rules and the database-semantic information we used :

- Titles such as 'Mr' or 'President' are not considered part of a person name. However, appositives such as 'Jr.', 'III' will be marked up.
- Occupations are not to be tagged

The MUC specifications (Chinchor et al., 1999) are much more detailed, they include for example guidelines for fictional persons and names of animals, but, for the needs of Concerto, only the above have been considered relevant.

The database has the following semantic categories that appear in our person rules:

Category	Instances
Occupation	Secretary, Chairman, **Director, CEO**…
Person_ambig	Valentine…
Person_female	Maria, Sue, Jane, Catherine, ~~Sahara~~…
Person_full	Bill Clinton, Mao, Buddha…
Person_male	John, Bill, Don, George, ~~Von~~
Title	President, **PH.D, M.D., D.PHIL**…
Title_female	Miss, Mrs, Ms, Madam, Lady…
Title_male	Mr, Sir, Lord…
Title_mil	Captain, Sergeant …
Title_modifier	Chief, ex, former,

We have to note that the words in bold are suggestions for additions in the database. The 'Occupation' and 'Title' categories should be enriched, so as to cope with the needs of Concerto texts. Some instances of 'Person_male and female' categories are quite funny. Sahara is included in these categories; however, it can hardly -if ever- be found as a person name. We suggest that peculiar names should not be included in the database; the rules can help considerably in their recognition as such. 'Van' and 'Von' are usually middle names (of Dutch and German people) and

not frequent male names. We suggest their exclusion from the database. The rules can cope with their presence in person names.

Our first rule trusts the database for the marking up of the names of well-known people. The 'O' value for the ORTH notations has been included as well, for cases of Surnames that contain an apostrophe e.g. "O'Neil" or mixed lower case and capital letters e.g. "McDonald"…

```
  [SYN=PROP, SEM=PER, ZONE=_Z] =>
\ [SEM=PERSON_FULL, ORTH=C|A|O, ZONE=_Z] /;
```

The next rule deals with cases of finding the first name in the database. This is a very common case, since the first name is usually reported in formal writing and the database contains hundreds of possible first names :

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, ZONE=_Z]
=>
\ [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [SEM=PERSON_AMBIG|PERSON_FEMALE|PERSON_MALE,ORTH=C|A,
   TOKEN=_F, ZONE=_Z],
  [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [SYN=NAME, TOKEN=_M]?,
  [ORTH=C|A|O, SYN=NAME, TOKEN=_S, ZONE=_Z],
  ([SYN=COMMA], [NORM="jr."|"sr."])? /;
```

The rule deals with the possible presence of initials before the first name or after it e.g. '*J. Franklin Jones'*, *'Peter N.C. Kellogg'*. The tagger assigns a NAME syntactic value to such tokens, so we take advantage of it. The same value is given to Middle names such as 'Van, Von, De' and other proper names that follow known first names. We have preferred to use a separate slot for 'Middle name' in the rule, instead of explicitly saying that the above three words may be present in the name, in order to achieve wider coverage and functionality. This slot can also be filled by a possible second name one may have or a second surname. So, the following cases can all be marked up: *'Karl de Shutter'*, *'John Von Armstrong'*, *'Maria Helena Salvadora'*, *'Maria Pino Marino'*, *'Peter N.C. Kellogg, Jr.*', *'Mona K. Van Shutten'*. In this rule we do not care for the presence of a 'Title' before the name; since such an element

does not form part of the name and since it does not help –in this case- for the recognition of the person name, it would be redundant to refer to its possible presence.

The next rules are for cases when the first name is not known from the database or it is not present at all. The first of them has mainly the structure of the previous rule, but the presence of an apposition is essential. In fact, it is this apposition that designates the person name as such : *'Panayiota N.C. Kellogg, Jr.*'

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, ZONE=_Z]
=>
\ [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
  [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [SYN=NAME, TOKEN=_M]?,
  [ORTH=C|A|O, SYN=PROP, TOKEN=_S, ZONE=_Z],
  ([SYN=COMMA], [NORM="jr."|"sr."]) /;
```

We have to note that since the first name is not known, the tagger does not necessarily identify it as a NAME, it only gives a PROPER value. The same is for the last name. This is the reason why, in the person rules from now on, a SYN=PROP attribute - value pair is asked for and not a SYN=NAME one.

One more rule that can lead to absolutely correct mark up is the one that seeks for the presence of a 'Title' designator before the name : 'Mr *Ahmed Sahran'*, 'Dr *Kalia Mertis'*, 'Miss *Cass*, Captain *G.P. Pastras* etc.

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, ZONE=_Z]
=>
  [SEM=TITLE_MIL|TITLE_FEMALE|TITLE_MALE, ZONE=_Z]
\ [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
  [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [SYN=NAME, TOKEN=_M]?,
  [ORTH=C|A|O, SYN=PROP, TOKEN=_S, ZONE=_Z, SOURCE!=RULE] /;
```

The next rule is based on the presence of words such as 'said', or 'stated' which are followed by a person name, in cases of reported speech. For more accurate

results we have defined a right hand side context; punctuation should follow the name. Our corpus has given strong evidence for the presence of punctuation after such expressions. Additionally, we have defined that the surname token, should not be a word that belongs to the 'Title' or 'Occupation' semantic category; sometimes the occupation or title of the person who spoke is given and not his/her name e.g '…said *Kalia Marino*…' but not 'stated Aviron President '…

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, ZONE=_Z]
=>
 [ROOT="say"|"state"]
\ [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
  [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [SYN=NAME, TOKEN=_M]?,
  [ORTH=C|A, SYN=PROP, SEM!=OCC|TITLE, TOKEN=_S, ZONE=_Z,
   SOURCE!=RULE] /
  [SYN=PUNCT];
```

The rules that follow do not lead a hundred percent to the right markings. According to our corpus, they do find the person names with good precision; nevertheless, inconsistencies in the database may also lead to the marking of tokens that are not actually person names:

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, OCC=_O,
  ZONE=_Z] =>

\ [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
  [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [SYN=NAME, TOKEN=_M]?,
  [ORTH=C|A, SYN=PROP, TOKEN=_S, ZONE=_Z, SEM!=OCC|TITLE,
   SOURCE!=RULE] /
  [NORM=","|"as"|"the"|"to"]+, [SEM=OCC|TITLE|TITLE_MODIFIER, TOKEN=_O];
```

The above rule deals with cases such as : '...*Mania Bellini* as chief...', '...*Nisha Sahran*, PHD...', '...*Kiro Nakasian*, Aviron's...'. In these cases, the name is followed by

an occupation or a title. When a 'Title' follows, the recognition of the person name is a hundred percent certain, but the same is not so with the 'Occupation' category. Consider the case: '…The President, executive officer and…'. Here, the word 'President' would be identified as a person name, if the rule did not define that the surname must not be an 'Occupation' or a 'Title'. One more example that we came across when refining the rules will prove how essential such restrictions are: "Pharmacopia's Chairman, President and CEO…". Initially, the word 'Chairman' was marked up as a Person expression; to deal with it the above mentioned restrictions had to be added in the rule…We believe that even these rules that are not absolutely accurate can perform extremely well, given some small refinements, that the corpus will dictate.

The last rule is for quite specific but frequent cases: '...president, *Lania Marita*,..'. Usually, after a token that denotes an occupation, a name enclosed in commas follows; it is the name of the person whose occupation was mentioned. It is a case of apposition and the rule strictly asks for the specified left and right context. To avoid cases of enumeration instead of apposition we restrict the token to be marked up as first name; it has to be unknown and to have a NIL value after the morphological analysis. We came across the following case which led us to such a strict restriction : '…president, Vice Chancellor,…' where 'Vice Chancellor' had been marked up as a person expression. In this rule, as well as in the above, the occupation is an added value to the properties of the expression that will be tagged.

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, OCC=_O,
  ZONE=_Z] =>
 [SEM=OCC, TOKEN=_O, ZONE=_Z], [NORM=","]?
\ [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [ORTH=C|A, SYN=PROP, MORPH=(("NIL")), TOKEN=_F, ZONE=_Z]?,
 [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [SYN=NAME, TOKEN=_M]?,
 [ORTH=C|A|O, SYN=PROP, TOKEN=_S, ZONE=_Z, SOURCE!=RULE] /
 [SYN=PUNCT];
```

The person rule set concludes with a co-reference rule. The co-reference is based on the surname of persons already identified. Sometimes, a person may be referred to just with their first name. In formal writing, in official reports, this is rare; it

may be found in cases of direct speech presented in quotes in the text, but we do not consider this case important enough to create a separate co-reference rule just for this. Through the co-reference rule, the person name marked up is related to its full form :

```
  [SYN=PROP, SEM=PER, FULL_FORM=_FF, ZONE=_Z] =>

\ [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
  [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [SYN=NAME, TOKEN=_M]?,
  [ORTH=C|A|O, SYN=PROP, TOKEN=_S, ZONE=_Z, SOURCE!=RULE] /
   >>
  [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S,
  SOURCE=RULE,    TOKEN=_FF, ZONE=_Z] ;
```

4.2 LOCATION RULES

The current rule file contained only one location rule; this rule simply trusted the database for the mark up of countries as locations. A few more rules have been suggested that deal with the following cases:

- USSTATE / Province confirmed by the database (McNaught, 2000)
- A proper name followed by comma and a location may be a location too (apposition) (McNaught, 2000)
- A rule to mark up the whole apposition (e.g. 'Cambridge, Massachusetts') (McNaught, 2000)

These rules do not cover basic location expressions, so they have to be enriched and complemented. The apposition case is indeed a case one may take advantage of in order to denote many location expressions that would otherwise be missed. We suggested a broader rule that deals with this case. We would also like to note that the marking up of a whole apposition seems redundant. It is of course certain that one would like to preserve the information an apposition carries, so as to use it in template building. Nevertheless, this does not necessarily require a separate rule, a rule that will mark up the apposition as a single location expression; we suggest in our rules that the parts of the apposition should be identified as such separately, but the information that one location is part of a larger one will be preserved in a 'Greater_region' property that we assign to the subordinate location. The explanation of the rules that follows will make things more clear…

Before presenting the rules though, one should be acquainted with the MUC specifications (Chinchor et al., 1999) followed -or not and why - :

- First of all, locations include: named heavenly bodies, continents, countries, provinces, counties, cities, regions, districts, towns, villages, airports, military bases, railways, …oceans, seas, straits, bays, rivers, islands, lakes, mountains …
- "Locative entity expressions listed in succession, with or without a separating comma, are to be tagged as separate instances of Location". E.g. '*Washington*', '*D.C'*. Here one sees that the guideline support what we mentioned earlier for the case of appositions.
- Locative designators and specifiers (e.g. river, city) are to be tagged if integrally associated with a place name, as part of the name. According to the guidelines,

such designators form part of the name mainly when capitalised. When they are not capitalised, they may be marked up only if the extent of the location name "cannot be determined from world knowledge or from the appropriate reference atlas". We have decided to use the capitalisation criterion as our main criterion for such cases in the rules.

- Sub-national regions, when referred to only by compass-point modifiers, are not to be tagged up as location expressions e.g. 'the northeast', 'the Southwest region'.
- "Directional modifiers ('north', west'…) and combinations thereof are taggable only when they are intrinsic parts of a location's official name as in 'North Dakota'.
- Adjectival forms of location names are not to be tagged as locations e.g. 'American exporters'.

Apart from these general principles that characterise our rules this specific category of named entities involves the bulk of information found in the database. A brief report of the semantic categories used is considered essential for understanding the rules. The table that follows may help :

Category	Instances
Airport	Hellinikon, Gatwick, Heathrow, ~~National~~
City	Athens, London, Paris, ~~Center~~
City_part	City, fort, great, san, station, town…
Continent	America, Asia, Australia, Europe…
Country	Greece, United Kingdom, Germany…
Country_adj	Greek, British, German…
Country_part	Republic of, federation of, territories, **democratic republic of**…
Facility	Airbase, barracks, base
Facility_head	Airport, stadium, headquarters, **railway**
Island	Crete, Malta, Rhodes, Zante…
Landregion_head	Area, coast, beach, desert, district, valley, mountain, island, hill…

Location_prefix	Camp, cape, coastal, continental, east, new, los, lower, upper, porto, puerto
Province	Columbia…
Province_head	County, district, province, state…
Region	Andes, Australasia, Caribbean, Balkans
Region_head	Region, belt, ~~countries~~
Street	Boulevard, square, **street, avenue, road**
Usstate	New Jersey, California, Massachusetts
Water	Adriatic, Ontario, Gulf of Mexico…
Water_head	Lake, canal, gulf, harbor, sea, bay…

Starting now from the most certain rules, the one that trusts the above semantic categories is the 'leader'. This rule covers all cases of known locations, either these are cities, states, regions, provinces, seas, continents, islands, airports or countries. They just have to be capitalised[20] and to have one of the above categories in the database :

```
[SYN=PROP, SEM=LOC, TYPE=_T, ZONE=_Z] =>

\[SEM=REGION_HEAD|LANDREGION_HEAD|WATERREGION_HEAD|CITY_PART|
  COUNTRY_PART|PROVINCE_HEAD|LOCATION_PREFIX, ORTH=C|A]?,
[SEM=REGION|USSTATE|WATER|CITY|CONTINENT|COUNTRY|PROVINCE|
  ISLAND|AIRPORT,  SEM=_T, ORTH=C|A|O, ZONE=_Z],
[SEM=REGION_HEAD|LANDREGION_HEAD|WATERREGION_HEAD|CITY_PART|
  COUNTRY_PART|PROVINCE_HEAD|STREET|FACILITY|FACILITY_HEAD,
  ORTH=C|A]? /;
```

The rule deals with all types of locations, since the TYPE property of the expression that will be marked up will have the value of the SEM attribute of the main token in the expression. So, if the expression is '*Gatwick*', the token will be found in the database in the 'Airport' category and this category will be its type as well, after having been identified. The main token of the expression may also be preceded or

[20] The 'O' value in ORTH is for cases such as :'U.S.' where the token is not considered capitalised.

followed by another token that is found in the database in location-related categories and is capitalised itself. Some examples would be:
'*Chicago* based company', '*California*'s', *'South America'*, *'Cape Town'*, '*Balkans*', '*Soviet Union'*, *'Beverly Hills'*, *'Palm Springs'*, *'Sahara Desert'*, *'Los Angeles'*, *'Democratic Republic of China'*, *'Rhode Island'*, *'Pearl Harbour'*, *'Lake District'*, *'North Carolina'* etc.

For locations that do not exist in the database, but are followed by a token that denotes location (e.g. *'Fellow City'*, *'Kokomo Valley'*, *'Asian territories'*, *'European counties*'), we have created the following rule:

```
[SYN=PROP, SEM=LOC, LOC_PRE=_L, LOC_HEAD=_L1, LOC_TYPE=_T, ZONE=_Z]
=>
\ [SEM=REGION_HEAD|LANDREGION_HEAD|WATERREGION_HEAD|CITY_PART|
   COUNTRY_PART|PROVINCE_HEAD|LOCATION_PREFIX,
   ORTH=C|A, NORM=_L]?,
  [ORTH=C|A, SOURCE!=RULE, NORM=_L1],
  [SEM=REGION_HEAD|LANDREGION_HEAD|WATERREGION_HEAD|CITY_PART|
    COUNTRY_PART|PROVINCE_HEAD|STREET|FACILITY|FACILITY_HEAD,
    ORTH=C|A, NORM=_T] /;
```

The known location token defines the type of the location expression. The NORM attribute is used instead of the TOKEN because of co-reference matters. One will notice that in previous rules we preferred the TOKEN attribute; this is because the token appears exactly as found (e.g. capitalised) and one may see it that way in the added-properties table. However, sometimes this is not convenient at all. It may happen that we find a location in a text, in all capitals and mark it up, and later on find the same location with only the first letter capitalised. If it co-reference is to work in such cases, then the NORM features have to match and not the TOKENs.

For locations that cannot be identified with any help from the database, a set of rules that cover appositions with location expressions may help. Apposition is a phenomenon that one finds in many cases e.g. Date expressions : 'Monday, 13 July'. However, we did not need a rule that would take advantage of the syntactic phenomenon, to deal with date expressions. With location expressions, one can easily come across a case when the only clue of identifying a string as a 'Location' is the

greater region that follows, a region that is considered well known and thus useful for locating the unknown string (cf. Wakao et al. 1996). This is a mechanism that one comes across frequently both in discourse and writing. Since this mechanism helps humans themselves to understand what is said and complement their real world knowledge, why not use it as well for artificial intelligence needs, for computers to spot the right expression and classify them properly?

Consider the cases: *'Kokomo, Indiana'*, *'Pech Tiqua, Israel'*. No one can possibly know all regions in the world, no database may contain them all. 'Kokomo' is a place in Indiana State and the only way one may know that is through apposition (except if one lives there or in the specific state, or has been there)[21]. We first thought of integrating this case in our general rules that we have already presented. The idea is that when an unknown proper name (or sequence of) is followed by comma and an expression already identified by a rule, then it inherits its properties. Unfortunately, this does not always work. If we do not specify that the following expression must be a location, one may come across something like this: 'MINNEAPOLIS, June 27' all marked up as 'Dates'. Here, 'comma' does not have a conjunctive function, does not lead to an explanation of the location; it separates expressions, enumeration of circumstances takes place. It acts as disjunction rather than a conjunction. For these reasons, we have preferred to create the following rules that pertain only to location expressions:

```
  [SYN=PROP, SEM=LOC, GREATER_REGION=_G, ZONE=_Z] =>
\ [ORTH=C|A, SOURCE!=RULE, ZONE=_Z] /
  [NORM=","], [SEM=LOC, SOURCE=RULE, TOKEN=_G, ZONE=_Z];


  [SYN=PROP, SEM=LOC, GREATER_REGION=_G,ZONE=_Z] =>
\ [ORTH=C|A, SOURCE!=RULE, ZONE=_Z]{2,2} /
  [NORM=","], [SEM=LOC, TOKEN=_G, SOURCE=RULE, ZONE=_Z];
```

[21] We deal with a mechanism of specifying items of an infinite set using a set of finite and thus known locations.

The first rule deals with single word locations, whereas the second deals with two word unknown locations. Unfortunately, the system works only when defining the exact number of tokens that will be identified; the '*' operator confuses things and leads the system to crash. This 'backwards' movement is quite restricted[22]. The SOURCE!=RULE attribute-value pair in the unknown tokens is absolutely necessary . While refining the rules, we came across the following phrase : 'Judith Meyers, M.D'. M.D. is both a title and the abbreviation of a U.S state in the database. So, the name was first identified as a person name, but then, because of the above rules, it was marked up as a location…This means that we must be quite specific and ask in the rules only for unclassified Proper names …

Sometimes, a known location is followed by another location (abbreviated or not) that is present just to avoid confusion with possible homonyms: *'Washington*, *D.C'* In this case, D.C is the greater location of Washington, but it is not present in the database. Hardly will one refer to Columbia using this abbreviation. If we are to mark it up as a location expression, which it is, we will need the following rule:

```
[SYN=PROP, SEM=LOC, ZONE=_Z] =>

[SEM=LOC, SOURCE=RULE, ZONE=_Z], [NORM=","]
\ [ORTH=C|A, SOURCE!=RULE, ZONE=_Z] /
[SYN=PUNCT] ;
```

This rule is, in a way, the reverse of the previous ones. The right hand side context is essential for the boundaries of the expressions not to be blurred. Punctuation does follow appositions; so such cases will be avoided: 'MINNEAPOLIS, June 28' all marked as location expressions.

The last location rule is a co-reference rule; a rule for cases when a location not in the database is explained once (full name) and then appears again partially : 'Crooks Valley.....Crooks', 'Waltham City...Waltham', 'New Psychikon Region...New Psychikon:

[22] This is also evident with co-reference rules: if something is mentioned earlier in the text, but there was not enough evidence to mark it up, the co-reference rule cannot compensate for its loss…

```
[SYN=PROP, SEM=LOC, FULL_FORM=_A] =>
\ [ORTH=C|A, NORM=_L]?, [ORTH=C|A, NORM=_L1] /
>>
[SEM=LOC, SOURCE=RULE, LOC_PRE=_L, LOC_HEAD=_L1, TOKEN=_A] ;
```

We have to note that NORMs and not TOKENs are used in the rule, for reasons we have already mentioned and only an ORTH attribute is used to define the Proper names to be marked. The latter is preferred rather than using a SYN=PROP attribute – value pair, because sometimes the tagger does not recognise a Proper name as such e.g. 'Blue Bell, Pa'…'Blue Bell'. 'Blue' is considered an ADJ and not a Proper name, since it happens to be an adjective when not followed by a Proper name. In order not to miss such cases we just use another attribute to describe what the rule should look for, and the same is true for the apposition rules presented above…

4.3 ARTIFACT RULES

'Artifacts' have a truly domain specific nature. Rules have therefore to be guided by the needs of the texts that are to be handled by the system. The current rule file of Concerto contains just three rules. These rules are based on a 'Trade_name' semantic category in the database and in fact they just confirm the information taken from the database. In actual texts, we found out that the rules did not perform well. The corpus that we examined in order to write artifact rules was the newswire texts for Pharmaceuticals / Biotechnology and Health field.

We have not used the 'Trade_name' category at all, since thorough examination of the contents of this category is necessary, before using it in a rule. Unfortunately, Common nouns have been added as 'Trade_names' and this may lead to marking up of irrelevant tokens in the texts as 'Artifacts'. We suggest that all common nouns should be taken out first and preserve only 'Trade_names' that have been found in the corpus, not names that 'could' possibly be found as such. McNaught (2000) has suggested the creation of two new semantic categories in the database : PHEAD (product head) and PORCAPP (product or company apposition) and we believe in the utility of these categories too:

Category	Instances
PHead	Tool, product, system, database, package, patent, trademark
Porcapp	Leading, foremost…

The above categories may be expanded to cover the needs of any subject area. For example, in Pharmaceuticals one comes across many Vaccines produced by different companies. The word 'vaccine' could be added in the 'Phead' category and used appropriately in rules, so as to designate a product.

We have seen, in our corpus, that very often the name of the product is mentioned in the title of the newswire text. So we created the following rule that uses the EDGENO attribute to denote that this is valid only for titles (the first 30 edges found in the text):

```
   [SYN=PROP, SEM=ARTIFACT, MAIN_NAME=_M, SPECIFIER=_P, ZONE=_Z] =>

   [SYN=POSS|DET]
\  [SYN=PROP|NN, EDGENO<30, ORTH=C|A|O, SOURCE!=RULE, SEM!=OCC,
   TOKEN=_M, ZONE=_Z],
   [ORTH=1, MORPH=(("CARD")), ZONE=_Z, TOKEN=_P]? /;
```

The rule refers to cases such as : 'Aviron's Multikine 700' which are very common in the titles of the documents Concerto deals with. With all the restrictions presented (SEM!=OCC etc.) the rule works very well. Nevertheless, all words in the title are capitalised , the tagger assigns to most of them the PROPER value and therefore we end up identifying many irrelevant tokens as 'Artifacts'. The case could be dealt with by a rule set that would pertain only to the 'Title' zone. Our rules deal with cases found in the body of the text mainly, so we have decided not to include this rule in the final rule set. It creates more problems than it solves –for the moment. When the text-zoner will be activated, the rule may be of more help.
Instead, we have written a rule with some of the features shown above, but with more restrictions, since it applies to the whole text:

```
   [SYN=PROP, SEM=ARTIFACT, MAIN_NAME=_M, ABBR=_A, ZONE=_Z] =>

   [SYN=POSS|DET]
 \ [SYN=PROP|NN, ORTH=C|A|O, MORPH=(("NIL")), SOURCE!=RULE, TOKEN=_M,
    ZONE=_Z],
   [ORTH=1|O|C|A, SYN!=PUNCT|DET|POSS, ZONE=_Z]{0,3} /
   [SYN!=NN|PREP|PROP|NUM],
   [SYN=OPEN]?, [ORTH=C|A, TOKEN=_A]?, [SYN=CLOSE]? ;
```

We have noted that 'Artifact names' have a distinct behaviour; whereas proper names that denote unique entities[23] are never preceded by a determiner or a possession expression, Artifacts are. So, we define a left hand side context for our rule, that asks for the presence of a determiner or possession expression before the 'Artifact token'. One may be surprised seeing a really strict morphological restriction in the properties

[23] These proper names as opposed to common names that are used as proper ones e.g. 'University' in 'University of Athens'…

that the token to be marked up should have. Apart from being an unknown noun, no morphological information must be available from the morphological analyser for this token. This restriction is essential if we are to avoid phrases such as 'the University of..', 'the Department of..' 'the BBC news' having their capitalised token marked up as an artifact expression. We look for names of artifacts that do not designate common nouns as well. McNaught (2000) has pointed out, in one of his rules, that artifacts are usually preceded by a Company name and a possession expression; however, we believe that it is the fact of being possessed that matters for our rule and not who the possessor is. That way, Organisation rules do not have to precede the artifact rules[24].

For cases of multiword names of Artifacts, we have used in our rule an up to three tokens iterator. The right hand side context is present in order to help defining the boundaries of the expression that will be marked up. A noun following right after the expression will most probably specify the 'kind of' of the expression; if the noun has specific characteristics that designate an artifact, then the rule will have marked the right phrase, if not, a mistake will have taken place. The former is dealt with by another rule, so any presence of a noun after the expression is forbidden. Prepositions, Proper nouns and Numbers exactly after the expression are thought negative evidence for its being classified as an artifact expression. We have also included in the right hand context the case of an abbreviation following the artifact.

What follows is the rule for cases when the trade-name is followed by a word that designates it as such :'Multikine system', 'GeneLex, Europe's foremost package'…

```
 [SYN=PROP, SEM=ARTIFACT, MAIN_NAME=_M, ABBR=_A, KIND_OF=_K,
    ZONE=_Z] =>

\[SYN=PROP|NN, SYN!=PUNCT|DET|POSS, ORTH=C|A|O, SOURCE!=RULE,
  TOKEN=_M, ZONE=_Z],
 [ORTH=1|O|C|A, SYN!=PUNCT|DET|POSS, ZONE=_Z]{0,3} /
 [NORM="","|"the"|"a"|"an"]*,
([SEM=LOC, SOURCE=RULE], [NORM="'s"])?,
 [SEM=PORCAPP]?, [SEM=PHEAD, ORTH=L, TOKEN=_K] ,
 [SYN=OPEN]?, [ORTH=C|A, TOKEN=_A]?, [SYN=CLOSE]? ;
```

[24] In fact, we have chosen to use the identification of artifacts in the rules for organisation identification.

The rule is for one or multiword artifacts that are followed by an apposition. The core of the rule is based on the respective rule suggested by Mr McNaught (July 2000) but more restrictions have been added for better results; the 'Porcapp' and 'Phead' categories are used and the case of a possible abbreviation following has also been included. The main difference with the rule that follows is that in this rule the 'Phead' token is not capitalised and therefore it does not form part of the proper name. Nevertheless, it conceals information on the kind of artifact in question. The next rule deals with cases when such designators are capitalised and do form part of the name of the product. In such cases, no 'Porcapp' token stands in between e.g.: 'Military Health System (MHS)'.

```
 [SYN=PROP, SEM=ARTIFACT, MAIN_NAME=_M, ABBR=_A, ZONE=_Z] =>

\ [SYN=PROP|NN, SYN!=PUNCT|DET|POSS, ORTH=C|A|O, SOURCE!=RULE,
  TOKEN=_M, ZONE=_Z],
  [ORTH=1|O|C|A, SYN!=PUNCT|DET|POSS, ZONE=_Z]{0,3},
  [SEM=PHEAD, ORTH=C|A] /
  [SYN=OPEN]?, [ORTH=C|A, TOKEN=_A]?, [SYN=CLOSE]? ;
```

Lastly, two co-reference rules are presented for artifacts. One is for co-reference when the main part of the name is repeated in the text and the other for cases of co-reference through the abbreviated form of the name:

```
 [SYN=PROP, SEM=ARTIFACT, FULL_FORM=_F, ZONE=_Z] =>

\ [ORTH=C|A|O, TOKEN=_M, ZONE=_Z],
 [ORTH=C|A|O|1, SYN!=PUNCT|DET|POSS,   ZONE=_Z]{0,3} /
  >>
  [SEM=ARTIFACT, SOURCE=RULE, MAIN_NAME=_M, TOKEN=_F] ;

  [SYN=PROP, SEM=ARTIFACT, FULL_FORM=_F, ZONE=_Z] =>

 \ [ORTH=C|A, TOKEN=_A, ZONE=_Z] /
   >>
   [SEM=ARTIFACT, SOURCE=RULE, ABBR=_A, TOKEN=_F] ;
```

The last co-reference rule was a need that we came across during the test runs, the refinement phase of the rule writing. In fact, it seems that abbreviations is a quite preferred way of referring to multiword artifact names…

4.4 ORGANISATION RULES

Rules for organisations are very difficult to write because of all the different types of organisations and names one comes across. One may write rules that lead to high precision; unfortunately these rules cover just a few cases. If one aims at high recall as well, many case specific rules have to be written, rules that may perform extremely well in terms of recall, but to the detriment of precision. The balance is really difficult to obtain and requires clear specification of structures the rules will look for. Interaction with the rest of the rule categories is also an important consideration. How much can one trust the semantic information obtained from the database? All these issues require the rule writers to make clear decisions on the characteristics their rules will have.

We have pre-decided that our rules will cover all possible 'certain' cases without significant help from the database. Unfortunately, Concerto's database has many inconsistencies and specific semantic categories cannot be trusted at all. For instance, the 'Company' category is thought of a category that should contain names of companies found in texts. The problem is that in aiming at covering all possible cases of company names, words such as 'the' have been assigned this semantic category. This entails that a rule that would trust the database for marking up an unknown proper name as 'Company' would just lead to over generation of wrong edges. Even if we ask for just the capitalised instances of these words, problems are not solved; capitalisation at the beginning of the sentence is still something that will lead to many mistakes. It seems, from the point of view of the rule writer, that this semantic category has to be built from scratch; all its entries should be company names found in real texts, names that would not be identified otherwise by the rules, names that cannot be confused with common nouns.Otherwise, we just have to avoid using the specific category in our rules, or write rules that will restrict the context in which such tokens of the database will be found and marked up. At present, we found it more cumbersome to write rules for avoiding mistakes as the ones mentioned above, than not using the category at all.

For the needs of Concerto, three types of organisations will be distinguished :
- Company, - Government agency, - Institute. A fourth category has also been suggested for 'Regulatory bodies'; nevertheless, we have not found enough evidence in our corpus to write rules for the fourth category. We suggest significant changes in the database, a re-organisation of the categories that pertain to organisations. The

following table illustrates what has to be preserved and what should be added or deleted :

Category	Instances
Cdg	Inc. , co. , ltd. …
Chead	**Provider, designer, supplier, publisher**…
Gov_agency	Ministry of Defence, ~~The house, Unit~~
Gov_head	Ministry, parliament, police, committee, court, embassy, ~~company, center, academy~~
~~Gov_key~~	~~Embassy, project, federal, national, academy…~~
Inst_head	**Organisation, association, center, centre, institute, group, laboratory, service …**
Org	**Nato, unesco, european community…**
~~Org_other~~	~~Nato, unesco, european community…~~
Uni_head	University, **school, college, academy, university college…**
University	University of Manchester, UMIST…

So, we suggest that the 'Gov_key' category should be deleted since its few constituents seem either repeated in the 'Gov_head' category or irrelevant with the 'Government agency' category. The 'Org_other' category also seems redundant. Its constituents should be instances of the 'Org' category which currently has almost no constituents. We suggest an 'Inst_head' category that will contain words that designate an institute name. 'Institutes' are considered all organisations that cannot be classified neither as 'Companies' nor as 'Government agencies'. The 'Chead' category is one proposed by Mr McNaught (July 2000, Rule File) and we support its creation as well.

Last, we would like to talk about one more 'problematic' category in the database called 'Comp_part'. This category contains many words that do not in fact designate a

company. If we are to use it in our rules – which would help a lot – we have to examine its contents thoroughly. In fact, we do use it in one of our rules; it helps very much to identify companies, but unfortunately it causes many problems as well. During the rule refining phase, it was the rule that was responsible for 90% of mistagged expressions. The table that follows suggests some deletions that should take place in this category and re-movements of some of its entries to different categories :

Category	Word for deletion	Where to be moved into
Comp_part	Administration	
	Affairs	
	Board	
	ADD : Board of directors	
	Business	
	Company	
	Conference	
	Corporation	
	Customs	
	Data	
	Defence	Gov_head
	Department	Inst_head
	Development	
	Division	
	Hospital	Inst_head
	Market	
	Marketing	
	Office	Inst_head
	Organisation	Inst_head
	Society	Inst_head
	ADD : subsidiary	
	Systems	Phead

One can see that the changes are significant; we have to note that more deletions – additions may be necessary, but we have just presented the cases we came across. We would also like to note that sometimes the rules that follow seem not to work if the

expression in question exists in the database in a category that we do not use e.g.: Though there is a rule for the classification of Hospitals as 'Institutes', the 'Children's Hospital' phrase was missed in our test runs. The problem is that 'Children's Hospital is an entry in the database with a 'Company' semantic category. It forms an edge in the multiword look up phase and since it is a multiword token, the word 'Hospital' on its own is not found as a designator of the 'Institution expression'.

On the other hand one may take advantage of this multiword look up feature of the Concerto BSEE analyser; usually, company names are followed by a parenthesis where the abbreviated name of the company is given as it appears in specified stock exchanges or bulletins: e.g. 'Medlex inc. (Nasdaq : MDL). When the name of the place where the abbreviation is used is not just one word, the rule has a difficulty in identifying the whole structure. Instead of trying to modify the rules by saying that a number of words may constitute the name of this place, we can just add it in the database as a multiword token. The case is quite rare: (OCT Bulletin Board : OMIM). When the string exists in the database as a whole, it is being treated in the rule application phase as a single token…

Before presenting the rules, let us see what exactly we look for and which guidelines have been followed :

Legislative bodies such as Congress, event organisers (e.g. committees), groups, parties, unions, armies, organisation related facilities (e.g. embassies, hospitals, universities), meeting places or places where organisational activities occur, all are considered 'Organisations' and have to be tagged appropriately. Capitalisation is our basic criterion for all these cases.

- A really interesting case is when locative entity strings are embedded in an Organisation Name. When a location follows an Organisation, it may or may not be part of the Organisation proper name. For such crucial decision points, the MUC guidelines (Chinchor et al., 1999) are clear :

- "If the location name is preceded by 'in' or 'at' the location name will be tagged separately"
- "If it is preceded by 'of' or no preposition at all", then tagging is determined as follows: 1) if there is an organisation designator or organisation designating proper /common noun, it marks the end of the name. 2) If there is none or if the

organisation is a bank or university, the location name part of the organisation name…

We have decided not to mark up the designators (e.g. inc. co.), but to take advantage of their existence by adding one more attribute to the marked up expressions, the KIND_OF attribute. Its instantiation will reveal the kind of Company one has to do with. The same property is used in cases when a common uncapitalised noun designating an organisation is present; it is not considered part of the proper name, but its value is useful and therefore preserved .

The current rule file of Concerto has 30 rules for organisations and mainly companies. We have revised and made more concise some of these rules. Some others seem too general to be used. The main cases covered are:

- Presence of designator ('Cdg') for one word or multiword company names.
- Long company name with plausible head (a Location) e.g. 'Oxford GlycoSciences'.
- Unknown proper name followed / preceded by an already tagged company (with possible comma or 'and' in between) – (too general, not always true)
- Unknown proper name followed by " 's" is an owner.
- Conditions under which this 'Owner is not a person but a company'- (too general, not always true).

Apart from these rules that we have revised and re-formed, McNaught (2000) has suggested another 10 rules for organisations :

- A rule for 'National institutes of health'.
- The X, Y center / institute / centre
- City / Company followed by a mixed unknown proper is a company.
- X, a / the leading Y.
- Companies buy things e.g. 'The Mirabilis purchase'.
- Company if followed by " 's + occupation"(cf also Wakao et al. 1996).

Some more rules for companies[25] have also been suggested by Thompson (2000):

- The X board of directors
- X's board of directors
- X's company_position (separate rules for company-positions have been created)
- Person, X company_position e.g. 'E. Smith, Aviron President…'

- X announced that its board of directors…
- X company_position, Person e.g. ‘Aviron’s president, E. Smith
- X names company_position e.g. ‘Aviron names President..’
- X (Abbreviation of specific type)
- Subsidiary of X
- X, a wholly owned subsidiary…

All these rules have provided us with many good example – cases and many good observations on what is needed. We have tried to make them more general, so as to cover a wider range of cases; they do have the ability to be extended and be more organised. Additionally, a really rich source of examples that must be covered by the named entity recognition rules and in particular examples for organisation identification is a technical report from Biovista, one of Concerto’s partners (Persidis, 2000). Aided by all mentioned above, we have managed to create 12 rules, one of which is of optional status, and cover even more cases than all 60 rules mentioned above. It is important to note, however, that the existence of the rules mentioned above did help significantly…

The first rule trusts the database (given the changes made above) for organisations that are known and have been classified e.g. 'Nato', 'University of Manchester'...:

```
[SYN=PROP, SEM=ORG, TYPE=_T, ZONE=_Z] =>
\ [SEM=LOC]?,
[SEM=GOV_AGENCY|ORG|UNIVERSITY, ORTH=C|A|O, ZONE=_Z, SEM=_T] /
([NORM!="of"],[SEM!=LOC]);
```

The type of the organisation is defined by the semantic category assigned to the expression in the database. If more than one category has been assigned to a certain expression, all will be presented in the properties table. For example, ‘Nato’ can be identified by the above rule. Its type will appear to be ‘Company’ or ‘Organisation’, since this entry has been assigned to both categories in the database. This variable ‘sharing’ in the rule, between the eventual type of the organisation expression and the

[25] It is about 20 rules since the cases mentioned cover both single word and multiword company names.

semantic category of the token to be marked up, has been preferred in order to create a concise rule; otherwise, one should write a separate rule for each type of known organisation. We believe that seeing all possible categories of a token in its type attribute is not a disadvantage; instead, it may help to decide which of them is most appropriate in a specific textual context.

A location may precede the known expression (e.g. 'U.S Army' where 'Army' is a known organisation and the location is part of the name) and if so, it must not appear separately tagged as a 'Location', but the whole expression has to be tagged altogether as an organisation. The right hand side context we have used is essential in order to avoid cases such as : '…Hyundai of Korea', where the location that follows does not form part of the name of the company. We have to note that the above rule could include tokens of the 'Company' semantic category, but for reasons that have to do with the inconsistencies of this category in the database, we have excluded it for the moment…

What follows is a pair of rules for single word and multiword companies followed by a 'Cdg', a company designator e.g. 'MedLex, inc', 'Boron, Lepore & Associates, co', 'Cell Therapeutics, inc. ("CTI") (Nasdaq: CTIC)' :

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F,
 ABBR=_A, IN=_I, ABBR2=_A2, IN=_I2, KIND_OF=_K, ZONE=_Z] =>

\ [ORTH=C|A|O, SYN!=PUNCT|DET, NORM=_F, ZONE=_Z] /
 [NORM=","]?, [SEM=CDG, NORM=_K],
 [NORM=","]?,
 [SYN=OPEN|QUOTE]{0,2}, ([ORTH=C|A|O, TOKEN=_I, ZONE=_Z], [NORM=":"])?,
 [ORTH=C|A|O, TOKEN=_A]?,  [SYN=OPEN|QUOTE]{0,2}, [NORM=","]?,
 ([NORM="("], [ORTH=C|A|O, TOKEN=_I2, ZONE=_Z], [NORM=":"], [ORTH=C|A|O,
  TOKEN=_A2],  [NORM=")"])?;
```

The above rule is for single word company names. The token to be tagged is defined through the ORTH attribute. Since the 'Other' value for this attribute covers many cases, we were forced to use some more restrictions. In fact the 'O' value is given both to proper names with mixed capital and lower case letters and to punctuation symbols. So, we ask for the exclusion of 'punctuation' tokens in the rule. The right hand side context is very large, one may notice. The presence of a 'Cdg' is essential; what may

follow this designator, may be one or two abbreviations, the first with an optional reference to the place the abbreviation is used. One may find several different ways of expressing the abbreviation; our rule deals with the format found in the texts Concerto is interested in. We have captured all information one may take from the right context: The abbreviation, where this abbreviation is used and the kind of company found; the latter is revealed through the 'Cdg' . As mentioned in other rule categories, whenever NORM is used instead of TOKEN it is for co-reference matters (e.g. 'EMBION inc ….Embion').

The next rule is the same as the previous one, but it handles multiword company names :

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, LAST=_L,
 ABBR=_A, IN=_I, ABBR2=_A2, IN=_I2, KIND_OF=_K, ZONE=_Z] =>

\ [ORTH=C|A|O, SYN!=DET|PUNCT, NORM=_F, ZONE=_Z],
  [ORTH=C|A|O|1, NORM!="."|"/"|"--", SOURCE!=RULE, ZONE=_Z]{0,3},
  [ORTH=C|A|O|1, SYN!=PUNCT|DET, NORM=_L, ZONE=_Z] /
  [NORM=","]?, [SEM=CDG, NORM=_K],
  [NORM=","]?,
  [SYN=OPEN|QUOTE]{0,2}, ([ORTH=C|A|O, TOKEN=_I, ZONE=_Z], [NORM=":"])?,
  [ORTH=C|A|O, TOKEN=_A]?,
  [SYN=OPEN|QUOTE]{0,2}, [NORM=","]?,
  ([NORM="("], [ORTH=C|A|O, TOKEN=_I2, ZONE=_Z], [NORM=":"], [ORTH=C|A|O,
   TOKEN=_A2],
  [NORM=")"])?;
```

The first word of the name may not be a determiner or punctuation. The following tokens may include even a number or a comma but no other punctuation. The tokens must be unknown; no rule must have identified them. This is very important if we are to avoid cases such as:'Bothell, Washington, Microvision inc' where all the phrase may be marked up as a company, if the restriction will not be included in the rule. The iterator has to be specific and we have defined a maximum of three tokens in between the first and last word in the company name. The last word of the name can also be a number, but not a determiner or punctuation.

The next group of rules deals with cases when part of the organisation name is a word in the database designating its type e.g.:
'OxyGen Medical Center', 'University of Athens', 'Case Western Reserve University', 'Fregit Gurtwinkle Center for Mushroom Spore Transduction', 'Royal Health Institute', 'Ministry of Defence', 'Embassy of France', 'Pennsylvania State Nurses Association', 'First Southwest Company', 'Food and Drug Administration'...

```
[SYN=PROP, SEM=ORG, TYPE=GOV_AGENCY, FIRST=_F, LAST=_L, ABBR=_A,
 ZONE=_Z] =>

\ [ORTH=C|A|O, SYN!=PUNCT|DET, NORM=_F, ZONE=_Z]{0,1},
  [ORTH=C|A|O|1, SYN!=PUNCT|DET, ZONE=_Z]{0,3},
  [SEM=GOV_HEAD, ORTH=C|A, NORM=_L, ZONE=_Z],
  [NORM="of"|"for"]?, [ORTH=C|A|O, SYN!=PUNCT, ZONE=_Z]{0,4} /
  [NORM=","]?,
  ([NORM="("], [ORTH=C|A|O, TOKEN=_A],
  [NORM=")"])?;
```

Starting from the rule for 'Government agencies', one has to look 'in media regli', in the middle of the rule. The third line is the core of the body of the rule; the capitalised token that belongs to the 'Gov_head' category in the database must be present in the expression. It may stand alone and be marked, but this is rarely the case. It is usually preceded or followed by other capitalised tokens that specify it. It may also be followed by an abbreviation.

A similar rule deals with companies. Because of the problems with the database, mentioned earlier, we have not used the 'Company_part' semantic category as a way to indicate companies. We have restricted the rule in cases when the word 'company' is explicitly present in the expression and forms part of it. It has to be preceded by another capitalised token; cases of the capitalised word 'company' that refer to a company already mentioned in full in the text, and thus known, will not be tagged that way (e.g. 'Medlex inc…….The Company will…'):

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, LAST=_L, ABBR=_A,
 ZONE=_Z] =>
```

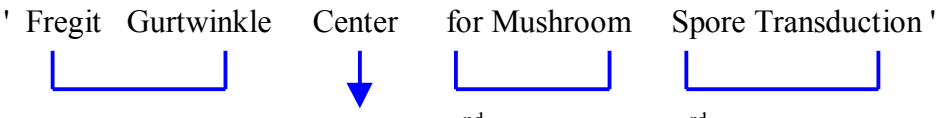
```
  \ [ORTH=C|A|O, SYN!=PUNCT|DET, NORM=_F, ZONE=_Z],
    [ORTH=C|A|O|1, SYN!=PUNCT|DET, ZONE=_Z]{0,3},
    [NORM="company", ORTH=C|A, NORM=_L, ZONE=_Z] /
    [NORM=","]?,   ([NORM="("], [ORTH=C|A|O, TOKEN=_A],   [NORM=")"])?;
```

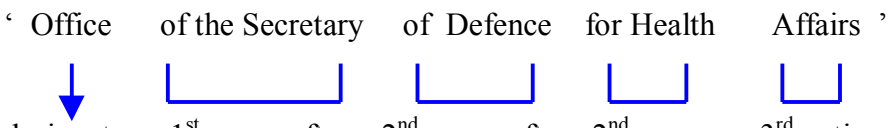
Last in this group of organisation rules is the one for institutes. This rule covers the majority of the cases found in texts, cases that have to do with institutes :

```
[SYN=PROP, SEM=ORG, TYPE=INSTITUTE, FIRST=_F, LAST=_L, ABBR=_A,
 ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=PUNCT|DET, SOURCE!=RULE, NORM=_F, ZONE=_Z]{0,1},
 [NORM="and"]?,
 [ORTH=C|A|O|1, NORM!=","|".", ZONE=_Z]{0,3},
 [SEM=UNIV_HEAD|INSTITUTE_HEAD, ORTH=C|A, NORM=_L, ZONE=_Z],
([NORM="of"|"for"|"and"], [NORM="the"], [ORTH=C|A, ZONE=_Z]){0,3},
([NORM="of"|"for"|"and"], [ORTH=C|A, ZONE=_Z]){0,3},
 [ORTH=C|A, ZONE=_Z]{0,4}/
 [NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_A],
 [NORM=")"])?;
```

The rule covers cases of both simple and very complex institute names. The main idea is that what precedes the token that denotes the organisation expression must not be a punctuation or determiner, nor an already identified expression. If more than one token precedes, they may be connected with ‘and’ capitalised or not. The tokens that may follow the designator have been grouped into: the ones that are combined with the designator through a specific preposition or conjunction followed by a determiner; the ones that do not contain a determiner; and the ones that follow directly with no connective tokens. A combination of all these is also possible. In our test runs we came across the following name that we have classified as ‘Institute’ : ‘Office of the Secretary of Defence for Health Affairs’. This case gave us an insight into possible complex combinations of Proper names when forming an institute name. The rule deals with such cases successfully. The following scheme illustrates how the rule works:





One more rule that leads to indisputable results is the one that looks for a Proper name or a sequence of, followed by a specific type of abbreviation. It is a rule for ‘companies’ e.g. 'IMS HEALTH (NYSE: RX)', 'Immune Response (OTCBB: IMUN)', 'Biovail (NYSE: BVF) (TSE: BVF)'...

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, ABBR=_A, IN=_I,
 ABBR2=_A2, IN=_I2, ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=PUNCT|DET, SEM!=CDG, SOURCE!=RULE, NORM=_F,
 ZONE=_Z],
[ORTH=C|A|O|1, SYN!=PUNCT|DET, SEM!=CDG, SOURCE!=RULE, ZONE=_Z]{0,4} /
[NORM=","]?,
[SYN=OPEN], [ORTH=C|A|O, TOKEN=_I, ZONE=_Z], [NORM=":"], [ORTH=C|A|O,
 TOKEN=_A],
[SYN=CLOSE], [NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_I2, ZONE=_Z], [NORM=":"],
 [ORTH=C|A|O, TOKEN=_A2],
 [NORM=")"])?;
```

In this rule, the existence of an abbreviation right after the expression to be tagged is obligatory. Only a specific type of abbreviation is being asked for because it is this and only this that designates companies: In parentheses, a capitalised token is followed by semicolon and an other capitalised token. All other forms of abbreviation do not guarantee that a company name precedes. The abbreviation may belong to an artifact that is mentioned or even to a multiword term (e.g. ‘the application program interface (API)…’). The proper names that will be tagged must not be company designators. This is said explicitly in the rule, so as to avoid company names being identified twice:

first by the ‘Cdg’ rules and then again by the present rule, if the names are followed by an abbreviation.

Sometimes, the company name is followed by a ‘Chead’ token : 'Limatex is a leading provider', 'GeneLex is Europe's leading supplier'. The rule following is based on McNaught’s suggestion, it has just been refined so as to interact better with the rest of the rules (cf also Wakao et al 1996). Syntactic and semantic restrictions on the properties of the tokens to be tagged are essential especially because of the optional existence of ‘COMMA’ in the right hand context which may lead to mistaggings of coincidental collocations (e.g. ‘… Aviron’s President, a provider of ..’ – here the Chead token refers to a person, not a company and ’President’ must not be tagged as a company name) :

```
  [SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, KIND_OF=_K,  ZONE=_Z]
=>
\ [ORTH=C|A|O, SYN!=PUNCT|DET|POSS,SEM!=OCC,SOURCE!=RULE,NORM=_F,
   ZONE=_Z],
  [ORTH=C|A|O|1, SYN!=PUNCT|DET, SEM!=OCC, SOURCE!=RULE, ZONE=_Z]{0,3} /
  [NORM=","|"is"|"the"|"a"|"an"]*,([SEM=LOC, SOURCE=RULE], [NORM="'s"])?,
  [SEM=PORCAPP]?, [SEM=CHEAD, TOKEN=_K] ;
```

The rules that follow needed many refinements in order to perform well, without resulting in wrong markings of tokens irrelevant to organisations. We have managed to achieve really good results, but some refinements may be needed according to the needs of the texts on which they will be tested[26].

The main idea in the rule that follows is that when a location precedes an unknown proper noun, a company name may be denoted : e.g. 'Oxford GlycoSciences', 'U.S Bacou Solutions'...

```
   [SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, ABBR=_A,
    LOC_INDICATION=_LOC, ZONE=_Z] =>

\  [SEM=LOC|COUNTRY_ADJ, TOKEN=_LOC, SOURCE=RULE],
   [ORTH=C|A|O, SYN!=PUNCT|DET|POSS,SEM!=OCC|CDG|PHEAD,
    SOURCE!=RULE, NORM=_F,ZONE=_Z],
```

[26] That is of course if a hundred percent precision is to be achieved.

```
[ORTH=C|A|O|1,SYN!=PUNCT|DET|POSS,SEM!=OCC|CDG|PHEAD,
 SOURCE!=RULE, ZONE=_Z]{0,3} /
[NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_A],
 [NORM=")"])?;
```

One may note that many restrictions have been put in the rule; the capitalised token that will follow the location must not form an edge –result of a rule-, it must not be a punctuation mark, a determiner or possession expression and it must not belong to an 'Occupation', 'Cdg' or 'Product head' semantic category. These restrictions are essential for the expression not to have been recognised by previous organisation (or other) rules too. Consider the case : 'Oxford Molecular Group Plc ("OMG"), (London: OMG)'. This company name will be identified by the rule that relies on the presence of the 'Cdg'. Since it starts with the name of a location, it can also be found by the present rule and satisfy all criteria in the rule. To avoid a doubly - redundant edge from being formed, we have to ask for words in the expression that are not company designators and for tokens that have not been found by other rules, they are unknown.

One more example on the matter refers to artifacts: 'Wisconsin Package ™'. Only with the restriction discussed above, the artifact mentioned will not be marked up as company name, by the rule in question. One may also note that the abbreviation that may follow the expressions in this rule is of a simple format, not the standard found after company names. This is because if the standard abbreviation follows, then the expression will be identified by the rule that relies on abbreviations and not by this one. It would be therefore redundant to mention something like that; it would be a burden for the rule…

It has been noted, in the text corpus, that company positions ('Occupations' in the database) or 'Company parts' usually precede a company name: 'The President of Latimex Biorad', 'The Board of Directors of Pinox', 'subsidiary of Wertex Neurosciences' (cf. also Wakao et al. 1996). For such cases we have written the following rule :

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, ABBR=_A, ZONE=_Z] =>

[SEM=OCC|COMPANY_PART],[NORM="at"|"of"]?
\[ORTH=C|A|O,SYN!=PUNCT|DET|POSS,SEM!=OCC|CDG,SOURCE!=RULE,
```

```
 NORM=_F, ZONE=_Z],
[ORTH=C|A|O|1,SYN!=PUNCT|DET|POSS, SEM!=OCC|CDG, SOURCE!=RULE,
 ZONE=_Z]{0,3} /
[NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_A],  [NORM=")"])?;
```

It is this rule that has proved really problematic during the test runs. The problems were mainly caused by words that had been assigned to the 'Comp_part' semantic category without actually pertaining there. We refer to over generation problems. All restrictions imposed are for reasons mentioned above.

At other times, the 'Occupation', the 'Company part' or even the artifact may follow: 'Lenox acquisition of', 'Lenox purchase of', 'Trinity BioSciences's clients' . We have to note that the necessary changes in the database are a prerequisite for the rule to work properly…

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, ZONE=_Z] =>

\ [ORTH=C|A|O, SYN!=PUNCT|DET|POSS|NUM|CONJ, SEM!=OCC|TITLE|CDG,
  SOURCE!=RULE, NORM=_F, ZONE=_Z],
 [ORTH=C|A|O|1,SYN!=PUNCT|DET|POSS, SEM!=OCC|TITLE, SOURCE!=RULE,
  ZONE=_Z]{0,3} /
 [NORM="'s"]?, [SEM=OCC|COMPANY_PART|ARTIFACT] ;
```

The SYN!=NUM restriction may seem peculiar, since orthographically a number in the place of the first token is not allowed. Nevertheless, we came across the following case : '4,600 physician members…' where the number had an 'O' value for the ORTH attribute and not a '1' value. So, we have to define explicitly that a number in that place is forbidden.

Last, organisation rules finish with two co-reference rules. The first is for co-reference with the main name of the company name and the other is for co-reference using the abbreviation :

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FULL_FORM=_F, ZONE=_Z] =>
\ [ORTH=C|A|O, SYN!=DET|PUNCT|POSS, NORM=_F1, ZONE=_Z],
 [ORTH=C|A|O, SYN!=DET|PUNCT|POSS, SEM!=OCC|TITLE, GOOD-
  MORPH=(("NIL")),
```

```
   SOURCE!=RULE,  ZONE=_Z ]{0,3}  /
  >>
   [SEM=ORG, SOURCE=RULE, TYPE=COMPANY, FIRST=_F1, TOKEN=_F,
     ZONE=_Z] ;


   [SYN=PROP, SEM=ORG, TYPE=_T, FULL_FORM=_F, ZONE=_Z] =>
  \ [ORTH=A|C|O, SYN!=DET|PUNCT|POSS, TOKEN=_A, ZONE=_Z] /
  >>
   [SEM=ORG, SOURCE=RULE, TYPE=_T, ABBR=_A, TOKEN=_F, ZONE=_Z] ;
```

Apart from the first/main word of the name one may use to refer to a multiword organisation name, some more words of the name may be used as well. Since we cannot know which and how many of them will be used, we have included in the first co-reference rule the description of these words that may follow the main name. They must be capitalised, not a punctuation, determiner or possession expression, they must not have an 'Occupation' or 'Title' semantic value, they must be completely unknown. Not even the morphological analyser must provide any information for them.

This strict restriction is used in order to avoid the marking up of irrelevant capitalised words following the repeated organisation name by chance e.g. 'Valentic inc. ……Valentic Annual Report' – 'Valentic' must be tagged by the co-reference rule, but not the other two words as well… The rule may miss a multiword co-reference instance when the tokens following the main name must be tagged and morphological information is available for them; in such cases, the co-reference will miss a part of the name[27].

This first rule is only for company names and not for 'Institutes' or 'Government agencies'; this is because such names are hardly – if ever – repeated using just some of the words of the name e.g. one may not say 'the Embassy' when one means 'the Embassy of France'. For really long institute names, the abbreviation is used if one does not want to repeat the whole name. Exceptions of course always exist, but the majority of cases found in formal writing, in texts Concerto is interested in, do support our decision. In support of our rule, one should also present the following case as well :

[27] The test runs have not indicate of such a case. Nevertheless, it is not impossible to come across something like that…

‘ …Pharmacy Data Transaction Service (PDTS)...(recognised as institute)

‘…National Mail Order **Pharmacy** (MNOP) (later in the same text)’.

If the co-reference rule covered institutes too, the word highlighted above would be found as a co-reference instance of the first institute, something that is definitely wrong… The second co-reference rule for abbreviations covers all types of organisation names.

CHAPTER 5

RULE ORDER

The order of the rules is significant and one should therefore comment on the matter. The processor, when invoked on a text, applies the rules strictly in sequence and "the words and phrases identified by a rule replace their constituents, which are no longer 'visible' to subsequent rules" (Black, 2000). Each rule may depend on the results of other rules that precede and in fact it can take advantage of them to achieve modularity.

So, we propose that some basic rules we have written must be placed first; these are rules that do not have obvious results, their constituents do not appear highlighted, they do not pertain to one of the known rule categories mentioned above. What they do is to reinforce the semantic (mainly) information found in the database and to make things simpler – normalise the tokens. We refer to rules such as the ones for Festivals, Relative dates, Abbreviated ordinals and Big cardinal numbers. Edges created by the application of these rules are used in rules that follow. For instance, the edges that have been assigned a 'Relative date' semantic value are invoked in a 'Date' rule and it is by this semantic category that they are being referred to in the rule…

Similarly, in the end of the rule file we suggest that the two General rules should be placed. These two are rules that can be used in order to mark up expressions pertaining to more than one named entity categories. Their structure is such that they rely upon the application of all other rules; after all other rules have been invoked, these two look for tokens (numbers or places) that precede or follow identified expressions and mark them up appropriately. In fact, it is exactly this strict, sequential rule application procedure that these rules take advantage of.

Now that the 'boundaries' of the rule set have been defined, the order in the main body of the rule file is of interest. The rules have been grouped according to the semantic feature that they assign to the expressions that they mark up. So, we have the 'Date rules', 'Time rules' and so on. Some of these rules make use of attribute – value pairs that are assigned to tokens by other rule categories e.g.: a rule for organisations (Companies) relies on the presence of an 'Artifact expression' after the token (or

sequence of tokens) that are to be marked up. This means that the 'Artifact rules' must have already been applied, before the turn of this rule comes. This 'dependency' matter guides our rule ordering; during rule writing, we were very careful not to complicate dependencies. We avoided mixing rule categories and attempted to restrict dependencies on other rule categories in general, not on specific rules of another rule category[28]. What we only need is the 'Artifact' and the 'Location' rules before the 'Organisation' ones, because some of the latter rules do depend on the categories mentioned. 'Location' rules must also precede 'Time' rules, since the Location semantic category is used in the latter.

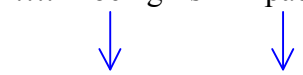
Last, the 'Person' rules must precede the 'Location' ones, but for reasons other than 'dependency': Quite surprisingly, we found that some tokens that denote 'first person names' are classified in the database not only as 'Person_male / female' names, but also as locations ('Cities', 'Regions'…). Since we have rules in both rule categories, that rely on this information found in the database, it is rule order that will tell which marking will be preferred : 'Helena Perez' is a person name found during our test runs. The token 'Helena' exists in the database both in the 'Person_female' semantic category and in the 'Region' one. The phrase will be tagged either as a 'Location' or as a 'Person' according to which rule category is invoked first. Since the case of a person name denoting a location is quite rare, we have preferred to put 'Person' rules before the 'Location' ones. So, the general structure of the rule file[29] is:

- Specific / database reinforcing rules
- Enamex Rules
  - Person rules
  - Location rules
  - Artifact rules
  - Organisation rules
- Timex rules
  - Date rules
  - Time rules
- Numex rules
  - Money rules -Percentage rules

[28] One may write for example an organisation rule that depends on a specific artifact rule and not on all artifact rules and put therefore this specific artifact rule just above the organisation rule in question…
[29] Appendix B presents the rule file with all rules in order as discussed in this section.

During the refinement phase of the rule-writing we came across a case that illustrates how well the rules interact when ordered this way :

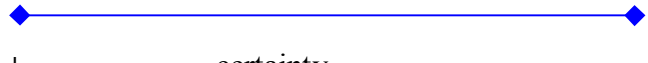
‘…. Boeing ‘s Apache …..’

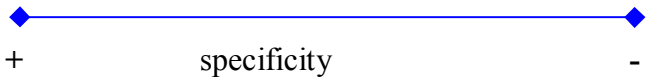


First, the rule for ‘Artifacts’ applies. Here the one used is the rule that looks for a proper name preceded by a possession expression. Then the ‘Organisation’ rule is invoked; the one that succeeds is the rule that asks for an ‘Artifact’ following the proper name to be identified as an organisation, with / or without a possession expression in between.

The internal order in all rule categories follows a descending scheme from the rules with the more certain results to the ones with the less. Rules that trust the database fully come first (e.g. Famous person names present in the database are marked up appropriately). Then, the ones that use some help from the database follow (e.g. the presence of designators –‘Cdg’, ‘Phead’ etc. - helps classifying an expression). Rules that look for specific immediate context come right after (e.g. an abbreviation must follow…).

Rules that presuppose that other rules of the same category will have been applied, come next (consider the case of location apposition rules: an unknown proper is considered a ‘Location’ if a ‘Location’ expression follows it…). Their place towards the end of the rule category set is not due to their ‘disputable’ results; on the contrary they perform very well. They just depend on other rules preceding, they have a more ‘general’ character. The same is for the co-reference rules that come last. Co-reference rules must apply to the results of all rules that exist for a category, if we are to make the most out of them. So, two schemes characterise internal rule order in our rule set :





CHAPTER 6

REMARKS - CONCLUSIONS

Basic semantic elements extraction is the first step in Information Extraction. Therefore, the rules written are very important for more advanced tasks to be realised. Immediate context, the results of the tokeniser, the POS tagger and the morphological analyser added to all information available in the database are valuable assets for the rule writer. These advantages can become a serious source of problems though, problems difficult –if possible- to encounter.

- When is information from these sources right?
- How much can one trust the results of the tagger?
- What can one do with inconsistencies in a large database?
- How shall tokenization problems encountered be dealt with? (Change some features of the tokenizer or not ?)
- What restrictions have to be imposed in a rule so as to avoid mistaggings?
- To what extent is the context considered immediate ?
- Recall or precision should come first, or both? How shall balance be attained?

The list of decisions one has to make goes on and seems endless; one has to write rules that will take advantage of all sources available accepting the fact that they cannot possibly be faultless and compensating for this fact.

Significant help can be obtained from MUC guidelines. When language shows its infinite character and taming even a small part of it seems impossible, these specifications help one to decide what the rules will cover or not. Whether the specifications will be followed or not is of no importance. Simply understanding that exhaustiveness is just incompatible with the nature of language is very important.

The methodology one will choose to follow when writing rules is also significant. Rules interact with each other and this is something one should take advantage of. Additionally, rules have to be written having in mind the needs of the texts against which they will be tested. They are domain specific; some categories more, some less. From our experience, ‘Artifacts’ are the most domain – dependent named entities. ‘Organisations’ are the expressions with an untameable variety. It is

not only the subject field the texts belong to, that affects rule writing. The style of the texts is equally important; formal writing has specific features one has to take into consideration. Not to mention the speech – text trade off, that may lead to totally different rules in information extraction tasks. Our rules take advantage of the fact that they are written for formal texts and cover only cases that conform to standard[30] formal writing conventions.

The corpus guides rule writing and if one is to create rules that will perform well, one has to test them and refine them as much as possible. The more the rules are tested on real texts and refined, the better their results will be. Chance collocations, that may suggest the deficiency of a rule, can only be found during test runs. Concordance tools cannot provide significant help, unless one finds a really efficient concordance program; that is a program that will not necessarily search for a specific word but for a specific part of speech. No significant results can be obtained when searching for a particular Named entity, except if one has a large corpus referring to that specific entity. A search through a feature such as 'Proper name' would help more. Given more time, such a systematic search for corpus evidence that would support a hundred percent our choices during rule writing could take place. However, conclusions / suggestions made in Concerto technical reports have been examined against our fifty - text corpus; test runs have taken place and it is these tests that prove our choices to be right…

Unfortunately, the MUC evaluation scoring software cannot be used to evaluate the performance of all the rules we have built, as the needs of the project meant that non MUC tags were used as well as MUC - like ones. Nevertheless, a systematic evaluation, should be performed, in the near feature, by the Concerto partners, possibly after systematic use of the rules in large textual collections.

Further work on more advanced information extraction tasks, such as template rule writing, may suggest changes in the basic rules, additions or deletions. Using the rules in various texts will also suggest changes and refinements. Refinement is after all an endless procedure. We would be satisfied if the rules simply prove to be a point of reference for further work in the BSEE analyser…

[30] Consider the case with the short formof date expressions e.g. 12-09-99

APPENDIX A

RULE NOTATION

COMPARISON OPERATORS	DESCRIPTION
=	Equal to
! =	Not equal to
<	Less than
>	More than
< =	Less or equal to
> =	More or equal to

ITERATORS	DESCRIPTION
*	Zero or more
+	One or more
?	Zero or one
{integer, integer}	A pair of integers in braces e.g {3,5} meaning the preceding item may be repeated from 3 to 5 times.
{integer, *}	The integer denotes the minimum times of iteration. No maximum times restriction is given.

OTHER SYMBOLS	DESCRIPTION
|	Disjunction (or)
,	Conjunction (and)
#	Comment
( )	Grouping operator
_Capital letter	Variable (Prolog style) e.g. _A
?	Wildcard for one character
*	Wildcard for zero or more characters.
\ /	Delimit left and right hand context.
>>	Delimiter. Whatever follows, is an antecedent…

ACCESSOR FUNCTION	TYPE	DESCRIPTION
Edge-id	Integer	Unique ID for the edge
Start	Integer	String offset in doc text section
End	Integer	String offset in doc text section
Sep	Integer (ASCI code)	What separates tokens
Zone	Symbol	Indicates the text section e.g title, body of the text etc.
Root	String in quotes	The root of a token as found by the morphological analyser
Token	String	Token as it occurs in text
Norm	String	Normalised form of token as specified in database or by the morphological analyser e.g. word in lower case letters.
Constituents	List	Contents of an edge. Nil if initial or database edge, list of edgenos otherwise.
Source	Symbol	D(atabase) R(ule) Default value : DOC(ument)
Ruleno	Integer	Rule used to build this edge from its immediate constituents
Ante(cedent)	Integer	Pointer to antecedent where this edge is a subsequent reference to a previous constituent.
Good-morph	List of strings	Set of strings compatible with the taggers' output
Other-morph	List of strings	Set of strings not compatible with the results of the tagger
Syn	Symbol	Syntactic POS tag as selected by tagger e.g. ADJ(ective), PROP(er), DET(erminer), PREP(osition)…

Orth	Symbol	Orthographic information for the tokens: C(apitalised) A(ll capitalised) L(ower case) I(nitial – single capital) 1(numbers) O(ther – any combination of symbols e.g punctuation, mixed capital and lower case letters).
Sem	List of symbols	Set of possible "semantic" tags: DATE TIME MNY (Money) PCT (Percentage) PER (Person) ARTIFACT LOC (Location) ORG (Organisation).
Type	Symbol	Subtype of a semantic class: For Locations : City Country Region … For Organisations : Company Institute Gov_agency Regulatory_body

A P P E N D I X B

RULE SET

The following rules are in the order suggested (cf. chapter 6). A capital letter that denotes the rule category and a number that denotes the order in the specific category are assigned to the rules to facilitate any reference to them.

```
#########################################################################
###########  Basic rules that complement - reinforce the database  ##############
```

(B1) : A rule for predictable festivals e.g. 'St. Patrick's Day'

```
 [SYN=PROP, SEM=FESTIVAL, ZONE=_Z] =>
 \[SYN=TITLE]?, [SEM=FESTIVAL]?, ([SYN=NAME], [SYN=POSS])?,
 [NORM="day", ORTH=C|A] /;

# Rules that define when a dateunit forms relative date expressions :
# STATUS=OPT>'last year', 'early this year', 'a year earlier',
#                    'two weeks earlier', 'last 16 years', 'three months ago',
#                    'next four years'...<
```

(B2) :

```
 [SYN=NP, SEM=REL_DATE, ZONE=_Z] =>
\[SEM=DATE_PRE|TIMEX_PRE]+, [SYN=NUM]?, [SEM=NUM]?, [SEM=DATEUNIT] /;
```

(B3) :

```
 [SYN=NP, SEM=REL_DATE, ZONE=_Z] =>
\ [SYN=NUM]?, [SEM=DATEUNIT], [SEM=TIMEX_PRE]+ /;
```

(B4) : Rule for abbreviated ordinals e.g. '3rd', '17th'

```
 [SYN=ORD, SEM=ORDABBR, ZONE=_Z] =>
```

\[SYN=NUM], [TOKEN="th"|"st"|"rd"]/;

(B5) : Normalisation of big cardinal numbers e.g. 'two hundred=2*100=200'

[SYN=NUM, SEM=CARD-NUM, NORM=(* _N1 _N2), ZONE=_Z] =>
\[SYN=NUM,NORM=_N1,ZONE=_Z],
[SYN=NUM,NORM=_N2,ZONE=_Z] / ;

(B6) : Normalisation of bigger cardinal numbers
e.g. '2 millions=2*1000000=2000000'

[SYN=NUM, SEM=CARD-NUM, NORM=(* _N1 _N2), ZONE=_Z] =>
\[SYN=NUM,NORM=_N1,ZONE=_Z],
[NORM=_N2,NORM=1000|1000000|1000000000, ZONE=_Z] / ;

##

##
############################## **P**erson Rules ############################

(P1) : A rule for famous people already in the database e.g. 'Clinton"...

[SYN=PROP, SEM=PER, ZONE=_Z] =>
\[SEM=PERSON_FULL, ORTH=C|A|O, ZONE=_Z] /;

(P2) : Rule for the case of finding the first name in the database e.g.
'Peter N. Kellogg', 'Karl de Shutter', 'John Von Armstrong',
'Katerina Pastra', 'Maria Helena Salvadora','Maria Pino Marino',
'Peter N.C. Kellogg, Jr.', 'J. Franklin Jones', 'Mona K. Van Shutten'...

[SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, ZONE=_Z]
=>
\[SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
[SEM=PERSON_AMBIG|PERSON_FEMALE|PERSON_MALE, ORTH=C|A,
TOKEN=_F, ZONE=_Z],

```
 [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [SYN=NAME, TOKEN=_M]?,
 [ORTH=C|A|O, SYN=NAME, TOKEN=_S, ZONE=_Z],
([SYN=COMMA], [NORM="jr."|"sr."])? /;
```

(P3) : A rule for cases when the first name is not known or present.

```
        # Whenever 'Jr.' or 'Sr.' follow a Proper, it is definitely a Person
        #  e.g. 'Panayiota N.C. Kellogg, Jr.'

 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, ZONE=_Z]
=>
\[SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
 [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [SYN=NAME, TOKEN=_M]?,
 [ORTH=C|A|O, SYN=PROP, TOKEN=_S, ZONE=_Z],
([SYN=COMMA], [NORM="jr."|"sr."]) /;

# Rules for the mark up of person names without known first name:
# The first restricts the cases by the presence of specific precedents
# and the second with a specific RHS context e.g.
# 'Mr Cass',  '...Mania Bellini as chief...',
# '...Nisha Sahran, MD...', '...Kiro Nakasian, Aviron's...'
```

(P4) :

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, ZONE=_Z]
=>
 [SEM=TITLE_MIL|TITLE_FEMALE|TITLE_MALE, ZONE=_Z]
\[SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
 [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
  [SYN=NAME, TOKEN=_M]?,
  [ORTH=C|A|O, SYN=PROP, TOKEN=_S, ZONE=_Z, SOURCE!=RULE] /;
```

(P5) :

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, OCC=_O,
  ZONE=_Z] =>

\[SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
 [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [SYN=NAME, TOKEN=_M]?,
 [ORTH=C|A, SYN=PROP, TOKEN=_S, ZONE=_Z, SEM!=OCC|TITLE,
  SOURCE!=RULE] /
 [NORM=","|"as"|"the"|"to"]+, [SEM=OCC|TITLE|TITLE_MODIFIER, TOKEN=_O];
```

(P6) : Rule for cases such as : '..." said Kalia Marino,'...

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, ZONE=_Z]
=>
 [ROOT="say"|"state"]
\[SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
 [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [SYN=NAME, TOKEN=_M]?,
 [ORTH=C|A, SYN=PROP, SEM!=OCC|TITLE, TOKEN=_S, ZONE=_Z,
  SOURCE!=RULE] /
 [SYN=PUNCT];
```

(P7) : A rule for cases such as : '...president, Lania Marita,..'

```
 [SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S, OCC=_O,
  ZONE=_Z] =>

 [SEM=OCC, TOKEN=_O, ZONE=_Z], [NORM=","]?
\[SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [ORTH=C|A, SYN=PROP, MORPH=(("NIL")), TOKEN=_F, ZONE=_Z]?,
 [SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
 [SYN=NAME, TOKEN=_M]?,
 [ORTH=C|A|O, SYN=PROP, TOKEN=_S, ZONE=_Z, SOURCE!=RULE] /
```

[SYN=PUNCT];

(P8) : Co-reference Rule E.G. 'Peter Kelogg...Pit Kellogg...Kelogg'

```
[SYN=PROP, SEM=PER, FULL_FORM=_FF, ZONE=_Z] =>

\[SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
[ORTH=C|A, SYN=PROP, TOKEN=_F, ZONE=_Z]?,
[SYN=NAME, ORTH=I|O, TOKEN=_I, ZONE=_Z]?,
[SYN=NAME, TOKEN=_M]?,
[ORTH=C|A|O, SYN=PROP, TOKEN=_S, ZONE=_Z, SOURCE!=RULE] /
>>
[SYN=PROP, SEM=PER, FIRST=_F, INITIALS=_I, MIDDLE=_M, LAST=_S,
 SOURCE=RULE, TOKEN=_FF, ZONE=_Z] ;
```

##

##
########################## **L**ocation rules ##########################

(L1) : Rule that marks up location found in the database e.g.
'Chicago based company', 'California's', 'South America', 'Cape Town',
'Balkans', Soviet Union', 'Beverly Hills','Palm Springs', 'Sahara Desert',
'Los Angeles', 'Democratic Republic of China', 'Rhode Island',
'Pearl Harbor', 'Lake District', 'North Carolina', 'Gatwick' e.t.c

```
[SYN=PROP, SEM=LOC, TYPE=_T, ZONE=_Z] =>

\[SEM=REGION_HEAD|LANDREGION_HEAD|WATERREGION_HEAD|CITY_PART|
        COUNTRY_PART|PROVINCE_HEAD|LOCATION_PREFIX, ORTH=C|A]?,

[SEM=REGION|USSTATE|WATER|CITY|CONTINENT|COUNTRY|PROVINCE|ISLAND
        |AIRPORT, SEM=_T, ORTH=C|A|O, ZONE=_Z],

[SEM=REGION_HEAD|LANDREGION_HEAD|WATERREGION_HEAD|CITY_PART|
        COUNTRY_PART|PROVINCE_HEAD|STREET|FACILITY|FACILITY_HEAD,
        ORTH=C|A]? /;
```

(L2) : Rule for locations that do not exist in the database but are followed by
a token that denotes location e.g. 'Fellow City', 'Kokomo Valley',

```
        # 'Asian territories', 'European counties'...


[SYN=PROP, SEM=LOC, LOC_PRE=_L, LOC_HEAD=_L1, LOC_TYPE=_T, ZONE=_Z]
=>

\[SEM=REGION_HEAD|LANDREGION_HEAD|WATERREGION_HEAD|CITY_PART|
        COUNTRY_PART|PROVINCE_HEAD|LOCATION_PREFIX,
 ORTH=C|A,NORM=_L]?,
[ORTH=C|A, SOURCE!=RULE, NORM=_L1],

 [SEM=REGION_HEAD|LANDREGION_HEAD|WATERREGION_HEAD|CITY_PART|
        COUNTRY_PART|PROVINCE_HEAD|STREET|FACILITY|FACILITY_HEAD,
 ORTH=C|A, NORM=_T] /;



#Rules for locations not identified as such by the above rules.
# Appositions (common with locations) can help e.g.
# 'Kokomo, Indiana', 'Washington, D.C'
```

(L3) :

```
  [SYN=PROP, SEM=LOC, GREATER_REGION=_G, ZONE=_Z] =>

\ [ORTH=C|A, SOURCE!=RULE, ZONE=_Z] /
  [NORM=","], [SEM=LOC, SOURCE=RULE, TOKEN=_G, ZONE=_Z];
```

(L4) :

```
 [SYN=PROP, SEM=LOC, GREATER_REGION=_G,ZONE=_Z] =>

\[ORTH=C|A, SOURCE!=RULE, ZONE=_Z]{2,2} /
 [NORM=","], [SEM=LOC, TOKEN=_G, SOURCE=RULE, ZONE=_Z];
```

(L5) :

```
 [SYN=PROP, SEM=LOC, ZONE=_Z] =>

 [SEM=LOC, SOURCE=RULE, ZONE=_Z], [NORM=","]
\[ORTH=C|A, SOURCE!=RULE, ZONE=_Z] /
 [SYN=PUNCT] ;
```

(L6) : A rule for co-reference - cases when a location not in the database is explained once (full name) and then referred again partially e.g

```
        # 'Crookes Valley.....Crookes', 'Waltham City...Waltham',
        # 'New Psychikon Region...New Psychikon...Psychikon'


 [SYN=PROP, SEM=LOC, FULL_FORM=_A] =>
```

```
\[ORTH=C|A, NORM=_L]?, [ORTH=C|A, NORM=_L1] /

>>

 [SEM=LOC, SOURCE=RULE, LOC_PRE=_L, LOC_HEAD=_L1, TOKEN=_A] ;


##########################################################################


##########################################################################
############################ Artifact   rules ##############################


(A1) :  Rule that identifies a trade-name in the title of the document (de-activated rule)
        # e.g. 'MULTIKINE(TM)', 'CLARITY 7000'...

# [SYN=PROP, SEM=ARTIFACT, MAIN_NAME=_M, SPECIFIER=_P, ZONE=_Z] =>

# [SYN=POSS|DET]
#\[SYN=PROP|NN, EDGENO<30, ORTH=C|A|O, SOURCE!=RULE, SEM!=OCC,
#  TOKEN=_M, ZONE=_Z],
# [ORTH=1, MORPH=(("CARD")), ZONE=_Z, TOKEN=_P]? /;


(A2) : Same as above, but for the rest of the text...

 [SYN=PROP, SEM=ARTIFACT, MAIN_NAME=_M, ABBR=_A, ZONE=_Z] =>

 [SYN=POSS|DET]
\[SYN=PROP|NN, ORTH=C|A|O, MORPH=(("NIL")), SOURCE!=RULE, TOKEN=_M,
  ZONE=_Z],
 [ORTH=1|O|C|A, SYN!=PUNCT|DET|POSS, ZONE=_Z]{0,3} /
 [SYN!=NN|PREP|PROP|NUM],
 [SYN=OPEN]?, [ORTH=C|A, TOKEN=_A]?, [SYN=CLOSE]? ;


(A3) :  Rule for the cases when the trade-name is followed by a word that indicates it..
        # e.g 'Clarity database', 'GeneLex, Europe's foremost package'...


 [SYN=PROP, SEM=ARTIFACT, MAIN_NAME=_M, ABBR=_A, KIND_OF=_K,
  ZONE=_Z] =>

\[SYN=PROP|NN, SYN!=PUNCT|DET|POSS, ORTH=C|A|O, SOURCE!=RULE,
  TOKEN=_M, ZONE=_Z],
 [ORTH=1|O|C|A, SYN!=PUNCT|DET|POSS, ZONE=_Z]{0,3} /
 [NORM=","|"the"|"a"|"an"]*,
([SEM=LOC, SOURCE=RULE], [NORM="'s"])?,
 [SEM=PORCAPP]?,
 [SEM=PHEAD, ORTH=L, TOKEN=_K] ,
 [SYN=OPEN]?, [ORTH=C|A, TOKEN=_A]?, [SYN=CLOSE]? ;
```

(A4) :

```
 [SYN=PROP, SEM=ARTIFACT, MAIN_NAME=_M, ABBR=_A, ZONE=_Z] =>

\[SYN=PROP|NN, SYN!=PUNCT|DET|POSS, ORTH=C|A|O, SOURCE!=RULE,
  TOKEN=_M, ZONE=_Z],
 [ORTH=1|O|C|A, SYN!=PUNCT|DET|POSS, ZONE=_Z]{0,3},
 [SEM=PHEAD, ORTH=C|A] /
 [SYN=OPEN]?, [ORTH=C|A, TOKEN=_A]?,[SYN=CLOSE]?;


# Two  Co-reference rules
# The first for cases when the main part of the name is repeated,
# the second for cases when the Artifact name is repeated abbreviated
# e.g. 'Multikine ™  (MT)….MT…'
```

(A5) :

```
 [SYN=PROP, SEM=ARTIFACT, FULL_FORM=_F, ZONE=_Z] =>

\[ORTH=C|A|O, TOKEN=_M, ZONE=_Z],
 [ORTH=C|A|O|1, SYN!=PUNCT|DET|POSS, ZONE=_Z]{0,3} /

>>

 [SEM=ARTIFACT, SOURCE=RULE, MAIN_NAME=_M, TOKEN=_F] ;
```

(A6) :

```
 [SYN=PROP, SEM=ARTIFACT, FULL_FORM=_F, ZONE=_Z] =>

\[ORTH=C|A, TOKEN=_A, ZONE=_Z] /

>>

 [SEM=ARTIFACT, SOURCE=RULE, ABBR=_A, TOKEN=_F] ;


##########################################################################


##########################################################################
########################## Organisation  rules ############################
```

(ORG1) : Rule for organisations that are present in the database
e.g. 'Nato', 'University of Manchester'...

```
 [SYN=PROP, SEM=ORG, TYPE=_T, ZONE=_Z] =>

\[SEM=LOC]?,
 [SEM=GOV_AGENCY|ORG|UNIVERSITY, ORTH=C|A|O, ZONE=_Z, SEM=_T] /
```

```
([NORM!="of"],[SEM!=LOC]);
 # Rules for one word and multiword companies followed by 'cdg'
#          e.g. 'MedLex, inc', 'Asset Biovista, ltd', Boron, Lepore & Associates, co',
#          'Cell Therapeutics, inc. ("CTI") (Nasdaq: CTIC)' ...
```

(ORG2) :

```
 [SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F,
  ABBR=_A, IN=_I, ABBR2=_A2, IN=_I2, KIND_OF=_K, ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=DET|PUNCT, NORM=_F, ZONE=_Z] /
 [NORM=","]?, [SEM=CDG, NORM=_K],
 [NORM=","]?,
 [SYN=OPEN|QUOTE]{0,2},
([ORTH=C|A|O, TOKEN=_I, ZONE=_Z], [NORM=":"])?,
 [ORTH=C|A|O, TOKEN=_A]?,
 [SYN=OPEN|QUOTE]{0,2}, [NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_I2, ZONE=_Z], [NORM=":"], [ORTH=C|A|O,
   TOKEN=_A2], [NORM=")"])?;
```

(ORG3) :

```
 [SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, LAST=_L,
  ABBR=_A, IN=_I, ABBR2=_A2, IN=_I2, KIND_OF=_K, ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=DET|PUNCT, NORM=_F, ZONE=_Z],
 [ORTH=C|A|O|1, NORM!="."|"/"|"--", SOURCE!=RULE, ZONE=_Z]{0,3},
 [ORTH=C|A|O|1, SYN!=PUNCT|DET, NORM=_L, ZONE=_Z] /
 [NORM=","]?, [SEM=CDG, NORM=_K],
 [NORM=","]?,
 [SYN=OPEN|QUOTE]{0,2},
([ORTH=C|A|O, TOKEN=_I, ZONE=_Z], [NORM=":"])?,
 [ORTH=C|A|O, TOKEN=_A]?,
 [SYN=OPEN|QUOTE]{0,2}, [NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_I2, ZONE=_Z], [NORM=":"], [ORTH=C|A|O,
   TOKEN=_A2], [NORM=")"])?;
```

(ORG4) : Rule for cases when part of the org-name is a word in the database

```
#    designating its type e.g.
#    'OxyGen Medical Center', 'University of Athens',
#    'Case Western Reserve University'
#    'Fregit Gurtwinkle Center for Mushroom Spore Transduction'
#    'Royal Health Institute', 'Ministry of Defence', 'Embassy of France'
#    'Pennsylvania State Nurses Association', 'First Southwest Company'...
#    'Food and Drug Administration'...


 [SYN=PROP, SEM=ORG, TYPE=GOV_AGENCY, FIRST=_F, LAST=_L, ABBR=_A,
  ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=PUNCT|DET, NORM=_F, ZONE=_Z]{0,1},
 [ORTH=C|A|O|1, SYN!=PUNCT|DET, ZONE=_Z]{0,3},
 [SEM=GOV_HEAD, ORTH=C|A, NORM=_L, ZONE=_Z],
```

```
 [NORM="of"|"for"]?, [ORTH=C|A|O, SYN!=PUNCT, ZONE=_Z]{0,4} /
 [NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_A],
 [NORM=")"])?;
```

(ORG5) :

```
 [SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, LAST=_L, ABBR=_A,
  ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=PUNCT|DET, NORM=_F, ZONE=_Z],
 [ORTH=C|A|O|1, SYN!=PUNCT|DET, ZONE=_Z]{0,3},
 [NORM="company", ORTH=C|A, NORM=_L, ZONE=_Z] /
 [NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_A],
 [NORM=")"])?;
```

(ORG6) :

```
[SYN=PROP, SEM=ORG, TYPE=INSTITUTE, FIRST=_F, LAST=_L, ABBR=_A,
 ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=PUNCT|DET, SOURCE!=RULE, NORM=_F, ZONE=_Z]{0,1},
 [NORM="and"]?,
 [ORTH=C|A|O|1, NORM!=","|".", ZONE=_Z]{0,3},
 [SEM=UNIV_HEAD|INSTITUTE_HEAD, ORTH=C|A, NORM=_L, ZONE=_Z],
([NORM="of"|"for"|"and"], [NORM="the"], [ORTH=C|A, ZONE=_Z]){0,3},
([NORM="of"|"for"|"and"], [ORTH=C|A, ZONE=_Z]){0,3}, [ORTH=C|A, ZONE=_Z]{0,4}/
 [NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_A],
 [NORM=")"])?;
```

(ORG7) : When a Proper name or a sequence of, is followed by a specific type of
abbreviation, then we have a company e.g.
'IMS HEALTH (NYSE: RX)',
'Biovail (NYSE: BVF) (TSE: BVF)',
'Immune Response (OTCBB: IMUN)'...

```
 [SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, ABBR=_A, IN=_I,
  ABBR2=_A2, IN=_I2, ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=PUNCT|DET, SEM!=CDG, SOURCE!=RULE, NORM=_F,
  ZONE=_Z],
 [ORTH=C|A|O|1, SYN!=PUNCT|DET, SEM!=CDG, SOURCE!=RULE, ZONE=_Z]{0,4} /
 [NORM=","]?,
 [SYN=OPEN], [ORTH=C|A|O, TOKEN=_I, ZONE=_Z], [NORM=":"], [ORTH=C|A|O,
  TOKEN=_A],
 [SYN=CLOSE], [NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_I2, ZONE=_Z], [NORM=":"], [ORTH=C|A|O,
   TOKEN=_A2], [NORM=")"])?;
```

(0RG8) : Rule for cases such as: 'Limatex is a leading provider',
'GeneLex is Europe's leading supplier'....

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, KIND_OF=_K, ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=PUNCT|DET|POSS,SEM!=OCC,SOURCE!=RULE,NORM=_F,
 ZONE=_Z],
[ORTH=C|A|O|1, SYN!=PUNCT|DET, SEM!=OCC, SOURCE!=RULE, ZONE=_Z]{0,3} /
[NORM=","|"is"|"the"|"a"|"an"]*,
([SEM=LOC, SOURCE=RULE], [NORM="'s"])?,
[SEM=PORCAPP]?, [SEM=CHEAD, TOKEN=_K] ;
```

(ORG9) : When a location precedes an unknown proper noun, a company name may
be denoted e.g. 'Oxford GlycoSciences', 'U.S Bacou Solutions'...

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, ABBR=_A,
 LOC_INDICATION=_LOC, ZONE=_Z] =>

\[SEM=LOC|COUNTRY_ADJ, TOKEN=_LOC, SOURCE=RULE],
[ORTH=C|A|O, SYN!=PUNCT|DET|POSS,SEM!=OCC|CDG|PHEAD,
 SOURCE!=RULE, NORM=_F,ZONE=_Z],
[ORTH=C|A|O|1,SYN!=PUNCT|DET|POSS, SEM!=OCC|CDG|PHEAD,
 SOURCE!=RULE, ZONE=_Z]{0,3} /
[NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_A], [NORM=")"])?;
```

(ORG10) : Rule for cases such as:
'The President of Latimex Biorad',
'The Board of Directors of Pinox',
‘subsidiary of Wertex Neurosciences'

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, ABBR=_A, ZONE=_Z] =>
[SEM=OCC|COMPANY_PART],[NORM="at"|"of"]?
\[ORTH=C|A|O,SYN!=PUNCT|DET|POSS,SEM!=OCC|CDG, SOURCE!=RULE,
 NORM=_F, ZONE=_Z],
[ORTH=C|A|O|1,SYN!=PUNCT|DET|POSS, SEM!=OCC|CDG, SOURCE!=RULE,
 ZONE=_Z]{0,3} /
[NORM=","]?,
([NORM="("], [ORTH=C|A|O, TOKEN=_A], [NORM=")"])?;
```

(ORG11) : Rule for cases such as: 'Lenox acquisition of', 'Lenox purchase of',
'Trinity BioSciences's clients'

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FIRST=_F, ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=PUNCT|DET|POSS|NUM|CONJ, SEM!=OCC|TITLE|CDG,
```

```
 SOURCE!=RULE, NORM=_F, ZONE=_Z],
[ORTH=C|A|O|1,SYN!=PUNCT|DET|POSS, SEM!=OCC|TITLE, SOURCE!=RULE,
  ZONE=_Z]{0,3} /
[NORM="'s"]?, [SEM=OCC|COMPANY_PART|ARTIFACT] ;
```

(ORG12) : Two co-reference rules: one just for companies and one for all types of
organisations when an abbreviation is used :

```
[SYN=PROP, SEM=ORG, TYPE=COMPANY, FULL_FORM=_F, ZONE=_Z] =>

\[ORTH=C|A|O, SYN!=DET|PUNCT|POSS, NORM=_F1, ZONE=_Z],
[ORTH=C|A|O, SYN!=DET|PUNCT|POSS, SEM!=OCC|TITLE,
 GOOD-MORPH(("NIL")),  SOURCE!=RULE,  ZONE=_Z ]{0,3}  /
>>
[SEM=ORG, SOURCE=RULE, TYPE=COMPANY, FIRST=_F1, TOKEN=_F,
 ZONE=_Z];
```

(ORG13) :

```
[SYN=PROP, SEM=ORG, TYPE=_T, FULL_FORM=_F, ZONE=_Z] =>

\[ORTH=A|C|O, SYN!=DET|PUNCT|POSS, TOKEN=_A, ZONE=_Z] /
>>
[SEM=ORG, SOURCE=RULE, TYPE=_T, ABBR=_A, TOKEN=_F, ZONE=_Z] ;
```

```
##########################################################################
```

```
##########################################################################
```
########################## **D**ate rules ##########################

(D1) : Variety of cases e.g.
'end of next June', 'this Monday morning', '3rd of July 2002'
'Christmas 2000' , 'yesterday evening', 'spring 1999', 'June 30'

```
[SYN=NP, SEM=DATE, ZONE=_Z] =>

\[SEM=DATE_PRE|TIMEX_PRE]*,
[SYN=NUM, GOOD-MORPH(("Card"))]?, ([SYN=ORD], [NORM="of"])?,
[SEM=MONTH|WEEKDAY|DATE|SEASON|FESTIVAL|REL_DATE, ZONE=_Z],
[SYN=NUM]?,
[SEM=TIME|TIMEUNIT]?,
([SYN=DET], [SYN=ORD])? /
[SYN=PUNCT|CONJ|POSS|PROP|NN|PREP] ;
```

(D2) : Rule that recognises just years e.g.
'in 1999', 'end of 2002' , 'in the late 1960s', 'The new Pirax 2000 Index'

```
 [SYN=NUM, SEM=DATE, ZONE=_Z] =>

 [SEM=DATE_PRE|TIMEX_PRE]*
\[ORTH=1, GOOD-MORPH=(("Card")), TOKEN="????", NORM<2100, ZONE=_Z],
 [NORM="s"]? /
 [SYN!=PL|NUM|ADJ];
```

(D3) : Rule for abbreviated format of dates e.g. '03-05-99'

```
 [SYN=NUM, SEM=DATE, ZONE=_Z] =>
\[SYN=NUM, ZONE=_Z, TOKEN="??-??-??"] /;
```

(D4) : Rule that defines when an event may form a date expression
e.g. 'end of World War II'

```
 [SYN=NP, SEM=DATE, ZONE=_Z] =>
\[SEM=DATE_PRE], [SEM=EVENT, ORTH=C|A] /;
```

(D5) : Rule that deals with centuries e.g.
'21st century', 'eighteenth century'

```
  [SYN=NP, SEM=DATE, ZONE=_Z] =>
\ [SYN=ORD], [NORM="century"] /;
```

(D6) : Rule mainly for fractions + date expressions e.g.
'last quarter of 1999', 'first half of the fiscal 2000'
'second half of eighteenth century',
'50th anniversary of the end of World war II'
'4th of July', 'first of June'

```
 [SYN=NP, SEM=DATE, ZONE=_Z] =>

 [SYN=PREP|DET]?
\[SEM=NUM|ORDABBR|EVENT]+, [SEM=FRAC]?,
 [SYN=PREP|DET]*,[NORM="fiscal"]?,
 [SEM=DATE, SOURCE=RULE, ZONE=_Z] /;
```

```
##########################################################################
```

```
##########################################################################
############################## Time   rules ##############################
```

(T1) : Rule for time expressions consisting mainly of a numeral e.g.
'2:50 p.m.' '10 a.m.', '13:30', '1:30 p.m. EST'...

```
 [SYN=NUM, SEM=TIME, ZONE=_Z] =>
\[SYN=TIME], [SEM=TIMEZONE]* /;
```

(T2) : Cases :‘last evening’, ‘8 in the morning’, ‘nine in the evening’,
'eight thirty in the evening', '2a.m', '1900EST'...

```
 [SYN=NP, SEM=TIME, ZONE=_Z] =>

 [SYN!=DET]
\[SEM=TIMEX_PRE]*,
([SYN=NUM], [SYN=PREP], [SYN=DET])?,
 [SYN=NUM]*,
 [SEM=TIME|TIMEUNIT|TIMEZONE, SYN!=PL, ZONE=_Z]+ /
 [SYN!=NN|ADJ];
```

(T3) : STATUS=OPT> Rule for periphrastic time expressions e.g.
‘a quarter to nine’, ‘half past seven’,
‘five minutes to eight, ‘8 o’clock’, ‘nine o’clock’<

```
 [SYN=NP, SEM=TIME, ZONE=_Z] =>

\[SYN=NUM]?, [SEM=FRAC|TIMEUNIT]?,
 [NORM="past"|"to"|"o'clock"], [SEM=TIMEUNIT]?, [SYN=NUM, SEM!=DATE]? /
 [NORM="."|","|"and"|"or"];
```

(T4) : Rule for 'location indication' in time expressions :
'local time', 'Greek time'

```
 [SYN=NP, SEM=TIME, LOC_INDICATION=_L, SPECIFIC_TIME=_S, ZONE=_Z] =>

 [SEM=TIME, SOURCE=RULE, TOKEN=_S,ZONE=_Z], [NORM=","]?
\[SEM=LOC|COUNTRY_ADJ, TOKEN=_L]?, [SYN=ADJ, TOKEN=_L]?,
 [NORM="-"]?, [NORM="time", ZONE=_Z] /;

##########################################################################
```

```
##########################################################################
```
########################## **M**oney rules ##########################

(M1) : Rule for money expressions with a currency sign e.g.
‘ $ 2.6 million ‘, ‘£400 bn’, ‘£175,000’, ‘- £50.3 m’...

```
 [SYN=NP, SEM=MNY, nkrl=(spec. money_ Curr(spec. amount_AMT)), ZONE=_Z] =>

\[TOKEN="-"|"minus"]?,
 [NORM=_Curr, TOKEN="£"|"$", ZONE=_Z],
 [SYN=NUM,  NORM=_AMT, ZONE=_Z],
 [TOKEN="m"|"bn"]? /;
```

(M2) : Rule for expressions with full currency reference e.g.
'200 million pounds', 'several thousand dollars',
'three hundred billion drachmas minus', '-200,000 Cypriot pounds'

```
 [SYN=NP, SEM=MNY, ZONE=_Z] =>

\[NORM="several"]?, [TOKEN="-"|"minus"]?,
 [SYN=NUM]?,[SEM=CARD-NUM|NUM, ZONE=_Z]*,[TOKEN="m"|"bn"]?,
 [SEM=COUNTRY_ADJ]?,
 [SEM=CURRENCY_UNIT|CURRENCY_ABBR, ZONE=_Z], [TOKEN="-" | "minus"]? /;

################################################################################

################################################################################
############################## Percentage rules ################################

(PCT1) : Rule for expressions with a percentage sign e.g. '30%'

 [SYN=NUM, SEM=PCT, ZONE=_Z] =>
\[SYN=NUM], [SEM=PERCENT] /;


(PCT2) : Rule for expressions with the word 'percent' e.g.
# '30 percent', '30 per cent', '30 per-cent'

 [SYN=NUM, SEM=PCT, VALUE=_R, ZONE=_Z] =>
\[SYN=PERCENT, ZONE=_Z, ROOT=_R] /;

################################################################################

################################################################################
############################## General rules ###################################

# Two rules for cases of elision e.g.
#               '6 - 7 April', 'April 5 and 7'
#               '50 to 60 percent',
#               '50 - 60 million dollars',
#               'North and South America'

(G1) :

 [SYN=_X, SEM=_S, ZONE=_Z] =>

\[SYN=NUM|PLACE, SYN=_X, ZONE=_Z, SOURCE!=RULE]+ /
 [NORM="-"|"and"|"or"|"to"], [SEM=_S, SOURCE=RULE, ZONE=_Z];
```

(G2) :

[SYN=_X, SEM=_S, ZONE=_Z] =>

[SEM=_S, SOURCE=RULE, ZONE=_Z], [NORM="-"|"and"|"or"|"to"]
\[SYN=NUM|PLACE, SYN=_X, ZONE=_Z, SOURCE!=RULE]+ /;

##

APPENDIX C

SAMPLE RESULTS

1) A text from PRNewswire, 9th of August, from the 'Today's News in Pharmaceuticals, Biotechnology and Health' section :

Key: Loc Artifact Per Date Mny Org

PRINCETON, **N.J**. and **OXFORD**, **England**, **Aug. 9** /PRNewswire/ -- **Pharmacopeia**, Inc. (**Nasdaq**: **PCOP**) and **Oxford Molecular Group** Plc ("**OMG**") (**London**: **OMG**) **today**, announced that they have signed a definitive agreement pursuant to which **Pharmacopeia** will acquire **OMG**'s software subsidiaries subject to **OMG** shareholder approval. Once completed, this acquisition will provide **Pharmacopeia** with three key assets: an industry leading position in the large and growing field of bioinformatics, a broad line of cheminformatics products that complement **Pharmacopeia**'s existing chemical database tools from **Synopsys Scientific Systems Ltd**, and a group of more than 120 talented employees, many of whom are highly skilled scientific software developers.

OMG**'s software business, which generated revenues from continuing products** of approximately **$15 million** in **1999**, designs and markets bioinformatics and cheminformatics products for the pharmaceutical, biotechnology and chemical industries. **OMG**'s **GCG** product line gives researchers the power to capture, manage, analyze, compare, and mine the huge volumes of genetic data created in the search for novel therapeutic targets. The planned acquisition will augment **Pharmacopeia**'s existing suite of products with state-of-the-art bioinformatics solutions that allow researchers to analyze DNA and protein sequence data. **OMG**'s bioinformatics solutions range from a single user desktop to enterprise-wide server licensing and include well-known products, such as **Winsconsin Package** (**TM**).

The planned acquisition will also provide **Pharmacopeia** with cheminformatics solutions that enable multidisciplinary research teams to capture, analyze and communicate the increasing volumes of biological and chemical data created in the search for new lead compounds and drug candidates. These products will complement **Pharmacopeia**'s existing cheminformatics products available through **Synopsys Scientific Systems Ltd**. **OMG** cheminformatics products include the **RS3** packages.

"The acquisition of **Oxford Molecular**'s software business is strategically very significant for **Pharmacopeia**," said **Joseph A. Mollica**, **Pharmacopeia**'s Chairman, President and CEO. "We believe their bioinformatics products are unmatched in quality and scientific utility. They serve an increasingly important function as efforts continue to sequence and understand the human genome. Two especially exciting areas of complementarity of expertise and product offerings are functional genomics and proteomics, where **GCG**'s sequence

analysis strength can be combined with our strength in assigning protein structure and its subsequent function. In addition, the critical mass we will achieve from **OMG**'s cheminformatics products will benefit our customers as they seek to leverage the important chemical data emerging from **today**'s drug discovery efforts. This acquisition should serve to fuel Pharmacopeia's continued growth."
"As we've stated again" continued Dr. **Mollica**, "our goal is to provide a broad spectrum of integrated modeling, simulation, cheminformatics, and bioinformatics products and services that help our customers accelerate drug discovery and chemical development. The acquisition of **OMG**'s software business will be another step toward achieving this goal."
Under the terms of the agreement, which is expected to close in **late August** and is conditional upon a majority vote of **OMG** shareholders, **Pharmacopeia** will acquire **OMG**'s software business for approximately **$27 million** comprised of cash and the assumption of certain liabilities. The acquisition will be accounted for using the purchase method of accounting.
Upon completion of the acquisition, **Pharmacopeia** will integrate **OMG**'s bioinformatics and cheminformatics products into its existing Life Sciences software business.

2) The Text we created for testing all the rules (cf. Part 2, Chapter 1, Methodology):

Key: Loc Artifact Per Date Mny Org Time PCT

17 - 18 October /PRNewswire/ **1999**.
In **April 6** - **7**, **Trinity** Ltd will sign the contract.
In **October the ninth**, the chairman resigned.
On **Christmas**, the employees will receive a present.
In **early Friday**, the president will make an announcement.
The assignment is due end of **this June**.
February 30, **2000** : The new **Clarity Database** will be a reality.
On **Monday morning**, all reports will have been completed.
The project is due **end of 2009** for all the participants.
The new Pirax **2000** Index is now available!
In **2008** or **2009**, the project will have finished.
In the **1960s**, the policy of the company was different.
Early this year, the company employed 0.1m people.
During the **last 16 years**, the company's revenue augmented significantly.
Three months ago, the stock market had serious problems.
In the **next four years**, the revenue will become double.
Last spring, revenues augmented **3%**.
In **today**'s news, there is information on our company's investments.
In **03-09-88**, the vice president resigned.
From **1999** to **2000**, the formula will have been finished.
In **spring 2009**, the project will be in its last phase.
Spring - **summer 1999**, is a 'dead' period.

The event will take place, on **Monday**, **June 16**, **2000**.
August, **last year**, the agreement was a reality.
In **1999** - **2000**, the sales will augment.
In the **second half of next year**, the project will have finished.
In the **first quarter of 1999**, the agreement was a reality.
Americans celebrate on the **fourth of July**.
Americans celebrate on the **4th of July**.
The revenues increased from 4.4 to 1.8 **last year**.
We have entered the **21st century**.
The **last quarter of the 18th century**, was a turning point in the history of Biotechnology.
At the **50th anniversary of the end of World War II**, many people went to **Auswitz**.
In **Valentine's Day**, people buy flowers.
Easter Monday, is a day people fasten.
It's **St. Patrick's Day**.
Boxing Day : All banks are closed.

At **1900EST**, the conference will have finished.
At **2:50 p.m**., everybody will be back at work.
It was **2:30 a.m. EST**, when the flood happened.
It was **2300 EST**, when the flood happened.
It was **8 in the morning**, when the door bell rung.
It was just eight in the morning, when the door - bell rung.
It's **2:50 p.m., New York City time** and everybody will be back at work soon.
At **1:30 p.m. local time**, the stock exchange closed.
At **13:30 Croatian time**, the man came out.
The conference will last from **13:30** until **23:00**.
It will be held at **5 past 7** or at **five to seven**.
It will be held at **five minutes to seven**.
The game starts at **half past six**.
The game starts at a **quarter to nine**.
The conference is at **9 o'clock**.
The conference is at **eight o'clock**, **in the morning**.
It happened at **six in the morning**, **local - time**.
It happened at **2:30 p.m. Crookes Valley time**.

The price was between **$50 million** and **$60 million**.
In **January 2000**, a new head of department was elected.
The transaction involved something like **£5,000** (**$10,000**).
It was just **$2.6 billion** that were invested.
Two hundred pounds is a big amount of money.
This car costs **several thousand dollars**.
500 hundred people died last year, in the hurricane.
The **five thousand dollar** investment was just the start.
The house cost approximately **$19,000**.
£0.15 was our company's profit.

50-80 percent of them suffers from ovarian cancer.
55 percent to **75 percent** of young people is shortsighted.
50 to **70 percent** of women suffer from osteoporosis.

30% of our lecturers is not qualified.
His account balance is : **-298,000 french francs**.

Final decisions will be made in **Washington**, **D.C**!
In **North** and **South America**, the percentage has augmented.
The company has partners all around the world :
Cape Town, **Balkans**, **Soviet Union**, **Beverly Hills**,
Los Angeles, **Democratic Republic of China**, **Rhode Island**,
Lake District, **North Carolina**,**Palm Springs** just to name some...
The company is near **New Psychikon Region**.
The headquarters are in **New York City**. **New Psychikon** is our base.
Gatwick is one of the biggest airports...
Crookes Valley is the place where the subsidiary is situated...
This **Crookes** based company has invested more than **$4,000m** in our products.
The headquarters are in **California**'s **Silikon Valley**.
The matter will be discussed in **Canada**'s **Parliament.**
It is **GeneExpress 718** system that we work with.
Dr. **Dudkevitch** from the **Department of Otolaryngology** at the **Rabin Medical Center** in **Pech Tiqva**, **Israel**, presented additional interim data in advanced head & neck cancer patients who were specially treated prior to surgery/radiation.
Our **GenElex 2000** database and **GeneExpress** system are impressive...
GeneLex, Europe's foremost package, is now available.

Biogen Inc. Appoints **Peter N. Kellogg** Vice President, Finance and Chief Financial Officer.

CAMBRIDGE, **Mass**., **July 10** /PRNewswire/ -- **Biogen, Inc**. (**Nasdaq**: **BGEN**) **today**, announced that **Peter N. Kellogg** has been appointed Vice President, Finance and Chief Financial Officer. Mr. **Kellogg** reports to Mr **C.K. Mullen**, **Biogen**'s President and Chief Executive Officer.
Karl de Shutter, **John Von Armstrong** and **Maria Helena Salvadora** are all employed in **Trinity ltd**..
J. Franklin Jones is the president.
Maria K. Van Shutten is the president.
Maria Pino Marino is the president.
Akweh N. Kinny, Jr., is the ambassador.
Mr **Cass** left suddenly.
Aviron's president, **Lania Marita**, and the board of directors appointed **Mania Bellini** as Chief Executive Officer.
Bellini and **Marita** have been collaborating since **1999**.
Nisha Sahran, PH.D., was appointed at the post.
Kiro Nakasian, the Executive officer, is responsible for the sector.
"It was a great success", said **Kalia Pertino**.
Bill Clinton will discuss the matter thoroughly.
Dr **Mahmud Al Farrad** is here.
Nakasian and **Al Farrad** will be examined by the board.
Nisha Sahran retired last week.

Medlex, Inc. announces the results of its **1990** - **2000** research.
OxyGen Medical Center is the biggest health center in the world.

The **University of Athens** is affiliated with **Case Western Reserve University** and **Temple University's Graduate School**.
Fregit Gurtwinkle Center for Mushroom Spore Transduction has many employees.
The **Royal Health Institute**, the **Ministry of Defence** and the **Embassy of France** are collaborating in the project.
Pennsylvania State Nurses Association, **Ingenix Pharmaceutical Services**, and **United Health Group** have already signed the agreement.
Boron, Lepore & Associates, inc. is one of the world's top suppliers.
Hyunday inc., of Korea sold more than 30,000 cars last year.
Oxford GlycoSciences announces the results of its research.
Lenox acquisition of **Radimel**(**TM**) system is a fact.
Trinity BioSciences's clients are satisfied.
Limatex is a leading provider of software programms.
The President of **Latimex Biorad** announced the collaboration.
The Board of Directors of **Pinox** will organise the meeting at the offices of the wholy owned subsidiary of **Wertex Neurosciences**.
The **Pinox** Board of directors will speak.
Cell Therapeutics, inc. (**CTI**) will announce the results.
Synaptic Pharmaceutical Corporation (**Nasdaq**: **SNAP**) today announced it.
Biovail (**NYSE**: **BVF**) (**TSE**: **BVF**) announced the results.
GlycoSciences will announce the results.

B I B L I O G R A P H Y

Allerton D.J (1987), *The linguistic and sociolinguistic status of Proper Names*, in : Journal of Linguistics, 11 (3).

Attar T. (2000), *The extraction of key words and phrases from MUC-7 texts,* Final Year Project, CCL, UMIST.

Bellinger G. (2000), *Knowledge Management – Emerging Perspectives*, an on-line article in : http://www.outsights.com/systems/kmgmt/kmgmt.htm

Black B. (2000), *BSE Manual – Guidelines for BSE Rule Development,* Technical Report for the UMIST team of Concerto.

Black B., McNaught J., Rinaldi F., Ferraro M., Gilardoni L., Mazza S., Zarri G.P., Brasher A., Persidis A. (1999), *Detailed specification of the text extraction and concept recognition components of the Concerto architecture*, D6 Deliverable, Doc. Version 1.2 .

Black B., Rinaldi F. (2000), *FACILE Pre-processor V3.0 – A User Guide*, Technical Report for Concerto and Facile Partners.

Black B., Rinaldi F., Mowatt D. (1998), *FACILE : Description of the NE system used for MUC-7*, Proceedings of the 7th MUC.

Brachman R., Anand T. (1996), *The process of Knowledge Discovery in databases,* in : *Advances in Knowledge Discovery and Data Mining,* Fayyad U., Piatesky-Shapiro G., Smyth P., Uthurusamy R. (eds), MIT Press.

Cashmore C., Lyall R. (1991), *Business information systems and strategies*, Prentice Hall International ltd.

Chinchor N., Brown E., Ferro L., Robinson P. (1999), *Named Entity Recognition Task Definition*, version1.4, The MITRE corporation and SAIC.

Chios K., Pedrycz W., Swiniarski R. (1998), *Data Mining : Methods for Knowledge Discovery*, Kluwer Academic Publishers.

Collins *Cobuild English Grammar* (1990), Harper Collins Publishers.

Cowie J., Lehnert W.(1996), *Information Extraction*, in Special NLP issue of commun ACM, Y. Wilks (ed).

Cowie J., Wilks Y. (2000), *Information Extraction*, in: *Handbook of Natural Language Processing*, R. Dale, H. Moisl, H. Somers (ed), Marcel Dekker inc.

Cunningham H. (1999), *Information Extraction – A user guide*, on-line article in: http://www.dcs.shef.ac.uk/~hamish

Davenport Th. (1998), *Some principles of Knowledge Management*, an on-line article in : http://www.bus.utexas.edu/kman/kmprin.htm

EAGLES (1996), Expert Advisory Group on Language Engineering Standards, *EAGLES Evaluation of Natural Language Processing systems - Final Report,* Technical report in: http://www.issco.unige.ch/projects/ewg96/ewg96.html

Fayyad U., Shapiro G., Smyth P. (1996), *From Data Mining to Knowledge Discovery. An Overview,* , in *Advances in Knowledge Discovery and Data Mining,* Fayyad U., Piatesky-Shapiro G., Smyth P., Uthurusamy R. (eds), MIT Press.

Findler N.V (1991), *An Artificial Intelligence technique for information and fact retrieval – an application in Medical Knowledge Process*, The MIT Press.

Forcada – Sanz V.M (1996), *Extraction of proper names from unrestricted text: an implementation*, MSc dissertation, UMIST, department of Language Engineering, supervised by B. Black.

Gaizauskas R., Robertson A. (1997), *Coupling Information Retrieval and Information Extraction: A new text technology for gathering information from the Web*, in Proceedings of RIAO 97 : Computer assisted information searching on the Internet.

Gerhart S. (1997), *Browsing in context*, an on-line article in : http://www.twurl.com/chi-browsing-in-context.htm

Glossary for Information Retrieval (1997), on-line article in: http://www.cs.jhn.edu/~weiss/glossary.html

Godbout A. (1996), *Information versus Knowledge*, an on-line article : http://www.km-forum.org/ajg-002.htm

Harris D. (1999), *Creating a knowledge centric Information Technology Enviroment*, an on-line article : http://www.techined.com

Hearst M. (1999), Text Data Mining, Issues, Techniques and the relation to Information Access, an on-line article in : http://www.sims.berkeley.edu/~hearst

Hearst M. (1999), *Untagling Text Data Mining*, in Proceedings of ACL 99.

Klosgen W., Zytkow J. (1996), *Knowledge Discovery in Databases Terminology*, in *Advances in Knowledge Discovery and Data Mining,* Fayyad U., Piatesky-Shapiro G., Smyth P., Uthurusamy R. (eds), MIT Press.

López – Trigueros (1998), Named Entity Recognition in the field of Crop Science, MSc dissertation, UMIST, department of Language Engineering, supervised by J. McNaught.

Macintonsh A. (1999), *Knowledge Management*, an on-line article : http://www.aiai.ed.ac.uk/~alm/kamlnks.html#ext

Makoul J. (1998), *Performance Measures for Information Extraction*, DARPA Broadcast News Workshop 1998.

Marsh E. (1998), *MUC-7 and MET-2 – Call for participation*, http://muc.saic.com

Mauldin M. (1991), *Conceptual information retrieval – a case study in adaptive parsing*, Kluwer Academic Publishers.

McNaught J. (2000), *Rules and Templates for Concerto,* Rule file for the UMIST team of Concerto.

McNaught J, Black W, Rinaldi F, Bertino E, Brasher A, Deavin D, Catania B, Silvestri D, Armani B, Persidis A, Semerano G, Esposito F, Candela V, Zarri G-P, Giraldoni L (2000), *Integrated Document and Knowledge Management for the Knowledge-based Enterprise,* Proceedings of the 3rd International conference on the practical application of Knowledge Management, The practical application company.

Moore J. (1999), *KM: Applications and Implications*, MSc Dissertation, UMIST, Department of Computation, supervised by I. Petrunias.

Newman B. (1996), *Knowledge Management versus Knowledge Engineering*, an on-line article : http://revolution.3-cities.com/~bonewman/kmvske.htm

NIST (1999), National institute of Standards and Technology, *The 1999 Information Extraction – Entity Recognition Evaluation Plan,* an on-line article in : http://www.itl.nist.gov/iaui/894.01

Persidis A. (2000), *List and analysis of examples of concepts in news items*, Technical Report, version 1.0, Doc. Ref. BVA-5_5-2.

PRNewswire (2000), *Today's* (9th of August) *News in Pharmaceuticals / Biotechnology and Health*, on-line news stories: http://www.prnewswire.com

SAIC (1998), *Definitions of terms used in Information Extraction*, in http://www.muc.saic.com/info/definitions.html

Scarbrough H., Swan J. (ed) (1999), *Case studies in Knowledge Management*, Institute of Personnel and Development, GB the Cromwell Press.

Smith P. (1988), *An introduction to LISP,* Chatwell – Bratt Ltd.

Thompson P. (2000), *Rules for Concerto*, Rule file for the UMIST team of Concerto.

Thuraisingham B. (1999), *Data Mining – Technologies, Techniques, Tools and Trends*, PHD, CRC Press.

Wakao T., Gaizauskas R., Wilks Y. (1996), *Evaluation of an algorithm for the recognition and classification of proper names*, in Proceedings of the 16th International Conference on CL (COLLING 96).

Wilks Y. (1997), *Information Extraction as core language technology*, in *Information Extraction*, M-T. Pazienza (ed.), Springer Verlag.

Wilks Y., Gaizauskas R. (1999), *LaSIE jumps the GATE*, in *Natural Language Information Retrieval*, T. Strzalkowski (ed), Kluwer Academic Publishers.

Wilks Y., Catizone R. (1999), *Can we make Information Extraction more adaptive?,* in Proceedings of the SCIE99 Workshop, M. Pazienza (ed.), Springer Verlag.

Zack M. (1999), *Managing codified knowledge*, in Sloan Management Review.