Institut für Informatik der Universität Zürich

Attempto Controlled English (ACE)

Language Manual

Version 3.0

Norbert E. Fuchs, Uta Schwertel, Rolf Schwitter

Nr. 99.03 August 1999

Contents

1 What is Attempto Controlled English (ACE)?

Attempto Controlled English (ACE) is a language specifically designed to write specifications. ACE is a controlled natural language (cf. http://www-uilots.let.ruu.nl/~Controlled-languages), i.e. a subset of English with a domain specific vocabulary and a restricted grammar in the form of a small set of construction and interpretation principles. This means that all ACE sentences are correct English, but that not all English sentences are allowed in ACE. The restriction of full natural language to a controlled subset is essential for ACE to be suitable for specification purposes. The main goals of this restriction are

- to support the writing of precise specifications,
- to reduce ambiguity and vagueness inherent in *full* natural language,
- to encourage domain specialists to deliberately choose a clear and unambiguous writing style so that readers of a specification understand it in the same way as the writer,
- to make specifications computer processable,
- to render specifications unambiguously translatable into formal specification languages, particularly into first-order logic,
- to make specifications executable.

In brief, ACE allows domain specialists to express specifications in familiar natural language and to combine this with the rigor of formal specification languages.

ACE has been used to specify a simple automatic teller machine, Kemmerer's library data base problem, Schubert's steamroller, and a number of smaller problems. Recently, ACE has also been used as the input language of a theorem prover, and first attempts have been made to interface it to a program synthesiser. Clearly, ACE can be adapted and extended for other purposes requiring precise input, e.g. writing technical documentation or updating databases.

The use of ACE presupposes only basic knowledge of English grammar. Note however, that because ACE is a *controlled* natural language not all standard grammatical notions are directly applicable or suitable for the description of the ACE grammar. Some grammatical notions have a restricted meaning in ACE, while other notions are especially coined for an effective description of the language. The divergences are kept to a minimum so that ACE can be easily learned extending basic grammatical knowledge. For linguistic concepts not known to the reader we refer to the "Lexicon of Linguistics" available over the internet from <http://www-uilots.let.ruu.nl/~Hans.Leidekker/lexicon/ll.html>.

2 ACE in a Nutshell

The following is intended as a quick overview of the main features of the language ACE. The complete language is described in sections 3 to 5. For the linguistic foundation of ACE check R. Schwitter, Kontrolliertes Englisch für Anforderungspezifikationen, PhD thesis, Department of Computer Science, University of Zurich, 1998 (http://www.ifi.unizh.ch/staff/schwitt/).

Vocabulary

The vocabulary of ACE (→ 4) comprises

- user-defined, domain-specific content words (nouns, verbs, adjectives, adverbs) (→ 4.2),
- predefined function words (e.g. determiners, conjunctions, prepositions) (→ 4.3).

Users can define content words with the help of a lexical editor that presupposes only basic grammatical knowledge. Alternatively, users can import existing lexica.

Grammar

The grammar of ACE (→ 3) defines and constrains the form and the meaning of ACE sentences and texts.

ACE Specifications

An ACE specification is a sequence of sentences. There are

- simple sentences, and
- composite sentences.

Furthermore, there are query sentences that allow users to interrogate the contents of specifications.

Simple Sentences

A simple sentence (→ 3.2) describes a situation that can be an event, e.g.

A customer inserts a card.

or a state, e.g.

A card is valid.

All elements of a simple sentence can be elaborated upon to describe the situation in more detail. To further specify the nouns *customer* and *card*, we could add ***adjectives***

A new customer inserts a valid card.

possessive nouns and ***of-prepositional phrases***

John's customer inserts a card of Mary.

or ***proper nouns*** and ***dynamic names*** as ***appositions***

The customer Mr Miller inserts a card A.

Other modifications of nouns are possible through ***relative sentences***

A customer who is new inserts a card that he owns.

which are described below since they make a sentence composite.

We can also detail the *insert* event, e.g. by adding an ***adverb***

A customer inserts a card manually.
— or equivalently
A customer manually inserts a card.

or by adding ***prepositional phrases***, e.g.

A customer inserts a card into a slot.
A customer inserts a card in the morning.

We can combine enhancements to arrive at

John's customer who is new inserts a valid card of Mary manually into a slot A.

This sentence has the following general structure that characterises all simple ACE sentences:

subject	+ verb	+ complements	+ adjuncts
John's customer who is new	*inserts*	*a valid card of Mary*	*manually into a slot A.*

Complements are necessary for transitive (*insert*) and ditransitive verbs (*give to*), whereas adjuncts are optional.

Composite Sentences

Composite sentences (→ 3.3) are recursively built from simpler sentences through coordination, subordination, quantification, and negation.

Coordination by *and* and *or* is possible between sentences and between phrases of the same syntactic type

A customer inserts a card and the machine checks the code.
A customer inserts a card or enters a code.
A new and trusted customer enters a card and a code in the morning or in the evening.

Note that coordination by *or* allows users to express non-determinism.

The *and*-coordination of prepositional phrases can be simplified by leaving out the *and*.

A customer inserts a card in the morning into the slot.

Coordination of verb phrases can be simplified by omitting the repeated (negated) verb. Instead of

A customer inserts a VisaCard and inserts a MasterCard.

we can write

A customer inserts a VisaCard and a MasterCard.

The sentence

A card is valid or is invalid.

can be simplified to

A card is valid or invalid.

The ACE system replaces the elided verb and thus reconstructs the conjunction or disjunction. The replacements are shown in a paraphrase.

A customer inserts a VisaCard and [inserts] a MasterCard.
A card is valid or [is] invalid.

If verbs are negated as in

A customer does not insert a card and a code.

the negated verb will be inserted to yield the paraphrase

A customer does not insert a card and [does not insert] a code.

Coordination by *and* and *or* is governed by the standard binding order of logic, i.e. *and* binds stronger than *or*. The sentence

A customer inserts a VisaCard or a MasterCard and a code.

means that the customer inserts a VisaCard, or the customer inserts a MasterCard and a code. Commas can be used to override the standard binding order. Thus the sentence

A customer inserts a VisaCard or a MasterCard, and a code.

means that the customer inserts a VisaCard and a code or a MasterCard and a code.

There are two forms of ***subordination***: relative sentences and *if-then* sentences.

Relative sentences starting with *who, which, that* allow to add detail to nouns, e.g.

A customer who is new inserts a card that he owns.

With the help of ***if-then*** sentences we can specify conditional or hypothetical situations, e.g.

If a card is valid then a customer inserts it.

The specification says that the customer inserts a card only if it is valid. Note the anaphoric reference via the pronoun *it* in the *then*-part to the noun phrase *a card* in the *if*-part.

Quantification allows us to speak about all objects of a certain class, or to denote explicitly the existence of at least one object of this class. To express that all involved customers insert cards we can write

Every customer inserts a card.

This sentence means that each customer inserts a card that may, or may not, be the same as the one inserted by another customer. To specify that all customers insert the same card – however unrealistic that situation seems – we can write

There is a card that every customer inserts.

ACE does not know the passive voice. To state that every card is inserted by a customer we write

For every card there is a customer who inserts it.

The textual occurrence of a quantifier opens its scope that extends to the end of the sentence, or — in coordinations — to the end of the respective coordinated sentence.

Negation allows us to express that something is not the case, e.g.

A customer does not insert a card.
A card is not valid.

To negate something for all objects of a certain class one uses *no*

No customer inserts a card.

or, equivalently, *there is no*

There is no customer who inserts a card.

Query Sentences

Query sentences (→ 3.4) permit us to interrogate the contents of a specification. Query sentences come as *yes/no*-queries and as *wh*-queries.

Yes/no-queries establish the existence or non-existence of a specified situation. If we specified

A customer inserts a card.

then we can ask

Does a customer insert a card?

to get a positive answer.

With the help of ***wh-queries***, i.e. queries with query words, we can interrogate a specification

for details of the specified situation. If we specified

A new customer inserts a valid card manually into the slot at 9 o'clock.

we can ask for each element of the sentence, e.g.

Who inserts a card?
Which customer inserts a card?
What does the customer insert?
How does the customer insert a card?
When does the customer insert a valid card?

Note, however, that we cannot ask for the verb itself.

Constraining Ambiguity

To constrain the ambiguity of full natural language ACE employs three simple means

- some ambiguous constructs are not part of the language; unambiguous alternatives are available in their place,
- all remaining ambiguous constructs are interpreted deterministically on the basis of a small number of interpretation principles; the interpretations are reflected in a paraphrase,
- users can either accept the assigned interpretation, or they must rephrase the input to obtain another one.

Avoidance of Ambiguity

Here is an example how ACE replaces ambiguous constructs by ***unambiguous constructs***.

In full natural language relative sentences combined with coordinations can introduce ambiguity, e.g.

A customer inserts a card that is valid and opens an account.

In ACE the sentence has the unequivocal meaning that the customer opens an account. This is reflected in the paraphrase by curly brackets

A customer inserts {a card that is valid} and opens an account.

To express the alternative — though not very realistic — meaning that the card opens an account the relative pronoun *that* must be repeated, thus yielding a coordination of relative sentences.

A customer inserts a card that is valid and that opens an account.

with the paraphrase

A customer inserts {a card that is valid and that opens an account}.

Interpretation Principles

However, not all ambiguities can be safely removed from ACE. To deterministically interpret otherwise syntactically correct ACE sentences we use about a dozen ***interpretation principles*** (→ 5).

Here are some examples. If we write

The customer inserts a card with a code.

we get the paraphrase

The customer {inserts a card with a code}.

that reflects ACE's interpretation principle that a prepositional phrase always modifies the verb.

However, this is probably not what we meant to say. To express that the code is associated with the card we can employ the interpretation principle that a relative sentence always modifies the immediately preceding noun phrase, and rephrase the input as

The customer inserts a card that carries a code.

yielding the paraphrase

The customer inserts {a card that carries a code}.

or — to specify that the customer inserts a card and then a code — as

The customer inserts a card and a code.

Adverbs can precede or follow the verb. To disambiguate the sentence

The customer who inserts a card manually enters a code.

we employ the interpretation principle that the postverbal position has priority.

The customer who {inserts a card manually} enters a code.

Anaphoric References

Usually specifications consist of more than one sentence (→ 3.5), e.g.

A customer enters a card and a code.
If a code is valid then SimpleMat accepts a card.
If a code is not valid then SimpleMat rejects a card.

To express in the sentences above that the occurrences of *card* and *code* should mean the same card and the same code, ACE provides ***anaphoric references*** via the definite article, i.e.

A customer enters a card and a code.
If the code is valid then SimpleMat accepts the card.
If the code is not valid then SimpleMat rejects the card.

or via personal pronouns, i.e.

A customer enters a card and a code.
If it is valid then SimpleMat accepts the card.
If it is not valid then SimpleMat rejects the card.

Note that proper nouns like SimpleMat always refer to the same object.

During the processing of ACE specifications the ACE system replaces any anaphoric reference by the most recent accessible noun phrase that agrees in gender and number, and displays the replacements in a paraphrase

A customer enters a card and a code.
If [the code] is valid then SimpleMat accepts [the card].
If [the code] is not valid then SimpleMat rejects [the card].

The search for antecedents of anaphora is governed by accessibility relations that are discussed in detail in section 3.5.2.5.

Meaning of ACE Specifications

We can associate meaning to ACE specifications in a way that is completely analogous to the model-theoretic interpretation of first-order logic formulas.

We can give each ACE sentence a ***truth value***, i.e. we call a sentence true if we can interpret it as describing an actual situation in the specified domain, or we label the sentence false if we cannot establish such an interpretation.

We can interpret simple sentences as descriptions of distinct events or states. The sentence

A customer inserts a card.

is true if the word *customer* can be interpreted as a relation `customer` that holds for an object A of the specified domain, the word *card* as a relation `card` that holds for an object B of the specified domain, and the word *insert* as a relation `inserting event` that holds between A and B. Otherwise, the sentence is false.

Once we have assigned meaning to simple sentences we can give meaning to composite sentences built from simpler sentences. Again, this is completely analogous to classical model theory. A conjunction of sentences is true if all conjoined sentences are true. A disjunction of sentences is true if at least one of the sentences of the disjunction is true. A sentence with a negation is true if the sentence without the negation is false. An *if-then* sentence is only false if the *if*-part is true and the *then*-part is false; otherwise, the *if-then* sentence is true. A sentence with a universal quantifier is true if the sentence without the universal quantifier is true for all objects denoted by the quantified phrase.

Time, Events and States

Each verb in an ACE sentence denotes an event or a state. Events occur instantaneously, while states — i.e. relations that hold or do not hold — last over a time period until they are explicitly terminated.

Each occurrence of a verb is implicitly associated with a ***time***. If the verb denotes an event this is the time point at which the event occurs; if the verb denotes a state this is the time point from which on the state holds.

Per default the ***textual order*** establishes the ***temporal order*** of the associated events or states. E.g. in

A customer inserts a card.
The customer enters a code.
The code is correct.

the *insert* event is followed by the *enter* event which is followed by the state of being *valid*. Users can override the default temporal order by explicitly specifying times through prepositional phrase like *at 9 o'clock*, or by adding prepositional phrases like *at the same time*. A future version of ACE will allow users to combine sentences by temporal prepositions like *before*, *after*, and *while*.

Domain Knowledge

The Attempto system is not associated with any specific application domain, nor with any particular software engineering method. By itself it does not contain any knowledge or ontology of the specified domain, of software engineering methods, or of the world in general. Thus users must explicitly define domain knowledge through ACE sentences like

A card is valid.

In this sentence the words *card* and *valid* are processed by the Attempto system as uninterpreted syntactic elements, i.e. any interpretation of these words is solely performed by the human writer or reader. Whatever understanding we may have of the concepts card and valid is not part of the ACE specification, unless we decide to add information explicitly, e.g. by sentences like

Every card that carries a code is valid.

As a consequence of its syntactic approach, the Attempto system can only detect explicit logical contradictions. For example, if the specification states in one place that a card is *valid*, and in another place that the same card is *not valid*, the contradiction can be detected. However, if

in one place the specification labels a card as *valid*, and in another place the same card as *invalid*, the contradiction can only be detected, if we explicitly define that *valid* and *invalid* exclude each other, e.g.

Every card that is not valid is invalid.

In summary, the only source of information about an ACE specification is the specification text itself.

3 Grammar of ACE

This section introduces the construction and the interpretation principles of ACE. The construction principles explain the admissible structures of ACE sentences and texts. The interpretation principles control the meaning of ACE texts.

3.1 Some Terminology

Words and Phrases

Sentences have an internal structure, i.e. they consist of smaller units. Take the simple ACE sentence

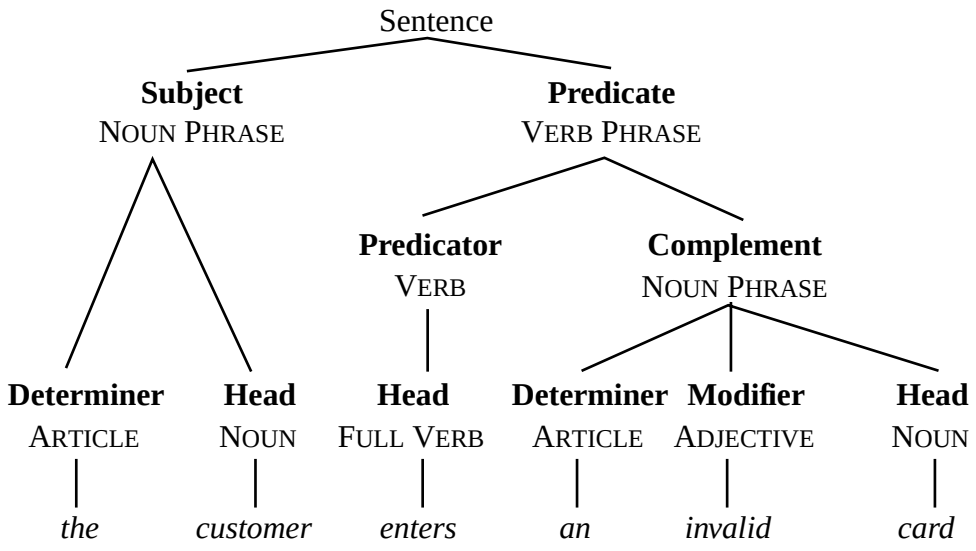
The customer enters an invalid card.

The components of the sentence are (i) individual ***words*** (e.g. *the*, *enters*) and (ii) ***phrases***. Phrases are relatively independent units of closely connected words (e.g. *the customer*, *an invalid card*). Phrases are hierarchically structured. The phrase *enters an invalid card* contains the smaller phrase *an invalid card*, which contains the phrase *invalid*.

Function and Form

Phrases are distinguished according to their syntactic ***function*** and ***form***. The function of the phrase *the customer* is that it is the 'subject' of the sentence, its syntactic form is that of a 'noun phrase'. The term 'noun phrase' derives from the main part of the phrase, the noun *customer*. The main part of a phrase is called the ***head*** of a phrase. The head of a noun phrase is a noun, the head of a verb phrase is a verb etc. Our sentence above contains two noun phrases (*the customer*, *an invalid card*), a verb phrase (*enters an invalid card*) and an adjective phrase (*invalid*). Note that the same type of phrase can have different syntactic functions: the noun phrase *the customer* functions as a ***subject*** of the sentence, the noun phrase *an invalid card* functions as a ***complement*** (also called object) of the verb.

The following tree summarises the hierarchical structure of a simple ACE sentence. The parts are labelled according to their function (indicated by bold font, e.g. **Subject**) and their form (indicated by small caps, e.g. NOUN PHRASE). All parts are explained in the following sections.



3.2 Simple Sentences

3.2.1 Basic Form

Path	Grammar of ACE → Simple Sentences → Basic Form
Aim	How to construct basic simple sentences in ACE.
Examples	*The employee works.* *John enters a card.* *The customer gives the letter to the agent.* *A card is invalid.*
Terminology	noun phrase, subject, predicate, predicator, complement, intransitive, transitive, ditransitive
Principles	—

Introductory Terminology

Noun Phrase

Noun phrases are proper nouns (→ 4.2.1.2) like *John*, *America*, or sequences consisting of a determiner (→ 4.3.2 and 3.3.5.1) and a common noun (→ 4.2.1.1) like *the man*, *a customer*, *every card*.

Construction

All sentences of ACE are built upon basic simple sentences which have the general structure:

subject	predicate	
	predicator	complement(s)
John	*enters*	*a card.*

In ACE the ***subject*** is always realised as a noun phrase, e.g. *the employee*, *John*, *a card*. The internal structure of the ***predicate*** depends on the verb. In ACE the ***predicator*** of a simple sentence is either a ***full verb*** (→ 4.2.2), e.g. *insert*, *enter*, *sleep*, or the ***copula verb*** *be* (→ 4.2.2, page 51).

Full verbs in ACE can have zero (intransitive verbs), one (transitive verbs) or two complements (ditransitive verbs). The ***complements*** of the verb are the parts that are necessary — apart from the subject — to form a complete sentence. For example, the verb *insert* requires one complement: *The customer inserts a VisaCard*, NOT: *The customer inserts*. The verb *work*, however, excludes a complement: *The customer works*, NOT: *The customer works a VisaCard*. Omitting a complement will in most cases result in an ungrammatical sentence. Less frequently, the sentence stays grammatical but the meaning of the verb changes (*The customer returns the book* vs. *The customer returns*). ACE only allows for noun phrases and for prepositional phrases (→ 4.3.1), e.g. *to the customer, on John*, as complements of full verbs.

The copula verb *be* has always one complement, which can be an adjective phrase (→ 4.2.3), e.g. *valid, identical to the manager,* an indefinite noun phrase (→ 4.3.2), e.g. *a customer*, or a prepositional phrase (→ 4.3.1), e.g. *in the bank*. ACE does not allow for other types of complements for the verb *be*.

Table 1 summarises the structure of basic simple sentences in ACE.

Table 1: Structure of basic simple sentences in ACE

Subject	Predicate		
	Predicator	Complement 1	Complement 2
The employee	*works*		
John	*enters*	*a card*	
The customer	*gives*	*the letter*	*to the agent*
The card	*is*	*invalid*	
Mary	*is*	*identical*	*to the customer*
The card	*is*	*in the slot*	
SimpleMat	*is*	*an automatic teller machine*	

Notes

In contrast to full natural language ACE imposes the following restrictions concerning admissible elements of a sentence:

- verbs are only used
  - in the active voice (no passive constructions: *The card is inserted by the customer.*)
  - in the indicative mood (no imperatives: *Enter the card.*, no subjunctives: *If I were you.*)
  - in the simple present tense (no past or future tense, no continuous forms)
  - in the third person singular form (NOT: *I enter the card.*)
- no plural noun phrases and plural verbs (NOT: *The customers sleep.*, NOT: *John enters five cards.*)
- no modal verbs (NOT: *may, can, must*), no intensional verbs (NOT: *hope, know, believe*)
- common nouns (*man, customer*) are always used with a determiner (article or quantifier)

3.2.2 Modifying Nouns and Noun Phrases

Path	Grammar of ACE → Simple Sentences → Modifying Nouns and Noun Phrases
Aim	How to add optional elements to noun phrases to express additional information.
Examples	*The old customer inserts the valid credit card.* *A customer who is old inserts a credit card that is valid.* *The customer of the bank enters a valid code.* *John's EuroCard is invalid.* *The customer Mr Miller enters a card.*
Terminology	modifier, attributive vs. predicative position, possessives, apposition
Principles	Principle of Right Association (→ page 23)

Introductory Terminology

Modifier

A modifier is an optional element of a sentence which gives additional information about the word or phrase which it modifies. ACE has noun modifiers (→ 3.2.2), e.g. *the valid card*, and verb phrase modifiers (→ 3.2.3), e.g. *The customer waits in the lounge*.

3.2.2.1 Adjectives

Construction

An adjective (→ 4.2.3) can be added in front of a common noun to further detail the noun.

The old customer inserts the valid credit card.

Here the adjective is used in ***attributive position*** (i.e. in front of the noun). Adjectives in attributive position function as a modifier of the noun.

Notes

Coordination (→ 3.3.2) allows users to put more than one adjective in front of a noun:

The customer enters the correct and new code.
NOT: *The customer enters the correct new code.*

3.2.2.2 Relative Sentences

Construction

Another possibility to detail a noun phrase are relative sentences.

A customer who is old inserts a credit card that is valid.

Relative sentences always relate to the rightmost noun preceding the relative pronoun.

See also

Sentences containing relative sentences are classified as composite sentences in ACE, therefore they will be dealt with in section 3.3.3.

3.2.2.3 '*Of*'-Prepositional Phrases

Construction

A third possibility to modify noun phrases are prepositional phrases beginning with the preposition *of*. *Of*-prepositional phrases are put directly after the noun phrase which they modify.

The customer of the bank enters a valid code.

In this sentence the prepositional phrase *of the bank* gives more information about the customer. It puts the customer into a relation to the bank.

Notes

In ACE *of* is the only preposition that can be used in prepositional phrases that function as modifiers of nouns.

See also

Section 4.3.1 gives a more detailed discussion of prepositional phrases in general and prepositional phrases with *of* in particular.

3.2.2.4 Possessive Nouns

Construction

A noun can be modified by adding a possessive noun (also called a genitive) in front of it, e.g. *John's EuroCard*, *the customer's card*, *James' card*. A possessive noun consists of a name or a noun to which an *'s* is added. With possessive nouns you can say who something belongs to or who something is associated with.

Notes

In ACE *John's EuroCard* is interpreted identical to *the EuroCard of John*. There is a structural difference however when the constructions are combined with relative sentences. Relative sentences always relate to the rightmost noun that precedes the relative pronoun (→ page 23, "Principle of Right Association"). Thus the following sentences are interpreted differently in ACE (though the interpretation of the second sentence is very unnatural):

SimpleMat checks John's card that is old.	*that* relates to *card*, i.e. the card is old
SimpleMat checks the card of John that is old.	*that* relates to *John*, i.e. John is old

3.2.2.5 Appositions

Construction

A final possibility to refine noun phrases is to use a construction called ***apposition***: two noun phrases are said to be in apposition when they occur next to each other and refer to the same person or thing. In ACE two types of noun phrases can be used in appositive position.

- proper nouns that are in the lexicon (→ 4.2.1.2)
  the insurance company Watertight
  the customer Mr Miller
  the message 'Enter your credit card.'
- dynamic names which are not in the lexicon (→ 4.2.1.3)
  a region ABC

Notes

In ACE appositions are never separated by a comma from the preceding noun.

3.2.3 Modifying Verb Phrases

Path	Grammar of ACE → Simple Sentences → Modifying Verb Phrases
Aim	How to add optional elements to verb phrases to express additional information
Examples	*The customer enters a card manually.* *John inserts the card into the slot.*
Terminology	adjunct, attachment ambiguity
Principles	• Adverbial Modification: Priority of Postverbal Position • Minimal Attachment of Prepositional Phrases

Introductory Terminology

Adjunct

An adjunct is a modifier (→ 3.2.2, page 12) of a verb phrase. It gives additional information about the event or state (→ 4.2.2, page 51) described by the verb phrase. Adjuncts are optional elements of a sentence.

3.2.3.1 Adverbs

Construction

Adverbs, e.g. *quickly, manually,* can be added as adjuncts to further detail the verb phrase:

The customer enters a card manually.

Adverbs can be used in three positions in ACE sentences:

- after the verb if there is no complement
  John drives carefully.
- after complements
  John enters the card quickly.
  John gives Mary a book today.
- immediately before the verb
  John correctly uses the credit card.

ACE does not allow sentence initial position of adverbs (NOT: *Carefully, the customer enters the credit card.*).

Principle — Priority of Postverbal Position

When adverbs occur together with relative sentences (→ 3.3.3) ambiguities can occur. In the sentence

The customer who inserts a card slowly enters a code.

it is not clear whether the adverb *slowly* modifies the verb *inserts* or the verb *enters*. To avoid this kind of ambiguity ACE uses an interpretation principle

Adverbial Modification: Priority of Postverbal Position

In case of a conflict as to whether an adverb modifies the preceding or the following verb ACE uses the default that the adverb refers to the preceding verb.

The sentence above is therefore interpreted with *slowly* modifying *inserts*:

The customer who {inserts a card slowly} enters a code.

Curly brackets in the paraphrase indicate the intended structuring.

To express that *slowly* modifies *enter* you have to restructure the sentence and put the adverb at the end:

The customer who inserts a card {enters a code slowly}.

Notes

Note that adverbs in ACE can only modify verb phrases but do not modify other constructions like sentences (NOT: *Fortunately, the code is valid*), adjectives (NOT: *John owns an extremely old book*) or other adverbs (NOT: *John drives extremely carefully*).

See also

Section 4.2.4 lists different types of adverbs with which you can add different types of additional information (e.g. information about time, location etc.).

3.2.3.2 Prepositional Phrases

Construction

Prepositional phrases (→ 4.3.1), e.g. *in the bank, to the customer,* can be added as adjuncts to further detail a verb phrase.

John inserts the card into the slot.

In ACE prepositional adjuncts are always placed after the complements; more than one prepositional phrase is possible:

The train arrives in the station at 5 o'clock.

Prepositional phrases after complements can also be coordinated (→ 3.3.2) leading to a composite sentence:

The train arrives on track 11 or on track 12.

If the coordinator is the conjunction *and* as in

The train arrives at 5 o'clock and on track 12.

the interpretation is equivalent to the sentence without *and*.

The choice of the preposition can vary depending on the type of information the user wants to add. For example, in the sentence

The customer works in a library.

the prepositional phrase *in the library* tells us more about the location where the customer works (→ page 55 for a complete list of modification types).

Principle — Minimal Attachment

In full English it is not always clear which part of the sentence prepositional phrases modify. The example

The bank employee observes the customer with a video camera.

has two interpretations in full English. The prepositional phrase *with a video camera* can function as a verb phrase modifier or as a noun modifier. If it functions as a verb phrase modifier it gives more information about the observing event. It expresses that the instrument of observing the customer is a video camera. If it functions as a noun modifier it gives more information about the noun *customer*, viz. that the customer is carrying a video camera. Ambiguities of this kind are called ***attachment ambiguities*** because it is not clear which element of the sentence the prepositional phrase is attached to.

To avoid these attachment ambiguities ACE implements a restriction which states that prepositional phrases (except *of*-prepositional phrases) cannot be used as modifiers of noun phrases. The following principle covers this restriction.

> **Minimal Attachment of Prepositional Phrases**
>
> In ACE all prepositional phrases (except *of*-phrases) that are used as modifiers relate to the closest preceding verb phrase not to noun phrases.

Therefore ACE interprets the prepositional phrase *with a video camera* in the sentence above unambiguously as modifying the verb phrase. In the ACE paraphrase this interpretation is optionally signalled to the user by curly brackets around the elements belonging together:

The bank employee {observes the customer with a video camera}.

Notes

As a consequence of the minimal attachment principle the following sentence is not possible in ACE: *The man in the car works.* This kind of noun phrase modification has to be expressed differently in ACE, e.g. with a relative sentence: *The man who is in the car works.* See section 6 for more examples.

See also

The special role of *of*-prepositional phrases, which modify preceding nouns, is discussed in section 3.2.2.3 and section 4.3.1.

3.2.4 Summary: Refining Simple Sentences

Table 2 summarises the possibilities to extend basic simple sentences of ACE. The added elements are underlined. Note that the table includes two sentences containing relative sentences. These sentences do not count as simple sentences, however, they are listed for a better overview. Composite sentences will be dealt with in full detail in section 3.3.

Table 2: Modifying basic simple sentences in ACE

Subject	**Predicate**						
	Adjunct	**Predicator**	**Complement(s)**		**Adjunct(s)**		
A customer		*inserts*	*a card*		*into the slot.*		
A customer		*enters*	*a code*		*manually.*		
An employee		*sends*	*the bill*	*to the customer*	*at the end of a year.*		

Table 2: Modifying basic simple sentences in ACE

SimpleMat		*rejects*	*a card that is invalid.*				
The customer	*correctly*	*enters*	*the code.*				
The customer		*enters*	*the code*		*correctly.*		
The blue card		*has*	*a valid number.*				
John		*is*	*a manager of the bank.*				
The customer's card		*is*	*invalid.*				
SimpleMat		*retains*	*John's card that is invalid.*				
John		*presses*	*the key 'A'.*				
John		*presses*	*the button 'A' of the machine.*				
The fast train		*arrives*			*punctually*	*on Monday*	*in the station.*

3.3 Composite Sentences

3.3.1 Introduction

In ACE users can build composite sentences from simple sentences, or composite phrases from simpler phrases, with the help of so-called constructors. ACE provides four different types of constructors (→ 4.3.4):

- coordinators (*and*, *or*)
- subordinators (relative pronouns: *who*, *which*, *that*, subordinating conjunction: *if...then*)
- quantifiers (*every, for every, each*, *no*, *there is a*, *there is no*)
- negators (*no*, *does not*, *is not*)

Coordination (→ 3.3.2) means combining elements that have the same syntactic form and function, while subordination means combining elements of unequal syntactic status: there is a superordinate element (the main clause) and a dependent subordinated element (the subclause). In ACE there are two types of subordination: relative sentences (→ 3.3.3) and *if-then* sentences (→ 3.3.4). Quantification (→ 3.3.5) and negation (→ 3.3.6) do not directly combine simpler phrases to composite phrases but lead to sentences having a more complex logical structure than simple sentences.

Every ACE sentence ends with a full stop, or — if the sentence is a query (→ 3.4) — with a question mark. Commas can be introduced to override the binding hierarchy in coordinated sentences (→ 3.3.3.2). Other uses of commas and other punctuation marks (semicolons, dashes, colons etc.) are not allowed.

3.3.2 Coordination

Path	Grammar of ACE → Composite Sentences → Coordination
Aim	How to use and interpret coordination.
Examples	*A customer enters a card and a code.* *The code is valid or invalid.*
Terminology	conjunction, disjunction, conjunct, disjunct, coordination, comma
Principles	• Binding Hierarchy • Distribution • Verb Phrase Coordination and Relative Sentences

3.3.2.1 Construction

Conjunction and Disjunction

ACE distinguishes two types of coordination:

- conjunction (*and*)
  John enters a VisaCard and Mary enters a MasterCard.
  John enters a card and a code.
- disjunction (*or*)
  John has a salary account or a savings account.

In ACE only sentences and phrases of equal syntactic form may be coordinated — with additional exceptions listed below. The elements of a conjunction are called conjuncts; the elements of a disjunction are called disjuncts.

- coordination of sentences
  John inserts the card and Mary enters the code.
  The card is damaged or the code is invalid.
- coordination of relative sentences
  The customer enters the card that is valid and that is correct.
- coordination of verb phrases
  The customer enters the card and inserts the code.
- coordination of noun phrases
  (if coordinated noun phrases do not function as subject of the sentence)
  The customer enters the user name and a password.
  SimpleMat displays the message 'Withdraw Money Without Receipt - Key 'A" and the message 'Terminate Transaction - Terminate'.
- coordination of prepositional phrases
  (except for prepositional phrases with *of*)
  The customer pays the bill with the credit card or with a traveller's cheque.
  The card is in the slot or on the table.
  John works with Word and with FrameMaker.
- coordination of adjectives
  The card is old and invalid.
  The old and invalid card is in the slot.
- coordination of adverbs

The customer inserts the card correctly and manually.

Note

The following phenomena cannot yet be dealt with in ACE. An analysis is postponed until there is a treatment of plurals in ACE. ACE does not allow for noun phrase coordination in subject position of a sentence (NOT: *John and Mary enter a credit card*). Furthermore, noun phrase coordination is not possible in combination with plural nouns: The sentence *The customer presses the keys 'On' or 'Off'* has to be reformulated by repeating the noun phrase *the key* in each conjunct: *The customer presses the key 'On' or the key 'Off'*. Similarly, the coordination of noun phrases that are complements of a preposition is not allowed: *The customer signs the contract with a black pen or a blue pen* has to be reformulated by repeating the preposition: *The customer signs the contract with a black pen or with a blue pen.* Finally, *of*-prepositional phrases cannot be coordinated NOT: *A book of John and of Mary is in the library.* In this sentence it is not clear whether there is a joint book of John and Mary that is in the library or whether there is a book of John that is in the library and there is a book of Mary that is also in the library.

3.3.2.2 Principles

Principle — Binding Hierarchy in Coordinations

Sentences can contain two or more different coordinators (e.g. *and* and *or*). In full natural language this can lead to ambiguities because it is not clear in which order the elements are joined. The sentence

The customer enters a VisaCard or a MasterCard and a personal code.

can be understood as

The customer enters a VisaCard or a MasterCard
and the customer enters a personal code.

or as

The customer enters a VisaCard
or the customer enters a MasterCard and a personal code.

To avoid this ambiguity ACE uses the binding hierarchy principle which controls which elements belong closer to each other than others. This binding hierarchy states that *and* binds more strongly than *or*. Furthermore, as we will see later, the negators (→ 3.3.6) bind most strongly, whereas the subordinator *if...then* (→ 3.3.3) binds least strongly.

> **Binding Hierarchy Principle**
> Negation > Conjunction > Disjunction > Implication
> (where ">" stand for "binds stronger than").

Note that this hierarchy corresponds to the standard binding order in logic.

As elements coordinated by *and* belong more closely to each other than elements coordinated by *or* ACE interprets the above sentence in the sense of the second meaning. This structuring is not immediately reflected in the paraphrase. It is just derivable from the application of the binding hierarchy.

It is possible to override the binding hierarchy by introducing ***commas*** to express the intended structuring. If the user writes the sentence with a comma

The customer enters a VisaCard or a MasterCard, and a personal code.

its intended interpretation corresponds to

The customer enters a VisaCard or a MasterCard
and the customer enters a personal code.

where *a VisaCard* and *a MasterCard* belong closer together.

Commas are similar to brackets known from formal languages; they are used to unambiguously indicate what belongs together.

There are other possibilities to circumvent the binding hierarchy. You can e.g. split our first sentence into two simpler sentences:

The customer enters a VisaCard or a MasterCard.
The customer enters a personal code.

Or you can distribute the conjuncts and thus make the structuring explicit:

The customer enters a VisaCard and a personal code or a MasterCard and a personal code.

In general, we recommend that users avoid multiple coordinations for the sake of simplicity and clarity. If multiple coordinations cannot be avoided we refer users to the examples and hints in section 6.

Principle — Distribution in Coordinations

ACE does currently not allow for noun phrase coordination in subject position of a sentence. For the treatment of noun phrase coordination in complement position of a sentence ACE provides a special principle — the distribution principle. This principle is also valid for other forms of complement coordination (e.g. if complements of the copula *be* are coordinated):

> **Distribution Principle**
>
> If the complement of a (negated) verb consists of a coordination of phrases then the (negated) verb is distributed to each phrase. The following elements can be distributed: *is*, *is not*, finite full verb, *does not* + full verb, *does not*. Non-finite elements (e.g. *not* alone, adjectives alone etc.) cannot be distributed.

The following examples and their paraphrases illustrate the distribution principle. In the paraphrases the distributed element is enclosed in square brackets. Note that the distribution principle is applied before the principle of minimal attachment.

- is
  - *The card is red and green.*
    The card is red and [is] green.
  - *The card is not red and not green.*
    The card is not red and [is] not green.
  - *The card is red and not green.*
    The card is red and [is] not green.
- is not
  - *The card is not red and green.*
    The card is not red and [is not] green.
- full verb
  - *John enters a card and a code and a password.*

John enters a card and [enters] a code and [enters] a password.
- *John enters a card and a code in the morning.*
  John enters a card and {[enters] a code in the morning}.

- does not + full verb
  - *The customer does not enter a card and a code.*
    The customer does not enter a card and [does not enter] a code.
  - *The man enters a card and does not enter a code and a message.*
    The man enters a card and does not enter a code and [does not enter] a message.
- does not
  - *The customer does not insert a card and enter a code.*
    The customer does not insert a card and [does not] enter a code.
  - HOWEVER: *The customer does not insert a card and enters a code.*

Note that the following elements are NOT distributed:

- verb + complement(s):
  - NOT: *The customer does not give a card to the man and to the woman.*
    USE INSTEAD: *The customer does not give a card to the man and does not give a card to the woman.*
  - NOT: *John gives a card to Mary and Yolande and a customer.*
    USE INSTEAD: *John gives a card to Mary and gives a card to Yolande and gives a card to a customer.*
  - NOT: *The book is checked out to the customer and to John.*
    USE INSTEAD: *The book is checked out to the customer and is checked out to John.*
- discontinuous elements
  - *The book is not checked out to the customer and not to John.*
    NOT: *The book is not checked out to the customer and is not checked out to John.*

The distribution principle will have consequences for the temporal interpretation of coordinated phrases. See section 3.5.3 for a discussion. The principle is also important for the combination of coordination with quantification as discussed in section 3.3.5.3, page 30.

Principle — Coordination and Relative Sentences

Sentences containing relative sentences (→ 3.3.3) and verb phrase coordination like

The customer enters a card that is valid and has a code.

can be ambiguous as to whether the verb phrase *has a code* belongs to the main clause or to the relative sentence. ACE employs the following principle to disambiguate these sentences:

Verb Phrase Coordination and Relative Sentences

Per default a coordinated verb phrase belongs to the main clause, not to the relative sentence.

The customer {enters a card that is valid} and {has a code}.

To express coordination within the relative sentence the relative pronoun has to be repeated.

The customer enters {a card that is valid and that has a code}.

The repeated relative pronouns refer to the same object as the initial relative pronoun.

3.3.2.3 Notes

Coordination and Adjuncts

The distribution principle applies before adjuncts (adverbs or prepositional phrases) are related to the closest preceding verb. The sentences

John enters a card and a code carefully.
John enters a card and a code in the morning.

are paraphrased and interpreted as

John enters a card and {[enters] a code carefully}.
John enters a card and {[enters] a code in the morning}.

To express that the adjunct belongs to both parts of the conjunct you have to repeat the verb and the adjunct for each conjunct.

John enters a card carefully and enters a code carefully.
John enters a card in the morning and enters a code in the morning.

Coordination and Temporal Order

For a general introduction to the principles of temporal order in an ACE specification see section 3.5, in particular section 3.5.3.

Interpretation of Conjunction

The conjunction of sentences or verb phrases is interpreted as 'first one and then the other', i.e. it implies a temporal succession. For example, the composite sentence

John enters a VisaCard and Mary enters a MasterCard.

means that first it happens that John enters a VisaCard and then it happens that Mary enters a MasterCard. The conjunction of verb phrases is treated analogous. The sentence

John inserts a card and enters a code.

means that John first inserts a card and then he enters a code.

Due to the distribution principle (→ page 20) the conjunction of noun phrases in complement position of a verb is equivalent to verb phrase conjunction. The sentence

John enters a card and a code.

means that John first enters a card and then he enters a code.

The default temporal succession can be overridden by special predefined adjuncts:

John enters a card and inserts a code at the same time.
John enters a card and a code in any temporal order.

Interpretation of Disjunction

The disjunction *or* is used in ACE with the meaning 'one or the other or both'. E.g. the sentence

John has a salary account or a savings account.

which is paraphrased as

John has a salary account or [has] a savings account.

is true if John has a salary account or if John has a savings account or if John has both, a salary account and a savings account. This corresponds to the logical concept of inclusive disjunction. There is no fixed temporal relationship between the two disjuncts.

In ACE sentences with disjunctions express non-determinism, i.e. it is not determined which of the two disjuncts is chosen. Attempto treats this non-determinism by selecting one of the alternatives at random. The user cannot control which alternative is chosen.

Note, that ACE does not implement a special constructor to express exclusive disjunction, i.e. a disjunction where one of the disjuncts has to be the case, but the other disjuncts are false. Users have to express this by explicitly listing the possibilities:

The book is available and is not checked out or the book is not available and is checked out.

3.3.3 Subordination — Relative Sentences

Path	Grammar of ACE → Composite Sentences → Subordination — Relative Sentences
Aim	How to construct and interpret relative sentences.
Examples	*The customer who enters a card types a code.*
Terminology	relative pronoun, non-determinism
Principles	Right Association

3.3.3.1 Construction

All relative sentences in ACE begin with one of the following relative pronouns:

- *who, which, that*

The relative pronoun is put immediately after the noun to which it refers. *Who* refers to people, *which* refers to things, *that* can refer to both people or things.

The customer who enters a card types a code.
The code which the customer enters is correct.
The customer enters a code that is valid.

Relative sentences can be put after nouns that are in subject position of the main clause, e.g.

The customer who enters a card types a code.

or after nouns that are in complement position, e.g.

The customer enters a card that has a valid code.
The customer gives a book to the employee who checks it.
The customer enters the VisaCard into a machine that displays the Visa sign.

Relative sentences in ACE function as modifiers of nouns (→ 3.2.2), i.e. they give further information about the person or object the noun refers to.

ACE does not allow to omit the relative pronoun (Not: *The card John enters is not valid.*), nor does ACE allow to use relative pronouns not listed above (e.g. *whose* is not allowed).

3.3.3.2 Principles

Principle — Right Association

To avoid ambiguities relative sentences in ACE are analysed according to the principle of right association.

Right Association

A relative pronoun not preceded by a coordinator (→ 3.3.2) relates to the rightmost noun that immediately precedes the relative pronoun.
A relative pronoun after a coordinator relates to the same noun as the closest preceding equal relative pronoun.

In the sentence

The machine rejects a card which is broken.

the relative pronoun *which* can only relate to *card*, not to *machine*. In the paraphrase

The machine rejects {a card which is broken}.

this principle is optionally visualised by curly brackets ({}) around the whole noun phrase.

In the sentence

The customer enters the card that is valid and that is correct.

the second *that* relates to the same noun as the first *that*, i.e. to *card*. The paraphrase is

The customer enters {the card that is valid and that is correct}.

The principle of right association eliminates ambiguities. In some cases, however, the principle can lead to a conflict between the intuitive interpretation and the ACE interpretation. In the following sentence the underlined relative pronoun *that* is intended to relate to the noun phrase *a copy.*

John returns a copy of a book <u>that</u> is damaged.
ACE paraphrase: *John returns a copy of {a book <u>that</u> is damaged}.*

In ACE, however, *that* relates to the immediately preceding noun *book*. To get the intended interpretation you can split the sentence into two sentences as follows:

John returns a copy of a book. The copy is damaged.

Other possibilities to reformulate relative sentences that do not relate to the intended noun phrase can be found in section 6.

3.3.3.3 Notes

We add a remark about the use of phrasal and prepositional verbs (→ 4.2.2) in relative sentences. If the verb in the relative sentence is a phrasal verb (e.g. *fill in*) the preposition or adverb stays after the verb and at the end of the sentence:

The employee checks the form that the customer fills in.

If the verb in the relative sentence is a prepositional verb (e.g. *apply for*) the preposition can either remain after the verb

The course that the employee applies for is full.

or it is possible to move the preposition to initial position when forming relative sentences:

The course for which the employee applies is full.

3.3.4 Subordination — *'If-Then'* Sentences

Path	Grammar of ACE → Composite Sentences → Subordination — *'If-Then'* Sentences
Aim	How to construct and interpret *if-then* sentences.
Examples	*If the card is valid then SimpleMat accepts the card.* *If the card is not valid then SimpleMat ejects the card.*
Terminology	implication, condition, consequence, non-determinism
Principles	—

3.3.4.1 Construction

If-then sentences in ACE always have the form

If ⟨CONDITIONS⟩ *then* ⟨CONSEQUENCES⟩

That means the conditions part — introduced by *if* — always comes first followed by the consequence part — introduced by the word *then*; the order cannot be reversed. Note, that the word *then* cannot be omitted. Conditions as well as consequences can consist of simple or composite sentences.

In ACE the constructor *if ... then* is used to express under which conditions certain consequences are the case. The sentence

If the code is correct then the machine accepts the card.

states that under the condition that the code is correct the machine will accept the card. *If-then* sentences are also called implications or conditional sentences in ACE.

In ACE all conditions have to be made explicit. There is no construct *else* which is known from many computer languages.

SM checks the code.
If the code is correct then SM accepts the card.
If the code is not correct then SM rejects the card.

3.3.4.2 Notes

The implication (*if … then)* binds (→ page 19) less strongly than the coordinators *and*, *or*. That means e.g. that in the following sentence

If the customer pushes the key 'Terminate'
then the machine displays the message 'Transaction Terminated. Retract Card'
and SimpleMat expels the card
and SimpleMat closes the door.

the parts *and SimpleMat expels the card* and *and SimpleMat closes the door* belong to the consequence part of the *if-then* sentence and do not count as separate conjuncts to a complete *if-then* sentence.

Note furthermore the following interactions with *or*-coordinated sentences (→ section 3.3.2). The meaning of the following sentence where *or* occurs in the conditions part

If the card is invalid or is not readable
then SimpleMat retains the card and closes the access door.

is equivalent to

If the card is invalid then SimpleMat retains the card and closes the access door.
If the card is not readable then SimpleMat retains the card and closes the access door.

The sentence which contains the disjunction *or* in the consequence part

If the card is invalid
then SimpleMat retains the card
or SimpleMat rejects the card and displays the message 'Invalid card'.

expresses non-determinism, i.e. it is not determined which of the two consequences will be taken when the condition is true. In the Attempto system non-determinism is treated by selecting one of the alternatives at random. The user cannot control which consequence is chosen.

3.3.5 Quantified Sentences

Path	Grammar of ACE → Composite Sentences → Quantified Sentences
Aim	How to construct and interpret quantified sentences.
Examples	*Every telephone box is red.* *No customer enters a valid card.* *There is a machine that accepts every VisaCard.* *There is no machine that accepts every card.*
Terminology	universal and existential quantification, scope
Principles	Surface Order Determines Quantifier Scope

ACE is supplied with predefined quantifiers that allow to express universal or existential quantification.

- universal quantifiers: *every, each, for every, no*
- existential quantifiers: *there is a, there is no, a/an*

Quantified sentences are used to make assertions about the number of persons or objects that are involved in an event or state. Universal quantification is used when you are talking about all members of a set of persons or objects. Existential quantification is used to express that at least one person or object with certain properties exists.

Note that ACE does not yet allow for other types of quantification (e.g. plural quantification in *all customers*, or cardinality statements like *three customers*). Note furthermore, that ACE will never allow for vague quantifiers like *most customers* etc.

3.3.5.1 Construction

'Every' and 'each' as Quantifiers

The quantifiers *every* and *each*, which are used equivalently in ACE, express universal quantification. *Every* and *each* combine with a countable noun (→ 4.2.1, page 45) to form a universally quantified noun phrase. The resulting noun phrase can be used in subject or complement position.

Every telephone box is red.
The machine checks each card that a customer inserts.

The program detects a virus that is on each computer.
If every university has a card phone then every student buys a phone card.

'For every' as a Quantifier

For every is a a special constructor to express universal quantification. *For every* can only be used in sentence initial position or directly after *if*. *For every* is succeeded by a (modified) noun after which a complete sentence follows. This construction allows to explicitly mark wide scope (→ 3.3.5.2, page 28) of the universally quantified noun phrase.

For every book that is checked out to the borrower the library database lists the title of the book.
For every regular customer there is a card.

'No' as a Quantifier

No is a constructor that combines properties of both, quantifiers and negation (→ 3.3.6). *No* is placed in front of a common noun and is used to negate certain properties for every single member of a set of people or things.

No customer has a valid code.
The automatic teller accepts no phone card.
If no university has a card phone then no student buys a phone card.

The first sentence means that there is no customer who has a valid code, i.e. if you look at every single person who is a customer you will not find a person that has a valid code. (See section 3.3.6.2 for more information on quantification and negation.)

'There is a' as a Quantifier

One way to express existential quantification is to use *there is a*. The constructor *there is a* has the meaning of "there is at least one", i.e. the existence of at least one person or object that has certain properties is stated. In ACE *there is a* has to be followed by a (modified) noun:

There is a customer.
There is an invalid credit card.
There is an automatic teller that accepts every EuroCard.
For every customer there is a card that the customer enters.
If there is a card that is not valid then SimpleMat retains the card.

ACE does not allow that sentences of the form "*there is a* + noun" are followed by prepositional adjuncts. These sentences have to be reformulated using *a/an* or by introducing a relative sentence.

NOT: *There is an automatic teller in the university.*
USE INSTEAD:
An automatic teller is in the university.
There is an automatic teller that is in the university.

'There is no' as a Quantifier

If you want to state that something does not exist you use *there is no*. Analogous to *there is a*, *there is no* is followed by a (modified) noun.

There is no printer.
There is no valid credit card.
There is no automatic teller that accepts an EuroCard.
There is no automatic teller that accepts every card.
For every department there is no printer that the department owns.

Again, sentences with prepositional adjuncts have to be reformulated using *a/an* or by adding a relative sentence.

NOT: *If there is no paper in the printer then the warning lamp blinks.*
USE INSTEAD:
If no paper is in the printer then the warning lamp blinks.
If there is no paper that is in the printer then the warning lamp blinks.

Implicit Quantificational Effect of the Indefinite Article 'a/an'

The indefinite article *a/an* — though not classified as an existential quantifier in ACE — has an implicit quantificational effect similar to *there is a*.

If John enters an incorrect code then the machine retains the card.
(= If there is an incorrect code that John enters then the machine retains the card.)
Every card has a code.
(= For every card there is a code that the card has.)

The quantificational effects of *a/an* show up if *a/an* is used in combination with universal quantifiers, negation or in the condition part of an *if … then* construction. In these cases scope ambiguities (→ 3.3.5.2) — which are characteristic for quantifiers — can occur. The article *a/an* viewed as a quantifier has the same meaning as the quantifier *there is a*. The main difference is that *there is a* allows to move the quantified noun phrases to the beginning of the sentence and thus make scope relations explicit as discussed below.

3.3.5.2 Principles

Principle — Surface Order Determines Quantifier Scope

Scope of Quantifiers

Every quantifier has a so-called scope. The scope of a quantifier contains the parts of the sentence which are affected by the universal or existential quantification. In natural language the scope is not necessarily derivable from a quantifier's position in a sentence. If a sentence contains two or more quantifiers, their scopes overlap — independent of surface order there are different overlapping possibilities which can lead to different interpretations for the sentence. Therefore, in full natural language, sentences with two or more quantifiers are often ambiguous. For example, the sentence

A device controls every pump.

has two interpretations which can be described as follows (the underlined parts indicate the scope of the sentence initial quantifier).

There is a device that controls every pump.
— that is, there is at least one and the same device that controls every pump
For every pump there is a device that controls the pump.
— that is, every pump is controlled by a device, different pumps can be controlled by different devices

You get the first of these two interpretations when the existential quantifier has scope over the universal quantifier, i.e. the existential quantifier has wide scope and the universal quantifier has narrow scope.

You get the second interpretation when the universal quantifier has scope over the existential quantifier, i.e. the universal quantifier has wide scope and the existential quantifier has narrow scope.

Principle of ACE to Determine Quantifier Scope

To avoid scope ambiguities ACE uses the "principle of surface order" which makes the scope of a quantifier uniquely predictable from the quantifier's position in the input sentence.

> **Surface Order Determines Quantifier Scope**
>
> The relative scope of a quantifier corresponds to its surface position. The scope opens at the textual position of the quantified noun phrase and extends to the end of the sentence. If complete sentences are coordinated, the scope of a quantifier extends only to the end of the conjunct/disjunct containing the quantifier. We say that a textually preceding quantifier has *wide scope* with respect to a succeeding quantifier that has *narrow scope*.

Due to the principle of surface order the sentence discussed above

A device controls every pump.

gets the first interpretation with wide scope of the existential quantifier.

To guarantee sufficient expressive power the principle of surface order presupposes methods for syntactic restructuring. ACE does not allow for passives which would reverse the order of the noun phrases automatically. Instead, ACE supplies the quantifiers *for every* and *there is a* which allow to move quantifiers to sentence initial position and thus automatically give them wide scope. Therefore the sentences

A device controls every pump.
There is a device that controls every pump.

express the same meaning with wide scope of the existential quantifier. To give the existential quantifier narrow scope you have to use the sentence

For every pump there is a device that controls the pump.

The following sentence

John assigns a personal code to every customer.

is a typical example where the intended scoping of the quantifiers does certainly not correspond to the surface order of the quantifiers.

ACE interpretation:
There is a personal code that John assigns to every customer.
— that is, every customer gets the same code.

To avoid this unintended interpretation you can use the following ACE reformulation

ACE Reformulation:
For every customer there is a personal code that John assigns to the customer.
— that is every customer gets a possibly different code.

Scoping effects also occur in combination with definite noun phrases. If definite noun phrases are not used anaphorically (→ 3.5.2) in ACE they are interpreted like indefinite noun phrases (→ 4.3.2) which leads to the above mentioned scoping effects. E.g. in the sentence

SimpleMat checks the code of every customer.

the universal quantifier is in the scope of the definite noun phrase *the code* that establishes — as not used anaphorically — an existential quantifier. Hence the sentence above gets an interpretation where every customer has the same code. What you want to say, however, is that every customer has his or her own code which SimpleMat checks. In ACE this has to be

expressed as follows:

ACE Reformulation:
For every customer SimpleMat checks the code of the customer.

We add two further examples of sentences that are scope ambiguous in full natural language and show how the sentences can be unambiguously reformulated in ACE. Scoping 1 corresponds to the interpretation that ACE assigns to the examples without reformulation. Restructuring of quantifiers allows to express scoping 2.

A program calculates every personal code.
Scoping 1 (= ACE interpretation):
There is a program that calculates every personal code.
Scoping 2 *(*= ACE reformulation to reverse the scope relations*):*
For every personal code there is a program that calculates the personal code.

Every customer who has a VisaCard uses a teller machine.
Scoping 1(= ACE interpretation):
For every customer who has a VisaCard there is a teller machine that the customer uses.
Scoping 2 (= ACE reformulation to reverse the scope relations):
There is a teller machine that every customer who has a VisaCard uses.

Extending the Scope of a Quantifier

The scope of a quantifier extends to the end of a sentence. If complete sentences are coordinated by *and* or *or*, the scope of a quantifier in a conjunct/disjunct extends only to the end of the respective conjunct/disjunct. In the following sentences the scope of *every* is underlined.

Every customer has a card. He enters it.
Every customer has a card and he inserts it into the slot.

The pronouns *he* and *it* are outside of the scope of the universal quantifier and thus cannot refer back to *every customer* and to *a card*, respectively (→ section 3.5.2.5).

You can enlarge the scope of a quantifier by adjoining relative sentences

Every customer has a card that he enters.
Every customer has a card that he inserts into the slot.

References from *he* to *every customer* and from *it* to *a card* are now possible (→ section 3.5.2).

Similarly, you can enlarge the scope of a quantifier by coordination of non-sentential elements.

Every customer has a VisaCard and a MasterCard.
Every customer enters a password or shows the access card that he receives at the reception.

3.3.5.3 Notes

Sentences with Coordination and Quantification

The combination of quantifiers together with coordinators (→ 3.3.2) can lead to ambiguities in full natural language. ACE avoids the ambiguity by interpreting the sentence

Every student visits Rome or Paris.

as

Every student visits Rome or [visits] Paris.

meaning that every student visits at least one of the two cities Rome or Paris, possibly both. This interpretation is a result of the distribution principle (→ 3.3.2, page 20) — according to which just the verb *visits* is distributed — and the principle of surface order (→ page 28) — stating that the quantified noun phrase *every student* has scope over the whole sentence.

The alternative interpretation that all of the students visit the same city (it is just not known which of the two cities, Rome or Paris) can also be expressed in ACE by coordinating two sentences:

Every student visits Rome or every student visits Paris.

Further examples that illustrate the treatment of quantification and coordination in ACE can be found in section 6.

3.3.6 Negated Sentences

Path	Grammar of ACE → Composite Sentences → Negated Sentences
Aim	How to construct and interpret negated sentences.
Examples	*The card is not in the slot.* *No customer has a VisaCard.* *There is no customer who has a valid VisaCard.*
Terminology	verb phrase negation, noun phrase negation
Principles	Interaction of Principles

If you want to express that something is not happening, not true or not the case you use negative statements. Negative statements in ACE can be formed with the following constructors:

- verb phrase negation (*does not, not*)
- noun phrase negation (*no, there is no*)

3.3.6.1 Construction

Negating the Verb Phrase

Not is used to negate a verb phrase. If the verb is the copula *be* you simply put *not* after the forms of *be*.

The card is not in the slot.
If the code is not valid then SimpleMat rejects the card.

If the verb is not *be* you have to put *does not* before the base form of the verb.

SimpleMat does not accept a VisaCard.
If the code is not valid then SimpleMat does not accept the card.

Negating the Noun Phrase

No is used in front of nouns and expresses that something does not exist or is not available.

No customer has a VisaCard.
The customer has no VisaCard.
The machine accepts no VisaCard.

Note that the constructor *no* combines properties of both quantifiers and of negators. In ACE

no means the same as *there is no … that*. The following ACE sentences express the same meaning:

No customer has a VisaCard.
There is no customer who has a VisaCard.
If there is a customer then he does not have a VisaCard.

3.3.6.2 Notes

Interaction of Principles

To avoid ambiguities when negators are combined with coordinators and quantifiers ACE uses the three principles:

1. Surface order of quantifiers (→ page 28)
2. Distribution (→ page 20)
3. Binding hierarchy (→ page 19)

The interaction of negation with quantifiers is treated according to the principle of surface order, the interaction of negation with coordinators follows the distribution principle and the binding hierarchy principle, where the distribution principle has to be applied first. In the following sections examples show how the principles work together.

Sentences with Negation and Quantification

'Not' and Quantifiers

Applying the principle of surface order to the sentence

SimpleMat does not accept a VisaCard.

has the consequence that *a VisaCard* has narrow scope with respect to the negation *does not* — meaning that it is not the case that there is a VisaCard that SimpleMat accepts. This interpretation can be equivalently expressed as

There is no VisaCard that SimpleMat accepts.

To express the meaning with wide scope of the existentially quantified noun phrase *a VisaCard* the following reformulation is necessary:

There is a VisaCard that SimpleMat does not accept.

Note furthermore that ACE does not allow for sentences beginning with *not every*:

NOT: *Not every automatic teller accepts a VisaCard.*
USE INSTEAD: *There is an automatic teller that does not accept a VisaCard.*

'No' and Quantifiers

In ACE *no* expresses the same meaning as *there is no*. According to the principle of surface order the noun phrase *no customer* in the sentence

No customer enters a card.

has wide scope with respect to the existentially quantified *a card* leading to an interpretation that can be equivalently expressed as

There is no customer who enters a card.
Every customer does not enter a card.
If there is a customer then the customer does not enter a card.

If you want to give *a card* wide scope you have to use

There is a card that no customer enters.

The same principles hold for the combination of *no* with universally quantified noun phrases. The sentence

No machine accepts every credit card.

can be equivalently expressed as

There is no machine that accepts every credit card.

To give the universally quantified noun phrase *every credit card* wide scope you have to use

For every credit card there is no machine that accepts the credit card.

Logical Equivalences to Aid Reformulations

Users familiar with predicate logic have noticed that many suggested reformulations are based upon identities of first-order predicate logic, e.g. *there is no* is the same as *every … not* etc.

Sentences with Coordination and Negation

'Not' and Coordinators

The sentence

John does not visit Rome and Paris.

is paraphrased according to the distribution principle as

John does not visit Rome and [does not visit] Paris.

After distribution, the binding hierarchy — saying that *does not* binds more strongly than *and* — is applied leading to an interpretation that John visits neither Rome nor Paris.

If you want to make the weaker statement that John does not visit each of the two cities, though he may visit one of them you have to use

John does not visit Rome or does not visit Paris.

'No' and Coordinators

The sentence

No customer enters a VisaCard or a MasterCard.

is paraphrased as

No customer enters a VisaCard or [enters] a MasterCard.

according to the distribution principle. The sentence expresses that every customer enters neither a VisaCard nor a MasterCard. This interpretation can be equivalently expressed as

Every customer does not enter a VisaCard or a MasterCard.
There is no customer who enters a VisaCard or a MasterCard.

If you want to express the weaker statement that there is no customer who enters both a VisaCard and a MasterCard (though he may enter one of the cards) you can say

There is no customer who enters a VisaCard and a MasterCard.

Negation, Quantifiers and Coordinators

The treatment of combinations of negation, quantifiers and coordinators follows from the principles mentioned in this section. Note that first possible distributions have to take place, then binding hierarchy and surface scoping are applied to interpret a sentence.

Despite the disambiguating principles of ACE we recommend to avoid complex combinations

of constructors and rather try to be as explicit as possible by repeating elements or by using two sentences instead of one.

3.4 Query Sentences

Path	Grammar of ACE → Query Sentences
Aim	How to construct query sentences in ACE.
Examples	*Is the card valid?* *What does the customer enter?*
Terminology	*Yes/no*-queries, *Wh*-queries
Principles	—

With query sentences users form questions to examine the contents of the ACE specification. Query sentences are not used directly in a specification. Query sentences are classified into two main types.

- *Yes/no*-queries
- *Wh*-queries

Query sentences can be formed from simple sentences with so-called query words (→ 4.3.5) (e.g. *what?*, *how?* etc.), by inversion, or by adding *does*. Query words in ACE are function words and as such predefined in the system. Users cannot add or change existing query words.

3.4.1 'Yes/No'-Queries

Yes/no-queries can be answered by 'yes' or 'no'. With *yes/no*-queries the user checks the factual information expressed in the specification.

Yes/no-queries are derived from simple sentences either by inverting the subject and the verb *be*

Is the code valid?

or by inserting *do* or *does* if the verb is not *be*

Does the customer enter the correct code?

Table 7 on page 60 shows some *yes/no*-queries and the resulting answers in ACE.

3.4.2 'Wh'-Queries

Wh-queries cannot be answered by *yes* or *no* but expect a specific reply according to the query word used.

Wh-queries begin with a so-called *wh*-word such as *what, when, who, where* etc.

Who enters the card?
What does the customer enter?

Despite their name *wh*-words do not necessarily begin with *wh* (e.g. *how* is a *wh*-word as well). Queries can also be formed by the combinations *Who … to, What … to, To … whom, To … what*. The combination *Whom … to* is not allowed.

Who does John give a card to?

Table 8 on page 60 gives a complete list of the possible *wh*-words in ACE.

3.5 Specifications as Sets of Sentences

3.5.1 Specifications as Texts

A specification normally consist of more than one sentence. The following is an excerpt of the specification of an automatic teller machine called SimpleMat.

*/*SimpleMat*
Specification of an automatic teller machine/*
A customer inserts a card and enters a personal code.
SimpleMat checks the personal code.
If the personal code is valid then SimpleMat accepts the card.
If the personal code is not valid then SimpleMat rejects the card.
% End of example

Note first of all, that specifications can contain comments either of the form

/ TEXT */*

where the text can extend over more than one line, or of the form

% TEXT

where the comment ends at the end of the same line. Comments are not translated by the Attempto system.

The sentences not marked as comments are interrelated to create a coherent specification. There are two forms of interrelation, viz. ***references to previously mentioned objects*** (→ 3.5.2) and ***temporal order*** (→ 3.5.3).

3.5.2 References to Previously Mentioned Objects

Path	Sets of Sentences → Reference to Previously Mentioned Objects
Aim	How to establish references to previously mentioned objects in a specification.
Examples	*John enters a card. It is invalid.* *A customer has a VisaCard. The VisaCard is not valid.*
Terminology	anaphor, antecedent, accessibility
Principles	Personal Pronouns as Anaphors Definite Noun Phrases as Anaphors Dynamic Names as Anaphors

Terminology

Anaphor

An anaphor in ACE is a noun phrase that refers to a previously mentioned noun phrase and thus expresses that you are talking about the same object. Anaphors in ACE can be personal pronouns (→ 3.5.2.2 and 4.3.3), definite noun phrases (→ 3.5.2.3 and 4.3.2) or dynamic names (→ 3.5.2.4 and 4.2.1.3).

Accessibility

Only noun phrases occurring in certain positions in a sentence can function as antecedents for anaphors. We say that a noun phrase must be accessible for anaphoric reference. The so-called accessibility restrictions are defined in section 3.5.2.5.

3.5.2.1 Proper Nouns

Construction

Proper nouns (→ 4.2.1.2) like *SimpleMat* are defined in the lexicon and can be used freely everywhere in a specification, and always denote the same object.

SimpleMat is an automatic teller machine. John inserts a card into SimpleMat.

3.5.2.2 Personal Pronouns as Anaphors

Construction

Personal pronouns (→ 4.3.3) are words which function as a whole noun phrase. They are used as substitutes or replacements for previously mentioned noun phrases. In the sentence

A customer enters a card. It is valid.

the pronoun *it* is an anaphor that substitutes the noun phrase *a card*. *It* and *a card* refer to the same object.

Principle — Personal Pronouns as Anaphors

In full English it is not always clear what a personal pronoun refers to. In the sentence

A customer shows a defect card to an employee. He checks it.

the pronoun *it* unambiguously refers to the same object as the noun phrase *a defect card*. The pronoun *he*, however, can refer to *a customer* or to *an employee*. To avoid this ambiguity ACE uses the following interpretation principle.

> **Personal Pronouns as Anaphors**
> A personal pronoun always refers to the most recent accessible noun phrase that has the same number and gender. If no reference can be found an error is flagged.

Due to this principle the pronoun *he* in the example above unambiguously refers to the noun phrase *an employee*. The Attempto system visualises this principle in a paraphrase

A customer shows a defect card to an employee. [The employee] checks [the defect card].

where every personal pronoun is replaced with the complete noun phrase the pronoun stands for, signalling the replacement by the definite article *the*.

3.5.2.3 Definite Noun Phrases as Anaphors

Construction

Definite noun phrases (→ 4.3.2) can also be used as anaphors.

A customer shows a defect card to an employee. The employee checks it.

The definite noun phrase *the employee* refers back to the indefinite noun phrase *an employee*. Both noun phrases denote the same object.

Principle — Definite Noun Phrases as Anaphors

The interpretation of definite noun phrases used as anaphors is guided by the following interpretation principle:

> **Definite Noun Phrases as Anaphors**
>
> Definite noun phrases used anaphorically always refer to the most recent accessible noun phrase that is suitable, i.e. that has the same noun and at least the same adjectives, possessive noun, *of*-prepositional phrase, and apposition as the referring definite noun phrase; e.g. the definite noun phrase *John´s code* can refer to a preceding noun phrase *John´s correct code ´1234´*, while *the incorrect code* or *John´s old code* cannot. Note that the presence of relative sentences in the antecedent does currently not play a role in the determination of suitable antecedents.
>
> If no reference can be found, a definite noun phrase is treated as if it were an indefinite noun phrase introducing a new object.

Notes

Definite noun phrases can be used to establish anaphoric references that are impossible with personal pronouns.

John enters a card into a machine. It is damaged.

According to the above principle the Attempto system will generate the paraphrase

John enters a card into a machine. [The machine] is damaged.

while the user wanted to express that the card is damaged. Anaphoric reference with the help of a definite noun phrase

John enters a card into a machine. The card is damaged.

solves the problem.

3.5.2.4 Dynamic Names as Anaphors

Construction

Dynamic names (→ 4.2.1.3) are not defined in the lexicon, start with a capital letter and have to be introduced in appositive position:

A customer enters a card B and a card C.
B is not valid.

When used alone in a succeeding sentence dynamic names function as anaphors.

Principle — Dynamic Names as Anaphors

The interpretation of dynamic names obeys the following principle

> **Dynamic Names as Anaphors**
>
> A dynamic name used alone always refers to the most recent accessible noun phrase in which the dynamic name occurs. In the paraphrase the dynamic name is replaced with the complete noun phrase which introduced the dynamic name, i.e. the paraphrase contains at least the same noun, the same adjectives, possessive noun, *of*-prepositional phrase, and appositions as the noun phrase introducing the dynamic name.

For example the sentences

A customer XYZ enters a blue card B and enters a green card G.
B is valid.
G is not valid.

get the paraphrases

A customer XYZ enters a blue card B and enters a green card G.
[The blue card B] is valid.
[The green card G] is not valid.

3.5.2.5 Accessibility Restrictions

Construction

Not every noun phrase is accessible for an anaphoric reference, e.g. in the specification

*The customer does not **insert a card**.*
It is correct.

it cannot refer back to *a card*, because the noun phrase *a card* occurs within a negation and thus is not accessible for anaphoric reference.

Accessibility restrictions exist for negations, for implications and for universally quantified sentences. These restrictions are explained below.

Principles — Accessibility Restrictions

Accessibility Restrictions for Negations

When definite or indefinite noun phrases occur within a negation (→ 3.3.6) it is not possible to have anaphoric reference to these noun phrases from outside the negation. In the sentences

John does not have a VisaCard.
It is a MasterCard.

the pronoun *it* cannot refer back to *VisaCard*. In the sentences

No customer enters a VisaCard.
He enters a MasterCard.

the pronoun *he* cannot refer back to customer. We say that noun phrases (except proper nouns) within a negation are not accessible for anaphoric reference.

Accessibility Restrictions for Implications

In ACE, definite or indefinite noun phrases that occur in the conditions or in the consequence part of an implication are not accessible for anaphoric reference from sentences following the implication. In the sentences

If a customer pushes the key 'A' then the automatic teller prints a receipt.
It shows the date of the transaction.

it can neither refer back to *a receipt* nor to *the automatic teller*, nor to *the key 'A'* in the conditions part. You can use a relative sentence to express the possibly intended meaning:

If a customer pushes the key 'A' then the automatic teller prints a receipt that shows the date of the transaction.

If an anaphor occurs within an *if-then* sentence anaphoric reference to preceding noun phrases (within and outside the *if-then* sentence) is possible. The sentence

If the customer inserts a card into the machine then it checks the card.

is paraphrased as

If the customer inserts a card into the machine then [the machine] checks [the card].

Accessibility Restrictions for Universally Quantified Sentences

If an indefinite or definite noun phrase is in the scope of a universal quantifier it is not possible to have anaphoric reference to these noun phrases from a subsequent sentence:

Every customer has a correct personal code.
The correct personal code has a number.

In ACE the definite noun phrase *the correct personal code* cannot be used to refer back to *a personal code*. The reason is that *a personal code* is in the scope of the universal quantifier *every.* If you want to make such a connection you can use a relative sentence which extends the scope of the quantifier — as we have seen in section 3.3.5.2, page 30.

Every customer has a correct personal code that has a number.

The same effect of inaccessibility occurs with definite noun phrases inside the scope of quantifiers:

Every customer enters the personal code.
It is correct. (*It* does not refer back to *the personal code.*)

A possible reformulation is:

Every customer enters the personal code that is correct.

If an anaphoric noun phrase occurs within the scope of the quantifier it is possible to have anaphoric reference to noun phrases that are themselves in the scope of the same quantifier. This is important if you use e.g. *for every* at the beginning of a sentence

For every customer who is new the employee checks the signature of the customer.

where *the customer* is anaphoric to *for every customer.* Note the possibilities shown in 3.3.5.2 of how you can extend the scope of a quantifier.

Accessibility Restrictions for Coordinated Sentences

When sentences are coordinated anaphoric reference from later sentences to previous sentences of the coordination is possible. The accessibility restrictions are analogous to sequences of sentences separated by full stops:

The customer inserts a card or he enters a code.
The customer inserts a card and the employee checks it.

When phrases are coordinated, anaphoric reference from one phrase to a previous phrase is possible:

Every customer has a card and enters it.

3.5.3 Temporal Order

Path	Sets of Sentences → Temporal Order
Aim	How to establish temporal order in specifications.
Examples	*A customer inserts a card.* *The customer enters a code.*
Terminology	temporal order
Principles	Default Temporal Order

Construction

Each verb in a sentence denotes an event of state. Events occur instantaneously, while states — i.e. relations that hold or do not hold — last over a period of time.

Each occurrence of a verb is implicitly associated with a time. If the verb denotes an event this is the time point at which the event occurs; if the verb denotes a state this is the time point from which on the state holds.

Principle — Default Temporal Order

The default temporal order of events and states is given by the following ACE principle.

> **Default Temporal Order**
> The textual order of verbs establishes the relative temporal order of their associated events and states.

For example in

A customer inserts a card.
The customer enters a code.
The code is correct.

the event *insert* is temporally followed by the event *enter* which is temporally followed by the onset of the state of *be correct*. Similarly, in

A customer inserts a card and enters a code.

the event *insert* is temporally followed by the event *enter.*

In implications the preconditions in the *if*-part temporally precede the consequences in the *then*-part.

Sentences coordinated by *or* deserve special consideration (→ 3.3.2.3). All sentences are considered as temporally parallel. If the *or*-coordinated sentences appear in the *if*-part of an implication the textually first sentence that is true is selected. In all other cases we have nondeterminism: the system selects one sentence of the *or*-coordinated sentences at random.

Notes

Users can override the default temporal order by adding the prepositional phrase *at the same time* to express simultaneity, or *in any temporal order* to express that the temporal order is left open.

John enters a card and a code at the same time.
John enters a card and a code in any temporal order.

Finally, users can explicitly specify times through prepositional phrases.

The computer makes a backup at 2 pm.

4 Vocabulary of ACE

4.1 Basic Terminology

Path	Vocabulary of ACE → Basic Terminology
Aim	Overview of Word Classes in ACE
Examples	noun, verb, adjective, adverb
Terminology	vocabulary, lexicon, word class, content words, function words, constructor, compounds, synonyms, abbreviation

In this section the classification of the basic units of sentences, the words, is explained. We will call the set of words allowed in ACE its vocabulary or sometimes its lexicon.

4.1.1 Word Classes: Content Words and Function Words

The vocabulary of ACE is divided into the two main word classes

- content words
- function word

which are further subdivided into more specific word classes. Words belonging to the same word class show a syntactically analogous behaviour.

Content Words

Content words typically refer to objects, events, qualities, properties in the outside world. The content word-class is an 'open' word class in the sense that new members can be added, and words can change or disappear as language develops. The content word-classes can have very large numbers of members.

The content word classes of ACE are

- Noun (→ 4.2.1) *John has a card that carries a machine-readable code.*
- Verb (→ 4.2.2) *The customer enters the card which belongs to John.*
- Adjective (→ 4.2.3) *The card is readable and the customer enters the correct code.*
- Adverb (→ 4.2.4) *The customer enters the personal code correctly.*

In ACE all content words are defined by the user. The content words are incrementally added or modified during the specification process with the help of a lexical editor — a software tool that guides the input of new words. Thus, by adding content words, users create their own application specific lexicon. A careful classification of the content words is essential because correct analysis of the specification text depends on the correct input of new content words.

For every new content word users first have to determine the corresponding word class. After that users have to input additional information that characterises the new word more precisely. This process of adding a new word is guided by the lexical editor as explained below.

Note that there are word forms which can belong to more than one word class (e.g. the forms *loop* or *set* can both be a noun or a verb). It follows that the user cannot classify an instance of a word by considering it in isolation. The user has to examine how the word is used in that particular part of the specification. If a specification contains different uses of the same word form the user has to enter a separate entry for each possibility (e.g. one entry for *loop* as verb and

one entry for *loop* as noun).

Function Words

Function words are used to establish the syntactic or logical structure of sentences. The function word-class is a 'closed' class in the sense that it has a limited number of members. Function word classes change relatively little.

- The function word classes of ACE are: preposition (→ 4.3.1)*The customer of the bank inserts a card into the slot.*
- article (→ 4.3.2) *The customer inserts a card.*
- personal pronoun (→ 4.3.3) *He enters a card and the machine rejects it.*
- constructors (→ 4.3.4)
  - coordinator *The customer inserts a card and enters a code.*
  - relative pronoun *A customer enters a card that is not valid.*
  - implication *If a card is valid then SimpleMat accepts the card.*
  - quantifier *Every customer enters a card.*
  - negator *SimpleMat does not accept a VisaCard.*
- query word *Who enters a card?*

The term "constructor" is specifically coined for ACE to summarise words with which composite sentences can be formed from simpler sentences (→ 4.3.4 for a complete list of constructors).

Function words in ACE are predefined in the system. The user cannot add new function words or change the existing entries. The user only has to know which function words are provided in ACE and how they are used.

4.1.2 Compounds

The structure of the words in ACE can be simple, i.e. consisting of a single word (*customer*) or compound, i.e. consisting of two or more parts (*check code*). Compounds can be spelled as more than one word (*bank card*) or hyphenated (*bank-book*). In ACE compounds are treated as a single unit in the vocabulary, i.e. there is just one lexicon entry for e.g. *check code*. This has the consequence that *check code* is always treated like one word; the individual words *check* and *code* will not be identified by the system. The users defining the lexicon decide which combinations of words are treated as compounds. Classification criteria and examples for different forms of compound nouns, verbs, adjectives and adverbs are introduced in the corresponding sections.

4.1.3 Synonyms and Abbreviations

When a new word is entered we distinguish between the ***primary word*** (the new word) and its ***synonyms.*** In ACE it is possible to associate each primary word with a set of synonyms — words having the same meaning (e.g. *customer, client, patron*). Note that ACE employs a *strict* sense of synonymy, i.e. the primary word and its synonyms can be substituted for each other in all contexts while the meaning of the sentence does not change.

In ACE ***abbreviations*** are a special case of synonyms. In specifications technical terms often consist of long compounds or multi-word groupings. Introducing abbreviations for these words allows for a more economical treatment, especially when the same words are frequently used. In this manual, for example, we introduced the abbreviation 'ACE' as a shorter form for 'Attempto Controlled English'. Whenever the abbreviation 'ACE' occurs it can be replaced

with the equivalent longer term.

In the course of the internal processing of a specification every synonym or abbreviation is replaced with the corresponding primary word. This substitution is made visible to the user by creating a paraphrase where the primary word is enclosed in square brackets. For example, *ACE is a controlled natural language* is paraphrased as [*Attempto Controlled English*] *is a controlled language*.

Note that users can define, change or delete synonyms or abbreviations for all primary content words. The synonyms of function words can only be changed by specialists maintaining the Attempto system.

4.1.4 Comments

Every word in the vocabulary is assigned a field for comments where the user can input text, which does not have to be in a particular format or follow a particular grammar. The comment field can store information about the intended meaning or the usage of the word, or any other comments. Users can only add comments to content words.

4.2 Content Words

4.2.1 Nouns

Path	Vocabulary of ACE → Content Words → Nouns
Aim	How to define nouns in ACE.
Examples	*man, SimpleMat, Mr Smith, 'Enter your code.', A*
Terminology	common noun, proper noun, quoted string, dynamic name, countability, type, gender, number

Nouns are used to designate the objects a specification talks about. In ACE three subclasses of nouns are distinguished:

- common nouns (→ 4.2.1.1) (*customer, book, credit card, registration number*)
- proper nouns (→ 4.2.1.2)
  - ordinary (*London, John, Central Library*)
  - quoted strings (*'Enter the personal code'*)
- dynamic names (→ 4.2.1.3) (*A, XYZ, Xyz*)

Note that nouns can be simple (*library*) or compound (*Central Library*).

4.2.1.1 Common Nouns

Common nouns (*customer, book, credit card, registration number*) are nouns used to refer to persons, objects or substances.

If users classify a word as a common noun they additionally have to enter how the common noun behaves with respect to the following features:

- countability (COUNT or MASS)
- type (e.g. PERSON, TIME, OBJECT)

- gender (MASCULINE, FEMININE, MASC/FEM, NEUTER)
- number (information about singular and plural form of noun)

Countability

All common nouns in ACE fall into one of two subclasses:

- count nouns: *account, number, credit card*
- mass nouns: *equipment, time, air-traffic control*

Count Nouns

Count nouns describe objects that can be counted, e.g. *card, book*. In ACE the user can identify count nouns by the following tests. Count nouns

- can be singular or plural (*card*/*cards*)
- can be used with an indefinite article (*a card*)
- can be used with numbers (*three cards*)
- can answer queries like *How many?*

Mass Nouns

Mass nouns refer to things not usually counted, e.g. *earth, water.* These are general things such as qualities, substances, processes, and topics rather than individual items or events. Mass nouns are identified by the following tests. Mass nouns

- do not have an indefinite article (NOT: *a money*, NOT: *a meat*)
- are not used with numbers (NOT: *three health*, NOT: *two meat*)
- have only one form (usually no plural form) (*weather*/NOT: *weathers*)

Note

Many nouns have a number of different meanings, and so can be, for example a count noun for one meaning, and a mass noun for another (e.g. *conflict, victory*). Users have to define each meaning separately.

Type

Common nouns (and proper nouns) in ACE have to be classified according to their semantic type. The notion of semantic type is not usually found in school grammars but is adopted from theoretical linguistics and computer science. Semantic types express what kind of object the noun denotes. Currently, ACE employs a coarse classification into three types:

- person: *customer, manager*
- time: *month, morning*
- object (concrete or abstract): *card, bank, idea*

Users have to assign the types according to the following classification criteria:

Type — Person

- nouns of type “person” denote entities that can be referred to by the pronouns *he* or *she*, but not by the pronoun *it*. Usually these entities are human beings. (Example: *The customer sings. He is happy.)*
- nouns of type “person” allow for queries containing *Who? (*Example*: The customer sings. Who sings?)*

Type — Time

- nouns of type “time” denote entities that have a temporal duration or denote a time point (*morning*, *Monday*, *9 o’clock*, *hour*)
- temporal nouns in phrases like *in the morning*, *on Monday*, *at 9 o’clock*, have the effect that the phrases denote a certain time or time span (and not a location like *in the bank*)
- temporal nouns can be used in phrases answering temporal queries like *When?*, *How long?*, *Until when?*, etc.

Type — Object

- In the current type classification of ACE every noun that is neither of type “person” nor of type “time” falls in the class “object”, i.e. the nouns are impersonal and non-temporal.
- nouns of type object can denote physical objects with spatial properties (e.g. *rock*, *tree*, *bank*) or abstract objects (e.g. *idea*, *happiness*, *equipment*)
- nouns of type object can be queried with *What?*

Gender

In ACE for each common noun (and proper noun) the user has to determine its gender. The gender of a noun tells us if the noun refers to a male (*father*, *brother*), a female (*mother*, *waitress*, *sportswoman*), if the noun can be used for both male and female (*teacher*, *doctor*), or if the noun refers to something which is neuter, i.e. neither male nor female (*book*, *card*, *bank*). For each common noun the user has to enter exactly one of the following values for the gender feature.

- MASCULINE: noun refers to males only
  (*father*, *brother*, *son*)
- FEMININE: noun refers to females only
  (*mother*, *sister*, *waitress*, *policewoman*, *stewardess*)
- MASC/FEM (= masculine or feminine): noun refers to males or females
  (*teacher*, *doctor*, *writer*)
- NEUTER: noun refers to neither males nor females
  (*book*, *card*, *computer*)

Number

Entering a common noun the user has to give additional number information for the noun. For count nouns the user has to input the singular and plural form of the noun (*book*/*books*, *man*/*men*). For mass nouns there is only one noun form which is in the singular (*childhood*, *equipment*). The input of the number information is guided by the lexical editor. Currently, ACE does not yet allow for plural nouns. Nevertheless, the user has to input plural forms of nouns so that the vocabulary is compatible with future extensions of ACE.

Note that there are some common nouns, for which only a singular form is used (*past*, *future*, *air*). These nouns are called **singular nouns**. Singular nouns are always used with an article, because they behave like the singular form of a count noun. In ACE these nouns are classified as count nouns (e.g. *future*), but no plural form is entered. Conversely, there are also nouns which have only a plural form. For example, you can buy *goods*, but not *a good*. These nouns are called **plural nouns**. Examples are: *goods*, *clothes*, *weather conditions*, *trousers*, *glasses*. In ACE for these words only the plural form is entered.

The following table summarises the classification process for common nouns.

Table 3: Classification of Common Nouns

Feature	Possible Values		
countability	COUNT or MASS		
type	currently: either PERSON or TIME or OBJECT		
gender	Either MASCULINE or FEMININE or MASC/FEM or NEUTER		
number	singular form	plural form	
	man	*men*	noun has singular and plural form
	future	-	singular noun, no plural form
	-	*trousers*	plural noun, no singular form

4.2.1.2 Proper Nouns

Proper nouns are used to refer to a particular person, place or institution. We say that proper nouns are understood to have a unique reference: *Zurich* refers to one particular city and *John* (in a given context) refers to one particular person. Proper nouns fall into two subclasses, "ordinary" proper nouns and "quoted strings".

Ordinary Proper Nouns

In ACE ordinary proper nouns can be identified by the following tests. Proper nouns

- are spelled with a capital letter (*Washington*)
- have no article (*January, SimpleMat*)
- can be simple (*John*) or compound (*John Smith*) which has to be specified in the lexicon

Proper nouns can e.g. be personal names (without title or with a title that must not end with a dot): *John*, *Shakespeare*, *Mr Smith*, *Dr Johnson*, calendar items: *Christmas*, *Independence Day, January, Monday, 9 o'clock*, geographical names: *Europe*, *North America*, *Zurich*, *Lake Michigan*, *Mount Everest*, institutions: *Thames Polytechnic*, *The South China Morning Post.*

Proper nouns additionally have to be classified with respect to the features

- type
- gender
- number

The type, gender and number information for proper nouns is analogous to that of common nouns (→ 4.2.1.1).

Notes

In full natural language some proper nouns are used with the definite article *the*. This is especially the case with plural names — which are not yet allowed in ACE (*The Netherlands*, *the Canaries*, *the Alps*, *the Millers*), names of organizations (*the United Nations*), institutions (*the Hilton, the British Museum*), newspapers (*The Daily Express*, *The New York Times*), some geographical names (*the Thames*, *the Pacific*) etc. Currently, this usage is only possible in ACE if the definite article starts with a capital letter and the proper noun is entered like a compound proper noun in the lexicon. If the article is not spelled with a capital letter the noun phrase will

be dealt like a normal definite noun phrase, which will probably cause an error message if the noun is undefined.

Quoted Strings

As a special subclass of proper nouns ACE introduces quoted strings mainly to deal with application specific texts like messages, instructions, titles, reference to buttons or keys on a machine etc. Quoted strings are stored as a unit in the vocabulary even if they are not "words" in the usual sense. The text of the quoted string has to be enclosed in single quotes (`'text'`). Inside the quotes any text is admitted: a single letter or number, one ore more words, one or more sentences etc. The underlined parts of the following sentence are examples for quoted strings.

> *If the customer pushes the key 'Terminate' then SimpleMat displays the message 'Amount is being prepared. Retract Card.'.*

Note that quoted strings can only be used in appositive position, i.e. the quoted string has to occur next to another noun phrase which refers to the same person or thing. The underlined parts below are examples for quoted strings in appositive position:

> *the key 'A'*
> *the message 'Amount is Being Prepared. Retract Card.'*
> *the novel 'Moby Dick'*

4.2.1.3 Dynamic Names

Dynamic names are introduced locally in the specification, they must not be defined in the lexicon. Dynamic names are non-compound words without definite article and start with a capital letter. Dynamic names are introduced in appositive position and automatically get the number, gender and countability information of the preceding common noun.

> *The graph colouring maps a region RA into a colour CA.*

Dynamic names in appositive position can be used to distinguish single instances of the set of objects denoted by the preceding common noun.

> *Every animal A1 eats every animal A2 that is smaller than A1.*

When used alone in the succeeding text dynamic names are used anaphorically (→ 3.5.2.4).

4.2.2 Verbs

Path	Vocabulary of ACE → Content Words → Verbs
Aim	How to define and use verbs in ACE.
Examples	*walk, insert, give to, look after*
Terminology	phrasal and prepositional verbs, intransitive, transitive, ditransitive verbs, copula *be*, event, state

Verbs are used to say what someone or something does (e.g. *insert*, *walk*, *check*) or what happens to someone or something (e.g. *die*) or which states hold (*have*, *own*, *stay*). Additionally, there is one special verb, the copula verb *be*, that is predefined in the system (→ page 51). The following points are relevant for adding a new verb to the content word lexicon.

Verb Forms

As a specification language ACE does not need the full range of verb forms. The following verb forms are available:

- only third person singular (*she goes*) and third person plural (*they go*)
  (Note: plural verbs are not yet allowed in ACE. Nevertheless, the third person plural form of the verb has to be entered, so that the vocabulary supports future extensions of ACE.)
- only indicative mood (*he goes*),
  no imperative (NOT: *Go away!*), no subjunctive (NOT: *They demanded that he should go*, *If I were you ...*)
- only present tense (*go*, *enter*),
  no future or past tense (NOT: *will go*, *went*, *has gone* etc.)
- only simple present (*enters*),
  no continuous form (NOT: *is entering*, *has been entering*)
- only active voice (*enters*),
  no passive voice (NOT: *is entered*)
- no modal auxiliaries (NOT: *may*, *might*, *can*, *could*, *must* etc.)

The two possible resulting verb forms have to be entered by the user: e.g.

(he/she/it) enters – (they) enter,
(he/she/it) goes – (they) go,
(he/she/it) deals with – (they) deal with.

Simple and Compound, Phrasal and Prepositional Verbs

Verbs in ACE can be simple (*enter*, *give*, *insert*, *check*) or compound, i.e. consisting of two or more words (*hitch-hike*, *dry clean*, *proof-read*).

ACE allows for ***phrasal verbs*** (*look after*, *come across*, *get up*, *deal with*) which are treated like compound verbs. Phrasal verbs are verbs that are used together with an obligatory particle, i.e. a preposition like *on*, *after* or an adverb like *out*, *up* which must not be separated from the verb. Users have to enter the verb together with its particle into the lexicon. Intransitive (→ page 50) phrasal verbs (*take off*) take no complement, transitive (→ page 50) phrasal verbs (*fill out*) take noun phrases as complements.

The plane takes off.
The employee hands out the money.
John fills out the form.

Phrasal verbs have to be distinguished from ***prepositional verbs*** (*call [for], give [to]*) which are also allowed in ACE but which are *not* treated like compounds. Whereas phrasal verbs take noun phrases as complements, prepositional verbs take prepositional phrases as complements. The preposition associated with the verb is determined by the verb and does not have an identifiable independent meaning of its own.

John calls for an inquiry.
John calls on a friend.
John applies for a job.
John gives a book to Mary

In the lexicon, users have to specify *that* the verb takes a prepositional phrase as complement and *which* preposition is determined by the verb. Note that different prepositions associated with a verb lead to different meanings and therefore to different lexical entries (*call [for]), call [on]*).

At first sight, phrasal verbs and prepositional verbs may look similar. Therefore, the following table gives a test that helps to distinguish them.

Table 4: Phrasal vs. Prepositional Verbs

Phrasal verbs	Prepositional verbs
It is *not* possible to move the particle (preposition or adverb) to initial position when forming relative sentences or queries. NOT: *the money out which the employee hands* *Out which money does the employee hand*?	It is possible to move the preposition to initial position when forming relative sentences or queries. OK: *a job for which John applies* *For which job does John apply*?
Examples	
look after, come across get up, blow up, bring about, call off, fill out, turn on,	*rely [on], consist [of], apply [for], give ...[to]*

Complementation Patterns

As introduced in section 3.1 verbs can be subclassified according to the number and kinds of complements they take.

Full verbs in ACE can have zero (intransitive verbs), one (transitive verbs) or two complements (ditransitive verbs). ACE — in contrast to full English — only allows for noun phrases (*the card, a bank*) and prepositional phrases (*to the customer, on John*) as complements of full verbs. This results in the following complementation patterns for full verbs. These patterns have to be specified by the users of ACE

Intransitive Verbs

Intransitive verbs are verbs that have no complement. The intransitive verb can be a simple verb (*wait*), a compound verb (*hitch-hike*) or a phrasal verb (*give up*).

The customer waits.
The card disappears.
The plane takes off.
The tank blows up.

Transitive Verbs

Transitive verbs are verbs that take one complement called “direct object”. There are different possibilities to realise the complement:

- The verb is simple or compound, but not phrasal or prepositional, and the complement is a noun phrase.
  John inserts a card.
  The machine checks the code.
  John dry cleans the trousers.
  John proof-reads the report.
- The verb is a phrasal verb and the complement is a noun phrase.
  The employee hands out the money.
  John fills out the form.

- The verb is a prepositional verb and the complement is a prepositional phrase.
  John applies for a job.
  John calls for an inquiry.

Ditransitive Verbs

Ditransitive verbs are verbs that take two complements called “direct object” and “indirect object”, respectively. In the sentence

The employee gives the money to John.

the complement, *to John*, is the indirect object. The indirect object indicates who or what benefits from an action. The complement, *the money*, is the direct object. ACE only allows for one complementation pattern for ditransitive verbs:

- The verb takes a noun phrase as a direct object and a prepositional phrase with the preposition *to* as an indirect object:
  John gives [the money] [to the customer].
  Examples:
  offer, send, show, teach, write

ACE does not allow for the complementation pattern in which the ditransitive verb takes two noun phrases as complements (NOT: *John gives the customer the money.*). To avoid possible sources for ambiguities the indirect object has always to be indicated within a *to*-prepositional phrase.

Event or State

In addition to syntactic information, verbs in ACE have to be specified concerning certain aspects of their meaning. In ACE, verbs referring to an event have to be distinguished from verbs referring to a state. An event is a happening thought of as a single occurrence, with a definite beginning and end. A state is a state of affairs which continues over a period, and need not have a well-defined beginning or end. Examples for state verbs are *live, stay*, typical examples for event verbs are *enter, check, insert.*

The Copula Verb ‘Be’

The copula or linking verb *be* is a special verb that is predefined in the ACE vocabulary. In ACE the copula verb *be* introduces a state and is used

- to assign a property to an individual
  The credit card is valid.
  John is a customer.
- to attribute a location, time etc. to an individual
  The man is in the bank.
  The meeting is in the morning.

In both cases the verb *be* takes a complement, i.e. an obligatory element to complete the sentence. The complement can only be an adjective (*valid*), an adjective phrase (*identical to the code*), an indefinite noun phrase (*a customer*) or a prepositional phrase (*in the bank*). In all three cases we say that *be* expresses predication.

4.2.3 Adjectives

Path	Vocabulary of ACE → Content Words→ Adjectives
Aim	How to define adjectives in ACE.
Examples	*big, valid, old*
Terminology	predicative and attributive position, degree modification

Adjectives are content words that give additional information about a person or thing, such as the appearance, colour, size or other properties, e.g. *the red book.*

Position, Function and Form of Adjectives

Adjectives in ACE can be used in two positions: in front of a noun (attributive position) or after the copula *be* (predicative position):

The old customer inserts the valid credit card.	attributive
The credit card is valid.	predicative

Adjectives in attributive position function as noun modifiers (→ 3.2.2). They are optional elements of a sentence. Adjectives in predicative position function as a complement of the copula *be* (→ 4.2.2, page 51). They are necessary elements of a sentence.

ACE allows for "normal" adjectives (*valid*), compound adjectives (*one-way, two-edged, second-class*) and for two-place adjectives, i.e. adjectives that take a prepositional phrase as complement:

The number is identical to the check code.

Users have to specify which preposition the adjective requires, e.g. *good at, compatible with, identical to, similar to*. Note, that ACE does not allow that adjectives take complements like *that*-clauses (NOT: *it is true that ...*, or *to*-infinitives (NOT: *it is difficult to understand*).

ACE allows to use the past participle forms of verbs as adjectives (*painted, broken, built-up, satisfied*) both in attributive and predicative position. *The painted door is closed.*

Users have to explicitly lexicalise these forms as adjectives. Note that the use of these adjectives must not be mixed up with passive constructions which are not allowed in ACE (e.g. *The door is painted by John* is NOT allowed in ACE).

Degree Modification

Adjectives in ACE can take comparative and superlative forms to express comparison or degree, respectively. We call these adjectives 'gradable'. The basic non-graded form of the adjective is called positive form.

- positive
  John is old.
  The car is expensive.
- comparative
  John is older than Mary.
  The credit card is more expensive than the EuroCard.
- superlative
  The oldest person is a woman.
  John has the most expensive car.

For each gradable adjective the user has to enter whether the comparison is expressed by adding *-er* and *-est* to the adjective, whether the comparison is irregular (*good, better, best*) or whether the comparison is expressed by placing *more* and *most* before the adjective. Some adjectives are not gradable, i.e. do not take a comparative or superlative form (e.g. *next, first*), a case which also has to be marked by the user.

Comparatives can only be used in predicative position (NOT: *The bigger man enters a card.*). They have to be followed by *than* followed by a noun phrase (*The MasterCard is bigger than the VisaCard.*). That means, the compared objects both have to be made explicit. It is not allowed to omit the second element (NOT: *The MasterCard is bigger.*).

Superlatives can only be used in attributive position, i.e. before a noun.

The biggest man enters a card.
John is the biggest man.

Superlatives may not be used alone (NOT: *The man is the biggest.*)

Note that ACE does not allow for other forms of gradability, e.g. indicating the degree with *very* or *rather* (NOT: *very expensive*, NOT: *rather uninteresting*, NOT*: extremely hot*).

4.2.4 Adverbs

Path	Vocabulary of ACE → Content Words → Adverbs
Aim	How to use and define adverbs in ACE.
Examples	*quickly, outside, constantly*
Terminology	modification type

Adverbs are words that give extra information about when, how, where, or in what circumstances something happens, e.g. *quickly, now.*

Position, Function and Form of Adverbs

Most adverbs are formed from adjectives with the suffix *-ly*: *quick*/*quickly, constant*/*constantly,* but there are other forms of adverbs as well (*yesterday, always*). Adverbs in ACE can be simple (*simply*) or compound (*out of doors*). Recall that adverbs in ACE can be used in three positions in the sentence (→ 3.2.3.1): immediately before the verb, after complements or after the verb if there is no complement. ACE does not allow sentence initial position of adverbs. Adverbs in ACE function as adjuncts, i.e. as modifiers of verb phrases (→ 3.2.3).

Modification Types

There are several types of adverbs expressing different kinds of modification, e.g. manner modification (*carefully, manually*) or time modification (*today, soon*). Users have to assign the appropriate modification type to each adverb. Currently, ACE implements the types listed below. Where users are not sure about the classification of an adverb they can form queries as indicated in brackets.

- adverbs of **location**: [Where?]
  The library is underground. [Where is the library?]
  Examples:
  downstairs, underneath, here, there, upstairs, inland, underground
- adverbs of **direction**: [In which direction?, Where ... to?]

The seat faces forward. [In which direction does the seat face?]
Examples*:*
abroad, inwards, out

- adverbs of **time**: [When?]
  The train arrives today. [When does the train arrive?]
  Examples*:*
  today, soon, now, again, then, afterwards, nowadays, tomorrow
- adverbs of **frequency**: [How often?]
  The train arrives daily. [How often does the train arrive?]
  Examples*:*
  daily, weekly, often, normally, generally, never, occasionally
- adverbs of **duration**: [How long?]
  It lasts long. [How long does it last?]
  Examples*:*
  always, indefinitely, briefly, permanently, temporarily
- adverbs of **manner**: [How?, With what?]
  She speaks correctly. [How does she speak?]
  Examples*:*
  carefully, quietly, well, beautifully, silently

Because ACE is designed to write precise and unambiguous specifications the following types of adverbs are not allowed: adverbs of degree (*almost, absolutely, simply*), modal adverbs *(perhaps),* linking adverbs (*however*), adverbs of reason (*consequently, furthermore),* sentence adverbs *(philosophically, apparently, indeed),* negative adverbs *(barely, hardly),* focusing adverbs *(especially, only).* Note further, that adverbs in ACE can not be graded (see page 52). (NOT: *more quickly, most quickly, very quickly*) and may not take complements.

4.3 Function Words

The function words of ACE are predefined in the system, i.e. users cannot add new function words or change the existing entries. Function words of ACE are divided into the following main word classes:

- prepositions (*to, in, with, ...*)
- articles (*a, an, the*)
- personal pronouns (*he, him, she, her, it*)
- constructors (→ 4.3.4)

4.3.1 Prepositions

Path	Vocabulary of ACE → Function Words → Prepositions
Aim	How to use prepositions in ACE.
Examples	*at, for, from, from ... to, in, into, of, on, through, to, until, with*
Terminology	prepositional phrase, modification type

Basics

ACE implements the following prepositions:

at, for, from, from ... to, in, into, of, on, through, to, until, with.

In the future, extensions to this set of prepositions are planned.

Prepositions of ACE mostly combine with noun phrases (→ 3.2.1) to form a so-called prepositional phrase (*into the slot, in the hand, with John*).

Functions of Prepositional Phrases

Prepositional phrases in ACE can have three basic functions.

Prepositional Phrases as Complements

Prepositional phrases that function as complements (→ 3.2.1) are necessary elements of a sentence. Prepositional phrases can be complements

- of prepositional verbs (→ page 49)
  John applies for a job.
- of the copula *be*
  The card is in the slot. (→ page 51)
- of adjectives (→ page 52)
  The number is identical to the personal code.

In the case of prepositional verbs and adjectives the choice of the preposition is determined by the verb or adjective. In the case of the copula *be* the choice of the preposition is free.

Prepositional Phrases as Modifiers of Verbs

Prepositional phrases that function as a modifier of a verb (→ 3.2.3.2), e.g.

John inserts the card into the slot.

give additional information, e.g. the time when something happens, or the place where something exists. In ACE these prepositional phrases are always placed after the complements; more than one prepositional phrase is possible (*The train arrives in the station at 5 o'clock.*). Prepositional phrases that modify verbs are optional element of a sentence. The choice of the preposition can vary; it depends on the type of information the user wants to add. For example, in the sentence *The customer works in a library* the prepositional phrase *in the library* tells us more about the location where the customer works. In the subsection "Modification Types" below other types of prepositional verb modifiers are introduced. Note, that the use of prepositional phrases as verb modifiers is analogous to that of adverbs discussed in section 4.2.4, page 53.

Prepositional Phrases with *of* as Modifiers of Nouns

In ACE *of* is the only preposition that can be used in prepositional phrases that function as modifiers of nouns:

The customer of the bank enters a valid code.

In the sentence above the prepositional phrase *of the bank* gives more information about the customer.

Modification Types

If prepositional phrases are used as modifiers of verb phrases they can express different kinds of additional information: we distinguish different so-called modification types. Each modification type is associated with one or more queries that allows the user to isolate the corresponding information. Note that adverbs (→ 4.2.4) are used to express similar types of modification (→ Table 5 on page 56). The prepositions of ACE can be used to express the fol-

lowing modification types:

- **location** [Where?]: *in, on, at*
  The card stays in the slot.
- **origin** [From where]: *from*
  The train departs from London.
- **direction** [Where ... (to)?, In which direction?]: *to, into, through*
  The customer inserts the card into the slot.
- **time** [When?]: *in, at, on*
  The visitor arrives in the morning.
- **start**: [From when?]: *from*
  The machine operates from morning until midnight
- **end**: [Until when?]: *until*
  The machine operates until midnight.
- **duration**: [How long?]: *for, from ... to*
  The machine displays the message 'Enter your personal code.' for one minute.
- **instrument** [With what?]: *with*
  John opens the door with a key.
- **comitative** (= accompaniment) [With whom?]: *with*
  The bank employee enters the bank with a policeman.

Predefined Prepositional Phrases

In ACE there are two predefined prepositional phrases with a fixed meaning.

- *at the same time*
- *in any temporal order*

They can be used to override the default temporal order of events. *At the same* time expresses simultaneity and *in any temporal order* expresses temporal non-determinism.

John enters a card and a code at the same time.
John enters a card and a code in any temporal order.

See section 3.5.3 for a more detailed discussion of the temporal order in ACE specifications.

Comparison of Adverbs and Prepositional Phrases

In ACE, both adverbs and prepositional phrases can function as modifiers of verbs. The following table summarises the different modification types and the corresponding queries:

Table 5: Modification Types in ACE

Modification Type	Adverb	Prepositional Phrase	Query
location	*underground*	*in the slot*	*Where?*
origin	—	*from London*	*Where ... from?*
direction	*forward*	*into the slot*	*Where ... to?*
time	*now*	*in the morning*	*When?*

Table 5: Modification Types in ACE

Modification Type	Adverb	Prepositional Phrase	Query
start	—	*from Tuesday until Sunday*	*Since when?*
end	—	*until Monday*	*Until when?*
duration	*permanently*	*for one hour*	*How long?*
frequency	*daily*	—	*How often?*
instrument	—	*with a key*	*With what?*
manner	*correctly*	—	*How?*
comitative	—	*with a customer*	*With whom?*

4.3.2 Articles

Path	Vocabulary of ACE → Function Words → Articles
Aim	How to use articles in ACE
Examples	*a/an, the*
Terminology	(definite/indefinite) noun phrase

Basics

In ACE there are two kinds of articles, the indefinite article *a/an* and the definite article *the*. Articles are used in front of common nouns (→ 4.2.1.1) to form a noun phrase (→ 3.2.1). We distinguish between so-called indefinite noun phrases (*a customer*) and definite noun phrases (*the customer*).

Indefinite Article

In ACE the indefinite article *a/an* serves two purposes:

- Each indefinite noun phrases introduces a new object into the specification. E.g. the sentences
  John inserts a credit card. He enters a code. Mary inserts a credit card.
  introduce one code and two credit cards as new objects into the specification.
- Indefinite noun phrases can function as complements of the copula *be*.
  John is a customer.
  In this case the indefinite noun phrase does not introduce or refer to a new object but has a descriptive role: it gives more information about the properties of the person John.

We have already seen that the indefinite article also has an implicit quantifications reading (→ 3.3.5.1, page 28).

Definite Article

In ACE the definite article *the* has two uses:

- When a person or object is previously introduced (e.g. with an indefinite noun phrase), the definite noun phrase can be used later in the text to refer back to the same person or object

(anaphoric use). This anaphoric use of definite noun phrases is similar to that of personal pronouns discussed in section 3.5.2 and section 4.3.3.
John enters a credit card into the slot. The credit card has a valid code.

- When the person or object is not previously introduced ACE treats the definite noun phrase like an indefinite noun phrase introducing a new object into the specification.

4.3.3 Personal Pronouns

Path	Vocabulary of ACE → Function Words → Personal Pronouns
Aim	How to use personal pronouns in ACE.
Examples	*he, she, it, him, her*
Terminology	personal pronoun, anaphor

Basics

In ACE the following predefined personal pronouns are available:

he, she, it, him, her.

The pronouns *he*, *she*, *it*, *they* are used in subject position of a sentence, the pronouns *him*, *her*, *it*, *them* are used in complement position of a sentence. As ACE only uses third person singular (and soon third person plural verbs) only these third person personal pronouns are needed.

Function of Personal Pronoun

Personal pronouns are words which function as a whole noun phrase (→ 3.2.1). They are used as anaphors (→ 3.5.2.2), i.e. as substitutes or replacements for previously mentioned noun phrases. In the sentence

The customer enters the card. It is valid.

the pronoun *it* substitutes the noun phrase *the card*. *It* and *the card* refer to the same object.

Remember the principle for the anaphoric interpretation of personal pronouns discussed in section 3.5.2.

4.3.4 Constructors

Path	Vocabulary of ACE → Function Words → Constructors
Aim	List of ACE constructors.
Examples	*and, or, if … then, who, every, there is a, is not*
Terminology	constructor

As explained in section 3.3 users can build composite sentences from simpler sentences or composite phrases from simpler phrases with the help of so-called constructors. ACE distinguishes four different types of constructors summarised in Table 6. Detailed information about the use of the constructors can be found in sections 3.3.2 to 3.3.6.

Table 6: Constructors in ACE

Constructor Types		Constructors	See
coordinator	conjunction	*and*	3.3.2
	disjunction	*or*	
subordinator	relative pronoun	*who, which, that*	3.3.3
	implication	*if … then*	3.3.4
quantifier	universal quantifier	*every, for every, each, no*	3.3.5
	existential quantifier	*there is a, there is no, a*	
negator	noun phrase negator	*no*	3.3.6
	verb phrase negator	*does not, is not*	

4.3.5 Query Words

Path	Vocabulary of ACE → Function Words → Query Words
Aim	List of query words in ACE
Examples	*wh*-query words: *who, whom, when, what, how, which, whose, where, when, until when, how long, how often,* *yes/no*-query: *do/does*
Terminology	*yes/no*-query, *wh*-query

Query words in ACE are predefined. They allow for two types of query sentences (→ 3.4):

- *Yes/no*-queries (→ Table 7)
- *Wh*-queries (→ Table 8)

Yes/no-queries expect the answer *yes* or *no*. *Wh*-queries ask for a specific part of a sentence.

Table 7 lists possible *yes/no*-queries. Table 8 shows which parts can be queried with which query words. The answers generated by the system are themselves valid ACE sentences; they correspond to the original input. The parts that are queried are underlined.

Table 7: *Yes/No* Queries

Specification	Query sentence	Answer
The card is valid.	*Is the card valid?*	*Yes.*
The card is not valid.	*Is the card valid?*	*No.*
There is no valid VisaCard.	*Is there a valid VisaCard?*	*No.*
John enters a card.	*Does John enter a card?*	*No.*
The customer does not enter a valid card.	*Does a/the customer enter a valid card?*	*No.*
The customer enters a card. He types the code of the card.	*Does the customer enter a card and type the code of the card?*	*Yes.*
The customer enters a VisaCard.	*Does the customer enter a VisaCard or a MasterCard?*	*Yes.*

Table 8: *Wh*-queries

Specification	Query sentence
query subject	
John enters a card. *The card is valid.*	*Who enters a card?* *What is valid?*
query direct object	
John enters a card. *The employee checks the customer.* *John gives a card to the customer.*	*What does John enter?* *Who does the employee check?* *What does John give to the customer?*
query noun modifier	
The green card is not valid. *A Swiss customer enters a VisaCard.* *John enters a card that is not valid.* *The customer presses the key 'A'.*	*Which card is not valid?* *Which customer enters a VisaCard?* *Which card does John enter?* *Which key does the customer press?*
query prepositional object and indirect object	
John gives a card to the customer.	*Who does John give a card to?* *To whom does John give a card?*
query complements of prepositions	
John works with an old computer. *John talks with a customer.*	*What does John work with?* *With what does John work?* *Who does John talk with?* *With whom does John talk?*
query subject and complements	

Table 8: *Wh*-queries

Specification	Query sentence
John enters a card. *A customer enters a VisaCard.* *John gives a book to Mary.*	*Who enters what?* *Who gives what to whom?*
query *of*-noun phrase	
The blue card of John is valid. *The code of the green card is valid.*	*Whose card is valid?* *The code of what is valid?*
query complements of copula	
The card is not valid. *John is a customer.* *John is the manager of the bank.*	*What is the card?* *What is John?* *What is John?*
query verb phrase modifiers	
location: *Where?*	
John works underground. *The card is in the slot.*	*Where does John work?* *Where is the card?*
origin: *Where ... from? From where*?	
The train arrives from London.	*Where does the train arrive from?* *From where does the train arrive?*
direction: *Where ... to? Where ... into?*	
John inserts the card into the slot. *The machine moves forward.*	*Where does John insert the card into?* *Where does the machine move to?*
time: *When?*	
The train arrives now. *The train arrives in the morning.*	*When does the train arrive?*
start: *Since when?*	
The employee works from morning until afternoon.	*Since when does the employee work?*
end: *Until when?*	
The shop is open until midnight.	*Until when is the shop open?*
duration: *How long?*	
The machine works permanently. *The train stops for one hour.*	*How long does the machine work?* *How long does the train stop?*
frequency: *How often?*	
The milkman comes daily.	*How often does the milkman come?*

Table 8: *Wh*-queries

Specification	Query sentence
instrument: *With what?*	
John opens the door with a key.	*With what does John open the door?*
manner: *How?*	
John inserts the card correctly.	*How does John insert the card?*
comitative: *With whom?*	
The employee enters the room with a customer.	*With whom does the employee enter the room?*

5 Summary of ACE Interpretation Principles

This section summarises the interpretation principles of ACE. All principles were already described in detail in preceding sections.

5.1 Deterministic Parsing and Paraphrases

To avoid ambiguity ACE applies a collection of interpretation principles such that each sentence is parsed deterministically and thus assigned one unambiguous interpretation.

For all sentences that are accepted by the system a paraphrase is created. In the paraphrase interpretations and additions that are made by the system are indicated through brackets. After removing the brackets the paraphrase is a valid ACE text. The user either accepts the interpretation, or rephrases the input to change it (see section 6 for different strategies to reformulate a sentence).

Square brackets ('[]') in the paraphrase identify elements replacing other elements. More precisely, square brackets are used to:

- indicate the replacement of omitted elements:

  Input: *John enters a card and a code.*
  Paraphrase: *John enters a card and [enters] a code.*
- show how anaphoric pronouns and anaphoric definite noun phrases are interpreted:

  Input: *John inserts a card. It is invalid.*
  Paraphrase: *John inserts a card. [The card] is invalid.*
- replace synonyms by the primary word:

  Input: *SM checks the card.*
  Paraphrase: *[SimpleMat] checks the card.*

Curly brackets ('{}') can be optionally inserted. They show which elements are interpreted as belonging together. Curly brackets

- indicate the principle of minimal attachment:

  Input: *John enters a card with the hand.*
  Paraphrase: *John {enters a card with the hand}.*
- indicate the principle of right association:

  Input: *John inserts a card that is invalid.*
  Paraphrase: *John inserts {a card that is invalid}.*

5.2 ACE Interpretation Principles

The following is an alphabetical list of ACE interpretation principles

Accessibility Restrictions (page 38)

Noun phrases must be accessible to be referred to by anaphora.

Proper nouns are accessible from everywhere. When being referred to anaphorically they are considered to be defined before any other definite or indefinite noun phrase.

For other noun phrases the following principles apply:

- noun phrases in simple sentences are accessible for anaphoric references from subsequent sentences. The sentences can be separated by a full stop or by *and*/*or*,

- noun phrases that are in the scope of a universal quantifier, that are used in an implication (*if ... then*), or that occur within a negation are not accessible for anaphoric reference from subsequent sentences,
- noun phrases in the *if*-part of an implication are accessible for anaphoric reference in the *then*-part of the same *if-then* sentence.

Adverbial Modification: Priority of Postverbal Position (page 15)

In case of a conflict as to whether an adverb modifies the preceding or the following verb ACE uses the default that the adverb refers to the preceding verb.

Anaphors (page 35)

Personal pronouns (*he*, *she*, *it*), definite noun phrases (*the card*) and dynamic names can be used anaphorically. The following principles apply:

- Personal Pronouns as Anaphors (page 36)
  A personal pronoun always refers to the most recent accessible noun phrase that has the same number and gender. If no reference can be found an error is flagged.

  Input: *The machine checks the personal code. It is valid.*
  Paraphrase: *The machine checks the personal code. [The personal code] is valid.*
- Definite Noun Phrases as Anaphors (page 37)
  Definite noun phrases used anaphorically always refer to the most recent accessible noun phrase that is suitable, i.e. that has the same noun, at least the same adjectives, possessive noun, *of*-prepositional phrase, and apposition as the referring definite noun phrase; e.g. the definite noun phrase *John´s code* can refer to a preceding noun phrase *John´s correct code ´1234´*, while *the incorrect code* or *John´s old code* cannot. Note that the presence of relative sentences in the antecedent does currently not play a role in the determination of suitable antecedents. If no reference can be found, a definite noun phrase is treated as if it were an indefinite noun phrase introducing a new object.

  Input: *The machine checks a personal code. The code is valid.*
  Paraphrase: *The machine checks a personal code. [The personal code] is valid.*
- Dynamic Names as Anaphors (page 37)
  A dynamic name used alone always refers to the most recent accessible noun phrase in which the dynamic name occurs. In the paraphrase the dynamic name is replaced with the complete noun phrase which introduced the dynamic name, i.e. the paraphrase contains at least the same noun, the same adjectives, possessive noun, *of*-prepositional phrase, and appositions as the noun phrase introducing the dynamic name.

  Input: *A customer XYZ enters a blue card B. B is valid.*
  Paraphrase: *A customer XYZ enters a blue card B. [The blue card B] is valid.*

Note that if a noun phrase is modified by an *of*-prepositional phrases (*the card of a customer*) or by a possessive noun (*the customer's card*) users can refer back to both nouns (*customer* and *card*) separately.

Binding Hierarchy (page 19)

The binding hierarchy controls which elements of a composite sentence belong closer together than others. The following principles apply:

- Negation > Conjunction > Disjunction > Implication
  (where ">" stand for "binds stronger than")
- The binding hierarchy is only applied after the distribution principle.

The default binding hierarchy can be overridden:

- by adding commas before *and* to indicate the chosen structuring when you use *and* and *or*

  John enters a VisaCard or a MasterCard, and a personal code.

- splitting the composite sentence into several sentences

  John enters a VisaCard or a MasterCard.
  John enters a personal code.

Disjunction (page 22)

The coordinator *or* is interpreted as inclusive disjunction, i.e. the meaning is ‘one or the other or both’:

John has a salary account or a savings account.

Exclusive disjunction has to be made explicit.

The book is available and is not checked out or the book is not available and is checked out.

Distribution Principle (page 20)

If the complement of a (negated) verb consists of a coordination of phrases then the (negated) verb is distributed to each phrase. The following elements can be distributed: *is, is not*, finite full verb, *does not* + full verb, *does not*. Non-finite elements (e.g. *not* alone, adjectives alone etc.) cannot be distributed. The distribution principle applies before the principle of minimal attachment.

Input: *The card is red and green.*
Paraphrase:*The card is red and [is] green.*

Input: *The customer does not enter a card and a code.*
Paraphrase:*The customer does not enter a card and [does not enter] a code.*

Minimal Attachment of Prepositional Phrases (page 15)

- In ACE all prepositional phrases (except *of*-phrases) that are used as modifiers relate to the closest preceding verb phrase not to noun phrases.
  Input: *The customer enters a card with a code.*
  Paraphrase:*The customer {enters a card with a code}.*
- The principle of minimal attachment applies after the distribution principle.
  Input: *The customer enters a card and a code in the morning.*
  Paraphrase:*The customer enters a card and {[enters] a code in the morning}.*

Right Association (page 24)

- A relative pronoun not preceded by a coordinator relates to the rightmost noun that immediately precedes the relative pronoun.

  Input: *The customer enters the code of a card that is invalid.*
  Paraphrase: *The customer enters the code of {a card that is invalid}.*

- A relative pronoun after a coordinator relates to the same noun as the closest preceding equal relative pronoun.

  Input: *The customer enters the code of a card that is invalid and that is expired.*
  Paraphrase: *The customer enters the code of {a card that is invalid and that is expired}.*

Surface Order Determines Quantifier Scope (page 29)

- The relative scope of a quantifier corresponds to its surface position. The scope opens at the textual position of the quantified noun phrase and extends to the end of the sentence. If complete sentences are coordinated, the scope of a quantifier extends only to the end of the conjunct/disjunct containing the quantifier. We say that a textually preceding quantifier has *wide scope* with respect to a succeeding quantifier that has *narrow scope*.
- The constructors *for every* and *there is a* allow to move quantified noun phrases to sentence initial position and thus give them wide scope.

  Input: *A device controls every pump. (= There is a device that controls every pump.)*
  Rephrasing: *For every pump there is a device that controls the pump.*

Verb Phrase Coordination and Relative Sentences (page 21)

- Per default a coordinated verb phrase belongs to the main clause, not to the relative sentence.
  The customer {enters a card that is valid} and {has a code}.
  To express coordination within the relative sentence the relative pronoun has to be repeated.
  The customer enters {a card that is valid and that has a code}.
  The repeated relative pronouns refer to the same object as the initial relative pronoun.

Verbs

Verbs denote events (*insert*) and states (*possess*). The following principles apply:

- events happen when they occur in the text; they have no definite duration,
- states begin when they occur in the text and persist until they are explicitly terminated,
- the textual order of verbs determines the default temporal order of the associated events and states,
- the default temporal order can be overridden by adding cue phrases: *at the same time* indicates simultaneity, *in any temporal order* indicates non-determinism,
- adverbial phrases and prepositional phrases (except *of*-prepositional phrases) qualify the event or state denoted by the verb.

6 Style Guide

ACE supplies the means to express specifications in a concise, unambiguous and easily understandable way. However, effective use of the language ACE needs some experience. This section shows typical problems in using ACE and gives advice how to avoid them.

6.1 General Hints

Vocabulary

- Enter words carefully, avoiding typos.
- Use the commentary field in the lexicon to designate the intended use and interpretation of a word.
- Try to avoid complicated or ambiguous word forms, specifically try to avoid phrasal or prepositional verbs and use synonymous one-word verbs instead, e.g. *return* instead of *give back*.
- Define abbreviations or synonyms for frequently occurring or long words.
- Make sure you comply with the definition, use and interpretation of ACE function words.
- Only use predefined function words.
- Report problems with function words to the maintainer of the system.

Construction

- Only use complete sentences.
- Put a full-stop after declarative sentences and a question-mark after query-sentences.
- Stick to the predefined word order of ACE.
- Though the Attempto system will unravel any syntactically correct sentence, however complex, you may have problems to do so. Thus keep your sentences short and simple. Specifically
  - avoid unnecessarily complex coordinations,
  - avoid unnecessarily involved combinations of negation, quantification and coordination,
  - use several short sentences instead of one complex sentence.
- Use only active sentences; give agents for all actions.
- Do not use plural verbs and plural noun phrases since they are not yet allowed in ACE.

Interpretation

- Remember that your specification text is the only source of information; there is no hidden information in the Attempto system.
- Be as explicit as possible.
- Avoid ambiguous or misleading constructions.
- Be aware that the textual order of events and states implies their default temporal order.
- Check that the paraphrase given by the Attempto system reflects your intended interpretation. If it does not, reformulate your sentence.
- In particular, recall the ACE interpretation principles when you use:
  - relative sentences — relative pronouns relate to the rightmost noun preceding the relative sentence,
  - personal pronouns — they relate to the most recent accessible noun phrase that agrees

in number and gender,
- anaphoric definite noun phrases — they relate to the most recent accessible noun phrase with the same noun and at least the same adjectives, possessive noun, *of*-phrase and/or apposition,
- prepositional phrases — they modify the verb not then noun,
- quantified statements — the relative scope of quantifiers extends from the textual occurrence of the quantifier to the end of the sentence, or — in coordinations of sentences— to the end of the respective conjunct/disjunct,
- negation together with quantification and different coordinators — remember first to apply the principle of surface order, then the principle of distribution and finally the binding hierarchy.

6.2 Troubleshooting

After the user has entered a text the Attempto system analyses it and returns a paraphrase. Two things can go wrong: Attempto does not accept (part of) the specification, or it accepts the specification but the paraphrase does not reflect the user's intended interpretation. In both cases the text has to be modified. The following sections show typical causes for errors and give strategies for reformulation.

6.2.1 What if Attempto does not Accept the User Input?

Sentences are rejected when they contain unknown or misspelled words, or when their structure does not correspond to the ACE grammar. In each case, the Attempto system processes your input up to the first incorrect sentence, and then generates an error message. Misspelled or unknown words are listed for your convenience. If there are syntax errors, the first erroneous sentence is displayed.

However, currently no further details of syntactical errors are given. The following checklist helps you to systematically look for errors.

Do You Use Words or Word Forms that Are not in the Vocabulary?

- You have misspelled a word, either in the text or in the lexicon.
- You use function words that are not predefined.
- You use content words that you have not defined at all.
- You have incorrectly classified a content word, i.e. in the specification you use the word not in accordance to your classification, e.g. wrong word class, wrong type, wrong number or category of complements etc.

Do You use Constructions that Are Not Available in the ACE Grammar?

- Do not use other tenses than the simple present, do not use plural noun phrases or verbs, modal constructions, passive forms.
- Do not put adverbs or prepositional phrases at the beginning of the sentence.
- Always start implications with the condition (*if*) followed by the consequence (*then*).
- Do only coordinate equal phrases.
- Do not use coordination in subject position.
- Always begin a relative sentences with one of the relative pronouns *who, that* or *which.*
- Do not use common nouns without an article.

- Do not use prepositional phrases – except *of-phrases* – as modifiers of nouns; use a relative sentence instead. (NOT: *The card in the slot is invalid*. USE INSTEAD: *The card that is in the slot is invalid*.)

How to Avoid Constructions that Are Not Allowed in ACE?

- How to avoid passives?
  - Always put an agent in subject position.
    NOT: *The code is entered.* or *The code is entered by the customer.*
    USE INSTEAD: *The customer enters the code.*
- How to rephrase plural phrases?
  ACE will soon be augmented with a treatment of plurals. Currently, you can avoid plurals in the following constructions:
  - Use *every* instead of *all.*
    NOT: *All customers enter the correct code*.
    USE INSTEAD: *Every customer enters the correct code*.
  - Distribute the verb.
    NOT: *If the VisaCard and the MasterCard are defect then the customer gets no money.*
    USE INSTEAD: *If the VisaCard is defect and the MasterCard is defect then the customer gets no money.*
  - Find set or group denoting expressions (like *amount*, *group*).
    NOT: *If less than three persons apply for a course then the teacher cancels the course*.
    No appropriate alternative yet. You could use something like: *If the attendance of the course is smaller than a predefined amount then the teacher cancels the course.*
- Do not use the combination *not every.*
  - Depending on what you want to express use *there is ... not* or *no* instead.
    NOT: *Not every card is correct.*
    USE INSTEAD: *There is a card that is not correct*. OR *A card is not correct.* OR (to express that every card is incorrect) *No card is correct.*

6.2.2 What if Sentences Do Not Reflect the Intended Interpretation?

If a sentence is accepted by the system but the paraphrase does not correspond to the intended interpretation the user has to reformulate the sentence. Here are some hints.

Prepositional Phrases

Prepositional phrases used as adjuncts modify the verb not the noun. If you want to modify the noun you have to use a relative sentence. In the relative sentence you have to introduce a verb or an adjective that captures the intended meaning.

- Use a relative sentence to modify the noun (except for *of*-constructions).
  - Input: *The bank accepts every card with a Visa sign.*
    Paraphrase: *The bank {accepts every card with a Visa sign}.*

    This interpretation is not intended, since it means that the instrument of accepting the card is the Visa sign.
    Reformulation: *The bank accepts every card that has a Visa sign*.
    Paraphrase: *The bank accepts {every card that has a Visa sign}.*
  - Input: *Every customer has a personal code for the card.*
    The intended meaning is that *for the card* modifies *code* which is not possible in ACE.
    Use instead: *Every customer who has a card has a personal code that is associated with the card.* Or perhaps even: *Every customer has a card and a code.*

Anaphoric Reference

You can use proper nouns or anaphors – personal pronouns, definite noun phrases and dynamic names – to refer to a previously mentioned noun phrase. To resolve an anaphoric reference the system always chooses the most recent suitable noun phrase that has the same number and gender, and that is accessible. Sometimes you may find that this choice does not reflect your intended interpretation. The following sentences show techniques how you can manipulate the sentence structure to get the intended connection.

- Use a definite noun phrase instead of a pronoun.
  - Input: *If a credit card does not have a signature then it is invalid.*
    Paraphrase: *If a credit card does not have a signature then [the signature] is invalid.*
    Reformulation: *If a credit card does not have a signature then the credit card is invalid.*
- Use a relative sentence.
  - Input:
    If a customer pushes the key 'A' then the automatic teller prints a receipt.
    It shows the date of the transaction.
    ACE does not accept the second sentence because there is no accessible noun phrase in the first sentence to replace the personal pronoun *it*. Definite and indefinite noun phrases within implications are not accessible from succeeding sentences.
    Reformulation:
    If a customer pushes the key 'A' then the automatic teller prints a receipt that shows the date of the transaction.
  - Input:
    Every customer has a card and a personal code. The personal code is associated with the card.
    There is no anaphoric connection between *the personal code* and *a personal code*, because indefinite and definite noun phrases in the scope of *every* are not accessible from subsequent sentences.
    Reformulation:
    Every customer has a card and a personal code that is associated with the card.

Constructions with the Preposition 'of'

- It is possible to refer back to both parts of an *of*-construction separately.
  - *A student gives a demonstration of a program. The program runs on a Macintosh. The demonstration is in the Audimax.*
  - *If a staff user checks out a copy of a book then the copy is not available.*

Binding Hierarchy

If you combine *and* and *or* you have to be aware of the binding hierarchy which controls which parts belong more closely to each other. By default *and* binds stronger than *or*. However, you can override the default binding in various ways.

- Input: *The customer enters a VisaCard or a MasterCard and a personal code.*
  ACE paraphrase: *The customer enters a VisaCard or [enters] a MasterCard and [enters] a personal code.*
  ACE interprets this sentence as expressing two alternatives. The customer entering a VisaCard is the first alternative. The customer entering a MasterCard and then a personal code is the second alternative. If this interpretation does not correspond to your intention there are several possibilities for reformulation.
  - Use commas to override the binding hierarchy and explicitly show the intended struc-

turing.
The customer enters a VisaCard or a MasterCard, and a personal code.
Paraphrase: *The customer enters a VisaCard or a MasterCard, and a personal code.*
ACE interprets this as two sequential events. In the first event the customer enters a VisaCard, or alternatively, a MasterCard. In the second event the customer enters a personal code.

- Make two simpler sentences.
  The customer enters a VisaCard or a MasterCard.
  The customer enters a personal code.
- Coordinate two complete sentences with *and*.
  The customer enters a VisaCard or a MasterCard and the customer enters a personal code.
- Repeat conjuncts.
  The customer enters a VisaCard and a personal code or a MasterCard and a personal code.

How to Express Exclusive Disjunction?

In ACE *or* expresses inclusive disjunction, i.e. the sentence *If the customer works in a bank then he gets a VisaCard or a MasterCard* means that the customer can get a VisaCard or a MasterCard, or both. If you want to exclude that both cards are handed out, i.e. if you want to express exclusive disjunction, you can reformulate the sentence:

- Express the exclusive disjunction explicitly by mentioning all combinations.
  If the customer works in a bank then he gets a VisaCard and no MasterCard, or he gets a MasterCard and no VisaCard.

Relative Sentences

Relative sentences (that do not follow a coordinator) always relate to the immediately preceding the noun phrase. This principle of right association eliminates ambiguities but may in some cases conflict with the intuitive interpretation — you have to reformulate your input. We recommend the following techniques. (The underlined parts illustrate what the relative sentence is intended to relate to.)

- Split the sentence into two sentences and repeat the intended noun phrase
  Input: *John returns <u>a copy</u> of a book <u>that is damaged</u>.*
  Paraphrase: *John returns a copy of {a book that is damaged}.*
  Reformulations:
  John returns a copy of a book. The copy of the book is damaged.
  John returns a copy of a book. The copy is damaged.
  John returns a copy of a book and the copy is damaged.
- Move the relative sentence to a different position.
  Input: *The customer enters <u>a card</u> into the slot <u>which is invalid</u>.*
  Paraphrase:
  The customer enters a card into {the slot which is invalid}.
  Reformulations:
  The customer enters a card which is invalid into the slot.
  The customer enters a card into the slot. The card is invalid.
  The customer enters a card into the slot and the card is invalid.
- Avoid a relative sentence and use an adjective instead.
  Input*: The customer enters <u>a card</u> into the slot <u>which is invalid</u>.*
  Paraphrase:

The customer enters a card into {the slot which is invalid}.
Reformulation:
The customer enters an invalid card into the slot.

- Replace the constructor *every* and use an implication instead.
  Input*: Every copy of a book that has a yellow point is not available.*
  Paraphrase*:*
  Every copy of {a book that has a yellow point} is not available.
  Reformulation*:*
  If a copy of a book has a yellow point then the copy is not available.

Relative Sentences and Coordination

Verb phrases preceded by a coordinator after a relative sentence belong to the main sentences not the relative sentence.

- To make the coordinated verb phrase belong to the relative sentence repeat the relative pronoun *that*
  Input: *The customer enters a card that is valid and has a code.*
  Paraphrase: The *customer enters {a card that is valid} and has a code.*
  Reformulation: *The customer enters {a card that is valid and that has a code}.*

Quantification

Make sure you express the intended scoping when you use quantifiers. Remember that the textual order of the quantifiers corresponds to their relative scope. That means you have to change the position of the quantifiers in the sentence to change their scope. This restructuring can be done by using the constructors *there is a* and *for every* — even though this may lead to sentences that sound less natural than the original ones. Note: The scoping is not explicitly indicated in the paraphrase. Scoping problems often occur when you use a universally quantified noun phrase within a prepositional phrase, e.g. *a copy of every book*.

- Input: *John assigns a personal code to every customer.*
  Paraphrase: *John assigns a personal code to every customer.*
  ACE interpretation: "There *is at least one and the same personal code that John assigns to every customer.*"
  Reformulation to change scope, i.e. to express that every customer gets a personal code that may, or may not, coincide with that of another customer:
  For every customer John assigns a personal code to the customer.
  For every customer there is a personal code that John assigns to the customer.
- Input: *Every employee gets an extra holiday.*
  Paraphrase: *Every employee gets an extra holiday.*
  ACE interpretation: "*For every employee there is an extra - possibly different - holiday that the employee gets.*"
  Reformulation to change scope, i.e. to express that at least one extra holiday is the same for every employee:
  There is an extra holiday that every employee gets.

Quantification and Coordination

Note that in ACE a sentence like *Every tourist has a VisaCard or a MasterCard* means that some tourists have a VisaCard, others have a MasterCard, some have possibly both, i.e. all tourists do have at least one of the two types of credit cards. The sentence does not mean that every tourist has the same type of credit card. This meaning would have to be expressed by using two universally quantified statements: *Every tourist has a VisaCard or every tourist has*

a MasterCard.

Negation

Combinations of negators (*no, does not, is not*) with coordinators or quantifiers must be used carefully. We recommend to be as explicit as possible when combining these constructors.

- Negation and coordination
  Input: SimpleMat *does not accept a VisaCard or a MasterCard.*
  Paraphrase: SimpleMat *does not accept a VisaCard or [does not accept] a MasterCard.*
  In ACE the sentence does not mean that SimpleMat accepts neither a VisaCard nor a MasterCard. If you want to express this meaning you have several possibilities.
  - Replace *or* by *and*.
    SimpleMat does not accept a VisaCard and a MasterCard.
  - Repeat the negation.
    SimpleMat does not accept a VisaCard and does not accept a MasterCard.
  - Make two separate sentences.
    SimpleMat does not accept a VisaCard. SimpleMat does not accept a MasterCard.
- Negation and quantification
  When you combine negation with quantifiers take into account that first the negation (*is not* or *does not*) is distributed, then surface order determines the scope of the quantifier. Users familiar with predicate logic may notice that reformulations are possible on the basis of predicate logical identities, e.g. (*there is) no* is the same as (*for) every … not* etc.

Index

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

Y