

ADAPT: A Developmental, Asemantic, and Procedural Model for Transcoding From Verbal to Arabic Numerals

Pierre Barrouillet, Valérie Camos, and
Pierre Perruchet
Université de Bourgogne

Xavier Seron
Catholic University of Louvain

This article presents a new model of transcoding numbers from verbal to arabic form. This model, called ADAPT, is developmental, asemantic, and procedural. The authors' main proposal is that the transcoding process shifts from an algorithmic strategy to the direct retrieval from memory of digital forms. Thus, the model is evolutive, adaptive, and particularly suited to account for learning and development. ADAPT, which has been implemented in a computer program, accounts for existing and new developmental data, for new data in adolescents with mild retardation, and for the neuropsychological cases described in the literature.

How do people write numbers? Classical models assume that transcoding numbers from their verbal to their digital forms involves the construction of a semantic representation of the quantity to which the different parts of the numeral refer along with a representation of their mathematical relations (McCloskey, Caramazza, & Basili, 1985; Power & Dal Martello, 1990; Power & Longuet-Higgins, 1978). Then, through the understanding of these product and sum relations, numerals could be transcribed in their digital form. Thus, the classical view of transcoding supposes that children have an understanding of numbers, even large numbers, that precedes and underpins transcoding. But, where does this complete understanding come from in young children? Actually, in their seminal article, Power and Longuet-Higgins (1978) acknowledged that their model was not intended as a model of how children learn to count.

By contrast, we assume that transcoding numbers does not necessitate such a complex understanding. This article presents an asemantic procedural transcoding system that accounts for the way children learn to transcode numerals without accessing their meaning and for the evolution of transcoding processes through prac-

tice. We call this model ADAPT, with D, A, and P standing for developmental, asemantic, and procedural, respectively.

Models of Transcoding

Many studies have investigated the learning and use of verbal number systems (Fuson, Richards, & Briars, 1982; Siegler & Robinson, 1982). Most verbal systems contain a limited lexicon that designates only a few quantities and a syntax that governs the notation of the addition and multiplication relations that permits the representation of all the quantities. The verbal lexicon of numbers is organized into different lexical classes. In French, there are the units from *un* [one] to *neuf* [nine], which represent the basic number values corresponding to 1 to 9; the decades from *dix* [ten] to *quatre-vingt-dix* [ninety] in France and Québec or *nonante* in Belgium, which represent the basic numbers multiplied by ten; and the particulars from *onze* [eleven] to *seize* [sixteen], which represent the basic number values plus ten. This latter category corresponds approximately to the teens in English. However, the English teens *seventeen*, *eighteen*, and *nineteen* are expressed in French using the decade-unit forms *dix-sept*, *dix-huit*, and *dix-neuf*, respectively (literally, *ten-seven*, *ten-eight*, and *ten-nine*, respectively). There are also *cent* [hundred], *mille* [thousand], and *million* [million] that, although themselves representing a number value, possess the primary property of entering into syntactic relations to form sums and products with other elements of the lexicon. Thus, *two hundred* signifies the number value 2×100 , whereas *hundred and two* refers to $100 + 2$. It is possible to describe the syntax that makes it possible to generate all, and nothing other than, the acceptable verbal expressions (Hurford, 1987; Power & Longuet-Higgins, 1978).

In contrast, little work has been done into the real operation and learning of the number system written in arabic numerals (referred to as digits hereafter; Hughes, 1986; Sinclair, Siegrist, & Sinclair, 1983). There are two facts that may explain the scarcity of these developmental studies. The first is that the written decimal system is formally simple, consisting of only 10 elements (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) and a single principle, positional notation. The second is that this code is the subject of systematic tuition. Nev-

Pierre Barrouillet, Valérie Camos, and Pierre Perruchet, Department of Psychology, Université de Bourgogne, Dijon, France; Xavier Seron, Cognitive Neuropsychology Department, Catholic University of Louvain, Louvain-la-Neuve, Belgium.

This research was supported by the Action Concertée Incitative COGNITIVE of the Ministère de l'Éducation Nationale, de la Recherche et de la Technologie, France, and by the Conseil Régional de Bourgogne, France. We thank Margaret Delazer and Marie-Pascale Noël for providing us with patients' records. We also thank all the schoolteachers who conducted number dictations in their classrooms as well as the children for their participation.

Correspondence concerning this article should be addressed to Pierre Barrouillet, Université de Bourgogne, LEAD-CNRS, Faculté des Sciences Gabriel, 6 Bld Gabriel, 21000 Dijon, France, or to Valérie Camos, who is now at the Université René Descartes—Paris V, Laboratoire Cognition et Développement, 71 Avenue Edouard Vaillant, 92774 Boulogne-Billancourt Cedex, France. E-mail: barouil@u-bourgogne.fr or camos@psycho.univ-paris5.fr

ertheless, the use of positional notation raises many difficulties in children and adults.

Difficulties Involved in Transcoding Into Digits

In the field of developmental studies, Miura, Okamoto, Kim, Steere, and Fayol (1993, 1994) have shown that young Chinese, Korean, and Japanese children, whose languages facilitate the perception of decimality (*thirty-seven* is formed as *three-ten-seven*), are better able, and at a younger age, to write in digits than their American, French, or Swedish counterparts, whose languages are less transparent in this respect. Power and Dal Martello (1990), who studied Italian, and Seron, Deloche, and Noël (1992) and Seron and Fayol (1994), who studied French (in France and Belgium), have shown that between the ages of 7 and 9 years, children make mistakes when transcribing the phonological code to the digital code. These errors consist either of substitutions of digits (e.g., *one hundred and twenty-four* → 134), so-called *lexical* errors, or of additions and suppressions of 0s (e.g., *one hundred and twenty-four* → 10024 or *one thousand and six* → 106), so-called *syntactic* errors, that affect the relations among the elements forming the number. Moreover, young French children, who use the irregular denominations for seventy (*soixante-dix*, literally *sixty-ten*) and ninety (*quatre-vingt-dix*, literally *four-twenty-ten*), make more mistakes on these decades (e.g., *soixante-dix huit*, literally *sixty-ten eight* written 6018) than their Belgian counterparts, who use, respectively, *septante* for seventy and *nonante* for ninety.

These results are consistent with those obtained for patients suffering from acalculias. The syntactic errors consist of adding intermediary 0s (e.g., *three hundred and fifty thousand three hundred and two* → 350000302), even though none of the digits are changed. As far as the lexical errors are concerned, two types of errors have been observed. Certain patients systematically transcode *two* as 12 or 20 or *thirty* as 3 or 13, whereas others transcode *two* as 4 or 5 or *thirty* as 20 or 40 (Caramazza & McCloskey, 1987; Cipolotti, Butterworth, & Warrington, 1994; Deloche & Seron, 1982a, 1982b; McCloskey, 1992; McCloskey & Caramazza, 1987; Noël & Seron, 1995).

Several models of transcoding of numerals have been advanced in recent years on the basis of both neuropsychological (Deloche & Seron, 1982a, 1982b; McCloskey et al., 1985) and developmental data (Power & Dal Martello, 1990; Seron & Fayol, 1994). However, although the different models uniformly agree that lexical and syntactical information concerning numerals is processed separately, they diverge as to whether it is necessary to compute an intermediary semantic representation between the input and the output code. There thus exist two general families of transcoding models, semantic and asemantic.¹

Semantic Models

The most elaborated and influential semantic transcoding model has been proposed by McCloskey (1992; McCloskey et al., 1985). In this model, it is assumed that all numerical inputs are translated through notationally specific comprehension modules into an amodal and abstract representation of quantity. Conversely, the production of numbers requires the activation of an internal abstract representation, which is then translated into a specific output

notation via notation-specific production modules. As a result, transcoding requires the activation of an abstract semantic representation, which forms a bottleneck between the satellite comprehension and production modules. This semantic representation is assumed to specify in abstract form the basic quantities in a number and the power of 10 associated with each of them. For example, the verbal number comprehension process would generate, on the basis of the stimulus *four thousand and thirty*, the semantic representation $(4) 10 \exp 3, (3) 10 \exp 1$. As pointed out by Deloche and Seron (1987), this kind of semantic formalization matches the digit-code system organization, which is a base-10 system.

It has also been suggested that the transcoding processes involve semantic representations that capture some aspect of the linguistic lexical and semantic organization of the word-numeral system. A transcoding algorithm that operates on the basis of such a semantic verbal code has been formally described by Power and Longuet-Higgins (1978) and used as a theoretical framework by Power and Dal Martello (1990) and Seron and Fayol (1994) to interpret transcoding errors made by normal children as well as by Cipolotti et al. (1994) in a neuropsychological study. In this model (Power & Dal Martello, 1990), the semantic representations associated with 300, 1040, and 42000 are respectively $(C100 \times C3)$, $(C1000 + [C10 \times C4])$, and $(C1000 \times [(C10 \times C4) + C2])$. Transformation rules should be applied to these representations to obtain the surface forms. These rules differ for multiplicative and additive relations: a concatenation rule ($3 \& 00 = 300$) for multiplicative relations and an overwriting rule ($1000 \# 2 = 1002$) for additive relations. Each of these rules is associated with specific errors. For 7-year-old children, Power and Dal Martello (1990) mainly reported overwriting errors. For their part, Seron and Fayol (1994) identified errors relating to the multiplicative relations (e.g., *two hundred* and *six thousand* written 2100 and 61000, respectively).

Deloche and Seron's (1987) Asemantic Model

In contrast to the semantic models, the model proposed by Deloche and Seron (1987) supposes that the transcoding of numbers does not require the semantic representation of the quantity represented in the source code, irrespective of whether this code is phonemic, alphabetical, or arabic. According to Deloche and Seron (1982a, 1982b, 1987), it is possible to distinguish between four functional processing components involved in transcoding numbers from their verbal to their arabic form: parsing, categorization of primitives, transcoding, and production. The parsing process operates from left to right in the case of alphabetical code, or from the start toward the end in the case of phonological code, and

¹ There also exist some important multiple-route models, which consider that there are semantic and asemantic routes for the processing of numerals (Cipolotti, 1995; Cipolotti & Butterworth, 1995; Cohen, Dehaene, & Verstichel, 1994). However, these multiple-route models generally consider that the transcoding of numerals is usually performed via the asemantic route, whereas the semantic route is used for other kinds of arithmetic activities such as number comparison, calculation, or retrieval of encyclopedic numerical knowledge. Thus, as far as the transcoding activities are concerned, the multiple-route models can be considered as functionally equivalent to the asemantic ones.

identifies the lexical primitives through a search in a lexicon. The categorization process identifies the primitives yielded by the preceding process and the parameters necessary for the implementation of the transcoding rules. These parameters concern (a) the nature of the class to which the primitive belongs (units, teens, or decades) and (b) the position of the primitive (1st, 2nd, 3rd, 4th, etc.) in this lexical class (e.g., 12 is in the 2nd position in the lexical class of the teens). This process also identifies the word *hundred*, which triggers the implementation of specific rules, and the separators *thousand* and *million*, the identification of which controls the reiteration of the transcoding algorithm. The transcoding process itself is conceived of in the form of a set of rules that is generally activated by information relating to the class to which the lexical primitive belongs or through precise identification in the case of *hundred*, *thousand*, and *million*. The aim of these processes is to fill three-slot frames with the correct digits provided by information related to the position in the lexical class; the correct digits are entered in the correct position as specified by the transcoding rules. Finally, the encoding and production processes result in the writing down of the number in digits.

This model has the advantage of providing a functional description of the transcoding processes based on a limited number of rules. Deloche and Seron (1987) have also shown that it accounts for errors observed both in children and in mentally disturbed patients. However, this model has certain limitations that make it unsuitable for accounting for learning and developmental phenomena.

On the one hand, the application of the majority of the rules supposes that a three-slot frame has already been generated, whatever the number to be transcribed. Consequently, the model supposes that the transcription of the simplest numbers (i.e., one or two digits) requires the same knowledge as the transcription of more complex numbers. On the other hand, Deloche and Seron's (1987) model leaves little room for the learning of declarative knowledge given that a lexicon limited only to the lexical primitives can be stored in long-term memory (LTM). Now, children first learn to transcribe small numbers. The prolonged use of transcription rules for two-digit numbers, along with the direct encoding of correspondences between verbal and digit forms (e.g., *twenty-five* is written 25 or *hundred* is written 100), should create associations between the two forms in LTM.

Thus, although Deloche and Seron's (1987) asemantic model has attractive features, we do not adhere to the strong form of the hypothesis that posits the initial and systematic generation of the three-slot frame (0.0.0) irrespective of the form that is to be transcribed. The model proposed below, ADAPT, transcodes numbers from their verbal (phonological or alphabetical) to their arabic form through the use of procedures that produce and fill frames, the size of which vary during the transcoding process. These frames are filled by digit chains of varying size retrieved from LTM. More generally, this model is an evolutive model able to account for developmental and learning processes in number transcoding.

ADAPT: A Developmental, Asemantic, and Procedural Transcoding Model

The model proposed here supposes that learning proceeds (a) through the initial acquisition of elementary rules that are neces-

sary for the transcoding of small numbers (e.g., two-digit numbers), (b) through the successive addition of new rules to transcode increasingly large numbers, and (c) through the abandonment of the use of the most primitive rules in favor of processes for the direct retrieval from memory of the simplest and most frequent digital forms. This latter point results from the fact that the lexicon, which associates lexical primitives with their digital form, grows richer as the system operates.

The transition from algorithmic processes to a direct memory retrieval process is a general phenomenon within the learning processes (Logan, 1988) and constitutes one of the key points of cognitive development in general, and in the arithmetic field in particular (Barrouillet & Fayol, 1998; Compton & Logan, 1991; Logan & Klapp, 1991; Siegler, 1996). The low frequency of occurrence of the large numbers restricts the direct retrieval process to only a few forms (probably the two-digit numbers and certain special forms such as dates, the current year, etc.). However, these forms are particularly important because most of them are also involved in the transcoding of large numbers (e.g., the transcoding of *seven hundred and twenty-five* requires the transcoding of *twenty-five*).

General Overview of the Model

Like the models proposed by Deloche and Seron (1987) and Power and Dal Martello (1997), ADAPT is an asemantic model; the transcoding of a verbal form into digits and the use of positional notation do not require the representation of the quantity corresponding to the verbal numeral.² The model postulates the existence of a process for the encoding of the verbal string in a phonological form and a parsing process that segments this string into units that can be processed sequentially from the start of the string to the end by means of a production system. This production system consists of condition and action procedures. A procedure is triggered when the current contents of working memory (WM) match the conditions of application of the procedure. The contents of WM consist of declarative knowledge, which takes the form of (a) representations of the components of the situation that is to be processed (i.e., the units isolated by the parser), (b) knowledge retrieved from LTM, and (c) representations constructed by the procedures that have already been applied. We suppose that the knowledge in LTM that can be activated in WM associates certain linguistic units isolated by the parser with their digital representations (e.g., the linguistic unit *twenty-five* would be associated in LTM with its digital form 25). The WM representations constructed by the procedures consist of an ordered sequence of digits and possibly blank spaces (frames) that have to be filled by the subsequent procedures. For example, at a given moment, this representation may have the form 12 __, where __ refers to two empty slots to be filled. This process results in the construction in WM of an ordered sequence of digits corresponding to the digital form of the verbal string. Finally, this representation is thought to

² The asemantic nature of the model clearly does not mean we think it is impossible for a semantic representation to be calculated in parallel on the basis of the input form. We simply assume that in transcoding activities, the output form is not constructed on the basis of this semantic representation (see Noël & Seron, 1997, for a similar position).

be stored in a buffer prior to its subsequent transcription by the graphomotor procedures.

ADAPT is a general model for transcoding numbers. We present its application to French language. However, this model can be easily adapted to English by just deleting a few transcoding rules that process complex decades and the one deletion rule for *hundred* that are peculiar to French.³

Functioning of the Model

Temporary Storage in WM

We suppose that when individuals hear a verbal string corresponding to a number, they first have to store it in a phonological buffer. The difficulty of storage depends on the number of syllables in the string and the degree of phonological similarity between the items (Baddeley, 1986; Eriksen, Pollack, & Montague, 1970; Klapp, 1971). It is also conceivable that, given the relative frequency of the items in the number lexicon (Dehaene & Mehler, 1992), some entail a greater storage cost than others.

Construction of Internal Representations

The parsing process segments the verbal string into units that can be processed by the production system. Before the systematic learning of transcoding into digits, we assume that the parsing of spoken language results, insofar as numbers are concerned, in a set of lexical primitives consisting of units, teens, decades, and the separators *hundred* and *thousand*. At the start of learning, these primitives result in a parsing of the verbal string in which each unit contains a single word. The ADAPT^{BASIC} model (see Table 1), which writes numbers up to 99, is based on an assignment of each lexical primitive to the associated category (its lexical class). As Deloche and Seron (1987) suggested, knowledge in LTM associates the primitive with its position in the lexical class (e.g., *thirteen* is associated with the 3rd position in teens, and *forty* with the 4th position in decades). These positional values (PVs, in the model) that are retrieved from LTM are set by the procedures within the sequence of digits that is being constructed in WM (chain, in the model). The procedures (Pa to Pe; see Table 1) of ADAPT^{BASIC} are triggered by only two types of condition: the nature of the entering lexical primitive (input, in the model) and the presence of a frame (i.e., empty slots) in the sequence of digits in WM. Each of these procedures implements a series of actions out of eight possible ones:

1. "Find PV in LTM" controls the search for the positional value of the lexical primitive in its lexical class and retrieves the digit associated with it.
2. "Set PV" creates the sequence of digits by placing this digit in it or, alternatively, by placing it in the empty slot of a sequence that has already been formed.
3. "Set frame" adds an empty slot to the right of the chain of digits.
4. "Set 1" creates a sequence of digits starting with a 1.
5. "Add 1 to the number in chain" adds 1 to the digit already placed in the sequence.

Table 1
ADAPT^{BASIC}: Procedural Transcoding for Numbers up to 99

Procedure	Condition	Action
Pa	Input = unit Frame = no	Find PV in LTM Set PV in chain Stop
Pa'	Input = unit Frame = yes	Find PV in LTM Set PV in empty slot of the frame Stop
Pb	Input = dec Frame = no	Find PV in LTM Set PV in chain Set frame _ in chain Read next input
Pc	Input = dix [ten] Frame = yes	Add 1 to the number in chain Read next input
Pd	Input = teen Frame = no	Find PV in LTM Set 1 in chain Set frame _ in chain Step PV in empty slot Stop
Pd'	Input = teen Frame = yes	Find PV in LTM Add 1 to the number in chain Set PV in empty slot of the frame Stop
Pe	Input = end Frame = no	Stop
Pe'	Input = end Frame = yes	Fill empty slot with 0 Stop

Note. The procedures are grouped according to their triggering lexical primitive: a = units, b = decades, c = dix [ten], d = teens, and e = end-of-string rules. The transcoding of numbers in English requires only the Pa, Pa', Pb, Pd, Pe, and Pe' rules. unit = units; teen = teens; dec = decades; PV = positional value of the lexical primitive in its class (e.g., the PV for *thirteen* is 3, and for *forty*, it is 4); LTM = long-term memory; Frame = absence (i.e., "no") or presence (i.e., "yes") of empty slots in the chain of digits being constructed in working memory; chain (e.g., in "Set PV in chain") = chain of digits being constructed in working memory; end = end-of-string signal sent by the parser to ADAPT.

6. "Fill empty slot with 0" places a 0 in the empty slot in the sequence of digits.
7. "Read next input" enters the following lexical primitive into the transcoding system.
8. "Stop" determines the end of the construction of the chain of digits and sends it to the graphical transcription system.

However, in accordance with the recent suggestions presented by Perruchet and Vinter (1998, 2002) in their PARSER model, groups of primitives that are frequently perceived in direct temporal succession result in the formation of new representational units. These new units then guide parsing in their turn and should result in their isolation within the verbal strings that are to be segmented. Thus, with development, groups of primitives such as

³ In French, and unlike other languages such as English, when *cent* [hundred] is assigned the quantifier 1, this latter is never verbally marked and the word *cent* is not preceded by the name of any unit. For example, 134 is always verbalized as *cent trente quatre* [hundred and thirty four] and not as *un cent trente quatre* [one hundred thirty four].

quatre-vingt [eighty], *soixante-dix* [seventy], or *quatre-vingt-dix* [ninety] are represented in LTM as units and are isolated as such by the parsing process. This makes, for example, it no longer necessary to rely on the procedure Pc to deal with the appearance of the primitive *dix* [ten] after a decade in the complex decades *soixante-dix* [seventy] and *quatre-vingt-dix* [ninety]. Similarly, we can suppose that the first forms of decade–unit (DU) type (i.e., *dix-sept* [seventeen], *dix-huit* [eighteen], and *dix-neuf* [nineteen]) as well as all the decades (e.g., *trente* [thirty] and *quarante* [forty]) are, because of their frequency and early age of learning, processed as units by the parsing process and directly associated in LTM with their digital form (i.e., *dix-sept* → 17, *quarante* → 40). Their transcoding then simply requires the retrieval from LTM of the corresponding digital form instead of the application of the rules Pd for the teens or Pb and Pe' for the decades as suggested by the ADAPT^{BASIC} model.

Similarly, we can suppose that during learning many DU forms are represented as units in LTM and processed as such by the transcoding process. We have three main reasons for proposing this possibility. First, the syntax of large numbers is extremely constrained: Of the nine direct sequences possible between units (U), teens (T), and decades (D), a language like English only permits the DU sequences. Any other sequence requires the interpolation of a separator. Indeed, the sequences UU, UD, DD, or TU are not legal but the sequences UHU, UMU, UHD, UMD, DMD, THU, and TMU are, in which H stands for *hundred* and M for *thousand*. In French (as spoken in France), certain UD, DT, and DD sequences are possible, but there are only a few of these. Because DU sequences are the only ones present in the corpus, this increases the probability that they will be memorized as units. Second, numbers are learned and then written in their digital form as a function of their size. Initially, children learn to recite and then transcode numbers from 1 to 9, then from 10 to 99, and so on. As a result, children first learn the phonological and digital forms of numbers of the type DU. This primacy in learning should favor the storage of these forms in LTM. Finally, the presence of DU forms as representational units in LTM is attested to by the number comparison tasks that reveal that numbers up to at least 99 are processed as compact units (Dehaene, Dupoux, & Mehler, 1990; Reynvoet & Brysbaert, 1999).

In their study of the transcoding of numbers in 7-year-old children, Seron and Fayol (1994) observed no errors in DU, with the exception of the complex structures (DU 80 in French and Wallonian children and DU 70 and DU 90 in French children only). Even in complex forms, the errors did not affect the simplest forms. For example *soixante-dix-huit* (seventy-eight, literally *sixty-ten-eight*) was transcribed 6018 or 618 and not 60108, with the form 18 always being preserved. This suggests that these simple forms are processed correctly at an early age, probably by direct retrieval. In the same way, we suppose that the algorithmic transcription of large but frequent numbers (the current year, year of birth, certain important dates in history, etc.) leads to the association in LTM of their phonological form and their digital form. In this case, transcoding could simply be the result of direct memory retrieval.

Thus, the main proposal of the ADAPT model is that when a verbal string for transcoding corresponds to a representational unit stored in LTM, this string is processed as such, whereas its transcription is the result of the direct memory retrieval of its

digital form. Algorithmic transcoding constitutes a back-up strategy that is only used when direct memory retrieval fails (Siegler & Shrager, 1984), either because the digital form corresponding to the number for transcription is not known or because this knowledge is not accessible. Indeed, we assume that the retrieval is a probabilistic process depending on the strength of the associations stored in LTM and that this strength is lessened the rarer the form to transcode. When it is not possible for the whole verbal string to be globally processed, this algorithmic strategy then necessitates a parsing of the verbal string for transcription, which isolates the units that can be processed by the production system.

Sequential Parsing of the Verbal String

Parsing is governed by representational units stored in LTM and results in a succession of processing units that may contain several words and that trigger the production rules. These processing units can be either elements whose digital transcription can be directly retrieved from LTM or separators (*cent* [hundred] and *mille* [thousand]) that trigger the transcoding rules. When the verbal string does not correspond to a representational unit, the parsing process searches for separators. It operates hierarchically by successively isolating the parts of the string located on either side of the separator *mille* when the verbal string contains this separator, then of the separator *cent* when the substrings on either side of *mille* contain *cent*. At each stage, the parsing process for any given part of the verbal string is stopped as soon as the digital form of a chain is accessible in LTM. When the digital form of a substring is retrieved, this substring is processed by ADAPT as a processing unit, and its digital form is retrieved from LTM and temporarily stored in a buffer (in the model, WMS for working memory store). These digital forms are then placed in the chain of digits that is being constructed by the devolved procedures for the processing of the separators *cent* and *mille*.

Thus, the parsing of the string *two hundred thousand two hundred and thirty-four* by an expert will not necessarily result in the isolation of each primitive (i.e., *two / hundred / thousand / two / hundred / thirty / four*) but will rather result in a distinction between units that are transcribed, on the one hand, and separators indicating how these transcriptions should be managed, on the other. If we assume that the form *two hundred thousand two hundred and thirty-four* cannot be directly retrieved from LTM, which is a fairly safe bet, then parsing isolates the parts of the string on either side of *thousand* (i.e., *two hundred* and *two hundred and thirty-four*) and then tests sequentially whether the digital forms of *two hundred* and *two hundred and thirty-four* are available. If they are not (e.g., for *two hundred and thirty-four*), then these substrings are themselves split around the separator *hundred*, and the LTM presence of the resulting components is tested (i.e., *two* and *thirty-four*). When a substring contains no separator and is, nevertheless, not accessible in the lexicon, then this substring is itself split until the lexical primitives are identified (e.g., *thirty-four* into *thirty* and *four*). In our example, if we suppose that the digital transcriptions of the substrings *two hundred* (200), *two* (2), and *thirty-four* (34) are stored in LTM, the parsing process will isolate the following units: */two hundred / thousand / two / hundred / thirty-four/*. In summary, the algorithmic transcoding procedure used by ADAPT is a back-up strategy when direct memory retrieval fails not only for the string in its

entirety but also for the subsections of the string produced by prior stages of the parsing.

Thus, the ADAPT^{BASIC} model for writing numbers up to 99 would not simply be integrated “as is” in a more complex system that makes it possible to write larger numbers. Indeed, the functioning of ADAPT^{BASIC} results in the memorization of the forms that it produces. The direct retrieval of these forms rapidly renders the ADAPT^{BASIC} production rules obsolete, and these are then replaced by a simple rule (P1) for the retrieval of simple digital forms in a more advanced state of the model we call ADAPT^{ADV} (see Table 2). Version ADAPT^{ADV} of the model makes it possible to write all the numbers up to 999999 by supposing that the forms up to 99 give rise to direct retrieval from LTM. Some examples of the transcoding of numbers by ADAPT^{ADV} are given in Table 3.

Procedures

The procedures read the current content of WM and enter new representations in this WM or fill out the representations that already exist. As suggested by Anderson (1993), WM can contain three types of representation: (a) the elements provided by the situation, which are the verbal units selected by the parser; (b) knowledge retrieved from LTM, which relates to the digital forms of these units; and (c) representations entered in WM by the procedures, which relate to the ordered chain of digits that is currently being constructed. These three types of representations constitute the three classes of conditions that may trigger the procedures.

The condition part of the procedures. The condition part of all the procedures has the same structure with the exception of the procedure that retrieves, if they exist, the digital forms of the units isolated by the parser and enters these forms in the WMS (Procedure P1 in Table 3). This structure corresponds to the three types of knowledge that may be present in WM. The condition part evaluates (a) the nature of the verbal input element (“Input” in the tables), which can be of four types depending on whether it corresponds to an element for which there is a ready-made digital transcription in LTM, to the separator *hundred*, or to the separator *thousand* or, finally, if there is no further unit to process (i.e., processing has reached the end of the verbal string); (b) the presence of a digital form in WMS (“WMS = yes” or “WMS = no” in the tables); and (c) the presence of a frame programmed by a previously applied procedure and that now has to be completed (“Frame = yes” or “Frame = no” in the tables). The procedures can therefore be subdivided into four groups (P1, P2, P3, and P4; see Table 2) depending on the nature of the input unit, with each group itself containing four variants depending on whether the WMS contains a chain of digits that can be placed in this string and whether this string contains a frame for completion. The P4 procedures, which control the end-of-string operations, are an exception to this rule. Indeed, the P4 procedures empty the contents of the WMS or complete any frame when there is no new input to process. Procedures P4a and P4b therefore contain a stop instruction. When the application of P4c results in the completion of the digital chain (i.e., there are no further empty slots in the frame), then a stop rule would be required that is triggered by the absence of any input unit, an empty WMS, and the absence of any frame. This stop rule is not included in the model as it has no effective impact on production and therefore probably no implementation

Table 2
ADAPT^{ADV}: Procedural Transcoding for Numbers up to 999999

Procedure	Condition	Action
P1	Input = lexic	Find val in LTM Set val in WMS Read next input ^a
P2a	Input = <i>cent</i> [hundred] WMS = no Frame = no	Set 1 in chain ^b Set frame __ in chain ^c Read next input ^a
P2b	Input = <i>cent</i> WMS = yes Frame = no	Set WMS in chain ^b Clear WMS ^d Set frame __ in chain ^c Read next input ^a
P2c	Input = <i>cent</i> WMS = no Frame = yes	Set 1 left to frame __ Read next input ^a
P2d	Input = <i>cent</i> WMS = yes Frame = yes	Set WMS left to frame __ Clear WMS ^d Read next input ^a
P3a	Input = <i>mille</i> [thousand] WMS = no Frame = no	Set 1 in chain ^b Set frame ___ in chain ^c Read next input ^a
P3b	Input = <i>mille</i> WMS = yes Frame = no	Set WMS in chain ^b Clear WMS ^d Set frame ___ in chain ^c Read next input ^a
P3c	Input = <i>mille</i> WMS = no Frame = yes	Fill empty slots with 0s Set frame ___ in chain ^c Read next input ^a
P3d	Input = <i>mille</i> WMS = yes Frame = yes	Set WMS right in frame Clear WMS ^d Fill empty slots with 0s Set frame ___ in chain ^c Read next input ^a
P4a	Input = end WMS = yes Frame = no	Set WMS in chain ^b Stop
P4b	Input = end WMS = no Frame = yes	Fill empty slots with 0s Stop
P4c	Input = end WMS = yes Frame = yes	Set WMS right in frame Clear WMS ^d Read next input ^a

Note. The input condition tests the nature of the unit selected by the parser. Unless otherwise noted, the “Set” action sets digits to the right of those already in the chain of digits. The procedures P2a, P2c, P3a, and P3c, which are necessary because of the one deletion rule in French, can be removed to process numbers in English. lexic = representational unit stored in the lexicon; val = associated digital form of the representation unit stored in the lexicon; LTM = long-term memory; WMS = working memory store, which either contains a digital form (i.e., “yes”) or is empty (i.e., “no”); chain (e.g., in “Set 1 in chain”) = chain of digits being constructed in working memory; Frame = absence (i.e., “no”) or presence (i.e., “yes”) of empty slots in the chain of digits being constructed in working memory; end = end-of-string signal sent by the parser to ADAPT. ^a This action enters the following unit selected by the parser in working memory. ^b These actions set a 1 or the digital form stored in WMS in the chain of digits being constructed. ^c These actions set a frame of two or three empty slots to the right of the chain of digits. ^d This action empties the WMS.

cost. In contrast, we suppose that it routes the digital chain to a graphomotor transcription system, the operation of which does not form part of this model.

The action part of the procedures. As far as their action parts are concerned, the procedures have the following roles:

Table 3
Examples of ADAPT^{ADV} Functioning

Step	Enter	Procedure	Chain	WMS	Frame
<i>Deux mille huit cent trois (2803)</i>					
1	<i>Deux</i>	P1		2	No
2	<i>Mille</i>	P3b	2 _ _ _	No	Yes
3	<i>Huit</i>	P1	2 _ _ _	8	Yes
4	<i>Cent</i>	P2d	2 8 _ _	No	Yes
5	<i>Trois</i>	P1	2 8 _ _	3	Yes
6	End	P4c	2 8 _ 3	No	Yes
7	End	P4b	2 8 0 3	No	No
Stop					
<i>Trois cent mille (300000)</i>					
1	<i>Trois</i>	P1		3	No
2	<i>Cent</i>	P2b	3 _ _	No	Yes
3	<i>Mille</i>	P3c	300 _ _ _	No	Yes
4	End	P4b	300 000	No	No
Stop					
<i>Vingt quatre mille cent treize (24113)</i>					
1	<i>Vingt quatre</i>	P1		24	No
2	<i>Mille</i>	P3b	24 _ _ _	No	Yes
3	<i>Cent</i>	P2c	24 1 _ _	No	Yes
4	<i>Treize</i>	P1	24 1 _ _	13	Yes
5	End	P4c	24 113	No	No
Stop					
<i>Cent trois mille six cent trente neuf (103639)</i>					
1	<i>Cent</i>	P2a	1 _ _	No	Yes
2	<i>Trois</i>	P1	1 _ _	3	Yes
3	<i>Mille</i>	P3d	103 _ _ _	No	Yes
4	<i>Six</i>	P1	103 _ _ _	6	Yes
5	<i>Cent</i>	P2d	103 6 _ _	No	Yes
6	<i>Trente neuf</i>	P1	103 6 _ _	39	Yes
7	End	P4c	103 639	No	No
Stop					
<i>Seize cents (1600)</i>					
1	<i>Seize</i>	P1		16	No
2	<i>Cent</i>	P2b	16 _ _	No	Yes
3	End	P4b	1600	No	No
Stop					

Note. WMS = working memory store.

1. The retrieval from LTM of the digital forms corresponding to the units isolated by the parser, if such forms exist. Unlike ADAPT^{BASIC}, these digital forms correspond not only to the lexical primitives in their respective lexical classes but also to a sequence of digits associated with a processing unit. These sequences are designated as “val” in ADAPT^{ADV}. These forms are placed in the WMS so that they can be used directly by another procedure. Note that although a WMS is needed in the computational model, there is no need to postulate such a separate structure in the psychological model. Indeed, WM contains those LTM units that are active above threshold and readily available for processing (Cowan, 1999; Engle, Kane, & Tuholski, 1999).

2. The left-to-right concatenation of these digital forms and the planning of frames that have to be filled, either by forms that are

retrieved subsequently, if such forms exist, or by 0s. These actions result in the construction of the chain of digits in WM that, when it is considered to be complete, is placed in a buffer for its graphic transcription by graphomotor procedures.

Procedures for the retrieval from LTM. When it is necessary to parse the verbal string, each unit that results from this parsing is sent to the production system. Because parsing is controlled by representational units stored in LTM, each verbal unit that results from parsing necessarily corresponds to a representational unit whose digital form is available in the lexicon stored in LTM (“lexic” in Table 2). The identification of a representational unit by the parsing process triggers Procedure P1 in ADAPT^{ADV} (see Table 2). This rule retrieves the corresponding digital form from LTM, places it in the WMS ready for its subsequent positioning in the chain, and enters the next unit into WM.

Thus, the model permits the direct transcoding of parts of the verbal string by direct retrieval. For example, the transcoding of *two hundred and fifty thousand three hundred and seven* may start with the memory retrieval of the digital form of *two hundred and fifty* (250) if this part has already been constructed as a representational unit at this stage of development, whereas *three hundred and seven* may very well require new parsing steps. P1 places the form 250 in the WMS. The entry of *thousand* in ADAPT then triggers Procedure P3b that produces the chain 250 _ _ _ . The number and nature of the rules necessary for the transcoding of a given number therefore vary as a function of the knowledge stored in LTM.

Processing the separators. When the parsing process encounters a separator, it is sent directly to the production system, that is, without any test for the presence of a corresponding representational unit in the lexicon. Thus, the separators are not processed by Procedure P1, which can be triggered only by a positive outcome to the memory test performed by the parser. The separators are processed by special rules that form the heart of the procedural system. It should be noted, however, that the form of the words *hundred* and *thousand* can, under certain circumstances, be directly retrieved from memory. This is because the parser systematically attempts to search for the string in its entirety, or subsections of it, in LTM before performing any decomposition. Thus, the transcription of the number *one hundred* (in French, *cent* because there is the one deletion rule) can be performed through the direct retrieval of the form 100 if the parser identifies this representational unit in LTM. Otherwise, the string is broken down, and *cent* is identified as a separator that, when sent to the production system, triggers Procedure P2a (produced sequence: 1 _ _) followed by Procedure P4b (result: 100).

The separators *hundred* and *thousand* can both be processed by four procedures (P2a–P2d and P3a–P3d, respectively) depending on whether the WMS contains a digital form that has to be inserted in the chain of digits and whether this chain contains an empty frame to be filled. These procedures have two main functions. They enter the digital forms stored in the WMS in the chain and program a frame of two (in the case of *hundred*) or three slots (in the case of *thousand*) to the right of the chain that has already been formed. When the WMS is empty and there is no frame to be filled (i.e., when these separators are the first in the chain or the word *cent* is not assigned any quantifier), a 1 is placed in the chain of digits (P2a and P3a). These procedures make it unnecessary to hypothesize that a three-position frame is programmed whatever

the size of the number for transcription as Deloche and Seron's (1987) model suggests.

End-of-string procedures. Reaching the end of the verbal string triggers specific procedures that place any existing WMS content in the chain of digits and fill empty slots with 0s. As indicated above, these procedures permit the writing down of numbers whose digital form is known through a process of direct memory retrieval.

Partial transcriptions. The application of certain rules may permit direct transcription before the complete chain of digits has been formed. In particular, the rules P2a, P2b, P3a, and P3b, which manage the separators *hundred* and *thousand* if no frame is currently being filled, permit the direct transcription of the digit 1 or the content of WMS while retaining in memory the frame that they generate. For example, in the chain *twenty-four thousand three hundred and twelve*, Procedure P3b, which is triggered by the input of the word *thousand* and the presence of 24 in the WMS, may permit the direct transcription of 24 and the generation of a three-slot frame that is held in WM. This has the advantage of avoiding a WM overload but may also result in certain errors (see the case of L.R. described below).

Processes for Creation, Reinforcement, and Decay of Memory Traces

We assume that whenever the parser isolates a given verbal string for which there is no available representational unit in the lexicon (either because this representational unit does not exist or cannot be accessed), a goal is created in WM for the algorithmic transcoding of this verbal string. The creation of an item of knowledge in memory (i.e., a representational unit) then results from the association between a current goal and the outcome of the algorithmic processing used to reach it (Anderson & Lebière, 1998). Thus, any algorithmic resolution of the transcoding of the verbal string should result in the association in memory of this verbal string with the result of the processing, that is, the corresponding digital form. The strength of these memory traces should weaken with time. This strength would be increased each time the same problem is solved either algorithmically or through a process of direct memory retrieval. Thus, the mental lexicon is thought to contain only those forms whose frequency is sufficiently high to prevent them from being erased through the phenomenon of temporal decay. In the ADAPT model, this associative strength is given by the *strength accumulation equation*:

$$\text{Strength} = \sum tj^{-d},$$

where tj is the time that has passed since the j th occurrence of the item and the summation is over the n times the item has occurred (Anderson, Fincham, & Douglass, 1999). The parameter d was set at $-.50$, a value proposed by Anderson and Lebière (1998) in the ACT-R theory. The probability of a representational unit being retrieved from memory is a direct function of its strength. For the sake of simplification, the representational units whose probability of retrieval was less than $.05$ were removed from the lexicon.

However, this process does not lead solely to the association of the verbal form for transcription and its digital form. Indeed, when algorithmic solution is required (i.e., the digital form cannot be directly retrieved from memory), the parsing of the verbal string

results in the setting of subgoals that correspond to the transcription of substrings on either side of a separator. Thus, at every stage of parsing, representational units resulting from these subgoals would be created or reinforced, either by memory retrieval or through the use of the transcoding algorithm.

For example, when the digital form of *two hundred and fifty-four thousand three hundred and twenty-seven* is not accessible in memory, the current goal is to transcribe this string, which, at the end of the process, will be associated in memory with 254327. However, the first stage in parsing is to isolate the substrings *two hundred and fifty-four* and *three hundred and twenty-seven*, which constitute subgoals that are associated with the digital outcomes (i.e., 254 and 327). This recursive process results in the reinforcement or the formation of representational units that associate each verbal substring whose transcription has been planned as a subgoal with the corresponding digital form. Thus, the algorithmic transcription of *two hundred and fifty-four* can result in the following representational units:

two hundred and fifty-four → 254

two → 2

fifty-four → 54.

Whereas the verbal string *two hundred and fifty* within *two hundred and fifty-four* would not result in the formation or reinforcement of a representational unit because this string does not correspond to any subgoal generated during parsing. This process therefore makes the formation of a representational unit more likely for small numbers. Indeed, the frequency of occurrence of a number is approximately inversely proportional to its size.

Developmental Processes in ADAPT

Unlike the existing models, which describe only expert functioning, our model accounts for development. ADAPT makes it possible to understand how both the evolution of the procedural system and the increasing number of representational units in the lexicon contribute to improving performance with age and practice.

Creation of Procedures

In line with the ACT-R model, we assume that all knowledge starts out in declarative form. Thus, transcoding procedures in ADAPT derive from declarative knowledge on which analogy process applies. For example, knowing that five is transcribed 5 and twenty-five is transcribed 25, twenty-six can be transcribed 26 because six is 6. Thus, procedures result from compiling this analogy process through the creation of variables that would replace specific tokens. In our example, such compilation results in the creation of Procedure Pa' in ADAPT^{BASIC}. According to Anderson (1993), this compiling process occurs in learning by discovery but also in systematic tuition. The academic environment provides the most suited examples from which analogies can be mapped and thus facilitates learning. More advanced procedures would be constructed by coordinating already existing procedures because all the procedures of ADAPT have the same structure in their condition part. For example, Procedures P3a and

P3b for *thousand* in ADAPT^{ADV} are simply rewritings of Procedures P2a and P2b with the programming of a three-slot frame instead of a two-slot one. The creation of Procedures P2c and P2d is a minor diversification of Procedures P2a and P2b. In the first case (P2c), the writing of 1 and the production of a two-slot frame in P2a is transformed into the writing of the same 1 with the preservation of a two-slot frame on the right. The two procedures P2a and P2c produce the same local result (i.e., one part of the chain takes the form 1 _ _). An identical modification makes it possible to derive P2d from P2b. The writing of numbers above 100000 can be achieved by retaining the existing procedures and simply requires the derivation of two new procedures (i.e., P3c and P3d) when (a) the separator *thousand* is to be processed and (b) a frame has already been produced by a P2 rule. The P3c rule is obtained by adding P4b (i.e., “Fill empty slots with 0s”) to the action part, which is common to the already existing P3 rules (i.e., “Set frame _ _ _” and “Read next input”). P3d, whose conditions of activation are accumulated with the conditions of P3b and P3c, is a simple combination of the action part of these two rules. The model is therefore integrative in its psychological principles: the addition of new procedures resulting from the modifications and integration of existing procedures. However, the emergence of new rules has not been implemented in the computational version nor empirically tested.

Although it is limited to six-digit numbers, ADAPT^{ADV} could be developed beyond a million. The parsing process begins by searching for *million* and isolates the parts of the string located on either side of this separator. ADAPT^{ADV} processes then the left part, and when entering the production system, the separator *million* triggers Procedures PM, which have exactly the same structure as P3 rules for *thousand* except that they engender a six-slot instead of a three-slot frame (note that a PMa rule equivalent to P3a is not needed because there is no one deletion rule for *million* in French). The resulting chain (e.g., 23 _ _ _ _ _ _ for *twenty-three million two thousand*) is then temporarily stored in WM or written down through a partial transcription. In this latter case, the empty six-slot frame is the sole element to be stored. Then, ADAPT^{ADV} processes the remaining part of the string in a usual way, and the resulting chain is then treated as the content of the WMS and inserted in the form retrieved from the buffer by the stop rules of ADAPT^{ADV}. This process is referred to as the *buffer routine* in the computational model (see Table 4).

Evolution of the Lexicon

Second, our model accounts for the effects of the functioning of this system on the lexicon and for the use that this procedural system can make of an extension to this knowledge base. The small numbers soon form representational units that are identified as such by the parsing process and give rise to direct retrieval from memory. The number transcoding system would therefore develop through the addition of new procedures but also through the abandoning of entire sections of the production system (e.g., for numbers lower than 100), which would be replaced by memory retrieval processes. The same process would also account for the transcoding of special numbers such as the current year or the postal code. The frequency of occurrence of these special numbers would result in the formation of representational units and the *en bloc* retrieval of the corresponding chains (1789 and 2000).

Table 4
Procedures and Routine for the Separator Million in ADAPT^{ADV}

Procedure	Condition	Action
PMb	Input = million	Set WMS in chain
	WMS = yes	Clear WMS
	Frame = no	Set frame _ _ _ _ _ _ in chain
PMc	Input = million	Buffer routine
	WMS = no	Fill empty slots with 0s
	Frame = yes	Set frame _ _ _ _ _ _ in chain
PMd	Input = million	Buffer routine
	WMS = yes	Set WMS right in frame
	Frame = yes	Clear WMS
Buffer routine ^a		Fill empty slots with 0s
		Set frame _ _ _ _ _ _ in chain
		Buffer routine
		Set chain in buffer
		Clear chain
		Read next input
		ADAPT
	Set chain in WMS	
	Clear chain	
	Set buffer in chain	
	ADAPT (stop rules)	

Note. WMS = working memory store, which either contains a digital form (i.e., “yes”) or is empty (i.e., “no”); chain (e.g., “Set WMS in chain”) = chain of digits being constructed in working memory; Frame = absence (i.e., “no”) or presence (i.e., “yes”) of empty slots in the chain of digits being constructed in working memory.

^a The “ADAPT” action in this procedure indicates that the remaining part of the verbal string is sent to ADAPT, which processes it until its end. “ADAPT (stop rules)” indicates that the buffer routine refers back to ADAPT: Because the end of the string has been reached, only stop rules are triggered.

ADAPT Is Theoretically Grounded

The proposed model is based on a classical architecture, in which a production system uses knowledge stored in memory (Anderson, 1983, 1990, 1993; Anderson & Lebière, 1998). It assumes that the verbal string is kept active for processing in a phonological-type buffer, which could correspond to the articulatory loop in Baddeley’s (1990) model. Like Perruchet and Vinter’s (1998) PARSER model, the parsing process conforms to the elementary laws of memory and associative learning. In line with many models, the WM in which the digital string is constructed is thought of as the set of activated knowledge in LTM (Anderson, Reder, & Lebière, 1996; Cowan, 1995, 1999; Engle et al., 1999; Lovett, Reder, & Lebière, 1999). The three types of knowledge that can be held in WM (i.e., representations of the inputs, knowledge retrieved from LTM, and representations constructed by the procedures) determine the three types of conditions under which the procedures may be triggered. The formation of associations in LTM between verbal units and digital forms corresponds to an elementary process of learning and automatization (Logan, 1988). The reinforcement of these memory traces (Anderson, 1993; Siegler, 1996) or the multiplication of identical traces (Logan, 1988) would increase the probability of a direct retrieval with practice. ADAPT therefore supposes that the transcription of numbers into digits requires nothing other than general cognitive processes and is distinguished from other cognitive activities only in the specific nature of the procedures that are triggered and the knowledge that is mobilized.

The model is, furthermore, relatively independent of linguistic specificities and is in line with the recent doubts cast on the validity of linguistic relativism in the numerical field (Brybaert, Fias, & Noël, 1998; Saxton & Towse, 1998). Indeed, the model requires only minor modifications to account for the transcoding processes in so-called transparent languages, such as Japanese, which fully reflect the organization into powers of 10. For this language, and for numbers up to 99, such a model requires only (a) two rules identical to Pa and Pa' to process units, (b) two stop rules identical to Pe and Pe' in ADAPT^{BASIC}, and (c) one single rule to manage the equivalent of *ten*, whose only action is "Set frame _ in chain." The extreme simplicity of this model, linked to the fact that only one lexical class has to be accessed (the unit class), explains why Asian children perform better than western children at the start of learning (Miura et al., 1993, 1994). However, and despite this independence, it also accounts for differences in processing that are associated only with the linguistic form. For example, in a series of experiments, Noël and Seron (1997) have shown that *mille deux cents* (one thousand two hundred) and *douze cents* (twelve hundred; 1200) are processed differently even though an amodal semantic representation should be equivalent for the two forms. ADAPT accounts for such a difference in terms of the type and number of procedures used. For example, for *one thousand two hundred*, four procedures are required, (P3a, P1, P2d, and P4b), whereas only three are required for *twelve hundred* (P1, P2b, and P4b).

Thus, both the architecture and the functioning of ADAPT fit with theoretical accounts about the general structure of the cognitive system and its development. Moreover, the main characteristics of ADAPT allow several precise predictions.

Predictions From ADAPT

As we pointed out earlier, ADAPT is developmental, asemantic, and procedural in nature. Each of these three main characteristics permits precise predictions. First, concerning development and unlike any other transcoding models, the learning process inherent to ADAPT results in a system that uses an expanded and growing lexicon that goes far beyond the restricted set of lexical primitives. Thus, even young children should use the direct retrieval process to transcode forms that correspond to numbers, or even parts of numbers, that are frequently encountered. This retrieval process from an expanded lexicon permits precise predictions that have been tested by comparing computational data that indicate the forms that ADAPT learns first with behavioral data provided by a developmental study.

Second, ADAPT is an asemantic production system. Asemantic means that transcoding does not depend on the mathematical meaning of the verbal form but only on the lexical nature of the verbal input and the current state of the transcoding process (e.g., the presence of a frame to be filled). In consequence, the erroneous treatment of a given lexical input should result in different kinds of errors as a function of its position in the verbal string. This hypothesis has been tested in large samples of children in the learning phase and mildly retarded adolescents, thus providing us with a sufficient number of errors to permit accurate analyses.

Finally, ADAPT is a procedural system, the functioning of which is bounded by the general constraints of the cognitive architecture in which it is embedded. Thus, transcoding difficulties

experienced by young learners and mildly retarded adolescents as well as brain-damaged patients could result either from the procedural nature of the system or from general cognitive constraints. As far as the procedural nature of ADAPT is concerned, three main predictions can be proposed. First, like all production models, ADAPT predicts that the difficulty of transcoding a given number should depend on the number of processing steps its transcoding requires. Second, because ADAPT distinguishes between algorithmic and retrieval based transcoding, syntactic and lexical errors should result from the dysfunctioning of different parts of the transcoding system: Failures in algorithmic transcoding should result in syntactic errors, whereas difficulties in direct retrieval should lead to lexical errors. These predictions are tested on the errors produced by children and by mildly retarded adolescents. Third, specific degradations of the transcoding system in brain-damaged patients should correspond to the deterioration of a limited and identifiable subset of rules. Concerning the impact of general cognitive constraints on transcoding, the efficiency of the transcoding process should depend on the cognitive cost of its functioning, that is, of the rules the transcoding requires. The study of impaired functioning is especially suited to revealing this impact (i.e., in mildly retarded adolescents and brain-damaged patients).

These predictions concerning the developmental, asemantic, and procedural nature of ADAPT have been tested in different populations. A first study focused on development and involved 410 children (166 second graders and 244 third graders). A second study involved 965 adolescents with mild retardation who attended special courses at French high schools. Finally, the production of the neuropsychological cases available in the literature was analyzed within the ADAPT framework. These analyses were systematically compared with the concurrent interpretations proposed by the alternative models and tested through computational simulations.

Development

Procedural Aspects of ADAPT in Children in Their Learning Phase

Few studies have been devoted to the transcoding of numbers in children. Power and Dal Martello (1990) studied transcoding in 15 seven-year-old Italian children and observed that their transcoding errors primarily related to the sum relations that required what, in their model, they called an overwriting rule (e.g., *hundred and six*, $C100 + C6$, $\langle C100 \rangle \# \langle C6 \rangle = 100 \# 6 = 106$, written 1006 by children). In contrast, they observed no errors in product relations (e.g., *two hundred*, $C2 \times C100$, $\langle C2 \rangle \& \langle C100 \rangle = 2 \& 00 = 200$) that require a concatenation rule. However, Seron and Fayol (1994) reported errors in product relations in 20 seven-year-old French-speaking children. For example, these children transcribed *two hundred* as 2100. Seron et al. (1992) observed the same kind of errors and proposed a series of developmental stages in the transcoding of forms containing *hundred*. According to these authors, product relations are mastered, while the sum relations continue to give rise to literal transcriptions. Although providing some valuable insight about the first stages of transcoding learning and development, the data available in the literature thus appear to be inconsistent. Thus, we performed an experiment in which a

series of 92 numbers of two to four digits was dictated in a normal classroom setting to 410 young children (166 second graders, mean age = 7.6 years; 244 third graders, mean age = 8.6 years); the numbers dictated were the same as in the study with mildly retarded children (see below).

Rates of Errors and Procedures in ADAPT

The transcoding of the 92 numbers resulted in a total of 9731 errors and 935 nonresponses, that is, an error rate of 23.65% and a rate of nonresponses of 2.47%. The frequency of errors dropped drastically between second (errors plus nonresponses, 56.17%) and third graders (9.20%).

Our first prediction about the procedural nature of ADAPT was that the difficulty of transcoding a given number should depend on the number of processing steps its transcoding requires. Thus, the number of required production rules should predict the number of errors. ADAPT^{ADV} was applied to each of the 92 numbers dictated. This version of the model assumes that all the DU forms have already been stored in LTM and are thus transcribed through direct retrieval. Indeed, systematic tuition in number transcoding begins in French school during the last year of nursery school, and we assumed that even second graders should have stored DU forms in memory. However, we supposed that the difficulty inherent in complex decades in French (i.e., *soixante-dix* [seventy] and *quatre-vingt-dix* [ninety]) still necessitated an additional rule (either P1d or P1f in Table 5). This model proved to be highly predictive of children’s performance. When the entire sample of children was considered (*N* = 410), the correlation between the number of rules used by ADAPT^{ADV} and the number of errors observed was .903 (*p* < .001). This correlation reached .910 when the 16 numbers with complex decades that are specific to French were discarded.

ADAPT proved to be a better predictor than its competitors. For example, the number of concatenation and overwriting rules postulated by Power and Dal Martello’s (1990) semantic model was computed for each number dictated (numbers with complex decades peculiar to French were discarded from these analyses). This model only reached a correlation of .743 with the observed errors. Power and Dal Martello (1997) also proposed an asemantic model in which different concatenation and overwriting rules are postulated to process *hundred* on the one hand and *thousand* on the other. Although this model provided a better fit than its semantic predecessor, its predictive power remained lower than that of ADAPT (*r* = .863). The same findings were observed when second and third graders were considered separately. ADAPT provided an excellent fit with the performances observed in both groups (.873 and .844, respectively), whereas other models were less predictive (.839 and .775, respectively, for Power & Dal Martello’s, 1997, asemantic model; .722 and .667, respectively, for Power & Dal Martello’s, 1990, semantic model).

Nature of the Errors in Young Learners

To analyze the developmental changes in transcoding, we distinguished between syntactic and lexical errors. We defined pure syntactic errors as errors including all the digits other than 0, but with the addition or omission of one or more 0s (e.g., 30080 for 380, 50607 for 5607, 850 for 8500, or 79 for 709) or 1s (e.g., 5100

Table 5
ADAPT^{LD}: Procedural Transcoding for Numbers up to 9999
When the Decade–Unit Forms Are Not Available in Long-Term Memory (LTM)

Procedure	Condition	Action
P1a	Input = unit WMS = no	Find val in LTM Set val in WMS Read next input
P1b	Input = unit WMS = yes	Find val in LTM Replace 0 in WMS by val Read next input
P1c	Input = dec WMS = no	Find val in LTM Set val in WMS Read next input
P1d	Input = dec WMS = yes	Find val in LTM Add val to number in WMS Read next input
P1e	Input = teen WMS = no	Find val in LTM Set val in WMS Read next input
P1f	Input = teen WMS = yes	Find val in LTM Add val to number in WMS Read next input
P2a	Input = <i>cent</i> [hundred] WMS = no Frame = no	Set 1 in chain Set frame __ in chain Read next input
P2b	Input = <i>cent</i> WMS = yes Frame = no	Set WMS in chain Clear WMS Set frame __ in chain Read next input
P2c	Input = <i>cent</i> WMS = no Frame = yes	Set 1 left to frame __ Read next input
P2d	Input = <i>cent</i> WMS = yes Frame = yes	Set WMS left to frame __ Clear WMS Read next input
P3a	Input = <i>mille</i> [thousand] WMS = no	Set 1 in chain Set frame ___ in chain Read next input
P3b	Input = <i>mille</i> WMS = yes	Set WMS in chain Clear WMS Set frame ___ in chain Read next input
P4a	Input = end WMS = yes Frame = no	Set WMS in chain Stop
P4b	Input = end WMS = no Frame = yes	Fill empty slots with 0s Stop
P4c	Input = end WMS = yes Frame = yes	Set WMS right in frame Clear WMS Read next input

Note. Unit = units; teen = teens; dec = decades; val = associated digital form of the representation unit stored in the lexicon; WMS = working memory store, which either contains a digital form (i.e., “yes”) or is empty (i.e., “no”); chain (e.g., “Set 1 in chain”) = chain of digits being constructed in working memory; Frame = absence (i.e., “no”) or presence (i.e., “yes”) of empty slots in the chain of digits being constructed in working memory; end = end-of-string signal sent by the parser to ADAPT.

for 500 or 21005 for 205). A few, infrequent errors consisting of the elimination of a digit other than 0 were added to the preceding errors (e.g., 964 for 9364). Indeed, these latter errors also affect the magnitude of the number. We defined lexical errors as transcriptions containing the correct number of digits, but differing from the

correct response in one or more digits (e.g., 9354 or 9964 for 9364).

It should be noted that the great majority of the errors we observed were syntactic in nature. The total number of pure syntactic errors was 7149, that is, 73% of the observed errors (5938 and 1211 errors in second and third grade respectively; 76% and 64% of the observed errors), whereas the total number of pure lexical errors was only 1138, that is, 12% of the observed errors (577 and 561 errors in second and third grade respectively; 7% and 29% of the observed errors). There were also 834 mixed errors (i.e., syntactic and lexical) and 610 other errors. This pattern of results was ill-suited for testing our hypothesis of different roots for syntactic and lexical errors in this population. Indeed, the errors we classified as lexical did not exhibit the same regularity as the syntactic errors. For example, the mean number of occurrences of the most frequent syntactic error in each of the three-digit numbers was 34.5, whereas the mean rate of occurrence was only 1.50 for the most frequent lexical error. Moreover, there was in fact some doubt as to the lexical nature of these errors. For example, the two most frequent "lexical" errors in three-digit numbers were 215 for *two hundred and five* and 719 for *seven hundred and nine*, in which it is clear that the 1 refers to *hundred*, a type of error which is in fact syntactic. However, there were major differences between second and third graders, and these helped us clarify the developmental processes.

Our model predicts that errors in product relations should constitute the most primitive developmental level of transcoding. Indeed, when there is no available rule to deal with *hundred* as a separator, the parser isolates processing units for which there are corresponding digital forms that can be retrieved from LTM. Thus, if there is no production rule for *hundred* (i.e., P2b), the parser identifies *hundred* as a representational unit, and its digital form (100) is retrieved. Thus, *two hundred and three* is transcoded 21003, resulting in fully literal transcriptions. This declarative knowledge about the digital form of *cent* [hundred], and possibly of other UH forms, could induce a primitive rule for *hundred* that adds two 0s to the multiplicand. This kind of incorrect rule should in turn induce errors in the transcoding of sum relations and result in partial literal transcriptions (2003). The following step would be the constitution of a genuine transcription rule for *hundred* that adds empty frames instead of 0s (i.e., P2a and P2b).

Second graders mainly produced partial literal transcriptions or even fully literal transcriptions. In all of the three-digit and in 98% of the four-digit numbers except the UH and UM numbers (i.e., 300 or 2000), their most frequent error corresponded to a partial literal transcription in which the separators *hundred* and *thousand* and their respective multiplicands were processed as independent number words, the transcriptions of which were concatenated. Thus, as Power and Dal Martello (1990) observed, they wrote 90068 for 968, 6004 for 604, 600013 for 6013, or 300080052 for 3852. However, for the UH forms, the main error was a full literal transcription (3100, 5100, and 8100 for 300, 500, and 800, respectively), as already reported by Seron and Fayol (1994). In the UM forms, these literal transcriptions were also observed (e.g., 51000 for 5000) along with confusions with *hundred* (200 and 900 for 2000 and 9000, respectively).

These results suggest that the second graders produced errors resulting from a transcoding system that was only partially constructed. As ADAPT predicts, the children not only produced

errors in sum but also in product relations. Indeed, errors in transcoding product relations were frequent (.11 in UH forms, e.g., 8100 for *eight hundred*; .14 in the other three-digit forms, e.g., 710018 for *seven hundred and eighteen*). Moreover, they constituted a first developmental level. Indeed, these errors were mainly produced by the less skilled second graders as the comparison between the children who did and those who did not commit at least one error of the form U100 (e.g., 3100) in the three presented UH forms (300, 500, and 800) made clear. The former ($n = 34$) committed significantly more errors than the latter ($n = 132$) on the 89 remaining numbers (70.88 and 45.53 errors, respectively), $t(164) = 4.92, p < .001$. The more advanced second graders correctly transcribed product relations such as UH forms but failed to process sum relations correctly. For example, they wrote 300080052 for *three thousand eight hundred and fifty-two*. ADAPT produces these errors using incorrect P2b and P3b rules that add 0s to the multiplicand instead of empty slots to be filled (i.e., 300 or 3000 instead of 3__ or 3__ _). Thus, this incomplete and incorrect transcoding system resulted in very high global error rates in second graders for three- and four-digit numbers (57% and 67%, respectively).

These global error rates strongly decreased in third graders (5% and 13%, respectively), suggesting that the transcoding system is nearly acquired in third grade. Indeed, the most frequent errors no longer corresponded to the fully or partial literal transcriptions produced by second graders. For example, 65% of the syntactic errors produced by second graders in three-digit numbers were fully or partial literal transcriptions, whereas this rate dropped to 25% in third graders. In fact, the most frequent error in third graders resulted from the addition of a 0 after the hundreds digit in three-digit numbers (e.g., 3080 for 380) and after the thousands digit in four-digit numbers (e.g., 70510 for 7510). These low error rates and the nature of the errors suggest that they resulted from the dysfunctioning of a transcoding system in which production rules for separators have already been acquired.

In summary, literal transcriptions are predominant only in children who begin to learn to transcode large numbers. Errors in product relations constitute a first developmental step and result in fully literal transcriptions in which the digital forms of each lexical primitive are retrieved and concatenated (e.g., 310024 for *three hundred and twenty-four*). Difficulties in transcoding product relations are then overcome, while errors in sum relations are still frequent, thus resulting in partial literal transcriptions (i.e., *three hundred and twenty-four* is then transcribed 30024). However, the frequency of these literal transcriptions soon decreases (from .212 in second graders to .014 in third graders), and they are no longer predominant in third graders, who produce other types of errors (see the *Asemantic Aspects of Transcoding in Children* section, below, for a detailed analysis of third graders' errors in three-digit numbers). This developmental pattern suggests that literal transcriptions testify to a primitive developmental level of transcoding in which transcoding rules are not yet acquired.

Acquisition of the Lexicon in ADAPT

Contrary to all the other available models, the lexicon in ADAPT is not restricted to the lexical primitives but is rapidly enlarged to include the most frequent forms (i.e., small numbers and certain special forms). Their transcoding, which is initially

algorithmic, should progressively be achieved by a direct memory retrieval process. To test this hypothesis, we performed a computational simulation in which one million numbers between 1 and 999999 were presented in a written verbal code. The simulation recorded the nature of the transcoding process used (algorithm vs. direct retrieval). The results of this simulation were then compared with empirical data provided by the developmental study.

Computational Simulation

At the start of simulation, the memory register that mimics the lexicon in LTM contained only the lexical primitives: the set of units (nine forms), the particulars (from *onze* [eleven] to *seize* [sixteen] in French, six forms), and the decades (from *dix* [ten] to *quatre-vingt-dix* [ninety], nine forms). The implemented ADAPT^{ADV} version was the one presented in Table 2. Each procedure was implemented as an “if conditions then actions” rule in a Quick Basic Microsoft programming language. As explained above, production-rule conditions refer to the current state of the transcoding as well as the external situation, that is, to the verbal elements isolated by the parser. The production-rule actions create a chain of digits that constitute the output of the simulation.

However, because the DU forms were initially absent from the lexicon, an additional P1 rule was required for their algorithmic transcoding. Its condition of application was the presence of a representational unit in the lexicon when WMS already contained an element. This case occurred when the parser isolated a unit after having first isolated a decade (e.g., *five* after *forty* in transcoding *forty-five*) whose digital form had been retrieved and was located in the WMS (i.e., 40). The action of this new rule consists of replacing the final 0 in WMS with the digital form of the unit, thus transforming 40 into 45. Thus, at the start of the simulation, all the DU forms were transcribed algorithmically.

The presented numbers were randomly selected, respecting the distribution of their frequencies. On the one hand, a verbal structure was presented more frequently the fewer words it

contained (e.g., the presented structures were more likely to be of the form UH than the form UHDUMT). On the other hand, the probability of appearance of each unit, particular, and decade in the selected structure respected the values reported by Dehaene and Mehler (1992). These verbal structures were processed by the parser described in the *Sequential Parsing of the Verbal String* section. The processes of creation, reinforcement, and retrieval of memory traces were identical to those described in the *Processes for Creation, Reinforcement, and Decay of Memory Traces* section. A total of 100 simulations were performed, and in each of them, 10000 verbal forms were presented in 50 cycles of 200 numbers each. These cycles were intended to test the successive states of the system, thus mimicking training effects. At the end of each cycle, we recorded the rates of algorithmic transcoding. Naturally, the numbers whose digital forms constituted the initial state of the lexicon were excluded from these analyses because, from the very beginning, they were retrieved from memory. Thus, the following analysis of the two-digit numbers included the DU forms only. The reported values are averaged across the 100 simulations.

As expected, the rate of algorithmic solution decreased more rapidly and strongly with practice, the fewer digits the numbers for transcoding contained (see Figure 1). The relatively high rate of decrease observed in three- and four-digit numbers was due to the frequent retrieval of the HU, HT, and UH forms for three-digit numbers and the UM form for four-digit numbers (see Figure 2). The underlying principle of ADAPT is the transition from algorithmic transcoding to direct memory retrieval for the most frequent numbers, irrespective of whether they appear in isolation or as components of more complex numbers. In line with this principle, the DU forms no longer gave rise to algorithmic transcoding but were retrieved from memory at a very early stage in the simulation.

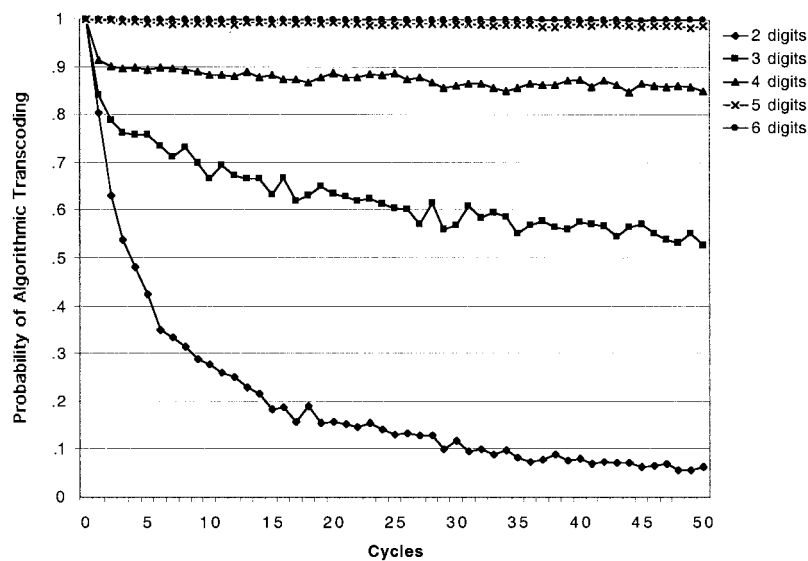


Figure 1. Probability of algorithmic transcoding by the computational simulation as a function of the number of cycles and the size of numbers.

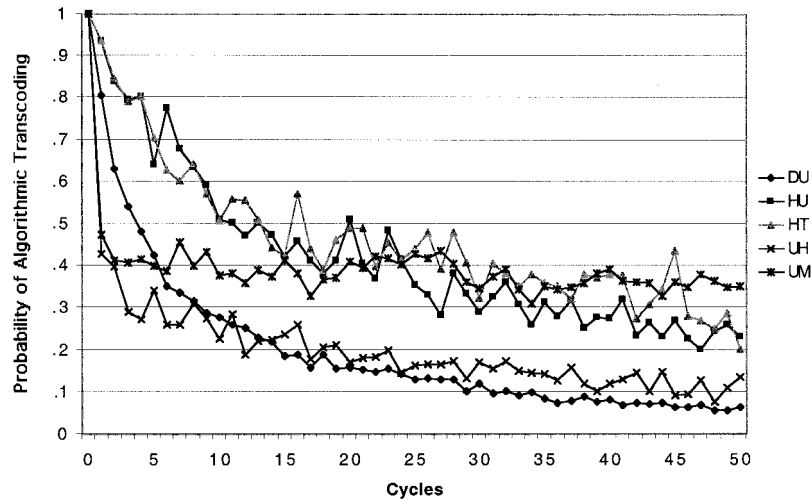


Figure 2. Probability of algorithmic transcoding by the computational simulation as a function of the number of cycles and the structure of the numbers. DU = decade unit (e.g., 46); HU = hundred unit (e.g., 103); HT = hundred teen (e.g., 114); UH = unit hundred (e.g., 400); UM = unit thousand (e.g., 2000).

Empirical Data

The results of the simulation suggest that early in development simple and frequent forms such as DU should be retrieved from LTM and then result in the same rates of errors as the transcoding of lexical primitives. By contrast, the other models predict that the transcoding of the DU forms should be more difficult than the transcoding of decades, particulars, and simple units. For example, in both their proposed models, Power and Dal Martello (1990, 1997) assumed that the transcoding of DU forms necessitates the overwriting rule to set the unit in place of the 0 of the decade. Similarly, Deloche and Seron (1987) assumed that processing the DU requires two distinct steps, one for the decade and another for the unit.

These opposed predictions have been tested using the results of our developmental study. As far as presentation in isolation is concerned, four DU (45, 52, 67, and 84) and four D forms (40, 50, 60, and 80) were dictated. In fact, DU and D forms resulted in the same error rates (.015 and .016, respectively). Among the large numbers presented, 14 three-digit and 27 four-digit numbers included either DU or D forms. Even when inserted in these large numbers, the transcoding of DU forms did not result in a significantly higher error rate than D forms (.322 and .312, respectively; $F < 1$), and this effect did not interact with the number of digits ($F < 1$). Thus, even in second and third grade children, there is no difference in transcoding difficulty between DU and D forms.

However, ADAPT permits only predictions of a null effect when DU and D forms are compared. More clear-cut and contrasted predictions are possible for large numbers ending with a DU, a D, or a U form. Indeed, according to Power and Dal Martello (1990, 1997), numbers ending with DU forms such as 9364 should elicit more errors than numbers such as 9250 or 9704 because the former require one more overwriting rule than the others in which the D and U terminal forms are lexical primitives. By contrast, ADAPT predicts that large numbers ending with U forms (e.g., 9704) should result in more errors than the others (e.g., 9250 or 9364). Indeed, although all the terminal forms are re-

trieved from memory, the former require an additional P4b rule to fill the empty slot with a 0.

We tested these predictions on the 17 dictated four-digit numbers containing a nonzero hundreds digit and ending with either a DU form (7 numbers), a D form (7 numbers), or a U form (3 numbers). The type of form had a significant effect on the error rates (.367, .393, and .428 for numbers with D, DU, and U forms, respectively), $F(1, 14) = 6.18, p = .01$. However, these results ruled out Power and Dal Martello's (1990, 1997) hypothesis. The planned comparison between numbers with DU forms, on the one hand, and those with D or U forms, on the other, was not significant ($F < 1$). By contrast, the prediction issuing from ADAPT was fully confirmed. The numbers with U forms elicited significantly more errors than numbers with DU and D forms, $F(1, 14) = 8.67, p < .01$, whereas these latter two types did not significantly differ, $F(1, 14) = 3.69, p > .05$.

In summary, all the predictions derived from the main hypothesis of a learning process in ADAPT were confirmed. Algorithmic transcoding of frequent forms results in the storage of memory traces that are reinforced by practice. This specificity of ADAPT makes our model more suitable than the others to account for the developmental data. Our results showed that as early as second grade the transcoding of DU forms is no longer any more difficult than the transcoding of lexical primitives that are directly retrieved from memory. This fact suggests that DU forms in turn become representational units and are treated as primitives by the system, even in the transcoding of larger numbers, as mimicked by the computational simulation.

Asemantic Aspects of Transcoding in Children

Apart from its procedural and developmental nature, ADAPT is also asemantic and thus differs from other transcoding models such as those proposed by Power and Dal Martello (1990) or McCloskey et al. (1985). Testing the semantic nature of the transcoding process might seem a rather complex affair. However,

the semantic hypothesis does permit one main prediction that, as we shall see, is ruled out by the empirical data.

The semantic models assume that the transcoding process involves an intermediary step in which an abstract representation of quantity is constructed. This internal abstract representation would, in turn, be used as an input by a production mechanism that translates it into a specific notation, here a digital form. Thus, these theories assume that the transcoding process depends on the quantities to which the verbal inputs refer (McCloskey, 1992; Power & Dal Martello, 1990; Power & Longuet-Higgins, 1978). For example, the word *hundred* should result in the same representation when it is inserted either in *two hundred* or in *three thousand two hundred*. Because the subsequent processing steps are based on these representations, the substring in which *hundred* appears should lead to the same local output from both numbers. This point is, for example, made clear in Power and Dal Martello's (1990) model. In the transcoding of *three thousand two hundred*, the local outputs for *three thousand* on the one hand (i.e., 3000) and *two hundred* on the other (i.e., 200) are calculated as if they appeared in isolation and are then simply overwritten. The model proposed by Power and Dal Martello (1997) presents the same characteristics. Thus, the main prediction that can be derived from these models is that the type of errors in transcoding numbers should remain unchanged when these numbers are either dictated in isolation or part of larger numbers. For example, the transcoding errors that are induced by three-digit numbers should be replicated when these three-digit numbers are part of four-digit numbers.

Conversely, ADAPT is asemantic. This means that it does not suppose that transcoding numbers necessarily requires a representation of the quantities to which the numbers refer. Instead, the successive lexical inputs selected by the parser trigger production rules that depend on the nature of this input but also on the current state of WM. For example, depending on the presence of a frame to be filled or on the presence of a digital form in WMS, a given input such as *hundred* triggers different production rules. Indeed, *hundred* triggers either P2a or P2b rules when it is embedded in three-digit numbers but P2c or P2d rules when it is embedded in four-digit numbers. As a consequence, ADAPT predicts that different errors should result from the incorrect transcoding of three-digit numbers when they are presented either in isolation or embedded in larger numbers.

These contrasting predictions have been tested in third graders only. Indeed, the above analysis of younger children's errors makes it clear that these were due to an incomplete transcoding system in which all the rules required to process *hundred* and *thousand* are not yet acquired. Conversely, the main errors in third graders were no longer the partial literal transcriptions that reveal incomplete transcoding systems, and the productions of these participants should be especially suited to testing the hypothesis in question.

As far as the 27 dictated three-digit numbers are concerned, third graders produced a total of 287 errors among which 215 were pure syntactic errors (frequency = .033). Ninety-one percent of these errors could be assigned to just three categories. The main error (70% of the syntactic errors) consisted in adding a 0 after the hundreds digit (e.g., 8015 for 815). The other two categories consisted in similarly adding either two 0s (i.e., 80015 for 815; 12% of the syntactic errors) or a 1 (i.e., 8115 for 815; 9% of the errors). We compared these errors with those that occurred in

three-digit numbers embedded in the dictated four-digit numbers ($n = 29$). Transcoding this part of the four-digit numbers resulted in the same rate of syntactic errors as in three-digit numbers dictated in isolation (.032 and .033, respectively). However, there was a major difference between the errors. Indeed, the three categories of errors observed on three-digit numbers represented only 32% of the syntactic errors observed when three-digit numbers were part of four-digit numbers. We performed an analysis of variance on the frequencies of the syntactic errors with the nature of the three-digit number (either in isolation or embedded in four-digit numbers) as a between-subjects factor, the type of error as a within-subject factor (the three categories), and the dictated numbers as the random factor. This analysis revealed that the types of errors observed in isolated three-digit numbers were significantly rarer in four-digit numbers (overall frequencies of .030 and .010, respectively), $F(1, 54) = 33.13, p < .001$. Moreover, this effect interacted with the type of category considered, $F(2, 108) = 29.42, p < .001$. The addition of a 0 after the hundreds digit was far rarer in four-digit than in isolated three-digit numbers (.006 and .023, respectively), whereas this effect was less pronounced for both the second category (adding two 0s, .003 and .004, respectively) and the third category of errors (adding a 1, .002 and .003, respectively).

Actually, four-digit numbers induced errors in the transcoding of hundreds that were never observed in three-digit numbers dictated in isolation. For example, numbers of the form UMUHU (i.e., 5607, 9704, and 8301) frequently induced (.068) the deletion of 0 in their UHU part (i.e., 607, 704, and 301 were transcribed 67, 74, and 31, respectively), an error never observed when numbers are dictated in isolation. By contrast, the main error induced by UHU forms was the addition of a 0 (i.e., 2005 for 205). In the same way, the hundreds of the UMUH forms (i.e., 2700, 4300, and 8500) were transcribed with the deletion of one or even two 0s (e.g., 700 in 2700 was transcribed either 70 or even 7, with a global frequency of .048). Of course, these errors never occurred in UH forms dictated in isolation, which instead, induced the addition of a 0 (e.g., 5000 for 500).

In summary, the pattern of errors does not fit with the major prediction that can be derived from the semantic approach. Thus, the architecture of the transcoding process is not modeled on the structure of the numerical system. Although 3200 is the sum of 3000 and 200, transcoding *three thousand two hundred* does not involve the computation and subsequent merging of the digital forms of its components *three thousand* and *two hundred*. Instead, as ADAPT suggests, transcoding is an asemantic process that sequentially processes verbal inputs. As a consequence, the processing of a given lexical primitive is constrained by the current state of the processing system.

Conclusion

In conclusion, the main predictions issuing from the procedural, developmental, and asemantic aspects of ADAPT were confirmed. The number of procedures used by ADAPT to transcode numbers is highly predictive of the error rates observed in children. It is noteworthy that this excellent fit was achieved by a version of ADAPT that retrieves DU forms from memory (ADAPT^{ADV}). Actually, empirical data confirmed that children as young as those in second grade have already shifted from algorithmic transcoding

to the direct retrieval of DU forms from memory, a shift that the computational simulation mimicked. Finally, the error patterns were at odds with a semantic view of transcoding. As ADAPT predicted, errors in forms that convey identical meanings differ according to the size of the numbers in which these forms are involved.

Although the developmental study strongly supported ADAPT's predictions, we have seen that our youngest participants had not yet acquired a complete transcoding system. Errors observed in older individuals who have acquired this system should provide us with additional and complementary evidence concerning ADAPT. Mildly retarded adolescents constitute such a population.

Adolescents With Mild Retardation

As far as we know, no study has as yet systematically investigated the errors produced by mildly retarded adolescents in number transcoding. Apart from the expected high error rates that should make it possible to gather a large corpus of errors to serve as a basis for statistical analysis, studying this population provides a critical test for ADAPT. Indeed, mental retardation and learning disabilities are associated with a difficulty in automatizing the retrieval of responses and with limited cognitive resources (Barrouillet, Fayol, & Lathulière, 1997; Geary, Brown, & Samaranayake, 1991; Geary, Widaman, Little, & Cormier, 1987). These peculiarities are of particular interest for testing our model because ADAPT involves a learning process in number transcoding and distinguishes between direct retrieval and algorithmic processing. Thus, as far as its developmental and procedural aspects are concerned, ADAPT permits specific predictions concerning this population.

First, mildly retarded adolescents should not have totally automatized the transcoding of DU forms, which would still require algorithmic transcription. Thus, a model that involves an algorithm to compute the DU forms should be highly and more predictive of the overall performance of mildly retarded adolescents than a model that retrieves these forms from LTM such as ADAPT^{ADV}. We therefore used a specific version of ADAPT (named ADAPT^{LD}) that combines procedures for writing simple numbers with procedures for the management of the separators *hundred* and *thousand*. Such a model is presented in Table 5. To Procedures P2, P3, and P4 presented above, this model adds procedures that make it possible to form chains for the numbers from 1 to 99 before they are positioned in the chain by the procedures that process the separators. Only Procedures P3a and P3b are included in this model because the numbers for transcription in this experiment did not exceed 9999. This model was compared with ADAPT^{ADV}, previously used to model performance in children.

Second, the retrieval difficulties experienced by these participants should result in frequent lexical errors. Moreover, the expected large number of lexical errors should, in turn, make it possible to test the hypothesis of different roots for lexical and syntactic errors. Lexical errors are thought to be due to difficulties in retrieving the forms from LTM and should thus be linked to the number of required retrievals (P1 rules). Conversely, syntactic errors should result from difficulties in applying the procedures that manage the positioning of the retrieved forms in LTM (P2, P3, and P4 rules). We assume that this difficulty depends mainly on the size of the frame the procedures fill. Indeed, maintaining large

frames and controlling their completion should be particularly costly for individuals with impaired cognitive resources.

Finally, as we demonstrated in normal development, the transcoding is asemantic. The analysis of the syntactic errors produced by the mildly retarded adolescents should provide additional evidence in favor of the asemantic hypothesis.

The same series of 92 numbers of two to four digits used in the developmental study was dictated to 965 mildly retarded adolescents aged between 12 and 16 years (Grades 6–9). This sample consisted of the entire population of special schools for adolescents (Section d'Enseignement Général et Professionnel Adapté) in the Côte d'Or, France. These adolescents were at least 2 years behind in their school grade and had been judged unable to attend normal school. Although, for ethical reasons, we could not access individual IQ information, all the participants suffered from mild mental deficiency (IQs ranging from 50 to 75). The number of participants studied was approximately the same in each grade.

Error Frequencies

The transcoding of the 92 numbers resulted in a total of 5816 errors and 987 nonresponses (see Table 6), that is, a mean of 63.22 errors⁴ ($SD = 38.12$) and 10.73 nonresponses ($SD = 8.54$) per number. The numbers of errors and nonresponses on the 92 numbers were highly correlated, $r(90) = .791, p < .001$.

Our first hypothesis was that the total error frequency (i.e., incorrect responses + nonresponses) should be globally determined by the number of procedures necessary to transcode a given number. ADAPT^{LD} (see Table 5), which uses algorithmic transcoding for DU forms, was applied to the 92 numbers. Each application of a P1 type rule involves one memory retrieval. The application of Procedures P1d and P1f, which are required to manage the complex decades in French (i.e., *soixante-dix* [seventy] and *quatre-vingt-dix* [ninety]), involves both a retrieval and an action on the chain that is being constructed ("Add val to number in WMS"). Consequently, Procedures P1d and P1f were assigned a weighting of 2, corresponding to their dual role, which involves both retrieval and calculation, whereas all the other procedures were assigned a weight of 1. This model proved to have a particularly high predictive power. The correlation between the number of procedures and the error frequency was .863 ($p < .001$; .835 when numbers with complex decades are removed). Thus, the more procedures the transcoding of the number requires, the greater the probability that an error will be committed. As we had observed in children, ADAPT provided a better fit for mildly retarded adolescents than its competitors. The semantic model proposed by Power and Dal Martello (1990) reached a correlation of .786 (.819 when their asemantic model was considered).

Two facts lent support to the hypothesis that mildly retarded adolescents often transcribed DU forms using the algorithmic process. First, as we predicted, ADAPT^{ADV}, which retrieves DU forms from memory, proved to be less predictive of the errors than ADAPT^{LD}, which transcodes DU forms algorithmically (correlations of .779 and .863, respectively). Second, and unlike the

⁴ Because the number of participants was very close to 1000 (965), we simply report the gross values, which can be easily converted into frequencies.

Table 6
Gross Numbers of Errors as a Function of Their Type for Each of the 92 Numbers and Overall Frequency of Errors in Adolescents With Mild Retardation

Number	Type of error						Total	No response	F
	Syntactic		Lexical		Mixed	Other			
	n	Main ^a	n	Main ^a					
40	0	—	0	—	0	2	2	0	.0021
45	5	405	1	—	1	0	7	5	.0128
50	3	500	1	—	1	2	7	2	.0096
52	3	502	3	—	0	0	6	2	.0087
60	0	—	1	—	1	0	2	2	.0043
67	7	607	8	77	1	7	23	12	.0376
70	5	610	8	60	9	0	22	6	.0301
76	17	616	13	66	5	3	38	10	.0513
80	1	—	2	—	0	1	4	3	.0079
84	9	804	7	94	3	1	20	4	.0256
90	5	810	7	80	7	1	20	3	.0245
91	20	8,011	14	81	10	0	44	11	.0588
190	14	1090	8	180	12	4	38	3	.0445
205	17	2005	7	250	2	0	26	2	.0304
250	19	2050	2	—	0	0	21	10	.0333
300	7	3000	0	—	1	1	9	2	.0124
312	37	3012	10	372	0	3	50	4	.0586
324	19	3024	8	424	3	1	31	7	.0406
380	17	3080	9	480	2	3	31	2	.0358
391	21	3091	24	381	18	3	66	9	.0801
413	25	4013	6	—	2	2	35	6	.0460
420	20	4020	4	—	2	2	28	6	.0369
457	23	4057	54	467	20	6	103	22	.1343
500	9	5000	1	—	4	0	14	3	.0181
560	32	5060	20	550	10	0	62	10	.0773
583	26	5083	16	543	11	1	54	12	.0705
604	21	6004	3	—	2	2	28	1	.0310
640	26	6040	2	—	4	4	36	9	.0481
642	32	6042	4	—	6	0	42	10	.0556
679	40	6079	15	669	24	34	113	15	.1388
709	28	7009	8	609	3	3	42	8	.0534
710	25	7010	7	510	6	4	42	1	.0466
718	28	7018	7	518	2	2	39	1	.0449
800	16	8000	1	—	3	0	20	4	.0256
815	37	8015	5	814	4	1	47	4	.0545
835	19	8035	13	635	1	1	34	6	.0434
930	42	9030	9	330	3	1	55	6	.0655
968	27	9068	24	908	9	2	62	15	.0823
970	48	9070	17	960	18	0	83	8	.0972
1040	40	10040	2	—	8	6	56	11	.0716
1054	43	10054	13	5054	3	24	83	17	.1074
1090	44	10090	18	1080	6	10	78	14	.0983
2000	24	200	3	8000	0	0	27	4	.0331
2011	41	211	2	—	2	5	50	6	.0606
2043	45	243	10	—	4	5	64	9	.0780
2534	27	20534	80	2334	22	4	133	25	.1688
2540	21	20540	39	2640	11	1	72	11	.0900
2700	29	20700	8	2077	2	25	64	5	.0741
3007	43	307	2	—	4	11	60	6	.0709
3190	27	30190	26	3090	7	2	62	19	.0876
3720	36	30720	14	3020	4	6	60	13	.0780
3852	12	30852	108	3862	10	3	133	23	.1692
4008	51	408	7	4004	4	9	71	11	.0863
4070	50	470	20	4670	10	15	95	19	.1224
4300	31	40300	11	3400	10	12	64	9	.0792
4572	14	472	117	4172	15	7	153	31	.1996
4812	21	40812	26	4612	12	1	60	13	.0820
4860	25	40860	55	4660	22	1	103	15	.1239
5000	19	500	3	—	2	2	26	1	.0303
5032	43	532	8	6032	6	4	61	10	.0760

Table 6 (continued)

Number	Type of error						Total	No response	F
	Syntactic		Lexical		Mixed	Other			
	n	Main ^a	n	Main ^a					
5080	37	580	16	6080	8	10	71	11	.0881
5185	17	50185	50	5085	6	5	78	25	.1155
5230	24	50230	21	1230	5	0	50	6	.0608
5380	20	50380	65	3380	8	2	95	16	.1243
5607	30	567	28	5707	9	11	78	6	.0944
6013	53	60013	3	2013	6	0	62	7	.0737
6020	45	620	5	6000	5	6	61	10	.0746
6050	61	650	6	6040	5	9	81	9	.0962
6054	58	654	39	6064	11	13	121	13	.1439
6278	31	678	103	6078	15	8	157	33	.2041
6315	29	60315	18	6615	5	2	54	5	.0634
6325	38	60325	27	3625	9	1	75	15	.0945
6470	24	60470	51	6490	6	6	87	40	.1351
7003	50	703	5	6003	7	6	68	8	.0815
7025	42	725	14	5025	10	4	70	8	.0876
7040	31	740	3	—	3	7	44	3	.0510
7196	34	70196	90	7096	30	7	161	24	.1968
7216	29	70216	45	7716	13	8	95	19	.1224
7510	21	70510	67	7710	9	4	101	11	.1258
7948	19	70948	86	7848	12	2	119	17	.1525
8015	58	815	6	—	3	7	74	8	.0881
8030	50	830	6	1030	2	3	61	8	.0741
8197	26	80197	96	8187	10	8	140	33	.1858
8301	58	80301	18	8001	3	23	102	5	.1143
8500	47	80500	16	9500	2	12	77	11	.0940
8719	35	80719	49	8779	13	15	112	14	.1353
9000	27	900	0	—	2	3	32	8	.0427
9060	39	90060	6	3060	5	10	60	10	.0747
9065	37	965	31	9075	10	18	96	14	.1232
9250	26	90250	47	9350	8	2	83	19	.1106
9364	31	90364	103	9354	15	10	159	44	.2132
9704	38	90704	52	9904	9	12	111	9	.1282

Note. Only those errors that were committed by at least 2 participants are reported. The overall frequency of errors is the number of incorrect responses plus the number of nonresponses. *F* = frequency.

^a This column shows the most frequently observed incorrect digital form that participants gave. Dashes indicate cases in which there was no main error (i.e., the same error was not produced by more than 1 participant).

children for whom the two error rates were equivalent, DU forms elicited more errors than D forms in the adolescents (19.8 and 5.5 errors per number, respectively). This difference just failed to reach significance, $t(6) = 2.25$, $p = .06$.

Prediction of Lexical and Syntactic Error Frequencies by ADAPT

The total number of pure syntactic errors in mildly retarded adolescents was 2601 (28.27 errors on average for each number dictated, $SD = 14.58$; 45% of the observed errors), and the number of lexical errors was 2093 (22.75 errors per number, $SD = 28.39$; 36% of the observed errors). In addition to these pure syntactic and lexical errors, there were 638 mixed errors (i.e., including both a syntactic and a lexical error) and 484 other errors. In total, the pure syntactic and lexical errors represented more than 80% of observed errors. As we predicted, mildly retarded adolescents, who experience difficulties in automatizing retrievals, were especially prone to lexical errors. Although they committed far fewer syntactic errors than third graders (syntactic error rates of .029 and .054,

respectively), $F(1, 91) = 66.54$, $p < .001$, they exhibited the same error rate for lexical errors (.025 and .024 in adolescents and third graders, respectively; $F < 1$).

Actually, the need to perform retrievals had a major impact on performance. We performed a step-by-step linear regression on the error frequency with (a) the number of memory retrievals (i.e., the number of applications of P1 type rules), (b) the number of procedures that manage frames (i.e., P2, P3, or P4 rules), and (c) the occurrence of a rule for the management of complex numbers (P1d and P1f) as independent variables. This analysis indicated that the number of memory retrievals was the factor of greatest significance in the global error frequency ($r = .753$), followed first by the need to process complex decades ($\Delta R^2 = .084$, $p < .001$) and then by the procedures ($\Delta R^2 = .112$, $p < .001$). Contrary to this result, a similar analysis performed on the error frequency observed in the previous developmental study indicated that the number of procedures required to process separators was the main factor explaining the errors in children ($r = .857$), followed by the number of memory retrievals ($\Delta R^2 = .073$, $p < .001$) and, finally,

the need to process the complex decades ($\Delta R^2 = .022, p < .01$). These results indicate that, unlike what was observed for children during learning, the main cause of errors in mildly retarded adolescents was not ignorance or the incorrect functioning of the procedures for the management of the chain of digits but difficulties in retrievals.

ADAPT predicted that lexical errors should depend on the number of required retrievals whereas syntactic errors should result from difficulties in processing frames. The numbers of lexical and syntactic errors produced for the 92 numbers were not correlated ($r = .038, p = .72$). This independence reinforces the hypothesis that the two types of error relate to different factors. In line with our hypotheses, the number of lexical errors was closely linked to the number of retrievals, that is, the number of required P1 procedures ($r = .732, p < .001$). This correlation was slightly higher ($r = .748$) when the numbers that contain complex decades that are peculiar to French were removed from the analysis. In contrast, the number of syntactic errors was not correlated with the number of retrievals ($r = .146, p > .10$).

By contrast, the syntactic errors should result from the difficulty of managing the positioning of the forms retrieved from LTM in the chain of digits that is being constructed. This difficulty is accentuated the greater the size of the frames that have to be manipulated by the production rules. We evaluated this difficulty by taking the mean size of the frames (mean Fr) manipulated by the various procedures required to transcode a given number. The rules that follow the application of a P2 rule manage the two-slot frame that this rule has just generated, whereas the rules that follow a P3 rule manage a three-slot frame. For example, transcoding the number *six thousand and thirteen* requires the rules P3b, P4c, and P4b. When P3b is triggered by *thousand*, there is no current frame (Fr = 0). Rule P3b places 6 in the chain that is currently being constructed and generates a three-slot frame. Rules P4c and P4b then manage this three-slot frame (Fr = 3, for both of these rules) by first inserting 13 and then 0. For this number, the mean Fr index of difficulty is equal to $(0 + 3 + 3)/3 = 2$. This mean Fr index was calculated for the 92 presented numbers. It was highly correlated with the number of pure syntactic errors ($r = .736, p < .001$; $.768$ when the numbers containing complex decades were removed) and much more weakly correlated with the number of lexical errors ($r = .291$; $.273$ when the numbers containing complex decades were removed). This mean Fr index proves to be a better predictor of the number of syntactic errors than the more trivial indices, such as the number of digits for transcoding ($r = .649, p < .001$). Indeed, the mean Fr index continued to be correlated with the syntactic errors even when the effect of the number of digits was partialled out ($r = .577, p < .001$). Thus, the ADAPT model is particularly effective in predicting not only the global probability of error occurrence but also the lexical or syntactic nature of these errors as a function of the type of rules the transcoding requires.

Analysis of Syntactic Errors and the Asemantic Nature of Transcoding

As we pointed out in the developmental section, semantic models predict that errors in isolated numbers should be replicated when they are inserted in larger numbers. We have seen that this prediction was ruled out by the results of the developmental study.

Syntactic errors in three-digit numbers did not correspond to the errors observed in the hundreds part of four-digit numbers. The same facts should be observed in the present study. To this end, we considered only the pure syntactic errors. We observed 675 errors of this type for the 27 dictated three-digit numbers ($M = 25$ per number). As far as three-digit numbers were concerned, the three main errors we observed in third graders were also the most frequent in mildly retarded adolescents. The addition of a 0 after the hundreds digit (e.g., 250 transcribed 2050) represented 68% of the observed syntactic errors, that is, an average of 17.0 errors of this type per number; the addition of two 0s (e.g., 250 transcribed 20050) represented 15% of the syntactic errors, that is, an average of 3.7 errors per number; and the addition of a 1 (e.g., 250 transcribed 2150) represented 4% of the syntactic errors, that is 1.1 errors per number.

As predicted by ADAPT and previously observed in children, these three types of errors were much more frequent in the three-digit than in the four-digit numbers that contain a nonzero hundreds digit (29 dictated numbers). These four-digit numbers elicited a mean of 28.3 syntactic errors. However, the addition of a 0 after the hundreds digit occurred only 2.3 times per number compared with 17.0 times in three-digit numbers, $F(1, 54) = 99.22, p < .001$. The addition of two 0s was also less frequent in four-digit than in three-digit numbers (average of 1.2 and 3.7 errors per number, respectively), $F(1, 54) = 17.06, p < .001$, whereas the addition of a 1 after the hundreds digit almost disappeared (0.3 times per number compared with 1.1 in three-digit numbers), $F(1, 54) = 6.55, p = .01$. Overall, these three types of errors that represented 87% of the syntactic errors in three-digit numbers represented only 34% of the syntactic errors observed on the hundreds part of the four-digit numbers. Thus, and contrary to the semantic hypothesis, the separator *hundred* did not elicit the same errors in three- and four-digit numbers.

The addition of one 0 after the hundreds digit (in the case of three-digit numbers) or the thousands digit (for the four-digit numbers) was the most frequent error in adolescents and was observed in the third graders in our developmental study as well as in the participants in the study by Seron et al. (1992). These authors accounted for this error in terms of the order of acquisition of transcoding rules during learning and supposed that the transcription of HU forms (e.g., 102) is overgeneralized to three-digit numbers (e.g., *one hundred and twelve* transcribed 1012). It is a simple matter to assume that the same phenomenon occurs when four-digit numbers are learned. The mildly retarded adolescents in our study might therefore be considered to exhibit a developmental retardation: The frequent addition of a 0 to the hundreds or thousands digit would result from their remaining at an immature stage of normal development.

Even though this explanation can account for some of the observed errors, it is not sufficient. Indeed, if the addition of a 0 after the hundreds digit was due to the use of an incorrect transcoding rule for *hundred*, which is generalized on the basis of HU forms, this error should occur all the more frequently when the word *hundred* appears in a three- or four-digit number. However, we have already seen that this is not the case. Thus, the appearance of an additional 0 after the hundreds digit in three-digit numbers and after the thousands digit in four-digit numbers could be linked, on the one hand, to an error in the action part of the rules responsible for programming frames (i.e., P2a, P2b, P3a, and P3b)

and, on the other hand, to errors in the procedures responsible for managing these frames. Errors in the production of frames would consist of programming a superfluous slot. The errors in frame management would, for their part, result in the incorrect positioning of the forms retrieved from LTM. According to ADAPT, the accidental programming of a four-slot frame by a P3 rule results in adding an unwanted 0 after the thousands digit. Indeed, the correct implementation of a P2c or P2d rule preserves a two-slot frame to the right of the digit it has positioned. Errors in the programming of frames are more frequent the lower the amplitude of this error is. For example, the addition of one, two, or three unwanted 0s after the thousands digit was observed 12.4, 3.2, and 1.0 times per number, respectively. It should also be noted that the frame errors affected Procedures P2 (a or b) and P3 (a or b) with comparable frequencies. Indeed, the addition of one 0 in the UH and UM forms (i.e., 500 and 2000 transcribed 5000 and 20000, respectively) was observed 8.3 and 6.7 times per number, respectively.

Thus, this pattern of error frequency suggests that the most frequent syntactic errors are due more to accidental errors in the transcoding algorithm than to the persistence of a primitive stage of normal development. The developmental study indicated that the main error in young children was literal transcriptions that resulted from a short-lived developmental stage in which there are no genuine transcoding rules for the processing of separators. By contrast, syntactic errors in mildly retarded adolescents appear to result from bugs in the application of correctly formed procedures. Indeed, when the 89 dictated forms that contained at least two lexical primitives and could thus result in literal transcriptions errors (*forty*, *fifty*, and *sixty* were discarded) were taken into account, the so-called literal transcription errors were particularly rare in our sample and reached a frequency of just .005. These literal transcriptions represented 17% of the pure syntactic errors in the mildly retarded adolescents, whereas they represented 53% and 24% of the syntactic errors in the second and third graders, respectively. Moreover, their gross frequency was far lower than in either second graders (.212), $F(1, 88) = 372.20, p < .001$, or third graders (.013), $F(1, 88) = 71.55, p < .001$.

Conclusion

The main predictions that ADAPT permitted concerning mildly retarded adolescents were confirmed. As we expected, the construction of the lexicon was impaired by their learning difficulties, and their performances in transcoding were better predicted by the ADAPT^{L-D} version, which processes DU forms by a transcoding algorithm rather than by direct retrieval. Contrary to what we observed in children, mildly retarded adolescents' performances depended mainly on the number of retrievals the transcoding required. Difficulties in retrieval processes resulted in a large number of lexical errors, the frequency of which was linked to the number of P1 rules devoted to retrieval in ADAPT^{L-D}. Similarly, these adolescents with limited cognitive resources encountered difficulties in planning and managing frames. Indeed, we demonstrated that syntactic errors depended on the cognitive cost of this processing. The analysis of the syntactic errors indicated that they are due, in mildly retarded adolescents, to bugs in the application of procedures that are otherwise probably correctly formed. Thus, the most frequent errors differ only slightly from the correct

transcription and very rarely correspond to the literal transcriptions that were frequently observed in our child participants. Finally, the pattern of errors in three- and four-digit numbers replicated the observations we made in children. Errors in transcoding hundreds differ according to the number of digits. This fact supports the asemantic nature of transcoding.

In summary, the general theoretical framework within which ADAPT was conceived takes into account the memory constraints that bear on the transcoding process and allows us to distinguish between errors that reveal a dysfunctioning within the transcoding system itself and errors due to general limitations of the cognitive system that constrain all algorithmic processes. This distinction guides our analysis of the neurological studies available in the literature.

Neurological Studies

Some neurological studies refer to patients who experience difficulty transcoding verbal numerals into arabic numerals. Noël and Seron (1995) referred to two cases reported by Singer and Low (1933), on the one hand, and McCloskey et al. (1985), on the other, but stated that these reports suffer from a lack of precision concerning the number and structure of the presented items as well as the level and type of observed errors. In contrast, three cases have been reported with a sufficient level of precision to permit an interpretation: the case of D.M. presented by Cipolotti et al. (1994), the case of L.R. presented by Noël and Seron (1995), and the case of C.K. presented by Delazer and Denes (1998). The interest of these three studies is that they contain, on the one hand, a detailed list of the errors made and, on the other, an attempt to interpret these errors on the basis of other theoretical frameworks: Seron and Deloche's (1987) model or Power and Dal Martello's (1990) model. It is therefore particularly important to examine the interpretations within the framework of these models. We then propose the interpretation provided by the ADAPT model.

These three studies reveal frequent and systematic partial literal transcription errors (e.g., *six thousand two hundred thirty* transcribed 6000230). These errors therefore differ from those made by mildly retarded adolescents who, as we have seen, produce very few literal transcriptions. This suggests that the disorders resulting from cerebral lesions are different in nature from those caused by mild retardation. The errors made by mildly retarded adolescents seem to be due to the accidental dysfunctioning of a production system that is otherwise correctly formed. Thus, these errors both are rare (e.g., the transcription of four-digit numbers resulted in only 9% of errors) and differ only slightly from the correct coding. In contrast, the errors made by patients are very frequent (e.g., 40% of errors when transcoding four-digit numbers by L.R.; 78% of errors on four-, five-, and six-digit numbers by D.M.; 100% of errors on three-digit numbers by C.K.). Moreover, they are relatively systematic and often very different from the correct transcription.

Within the ADAPT model, two main types of impairment could lie at the root of the systematic errors observed in the patients. The first is the deterioration of a component of the transcoding system: the parser, the lexicon, or the production system. As we show, D.M. and C.K. present a selective deterioration of the production system. The second is the deterioration of structures or processes

not specific to the transcoding system but necessary for its correct implementation. The case of L.R., who suffers from Alzheimer's disease, corresponds to this second type.

Disorders Specific to the Transcoding System

The Case of D.M. Reported by Cipolotti et al. (1994)

D.M. is a man aged 56 years who, after suffering a lesion of the left parietal lobe, developed an aphasia and transient disorders in the transcription of numbers while his understanding and reading of numbers remained intact (Cipolotti et al., 1994). This selective disorder resulted primarily in syntactic errors relating exclusively to numbers of four or more digits (i.e., containing *thousand*) and consisted of the addition of 0s. For example, for *three thousand two hundred*, D.M. wrote 3000200, and for *twenty-four thousand one hundred and five*, he wrote 24000105.

Interpretation of the errors according to Cipolotti et al. (1994). To account for this addition of incorrect 0s, Cipolotti et al. (1994) relied on the distinction made by Power and Dal Martello (1990) between concatenation and overwriting rules. The authors suggested that the errors made by D.M. directly correspond to the overwriting errors made by children during the learning phase. They explained that the overwriting errors occurred only for *thousand* and not *hundred* because the procedures for transcoding four-digit numbers are acquired later than those for three-digit numbers and are thus more fragile.

However, a detailed analysis of D.M.'s incorrect productions leads us to question the hypothesis of a deterioration of the overwriting rule. Indeed, 12 of the 35 errors made by D.M. consisted of the addition of three 0s to forms not requiring any overwriting. For example, *one thousand* was transcribed 1000000, *ten thousand* was transcribed 10000000, and *seventy thousand* was transcribed 70000000. It is therefore unlikely that the observed errors were due to a temporary deterioration of an overwriting rule. Instead, it seems that the occurrence of the word *thousand* systematically resulted in writing three 0s, irrespective of whether this word was located at the end of the string.

Interpretation within the ADAPT model. An initial interpretation of D.M.'s errors might suggest that the patient performs two successive transcodings. For example, when confronted with the string *thirty-two thousand and fifty*, he would initially transcode *thirty-two thousand* and then perform a second transcoding operation, independent of the first, on *fifty* (i.e., a partial literal transcoding). Although compatible with all the current transcoding models, this interpretation is unlikely because the simple concatenation of two successive transcodings should result in the arabic sequence 3200050 whereas the patient actually produced 32000050. In short, three additional 0s have been inserted into the correct transcription.

A second interpretation, which is also compatible with all the current models, would be to consider that the patient interprets *thousand* as meaning *million*. However, as Cipolotti et al. (1994) pointed out, this explanation is weakened by an observation reported by the authors. At the end of the dictation, D.M. correctly read the numbers he had written (e.g., 1000000 was read as *one million*), thus indicating that he realized that they contained more 0s than were present in the dictated forms. This suggests that the error is not one of comprehension.

Finally, a third interpretation, specific to the ADAPT model, is to suppose a specific dysfunctioning in the procedures for processing *thousand* (P3 rules) while procedures triggered by *hundred* (P2 rules) and the end of the string (P4 rules) are preserved. The activation of a P3 procedure would result in the accidental activation of another rule of P3 type by means of a partial matching process (Anderson & Lebière, 1998). For example, the transcription of *thirty-two thousand and fifty* requires the P1 rule, which retrieves 32 from LTM, and then the P3b rule, which places 32 in the chain of digits and produces a three-slot frame (i.e., 32 _ _ _). The status of the system at this moment consists of an empty WMS, the presence of a frame, and the input of *fifty*. We suggest that the P3c rule is then accidentally activated because two of its activation conditions are fulfilled ("WMS = no" and "Frame = yes"). Rule P3c fills the empty slots in the frame with 0s and generates a new three-slot frame, thus resulting in the chain 32 000 _ _ _ . The process then continues normally and results in the writing down of 32000050. The transcription of *one thousand* as 1000000 would be due to the same partial matching phenomenon, with the activation of P3a resulting in that of P3c.

The errors made by D.M. were simulated by ADAPT^{ADV}. The input of the simulation was the 35 numbers on which D.M. made an error that were reported by Cipolotti et al. (1994). The partial matching process that led to the firing of two consecutive P3 rules was mimicked by sending ADAPT^{ADV} verbal forms in which the main condition of activation of the P3 rules (i.e., the word *thousand*) was reiterated. For example, the form *three thousand and two hundred* was presented as *three thousand thousand and two hundred*. With the exception of this change, ADAPT^{ADV} ran an unaltered algorithm. In 74% of these 35 numbers, the simulation produced exactly the same outputs as D.M. As a comparison, Cipolotti et al.'s (1994) hypothesis of the dysfunctioning of an overwriting rule for *thousand* could only mimic 40% of the errors produced.

ADAPT's general theoretical framework therefore makes it possible to account for specific disorders that uniquely identify the processing of a separator because each separator activates different rules, thus avoiding the necessity of supposing the existence of a general overwriting rule that manages all the additive relations. A disorder of a comparable nature (i.e., deterioration of a particular type of rule) can be observed in the next case.

The Case of C.K. Reported by Delazer and Denes (1998)

C.K. is a woman aged 64 years who suffered a left-hemispheric ischemic stroke followed by an aphasia. At the time of her examination by Delazer and Denes (1998), she exhibited severe agraphia coupled with problems relating to the transcription of numbers. Like D.M., she primarily produced syntactic errors that gave rise to a full recovery and almost no lexical errors. The specific interest of this case is the development of the error types between the two observations (separated by a 6-month interval). During the first examination, C.K. transcribed one- or two-digit numbers without any errors and almost systematically failed on larger numbers (100%, 80%, and 100% errors for three-, four-, and five-digit numbers, respectively). The most frequent error consisted of transcribing all digits other than 0 on the left and then

adding one or two 0s (e.g., *five hundred and two* transcribed 5200 or *five hundred and twenty-three* transcribed 5230; 83% of three-digit number errors were of this type). Performance in the second examination differed from the preceding in two ways. First, the rate of correct transcriptions for three- and five-digit numbers remained practically zero (2% and 13%, respectively), whereas that for four-digit numbers was high (77%). Second, partial literal transcriptions became the most frequent error on three-digit numbers (e.g., *five hundred and twenty-three* transcribed 50023 or 5203; 72% of the errors on three-digit numbers were of this type). During a third examination conducted 2 months after the second, C.K. produced only a few errors, irrespective of the number size for transcription.

Interpretation of the errors by Delazer and Denes (1998). Delazer and Denes (1998) suggested that during the first examination C.K. generalized to all numbers the transcoding rule for two-digit numbers in which the digits other than 0 are written to the left. However, they commented that this explanation did not account for the systematic addition of 0(s) at the end of the number, thus leading C.K. to use four digits to transcribe three-digit numbers and five digits to transcribe four-digit numbers. Delazer and Denes (1998) interpreted the errors produced during the second examination as the result of the impairment of an overwriting rule. However, this interpretation is somewhat unsatisfactory given that the errors affect certain numbers and not others. Indeed, C.K. made far fewer errors on four-digit numbers (23%) than on three- or five-digit numbers (98% and 87% of errors, respectively). This fact is incompatible with any semantic model and with models that suppose that the same overwriting rule is used for three-digit numbers and the hundreds part of four-digit numbers (e.g., the two models proposed by Power and Dal Martello, 1990, 1997). It should also be noted that the idea of a variable error rate, depending on the chronology of acquisition (see the case of D.M. above), cannot be entertained here because the four-digit numbers are doubtlessly acquired later than the three-digit numbers. Thus, there was no available model that was able to account for C.K.'s errors and their evolution.

Interpretation within the ADAPT model. As we show, ADAPT accounts not only for the nature of C.K.'s errors but also for their development. Although spectacular in appearance, these errors actually result from a minor deterioration in the transcoding system, which primarily relates to the P2 type rules triggered by the separator *hundred*. This deterioration then gradually disappears.

The majority of errors observed during the initial examination can be explained in terms of a dysfunctioning of the action part of the P2 rules on the one hand and of the P4c rule on the other. All the P2 rules systematically produce a three- rather than a two-slot frame, and the P4c rule places the WMS contents not to the right but to the left of the existing frame. Thus, for *six hundred and seven*, the P2b rule produces a chain of the form 6 ____, in which the P4c rule places the 7 to the left (result: 67 __). If all the other rules are intact, the final product will be 6700. This explains why the three-digit numbers are transcribed with four digits. The same phenomenon occurs for the four-digit numbers, which are predominantly transcribed with five digits.

This disorder has been simulated using an ADAPT^{ADV} version with a lexicon containing all the DU and UH forms. Indeed, the developmental simulation suggests that adults have stored these forms in LTM. Only two changes were made: A slot was added in

the frame produced by the P2 rules, and the P4c action "set WMS right in frame" was changed to "set WMS left in frame." This simulation ran with the 86 numbers from one to four digits that Delazer and Denes (1998) presented to C.K. We did not try to simulate C.K.'s writing of the five-digit numbers (five items) because all the errors on these numbers involved also lexical errors. In 81% of the one- to four-digit numbers, the simulation produced exactly the same outputs as C.K.

The errors observed during the second examination result from a partial recovery from the disorder that we have analyzed and simulated above. Rule P4c is totally restored, as are Rules P2c and P2d. Rules P2a and P2b are also restored, but only partially. The frames they produce do not have three slots, but only two, as in normal functioning. However, the rules fill these slots with 0s. Thus, the four-digit numbers are generally correctly transcribed because they require only Rules P2c and P2d, which do not produce the frame but only manage a frame previously produced by Rule P3. Only the three-digit numbers that require Rules P2a or P2b give rise to errors, most frequently consisting of partial literal transcriptions (e.g., *one hundred and twenty-three* transcribed 10023). The modifications described above were made to the simulation, which produced the same results as C.K. for 76% of the 131 numbers of one to four digits. As in the simulation performed for the first examination, the five-digit numbers were excluded from the simulation because they were the object of lexical errors.

In summary, the cases of D.M. and C.K. fit uneasily with the hypothesis of the deterioration of an overwriting rule. As we have shown, the produced errors result from partial degradations of the transcoding system that affect only a few rules easily identified by ADAPT. The development of these neuropsychological cases reflects a gradual recovery from the initial deterioration, as the results of the computer simulations indicate. However, the systematic errors in number transcoding could also result from general factors affecting the mobilization of a transcoding system whose integrity is otherwise unimpaired.

General Disorders of the Cognitive System

Our model supposes that components specific to the transcoding system (the parser, the production system, or the lexicon) use nonspecific components of the cognitive architecture, such as short-term phonological memory or WM. Thus, any impairment of these components could have an impact on the transcoding process. This seems to be the case in patient L.R., whose case was reported by Noël and Seron (1995).

The Case of L.R. Reported by Noël and Seron (1995)

L.R. is a man aged 71 years at the time of the neurological examination whose diagnostic profile indicates Alzheimer's disease. L.R. suffers from memory disorders that affect short-term memory (STM) in particular (digit span of four; Wechsler Adult Intelligence Scale [Wechsler, 1955] norm of 5.7, $SD = 0.8$) as well as attentional disorders with indications of frontal dysfunction.

During the task of transcoding numbers into their arabic form, L.R. produced partial literal transcription errors, whose frequency increased with the digit size of the number (0%, 40%, and 62% of errors for three-, four-, and five-digit numbers, respectively).

These partial literal transcriptions primarily affected *thousand*. For example, *forty-eight thousand nine hundred and fourteen* was transcribed 48000914. These errors are reminiscent of those produced by D.M. but nevertheless differ from them in two important ways. First, forms of type X000 (e.g., *three thousand*), X0000 (e.g., *forty thousand*), and XX000 (e.g., *twenty-five thousand*) gave rise to no errors, in contrast to what was observed in D.M. Second, L.R.'s errors were much less regular than those produced by D.M., who added precisely three 0s following the multiplicand *thousand* in 27 of 35 lexical errors committed (77%). In comparison, L.R. added three 0s in only 8 of the 44 syntactic errors observed for four-digit numbers (17%) and in 30 of 77 errors observed for five-digit numbers (39%). For example, the most frequent error for numbers of the form XX0XX or XX0X0 was the addition of two and not three 0s (e.g., *eighty-nine thousand and thirty-one* was transcribed 89000031 and not 89000031).

Interpretation of the errors by Noël and Seron (1995). Noël and Seron (1995) discussed the possibility of accounting for the observed errors in terms of the hypothesis of a degradation of the overwriting rule (Power & Dal Martello, 1990), which processes the sum relations. They noted that this theoretical perspective is not able to account for the observed differences in the processing of *thousand* and *hundred* because the three-digit numbers were transcribed without error. They concluded that Power and Dal Martello's (1990) model, Deloche and Seron's (1987) asemantic model, and McCloskey's (1992) semantic view all have difficulties in explaining this patient's error pattern.

Interpretation within the ADAPT model. We have noted that L.R. made errors only on numbers consisting of more than three digits and that he was more frequently mistaken on five- than four-digit numbers. If the errors were due to a defect in an overwriting rule relating to the processing of *thousand*, no such difference should be observed. If we consider only those numbers in which a sum relation had to be processed after *thousand* (i.e., only those that gave rise to errors because no errors were observed for X,000 or XX,000 forms), the partial literal transcription errors on *thousand* affected 85% of five-digit numbers but only 38% of the four-digit numbers.

The partial literal transcription errors produced by L.R. could be due to the high cost resulting from the simultaneous maintenance of phonological information in WM, the process of retrieval from LTM, the implementation of the rules, and the maintenance in WM of the intermediate results provided by these rules (i.e., the chain of digits currently being constructed and the unfilled frames it contains). This cognitive load increases with the size of the numbers to be transcribed and changes as the transcription process operates. We hypothesize that the observed partial literal transcriptions result from a cognitive overload that leads L.R. to segment the verbal string into blocks whose transcriptions are then concatenated (e.g., *eighty-nine thousand two hundred and thirty-one* was segmented into *eighty-nine thousand* and *two hundred and thirty-one* and transcribed 89000231).

However, this hypothesis still requires us to explain why the string is segmented more frequently after *thousand* (89000231) than after *hundred* (8920031). ADAPT accounts for this phenomenon because the cognitive load during transcoding is greater after the processing of *thousand* than *hundred* as the following example shows. Table 7 describes the operation of ADAPT^{ADV} for the

Table 7
Successive Stages of the Transcoding Process for Mille Neuf Cent Quarante-Cinq (One Thousand Nine Hundred and Forty-Five) in ADAPT^{ADV}

Input	WMS	Frame	Procedure	STM	Chain
<i>Mille</i>	No	No	P3a	4	
<i>Neuf</i>	No	---	P1	3	1---
<i>Cent</i>	9	---	P2d	2	1---
<i>Quarante-cinq</i>	No	---	P1	1	19--
End	45	--	P4c	0	19--
End	No	No			1945 stop

Note. Before being processed by ADAPT^{ADV}, *mille neuf cent quarante-cinq* is parsed into *mille/neuf/cent/quarante-cinq* (thousand/nine/hundred/forty-five). *Input*, *WMS* (working memory store), and *Frame* refer to the three triggering conditions of the procedures. *STM* (short-term memory) refers to the number of representational units stored in the phonological store, and *Chain* refers to the form of the chain of digits being constructed in working memory.

string *mille neuf cent quarante-cinq* (one thousand nine hundred and forty-five). For each processing stage, the model specifies the content of the WMS, the presence of an unfilled frame in WM and its size, the nature of the triggered rule, the number of the representational units held in phonological STM, and the successive forms of the chain of digits being constructed. The cognitive load is a function of (a) the number of elements to be kept active in memory (i.e., the phonological information in STM and the chain of digits being constructed) and (b) the cost of mobilizing the rules, which primarily depends on the LTM retrievals they have to perform (Anderson, 1993). The cognitive load can be evaluated at each stage of processing as a function of (a) the size of the digit chain held in WM, (b) the number of units to be held in the phonological buffer and in WMS, and (c) the requirement for LTM retrievals.

If we assume that the cognitive load is greater the larger the size of the string, the more units to be held in STM, and the greater the requirement for LTM retrievals, then the optimum load in the transcription process results from the input of *neuf* [nine] in the production system, which should trigger the retrieval of 9 by Rule P1. At this stage, WM contains a four-element chain (including a three-slot frame), while three units (i.e., *neuf* [nine], *cent* [hundred], and *quarante-cinq* [forty-five]) have to be maintained in the phonological buffer. Our hypothesis is that in such cases the cognitive overload interrupts the transcoding process. This suspension of processing causes the system to consider that there is no input unit with an empty WMS and a frame that is currently being processed. This situation corresponds to the conditions that trigger Rule P4b, whose application results in filling the frame with 0s and proceeding to transcription, which, in this case, results in the transcription 1000. This partial transcription alleviates the current cognitive load. Processing then resumes with P1's retrieval of the form 9, which is set in WMS, and the transcription process then continues, resulting in the production of the chain 945. The resulting transcription is 1000945. In addition, if processing is not interrupted after *thousand*, it is also unlikely to be after *hun-*

dred, as the associated cognitive load is lower.⁵ However, it is possible that a very significant reduction in resources may lead to an initial interruption after *thousand* and a second one after *hundred*. Thus, errors of the types 1000945 or 100090045 should be more frequent than 190045.

We simulated this process. At every stage of processing that required a memory retrieval (i.e., Rule P1), the cognitive load was evaluated and compared with a target value that represented the patient's STM span. Whenever a cognitive overload occurred, ADAPT^{ADV} received as an input a stop signal that triggered the P4b rule, which fills the current frame with 0s. ADAPT^{ADV} was then restarted with a cognitive load of zero. For the sake of simplicity, when evaluating the cognitive load, we took account only of the size of the digit chain being constructed at the time of retrieval from LTM. The frame slots waiting to be filled were weighted 1, whereas already positioned digits, which are probably easier to maintain, were given 0.5 weighting. The STM span was fixed arbitrarily. To simulate the inevitable variance in attention, we varied the span value to which the cognitive load was compared around the fixed value as a normal distribution with a standard deviation of 0.5. ADAPT^{ADV} was preserved in its entirety, with the introduction of the described system used for computing the cognitive load.

The 401 numbers dictated to L.R. were entered in the simulation with a number of different span values. The mean span value resulting in the best fit between the number of observed errors ($N = 134$) and the numbers of errors produced by the simulation was 3.5. It should be noted that during the examination, L.R. had an STM span very close to this value (i.e., 4). We therefore performed 50 simulations with a span value of 3.5. The mean number of errors produced by ADAPT was 142.6 ($SD = 6.6$), compared with 134 errors made by L.R. The distribution of these errors as a function of the size of the numbers for transcription (one to five digits) was very close to that observed in L.R. (see Table 8). If we consider only the 5 simulations that produced a number of errors similar to L.R.'s (between 132 and 136), the distribution of these errors perfectly matched L.R.'s performance.

The performances produced by ADAPT in terms of successful or incorrect transcoding were very close to those of L.R. On average, the 50 simulations correctly transcribed 219 (82%) of the 267 numbers correctly transcribed by L.R. and, on average, committed 94 errors (70%) out of L.R.'s 134 incorrect transcriptions. Thus, ADAPT reproduced L.R.'s performance in 78% of cases. Furthermore, of the 120 numbers for which L.R. produced a purely syntactic error, 77 (64%) were transcribed by ADAPT in exactly the same way as L.R. It should be noted that there is no other model that is able to account for L.R.'s errors with such precision. Indeed, Noël and Seron (1995) discarded the hypothesis of a degradation of an overwriting rule in Power and Dal Martello's (1990) semantic model. In the same way, the asemanic model of the same authors (Power & Dal Martello, 1997) cannot account for L.R.'s difficulties in transcoding. Indeed, as we have shown, L.R.'s errors result from variations of the cognitive load during the transcoding process. However, such variations can be accounted for only by a model such as ADAPT that assumes that the transcoding proceeds sequentially from the beginning of the verbal string to its end. This is not the case with Power and Dal Martello's (1997) asemanic model.

Table 8
Rate of Errors Made by L.R. and ADAPT as a Function of the Number of Digits in Numerals

No. of digits	L.R.	Simulation	
		50 cycles	5 cycles
1	0	0	0
2	0	0	0
3	.010	.020	.022
4	.394	.450	.399
5	.669	.670	.659

Note. The correlation between the rate of errors made by L.R. and the 50-cycle simulation is .9976; the correlation between the rate of errors made by L.R. and the 5-cycle simulation is .9998.

Conclusion

We have suggested that transcoding difficulties could arise either following a specific deterioration of part of the production system or as a result of more general impairments that constrain its functioning. An examination of the cases published in the literature shows that ADAPT is able to account for these patients' errors by postulating the existence of such difficulties. The cases of D.M. and C.K. are due to a specific impairment of part of the procedural transcoding system that, in both cases, benefited from a full recovery, whereas the errors made by L.R. are due to a more general disorder that also affects transcoding. We have shown that ADAPT provides a more elegant interpretation of the errors than do the hypotheses issuing from other theoretical frameworks. Furthermore, the results of the simulations undoubtedly testify that ADAPT is also able to describe and interpret neuropsychological disorders.

General Conclusion

This article has presented a new model for transcoding numbers from verbal to arabic form. The main characteristics of this model, ADAPT, are the following:

1. ADAPT relies on a classical architecture of the mind, such as implemented in Anderson's (1983) ACT model. The cognitive system includes an LTM store, a WM, and a set of procedures that trigger a certain action when a set of conditions, defined by both the nature of the current input and the current content of WM, is fulfilled.
2. Transcoding is direct or asemanic. The verbal string is parsed as a sequence of units, each unit being the largest unit available in LTM that matches the current input. These units are processed sequentially and trigger the transcoding operations, without the

⁵ We suppose, for example, that the memory retention of frames is more difficult than that of a chain of digits, given that the various memory traces of an empty frame are probably highly similar and may therefore well result in confusion concerning the correct number of slots to be maintained. This hypothesis also predicts that numbers such as *eleven thousand one hundred and eleven* (11111) should cause difficulties during transcoding. It should also be noted that L.R. made errors apparently due to just such a confusion of memory traces. Indeed, he transcribed *four thousand four hundred and three* as 44403.

need of an intermediate stage in which their mathematical meaning would be activated.

3. The model accounts for learning or development. Improvement in transcoding abilities is determined first by a set of endogenous transformations of the processes engaged during transcoding and second by changes in the task demands, characterized by the need of dealing with larger and larger numbers while children progress in age.

4. Endogenous transformations mainly originate because repeated algorithmic solutions to a problem lead to the formation of new units in LTM, with each unit linking a problem to its solution. In keeping with Logan's (1988) model of automatism, the formation of new representational units allows a shift from algorithmic processing toward a direct retrieval for the most frequent digit strings.

5. Because the smallest numbers are the most frequent, this change makes the procedures specifically devoted to the transcoding of small numbers no longer necessary. As a consequence, these procedures can be removed from a model devised to simulate expert performance. Direct retrieval makes processing far simpler even for large numbers because large numbers include small numbers as components.

6. This simplification makes it possible to add new procedures, devised to process larger and larger numbers, without overloading the processing system.

Although these principles constitute a unified model of transcoding, they have been implemented in several computational versions, each of them being aimed at simulating processing occurring at a specific developmental level or in specific populations. In particular, ADAPT^{BASIC} simulates the learning of the transcoding of the numbers 1–99, whereas ADAPT^{ADV} takes for granted that transcoding of numbers 1–99 no longer involves algorithmic processing and deals with numbers up to 999999. ADAPT^{LD} borrows different components of these two models to address the case of mildly retarded subjects. ADAPT has been used to make precise predictions, which have been tested in two large-scale studies bearing on normal children in second and third grades on the one hand and on mildly retarded adolescents on the other. Earlier neurological studies reporting individual cases of brain-damaged patients have also been examined in light of the model.

Overall, predictions generated by ADAPT were confirmed with regard to the three main aspects of the model, pertaining to its developmental components, its asemantic option, and its procedural nature, respectively. First, developmental hypotheses were supported by several results in children and in mildly retarded adolescents. Specific predictions concerned errors in transcoding D forms (e.g., forty) and DU forms (e.g., forty-five) of numbers. Indeed, for alternative models, DU forms imply additional production rules with regard to D forms, and therefore, more errors on DU forms than on D forms should be expected if these rules were recruited. Actually, as predicted by ADAPT, error rates were identical in both cases. Similarly, although alternative models predict the reverse pattern, four-digit numbers ending by U forms result in more errors than those ending by DU or D. Hence, these results support ADAPT's predictions that DU forms were retrieved from memory after a moderate amount of learning, as also testified by the outputs of the simulation. It is interesting to note that mildly retarded subjects with difficulty in automatizing the retrieval of

responses show stronger error rates for DU forms than for D forms. This fact confirms that these subjects still relied on algorithmic processing for these numbers.

Second, contrary to McCloskey's (1992) and Power and Dal Martello's (1990) models, ADAPT proposes that transcoding is asemantic. Because they assume that the meaning of any lexical input serves as a necessary intermediary between the verbal and the numerical forms of any number, semantic models of transcoding anticipate that a given lexical input would generate a fixed type of errors, irrespective of its position in the verbal strings. Performance of normal children and mildly retarded adolescents ruled out this prediction. Indeed, the nature of errors appeared to be determined by both the present lexical input and the current state of the processing system, in a way predicted by ADAPT. Moreover, the errors produced by patient C.K. confirmed these experimental results. Indeed, she correctly transcribed four-digit numbers (77% of correct responses), whereas she systematically failed to transcode three-digit numbers. This fact, which is at odds with any semantic model, is successfully simulated by ADAPT.

Third, the procedural functioning of ADAPT, which is bounded by the general constraints of the cognitive architecture, permits specific predictions in children and in mildly retarded adolescents as well as in brain-damaged patients. The observed rate of transcoding errors was strongly correlated with the number of production rules required by ADAPT to transcode a number (.903 for normal children; .863 for mildly retarded adolescents). In each case, these values overcame the predictions of alternative models. Moreover, in ADAPT, so-called syntactical and lexical errors of transcoding may be selectively attributed to the dysfunctioning of algorithmic operations and direct retrieval, respectively. In line with ADAPT's predictions, syntactic and lexical errors of mildly retarded adolescents were independent from each other. Syntactic errors were correlated ($r = .736$) with a variable taken as an indicator of the level of difficulty of the production rules, whereas lexical errors were correlated ($r = .732$) with the number of retrievals required to transcode a given verbal string. Finally, specific degradations of this procedural system should result in specific pattern of errors. The errors produced by D.M. (Cipolotti et al., 1994) and C.K. (Delazer & Denes, 1998) turned out to be conveniently accounted for by the impairment of specific rules. Moreover, the way ADAPT is embedded in the general cognitive architecture allows us to predict the impact of general cognitive constraints on the transcoding process. Accordingly, it was demonstrated that errors produced by L.R. (Noël & Seron, 1995) correspond to what occurs when a general limitation in WM capacity prevents ADAPT from running correctly. In each neuropsychological case, simulation outputs fairly matched patients' production and accounted for more errors than any other alternative models or interpretation.

How do people write numbers? ADAPT provides evidence that even young children use a procedural system that evolves through learning and practice and does not require access to the meaning of numbers. Such an asemantic process would in turn have an impact on the further mathematical acquisitions. Indeed, the meaningful structure of the digital forms produced by this system would provide children with a basis for understanding how numbers are constructed, what they represent, and how they can be used.

References

- Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Erlbaum.
- Anderson, J. R. (1993). *The rules of the mind*. Hillsdale, NJ: Erlbaum.
- Anderson, J. R., Fincham, J. M., & Douglass, S. (1999). Practice and retention: A unifying analysis. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25, 1120–1136.
- Anderson, J. R., & Lebière, C. (1998). *Atomic components of thought*. Mahwah, NJ: Erlbaum.
- Anderson, J. R., Reder, L. M., & Lebière, C. (1996). Working memory: Activation limitations on retrieval. *Cognitive Psychology*, 30, 221–256.
- Baddeley, A. (1986). *Working memory*. Oxford, England: Oxford University Press.
- Baddeley, A. D. (1990). *Human memory: Theory and practice*. Hillsdale, NJ: Erlbaum.
- Barrouillet, P., & Fayol, M. (1998). From algorithmic computing to direct retrieval: Evidence from number and alphabetic arithmetic in children and adults. *Memory & Cognition*, 26, 355–368.
- Barrouillet, P., Fayol, M., & Lathulière, E. (1997). Selecting between competitors in multiplication tasks: An explanation of the errors produced by adolescents with learning difficulties. *International Journal of Behavioral Development*, 21, 253–275.
- Brysbart, M., Fias, W., & Noël, M.-P. (1998). The Whorfian hypothesis and numerical cognition: Is “twenty-four” processed in the same way as “four-and-twenty”? *Cognition*, 66, 51–77.
- Caramazza, A., & McCloskey, M. (1987). Dissociations of calculation processes. In G. Deloche & X. Seron (Eds.), *Mathematical disabilities* (pp. 221–234). Hillsdale, NJ: Erlbaum.
- Cipolotti, L. (1995). Multiple routes for reading words, why not numbers? Evidence from a case of arabic numeral dyslexia. *Cognitive Neuropsychology*, 12, 313–342.
- Cipolotti, L., & Butterworth, B. (1995). Towards a multiroute model of number processing: Impaired number transcoding with preserved calculation skills. *Journal of Experimental Psychology: General*, 124, 375–390.
- Cipolotti, L., Butterworth, B., & Warrington, E. K. (1994). From “one thousand nine hundred and forty five” to 1,000,945. *Neuropsychologia*, 32, 503–509.
- Cohen, L., Dehaene, S., & Verstichel, P. (1994). Number words and number non-words: A case of deep dyslexia extending to arabic numerals. *Brain*, 117, 267–279.
- Compton, B. J., & Logan, G. D. (1991). The transition from algorithm to retrieval in memory based theories of automaticity. *Memory & Cognition*, 19, 151–158.
- Cowan, N. (1995). *Attention and memory: An integrated framework*. New York: Oxford University Press.
- Cowan, N. (1999). An embedded-process model of working memory. In A. Miyake & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (pp. 62–101). Cambridge, England: Cambridge University Press.
- Dehaene, S., Dupoux, E., & Mehler, J. (1990). Is numerical comparison digital? Analogical and symbolic effects in two-digit number comparison. *Journal of Experimental Psychology: Human Perception and Performance*, 16, 626–641.
- Dehaene, S., & Mehler, J. (1992). Cross-linguistic regularities in the frequency of number words. *Cognition*, 43, 1–29.
- Delazer, M., & Denes, G. (1998). Writing arabic numerals in an agraphic patient. *Brain and Language*, 64, 257–266.
- Deloche, G., & Seron, X. (1982a). From one to 1: An analysis of a transcoding process by means of neuropsychological data. *Cognition*, 12, 119–149.
- Deloche, G., & Seron, X. (1982b). From three to 3: A differential analysis of skills in transcoding quantities between patients with Broca’s and Wenicke’s aphasia. *Brain*, 105, 719–733.
- Deloche, G., & Seron, X. (1987). Numerical transcoding: A general production model. In G. Deloche & X. Seron (Eds.), *Mathematical disabilities: A cognitive neuropsychological perspective* (pp. 137–170). Hillsdale, NJ: Erlbaum.
- Engle, R. W., Kane, M. J., & Tuholski, S. W. (1999). Individual differences in working memory capacity and what they tell us about controlled attention, general fluid intelligence, and functions of the prefrontal cortex. In A. Miyake & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (pp. 102–134). Cambridge, England: Cambridge University Press.
- Eriksen, C. W., Pollack, M. D., & Montague, W. E. (1970). Implicit speech: Mechanism in perceptual encoding? *Journal of Experimental Psychology*, 84, 502–507.
- Fuson, K. C., Richards, J., & Briars, D. J. (1982). The acquisition and elaboration of the number word sequence. In C. J. Brainerd (Ed.), *Children’s logical and mathematical cognition: Progress in cognitive development research* (pp. 33–92). New York: Springer-Verlag.
- Geary, D. C., Brown, S. C., & Samaranayake, V. A. (1991). Cognitive addition: A short longitudinal study of strategy choice and speed-of-processing differences in normal and mathematically disabled children. *Developmental Psychology*, 27, 787–797.
- Geary, D. C., Widaman, K. F., Little, T. D., & Cormier, P. (1987). Cognitive addition: Comparison of learning disabled and academically normal elementary school children. *Cognitive Development*, 2, 249–269.
- Hughes, M. (1986). *Children and number: Difficulties in learning*. New York: Basic Blackwell.
- Hurford, J. R. (1987). *Language and number*. Oxford, England: Basic Blackwell.
- Klapp, S. (1971). Implicit speech inferred from response latencies in same-different decisions. *Journal of Experimental Psychology*, 91, 262–267.
- Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, 95, 492–527.
- Logan, G. D., & Klapp, S. T. (1991). Automatizing alphabet arithmetic: I. Is extended practice necessary to produce automaticity? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 17, 179–195.
- Lovett, M. C., Reder, L. M., & Lebière, C. (1999). Modeling working memory in a unified architecture: An ACT-R perspective. In A. Miyake & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (pp. 135–182). Cambridge, England: Cambridge University Press.
- McCloskey, M. (1992). Cognitive mechanisms in numerical processing: Evidence from acquired dyscalculia. *Cognition*, 44, 107–157.
- McCloskey, M., & Caramazza, A. (1987). Cognitive mechanisms in normal and impaired number processing. In G. Deloche & X. Seron (Eds.), *Mathematical disabilities* (pp. 201–219). Hillsdale, NJ: Erlbaum.
- McCloskey, M., Caramazza, A., & Basili, A. (1985). Cognitive mechanisms in number-processing and calculation: Evidence from dyscalculia. *Brain and Cognition*, 4, 171–196.
- Miura, I. T., Okamoto, Y., Kim, C. C., Steere, M., & Fayol, M. (1993). First graders’ cognitive representation of numbers and understanding of place value: Cross-national comparison—France, Japan, Korea, Sweden, and the United States. *Journal of Educational Psychology*, 85, 24–30.
- Miura, I. T., Okamoto, Y., Kim, C. C., Steere, M., & Fayol, M. (1994). Comparisons of cognitive representation of number: China, France, Japan, Korea, Sweden, and the United States. *International Journal of Behavioral Development*, 17, 401–411.
- Noël, M.-P., & Seron, X. (1995). Lexicalization errors in writing arabic numerals. *Brain and Cognition*, 29, 151–179.
- Noël, M.-P., & Seron, X. (1997). On the existence of intermediate repre-

- sentations in numerical processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23, 697–720.
- Perruchet, P., & Vinter, A. (1998). PARSER: A model of word segmentation. *Journal of Memory and Language*, 39, 246–263.
- Perruchet, P., & Vinter, A. (2002). A self organized consciousness. *Behavioral and Brain Sciences*, 25, 297–388.
- Power, R. J. D., & Dal Martello, M. F. (1990). The dictation of Italian numerals. *Language and Cognitive Processes*, 5, 237–254.
- Power, R. J. D., & Dal Martello, M. F. (1997). From 834 to eighty thirty four: The reading of arabic numerals by seven-year-old children. *Mathematical Cognition*, 3, 63–85.
- Power, R. J. D., & Longuet-Higgins, J. C. (1978). Learning to count: A computational model of language acquisition. *Proceedings of the Royal Society of London, Series B*, 200, 391–417.
- Reynvoet, B., & Brysbaert, M. (1999). Single-digit and two-digit arabic numerals address the same semantic number line. *Cognition*, 72, 191–201.
- Saxton, M., & Towse, J. N. (1998). Linguistic relativity: The case of place value in multidigit numbers. *Journal of Experimental Child Psychology*, 69, 66–79.
- Seron, X., Deloche, G., & Noël, M. P. (1992). Number transcribing by children. In J. Bideaud, C. Meljac, & J. P. Fisher (Eds.), *Pathways to number* (pp. 245–264). Hillsdale, NJ: Erlbaum.
- Seron, X., & Fayol, M. (1994). Number transcoding in children: A functional analysis. *British Journal of Developmental Psychology*, 12, 281–300.
- Siegler, R. S. (1996). *Emerging minds: The process of change in children's thinking*. Oxford, England: Oxford University Press.
- Siegler, R. S., & Robinson, M. (1982). The development of numerical understanding. In H. W. Reese & L. P. Lipsitt (Eds.), *Advances in child development and behavior* (Vol. 16, pp. 241–312). New York: Academic Press.
- Siegler, R. S., & Shrager, J. (1984). Strategy choices in addition and subtraction: How do children know what to do? In C. Sophian (Ed.), *Origins of cognitive skills* (pp. 229–293). Hillsdale, NJ: Erlbaum.
- Sinclair, A., Siegrist, F., & Sinclair, H. (1983). Young children's ideas about the written number system. In D. Rogers & J. Sloboda (Eds.), *The acquisition of symbolic skills* (pp. 535–543). New York: Plenum Press.
- Singer, H. D., & Low, A. A. (1933). Acalculia (Henschen): A clinical study. *Archives of Neurology and Psychiatry*, 29, 476–498.
- Wechsler, D. (1955). *Wechsler Adult Intelligence Scale*. New York: The Psychological Corporation.

Received January 17, 2001

Revision received December 18, 2002

Accepted July 3, 2003 ■

Low Publication Prices for APA Members and Affiliates

Keeping you up-to-date. All APA Fellows, Members, Associates, and Student Affiliates receive—as part of their annual dues—subscriptions to the *American Psychologist* and *APA Monitor*. High School Teacher and International Affiliates receive subscriptions to the *APA Monitor*, and they may subscribe to the *American Psychologist* at a significantly reduced rate. In addition, all Members and Student Affiliates are eligible for savings of up to 60% (plus a journal credit) on all other APA journals, as well as significant discounts on subscriptions from cooperating societies and publishers (e.g., the American Association for Counseling and Development, Academic Press, and Human Sciences Press).

Essential resources. APA members and affiliates receive special rates for purchases of APA books, including the *Publication Manual of the American Psychological Association*, and on dozens of new topical books each year.

Other benefits of membership. Membership in APA also provides eligibility for competitive insurance plans, continuing education programs, reduced APA convention fees, and specialty divisions.

More information. Write to American Psychological Association, Membership Services, 750 First Street, NE, Washington, DC 20002-4242.