Exploring the Complexity and Structure of Acheulean Stoneknapping in Relation to Natural Language

ROBERT ALLEN MAHANEY
Department of Anthropology, Student Building 130, 701 E. Kirkwood Avenue, Indiana University, Bloomington, IN 47405-7100, USA; and Stone Age Institute, 1392 W. Dittemore Road, Gosport, IN 47433, USA; romahane@indiana.edu

submitted: 25 June 2014; accepted 19 October 2014

ABSTRACT
The intuition that there is a homology between sequenced action during stoneknapping and syntax in language is long-standing, but rarely explicitly analyzed. If valid, this proposed homology would allow paleoanthropologists to gain a handle on the timing and context of language emergence. Here, I present the results of three pilot studies performed to explore the methods such an analysis would require, as well as the issues that such an analysis would raise. The replication of an Acheulean handaxe was videotaped, then coded. This lithic reduction was analyzed using information theory, formal grammars, and Markov models. These three analyses found: (1) in terms of information entropy, the thinning phase of handaxe manufacture is as complex as many English language utterances; (2) the lithic reduction can be represented as a Context-Free Grammar (CFG), though in reality it only has limited embedding and is largely iterative in structure; and, (3) the lithic reduction also can be simulated by ‘mindless’ Markov models. These results raise a number of issues. First, it is not clear how to define and validate comparable units in stoneknapping and language. It is also not clear that the flow of actions performed by a stoneknapper can be easily segmented into discrete units. Second, in Studies One and Two, it was found that handaxe replication could be simulated by both a CFG and a Markov model instantiating a Finite State Grammar. The types of cognitive mechanisms capable of instantiating these are significantly different, with a CFG requiring memory resources not needed by the simpler Markov processes. These pilot studies indicate that it is possible to utilize these methods in the analysis of stoneknapping, but a number of basic conceptual and methodological issues remain to be clarified.

INTRODUCTION: STONEKNAPPING AND LANGUAGE EVOLUTION
The hypothesis that the sequential structure of stoneknapping is homologous to syntax in language is long-standing (Holloway, Jr, 1969; Moore 2010) with support from neuroimaging studies of modern knappers replicating Acheulean technology (Stout and Chaminade 2012; Stout et al. 2008; Uomini and Meyer 2013). Both population-level cerebral asymmetries associated with language use and handedness (Holloway 1976; 1983) and the Early Acheulean technocomplex (Asfaw et al. 1992; Lepre et al. 2011) emerged approximately 1.8 ma. However, little work has been done to move beyond analogical and conceptual analyses to compare the structure of stoneknapping to language. This paper is a contribution to the methodology of the structural analysis of stoneknapping.

Are there deep structural similarities between language syntax and stoneknapping? If so, then stoneknapping should exhibit an underlying grammar with actions organized into recursive, phrase-like structures that are in turn organized into overall sentence-like schemas (Figure 1). Such a structure would presumably mirror theoretical models of language and would require the hierarchical coordination of a diverse range of complex cognitive faculties (Barceló-Coblijn 2012).

In the 20th century, linguists working in generative grammar used formal languages to describe the underlying structure that all languages share. In particular, Chomsky (1959a) argued that a type of formal language called a context-free language (CFG, described below) captured all of the features required by the cognitive machinery powerful enough to generate the grammatical structures of observed (‘natural’) language. Chomsky’s conclusions regarding the cognitive basis of language were motivated and limited by contemporary debates between behaviorists and cognitivists (e.g., Chomsky 1959b). More recent work viewing language as a dynamic system (Beckner et al. 2009; Christiansen and Chater 2008; Port 2010; Port and Leary 2005; Tomasello 2000, 2006) have profoundly challenged Chomsky’s model of the language-capable brain. However, the comparison of sequential structures using formal languages remains a valid means of objectively comparing two sequential phenomena. If a CFG fits both natural languages and the actions involved in making a handaxe, this provides support for the intuitions of Holloway, Stout, and others.
Acheulean Stoneknapping and Language

observed sequence of action during the replication of an Acheulean handaxe: (1) should exhibit levels of complexity similar to that seen in actual linguistic utterances; and, (2) a context-free grammar should fit the observed sequences better than a simpler finite state grammar implemented in a simple Markov model.

**STUDY 1: RELATIVE COMPLEXITY OF LANGUAGE AND STONE KNPAPPING**

**INTRODUCTION**

Holloway (1969) proposed that the sequential structure of action in stoneknapping reflects the same underlying structure as syntax in language. He writes:

“Elements of a basic “vocabulary” of motor operations-flake detachment, rotation, preparation of striking platform, etc.-are used in different combinations to produce dissimilar tools, with different forms, and supposedly, different uses” (p. 54).

If this is the case, then the same metrics of relative complexity can be applied to both stoneknapping and a sample of utterances to determine how similar they are in terms of relative complexity.

There have been a number of attempts at finding a quantitative measure of technological complexity using graph theory (Mahaney 2013; Rugg 2011) or procedural
units (Perreault et al. 2013).

These approaches focus on the cultural recipes (in the sense of Mesoudi and O’Brien 2008; Shore 1996) underlying a particular technology. While this is a valid and interesting approach to the problem, a recipe is an abstraction. The ability to assess the complexity of actual lithic reductions performed by stoneknappers would provide an additional perspective on technological evolution.

Information Theory provides the ideal means of approach to measure the complexity of sequential events. The concept of information entropy (Pierce 1980; Shannon and Weaver 1949) was developed by electrical engineers as a means of determining the most efficient way to encode information in bits for transmission over noisy channels. Information entropy measures uncertainty in a signal. If I were to say the word “dog,” all other words in the language—such as “barks,” “flies,” “stoneknaps”—have a certain probability of following it. The more possibilities of subsequent words, the greater the listener’s uncertainty. More complex signals, or sequences, have higher levels of uncertainty. Entropy is a measure of that uncertainty.

Information entropy can be applied to any sequential structure from computer programs (Linz 2011), genetic sequences (Lin 1991), life histories (Gabadinho et al. 2011), the English language (Shannon 1951a, b), or the actions performed during stone knapping. In this analysis, entropy is used to explore the relative complexity of the replication of an Acheulean handaxe compared to English language utterances. While there may be methodological and conceptual issues with this approach that I will discuss below, the goal of this analysis is to determine if this approach is potentially valid and informative.

METHODS

An expert male stone knapper with over thirty years’ experience working stone was videotaped for forty minutes replicating a cordiform Acheulean handaxe similar to those found at site of Boxgrove, United Kingdom (Figure 2). The knapper was not informed of the nature of the study so as to not influence his behavior. A nodule of high quality flint was used for this replication, as well a stone hammer and antler billet. A coding scheme was developed based on the observation of repeated technical actions performed by the knapper (Figure 3). Actions were defined simply, such as...
Acheulean Stoneknapping and Language

These coded flake units were compared to a corpus of English language utterances transcribed from the 2nd US Presidential Debate in 2000 that were downloaded from the American National Corpus (Ide and Macleod 2001; Ide and Suderman 2002), an online archive for text-mining research. This corpus was segmented into sentences and punctuation removed. Both the stone knapping and language sequences were imported into the R 2.15.1 software (R Core Team 2012). Further processing and analyses were performed in the TraMiner package (Gabadinho et al. 2011; Gabadinho et al. 2009). Designed for analyzing life history sequences, TraMiner contains a number of analyses that can be applied to other forms of sequential data.

Shannon entropy was calculated for each flake unit and sentence. Entropy is being used as a relative measure of complexity and not an absolute measure. Shannon entropy is given by the formula:

$$H = - \sum p \log_2 p,$$

in which $p$ is the probability of an action or word occurring.

Figure 3: Coding scheme used in this study. In order to derive the grammar described in Study 2, the initial elements of the alphabet were specified. These included two non-terminal symbols (A and B) specifying the most basic components of a flake unit. It also included a coding scheme for each action performed by the knapper during each flake unit. These are terminal symbols. Coding resulted in descriptions of all of the actions performed during each flake unit. Starting with the first two non-terminal symbols, the grammar in Figure 8 was derived. The coded actions also served as the basis for Study 1 and 3.

The component actions were coded with letters representing one of sixteen possible actions listed in Figure 3. The entire sequence was represented as a string of actions that were then segmented into flake units (Moore 2010; 2011). A flake unit includes all of the actions leading to the removal of a flake. It was observed that the knapper often prepared the core for a series of related removals. He then removed two or more flakes in quick succession. In doing so, he was effectively concatenating a series of flakes into a single flake unit. In those cases, this sequence was treated as a single unit.

These coded flake units were compared to a corpus of English language utterances transcribed from the 2nd US Presidential Debate in 2000 that were downloaded from the American National Corpus (Ide and Macleod 2001; Ide and Suderman 2002), an online archive for text-mining research. This corpus was segmented into sentences and punctuation removed. Both the stone knapping and language sequences were imported into the R 2.15.1 software (R Core Team 2012). Further processing and analyses were performed in the TraMiner package (Gabadinho et al. 2011; Gabadinho et al. 2009). Designed for analyzing life history sequences, TraMiner contains a number of analyses that can be applied to other forms of sequential data.

Shannon entropy was calculated for each flake unit and sentence. Entropy is being used as a relative measure of complexity and not an absolute measure. Shannon entropy is given by the formula:

$$H = - \sum p \log_2 p,$$

in which $p$ is the probability of an action or word occurring.

Figure 3: Coding scheme used in this study. In order to derive the grammar described in Study 2, the initial elements of the alphabet were specified. These included two non-terminal symbols (A and B) specifying the most basic components of a flake unit. It also included a coding scheme for each action performed by the knapper during each flake unit. These are terminal symbols. Coding resulted in descriptions of all of the actions performed during each flake unit. Starting with the first two non-terminal symbols, the grammar in Figure 8 was derived. The coded actions also served as the basis for Study 1 and 3.

The component actions were coded with letters representing one of sixteen possible actions listed in Figure 3. The entire sequence was represented as a string of actions that were then segmented into flake units (Moore 2010; 2011). A flake unit includes all of the actions leading to the removal of a flake. It was observed that the knapper often prepared the core for a series of related removals. He then removed two or more flakes in quick succession. In doing so, he was effectively concatenating a series of flakes into a single flake unit. In those cases, this sequence was treated as a single unit.

These coded flake units were compared to a corpus of English language utterances transcribed from the 2nd US Presidential Debate in 2000 that were downloaded from the American National Corpus (Ide and Macleod 2001; Ide and Suderman 2002), an online archive for text-mining research. This corpus was segmented into sentences and punctuation removed. Both the stone knapping and language sequences were imported into the R 2.15.1 software (R Core Team 2012). Further processing and analyses were performed in the TraMiner package (Gabadinho et al. 2011; Gabadinho et al. 2009). Designed for analyzing life history sequences, TraMiner contains a number of analyses that can be applied to other forms of sequential data.

Shannon entropy was calculated for each flake unit and sentence. Entropy is being used as a relative measure of complexity and not an absolute measure. Shannon entropy is given by the formula:

$$H = - \sum p \log_2 p,$$

in which $p$ is the probability of an action or word occurring.

Figure 3: Coding scheme used in this study. In order to derive the grammar described in Study 2, the initial elements of the alphabet were specified. These included two non-terminal symbols (A and B) specifying the most basic components of a flake unit. It also included a coding scheme for each action performed by the knapper during each flake unit. These are terminal symbols. Coding resulted in descriptions of all of the actions performed during each flake unit. Starting with the first two non-terminal symbols, the grammar in Figure 8 was derived. The coded actions also served as the basis for Study 1 and 3.

The component actions were coded with letters representing one of sixteen possible actions listed in Figure 3. The entire sequence was represented as a string of actions that were then segmented into flake units (Moore 2010; 2011). A flake unit includes all of the actions leading to the removal of a flake. It was observed that the knapper often prepared the core for a series of related removals. He then removed two or more flakes in quick succession. In doing so, he was effectively concatenating a series of flakes into a single flake unit. In those cases, this sequence was treated as a single unit.

These coded flake units were compared to a corpus of English language utterances transcribed from the 2nd US Presidential Debate in 2000 that were downloaded from the American National Corpus (Ide and Macleod 2001; Ide and Suderman 2002), an online archive for text-mining research. This corpus was segmented into sentences and punctuation removed. Both the stone knapping and language sequences were imported into the R 2.15.1 software (R Core Team 2012). Further processing and analyses were performed in the TraMiner package (Gabadinho et al. 2011; Gabadinho et al. 2009). Designed for analyzing life history sequences, TraMiner contains a number of analyses that can be applied to other forms of sequential data.

Shannon entropy was calculated for each flake unit and sentence. Entropy is being used as a relative measure of complexity and not an absolute measure. Shannon entropy is given by the formula:

$$H = - \sum p \log_2 p,$$

in which $p$ is the probability of an action or word occurring.
at that point in the observed flake unit or sentence given the action or word preceding it. The resulting distribution of entropy values was not normal, so a three-sample Kruskal–Wallis test was performed over the aggregate samples to determine if the mean entropy differed significantly during the Presidential Debate, Acheulean Shaping phase, and Acheulean thinning phase.

RESULTS

Overall, the knapper performed 2,683 actions organized into 46 flake units. This falls within the range of flake units for the knapping of a handaxe reported in Steele et al. (1995) and Chazan (2012). The 2nd 2000 Presidential debate had 1,086 total sentences.

Figure 4 shows the distribution of Shannon entropy for sentences from the Presidential debate ($M=0.66$, $SD=0.50$, range=0.04–4.16) and flake units during Acheulean shaping ($n=23$, $M=0.18$, $SD=0.06$, range=0.12–0.34) and thinning ($n=23$, $M=0.78$, $SD=0.52$, range=0.14–1.47) phases of the replication. There was a significant difference in the medians of the three samples, $H=40.87$, $p<0.001$. Post hoc Mann-Whitney pairwise comparison found no significance difference between the Presidential debate and Acheulean thinning, with mean ranks of 541.9 and 13.05 respectively ($U=1.078E04$, $Z=-1.125$, $p=0.26$). However, it did find a significant difference between Acheulean shaping and the Presidential Debate with mean ranks of 552.1 and 2.87 respectively ($U=2901$, $Z=-6.31$, $p<0.001$). There was also a significant difference between Acheulean shaping and thinning with mean ranks of 15.83 and 7.67 respectively ($U=77$, $Z=-4.12$, $p<0.001$).

While entropy values for the presidential debate vary through the entire series, those for stoneknapping show the expected initial shaping phase followed by thinning (Figure 5). The shaping phase (hard hammer) lacks the careful platform preparation of the later thinning phase (antler billet and hammerstone). There is a base flake unit present in shaping composed of placing the blank on the thigh (a),
orienting the edge (b, c, d), gauging the blow (j), striking (k), and sweeping away debitage (p). This is obviously an invariant sequence. More variability, and thus higher entropy values, occurs during the thinning phase when additional platform preparation elements are added.

**DISCUSSION**

In terms of information entropy, the earlier thinning phase of handaxe replication is less complex than the corpus of English language utterances and handaxe thinning. Handaxe thinning is not statistically different from the Presidential debate corpus. This latter result was completely unexpected. Natural languages such as English are theoretically infinite in their ability to recursively embed phrases within phrases (Bar-Hillel et al. 1961; Ogden 1968). In reality, there are practical limitations on working memory (Miller 1956) that restrict the human capacity to understand or produce very long sentences. For generative linguists and many evolutionary psychologists (e.g., Fitch et al. 2005; Hauser et al. 2002; Pinker 1994; Pinker and Jackendoff 2005), it is this theoretically limitless recursivity that distinguishes language from all other animal communication systems. Figure 6 highlights this hypothesized design feature of natural language.

The greatest entropy value for the utterances is 4.16 bits while the greatest value for thinning a handaxe is 1.47. A flake unit is constrained to a lift-placement-orient-strike-sweep debitage structure. Additional platform preparation elements can only complicate the sequence so much until it becomes redundant and even counter-productive. Word order in language is not constrained in the same way, with subject-object-verb, subject-verb-object, and verb-subject-object being the most common word order (Tomlin 1986). While the perception and production of utterances are constrained by working memory capacity, phrase embedding in language can be more complex if there is a need for it to be. However, most of the time there may not be a need for higher levels of complexity.

This analysis and its methods raise important conceptual issues about research into the complexity and cognitive structure of stone knapping. It is not clear which units of action should be compared to particular linguistic structures, like phonemes, morphemes, sentences, or discourse. Is a flake unit comparable to a sentence? Looking at the graphical representations of knapping presented by Moore (2010) and Stout (2008, 2011) (see Figure 1), there does seem to be a similarity to the structure of a sentence—actions are grouped into phrase-like units that in turn are grouped into a sentence. However, does this graphical model map actually map onto stoneknapping or is it being imposed upon it? Hill (1972) speculated that stoneknapping was a simpler, shallower process than language. The actions out of which it is composed were homologous to consonants and not the morphemes that I have modeled them on in this pilot study. That said, the experimental work needed to determine what the appropriate units of analysis are has yet not been carried out.
The knapper, who has worked stone, such as flint, for twenty years, struck off the flake and swept away thedebitage.
Start with a structure constructed from terminal symbols
- Follow a series of rules (a grammar) that define how to transform these into terminal symbols

<table>
<thead>
<tr>
<th>Starting Schema</th>
<th>Simple Grammar</th>
<th>Example Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>S → AB</td>
<td>S → AB</td>
</tr>
<tr>
<td>A → aB</td>
<td>AB → aBB</td>
<td>AB → aBB</td>
</tr>
<tr>
<td>B → b</td>
<td>aBB → abb</td>
<td>aBB → abb</td>
</tr>
</tbody>
</table>

In this example we are given S and end with abb.

**RESULTS**

It is possible to generate a right-branching CFG capable of generating the observed flake units (Figure 8, see Appendix).

Overall, the resulting grammar demonstrates both recursive and chain-like behavior. First, it was common to concatenate (link in a chain-like manner) a series of flake removals onto a basic flake unit. The knapper would prepare a large region along the lateral edge of the blank, and then he would remove a series of flakes. This structure is captured by the rule $B \rightarrow EFCB$. After proceeding through states E, F, and C, the knapper arrives at B and can reinsert $EFCB (B \rightarrow EFCB \rightarrow EFCB \rightarrow EFCB \rightarrow \ldots)$.

This can be repeated as often as is necessary. For example, Appendix Example 3 can be verbalized as “The stoneknapper performs the basic flake unit in which he removes a flake (1–13) and he removes another flake (14–20) and he removes another flake (21–29).”

Next, there were two rules for embedding repeated platform preparation sequences. During the thinning phase, the rule $E \rightarrow HIE$ allowed for the insertion of platform preparation (HIE) in the $B \rightarrow EFCB$ sequence. Then chisel-like platform preparation ($H \rightarrow ENCAH$) and abrasion ($I \rightarrow EGCAI$) could be repeated as often as was necessary in a chain-like manner. Appendix Example 4 can be verbalized as “The stoneknapper, carefully preparing his platforms (5–41), performs the basic flake unit in which he removes a flake (1–48) and he removes another flake (50–55) and he removes another flake (56–61) and he removes...”

**METHODS**

The same lithic reduction used in the earlier study was reanalyzed in this study.

A formal grammar is comprised of a series of states $(A,B,C,\ldots)$, symbols $(a,b,c,\ldots)$, and rules for generating symbols from states (Figure 7) (Linz 2011; Revesz 1991). For instance, the rule $A \rightarrow aB$ means “If in state $A$ ‘print’ (or ‘emit’) $a$ and go into state $B$.” A series of states leads to the printing of strings like $abb$. The process halts when all states are exhausted (or when the system reaches a STOP state, a state not included in this analysis).

The ‘strings’ of action performed by the knapper were already segmented into flake units in the earlier study, providing the first element of structure in this grammar. At this point, a decision was made to structure flake units $(T)$ into three phases—placement of the blank $(A)$, the preparation and removal of the flake $(B)$, and the sweeping away of the debitage $(C)$. This gives the first rule, $T \rightarrow ABC$. Obviously, this was the imposition of an a priori structure on the strings. After formulating this initial structure, each flake unit was analyzed to derive the rules for moving from one action to the next (Appendix). Additional states were only added as necessary. At this point, it was found that the $T \rightarrow ABC$ flake unit schema appeared to inadequately represent the observed action and the rule was shortened to $T \rightarrow AB$ (see Appendix). The resulting rules were checked against the observed strings to verify that it could generate all observed flake units (see Appendix).
more interesting is the finding that this bout of stoneknapping is both a limited recursive grammar and a ‘string-of-beads.’ If natural language is a CFG, then this means that stone knapping only shares structural features with it to a limited extent. Yet, there are three important issues that this study raises.

First, it cannot be claimed that this grammar is the grammar for the replication of an Acheulean handaxe. Theoretically, there are an infinite number of possible grammars that could have produced these flake units. In addition, the actions as coded in this study are not the only approach to identifying the component actions. For instance, should an action be coded as “orienting the blank” or as “orienting the blank to a particular degree in a particular relation to the knapper”? Arguments can be made for both approaches. Evolutionary cognitive archaeologists need to develop conventions to address this methodological issue.

Second, the definitions of actions, flake units, and an overall schema for a basic flake unit ($T \rightarrow AB$) are necessary steps in formulating an action grammar. However, these units are imposed on the sequence following theoretically based assumptions. These are reasonable assumptions on paper, this grammar does satisfy the requirements of a CFG, it would be referred to as a ‘pumping lemma’ (Bar-Hillel et al. 1961; Ogden 1968). However, this result is slightly misleading. Remember that a CFG is potentially infinitely recursive according to generative linguists. However, in reality these sequences only demonstrate a single recursive level during the thinning phase. Platform preparation proceeds by concatenation ($H \rightarrow \text{ENCAH} \rightarrow \text{ENCAH} \rightarrow \text{ENCAH} \ldots$). In other words, neither a ‘string-of-beads’ structure nor a recursive action grammar appears to be a valid way to conceptualize stoneknapping. Instead, it appears to be a ‘string-of-beads’ with limited embedding of another ‘string-of-beads’ sequence.

DISCUSSION

It is possible to describe the operational sequence that produces an Acheulean handaxe as a CFG. As with the study of complexity above, this is a very interesting result. Perhaps another flake (62–69)."

On paper, this grammar does satisfy the requirements of a CFG, it would be referred to as a ‘pumping lemma’ (Bar-Hillel et al. 1961; Ogden 1968). However, this result is slightly misleading. Remember that a CFG is potentially infinitely recursive according to generative linguists. However, in reality these sequences only demonstrate a single recursive level during the thinning phase. Platform preparation proceeds by concatenation ($H \rightarrow \text{ENCAH} \rightarrow \text{ENCAH} \rightarrow \text{ENCAH} \ldots$). In other words, neither a ‘string-of-beads’ structure nor a recursive action grammar appears to be a valid way to conceptualize stoneknapping. Instead, it appears to be a ‘string-of-beads’ with limited embedding of another ‘string-of-beads’ sequence.

DISCUSSION

It is possible to describe the operational sequence that produces an Acheulean handaxe as a CFG. As with the study of complexity above, this is a very interesting result. Perhaps

more interesting is the finding that this bout of stoneknapping is both a limited recursive grammar and a ‘string-of-beads.’ If natural language is a CFG, then this means that stone knapping only shares structural features with it to a limited extent. Yet, there are three important issues that this study raises.

First, it cannot be claimed that this grammar is the grammar for the replication of an Acheulean handaxe. Theoretically, there are an infinite number of possible grammars that could have produced these flake units. In addition, the actions as coded in this study are not the only approach to identifying the component actions. For instance, should an action be coded as “orienting the blank” or as “orienting the blank to a particular degree in a particular relation to the knapper”? Arguments can be made for both approaches. Evolutionary cognitive archaeologists need to develop conventions to address this methodological issue.

Second, the definitions of actions, flake units, and an overall schema for a basic flake unit ($T \rightarrow AB$) are necessary steps in formulating an action grammar. However, these units are imposed on the sequence following theoretically based assumptions. These are reasonable assumptions
neuroscientists contend that CFGs are too powerful for observed natural languages. In fact, a FSG can generate the same string of symbols as a CFG. However, they lack the additional phrase-structure present in a CFG. Instead of deriving a FSG here, I examined the observed lithic reduction using Markov processes to simulate the observed sequences of actions performed by the stoneknapper. A Markov process can be thought of as a hypothetical machine, or automata ‘emitting’ or ‘printing’ strings of symbols as moves through a series of states probabilistically (Howard 2007). For each level of the Chomsky hierarchy of formal language there is a corresponding level in automata theory. CFGs are computed by pushdown automata equipped with a ‘stack’ onto which states such as the AB flake schema or the EFCB can be stored. FSGs are computed by Markov models. These automata lack the memory capacity that allows a pushdown automaton to recursively embed phrases into a string of symbols. With Markov processes, the only factor that determines the subsequent state is the state the system is in at the moment. A Markov process is modelled with two components, a transition matrix and an emission matrix. The transition matrix contains the probabilities of moving from one state to the next. The emission matrix contains the probabilities of emitting a particular symbol while in a particular state. In such a model, a state is comprised of both the condition of the stone knapper and the blank he is working. The emitted symbols represent the actions that he performed. In this third study, I used two simple Markov processes determine how well such a simple mindless process could simulate the sequenced actions performed by the stoneknapper replicating an Acheulean handaxe.

METHODS

The same coded lithic reduction used in the earlier studies was re-analyzed in this study. Two different Markov models were created for this analysis—a simple transition matrix model and a prior-knowledge model. For the Simple Markov (SMM) model, I determined the probabilities for transition from one action to another in the observed sequence, creating a transition matrix. Each ac-
tion was treated as a state. There were no emitted ‘symbols’ and the strings generated represented the progression of the state transitions. The model was run in R 2.15.1.

Containing only the frequencies of observed actions, the simple Markov model does not actually approximate the ‘string-of-beads’ model proposed by Moore (2010 and, in a different sense, by Wynn [1991]). This model contains latent variables, or states, that occur sequentially and emit an evoked response (the observed action). A state refers not only to the condition of the core, but also to the condition of the mind of the knapper. A Prior Knowledge (PKM) model was constructed by the author using knowledge of the process by which Acheulean handaxes are made. In this model, the transition matrix contains theoretical probabilities for the transfer from one state to the next. An emission matrix contains the theoretical probabilities for the emission of a particular action or set of actions when the system is in that state. Theoretical values were used for this study simply to test the feasibility of system like this, built on empirically derived archaeological theories, to simulate strings of actions that resemble observed strings. The shaping and thinning phase were built into this model, with a transition between shaping and thinning modeled as a low probability event. See the discussion below for further comment on this approach. This model was run in R using the HMM package (Himmelmann 2010). The PKM model is presented visually in Figure 10.

The performance of these two simple Markov models was assessed both qualitatively and quantitatively. Observed actions and simulated strings were assessed in terms of five features. Procedural coherence (1) refers to the all of the strings in a single run of a model. If the run had a simple shaping phase followed by a thinning phase, then it was procedurally coherent. Thinning is defined by the presence of chisel-like platform preparation and abrasion or grinding platform preparation. Each string was classified as shaping or thinning based on the presence or absence of these features. String coherence (2) refers to each individual string generated by a model. If the string follows a coherent flake schema (core placement, orientation, striking, flake removal), even if it is atypical of an actual flake removal episode as performed by the observed knapper, it is considered coherent. Chisel-like platform preparation/grinding preparation reversal (3) refers to the performance of these platform preparation techniques in a reversal of the expected order. Chisel-like platform preparation exclusive (4) refers to strings in which only this action is used to prepare a platform. Grinding platform preparation exclusive (5) refers to strings in which only this action is used to prepare a platform. Simple counts were made of the presence / absence of these features in both the observed and simulated action.

Next, I used two quantitative methods of determining how well the simulated strings approximated observed actions. Both methods use the concept of edit distance between strings. Levenshtein Similarity calculates the number of changes required to change one string of symbols, like “string,” into another, like “strung” (1 change) or “apple” (6 changes). These changes include both deletions and replacements. The distance between two strings \( \text{str}_A, \text{str}_B \) is then divided by the length of the longest string \( \max(\text{str}_A, \text{str}_B) \).
Acheulean Stoneknapping and Language

relative to the maximum possible complexity possible for that sequence given its length and the component alphabet. Gabadinho’s complexity is calculated using this formula:

\[ C(s) = \frac{(q(s)h(s))}{(q_{\text{max}}h_{\text{max}})} \]  

where \( s \) is the observed or simulated string, \( q(s) \) is the number of transitions in the sequence, \( h(s) \) is the within sequence entropy, \( q_{\text{max}} \) is the maximum number of transitions, \( h_{\text{max}} \) is the theoretical maximum entropy for this alphabet of coded actions:

\[ h_{\text{max}} = -\log \frac{1}{A} \]

Minimum values of zero can only be reached if the sequence is made up of only one state while maximum values represent a string with all possible component actions and maximum entropy.

**RESULTS**

Results are presented in Table 1 and Figure 10. Qualitatively, the ‘mindless’ Markov models could simulate sequences similar to those performed by the stone knapper.

<table>
<thead>
<tr>
<th>Class</th>
<th>Procedural Coherence</th>
<th>String Coherence</th>
<th>CPP-APP Reversal (%)</th>
<th>CPP Only (%)</th>
<th>APP Only (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed Lithic Reduction</td>
<td>Yes</td>
<td>Yes</td>
<td>0</td>
<td>0.02</td>
<td>0.11</td>
</tr>
<tr>
<td>Simple Markov</td>
<td>No</td>
<td>Yes</td>
<td>0.04</td>
<td>0.15</td>
<td>0.26</td>
</tr>
<tr>
<td>Prior-Knowledge Markov</td>
<td>Yes</td>
<td>Yes</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Finally, each observed or simulated sequence was analyzed using Gabadinho’s complexity index (Gabadinho et al. 2010). This index calculates the complexity of a sequence and subtracted from 1 to give an approximate similarity measure:

\[ 1 - \frac{d(str_1,str_2)}{\max(A,B)} \]

The Levenshtein Similarity (Levenshtein 1966) was calculated between: a) all string pairs in the observed lithic reduction and between b) each simulated string produced by the Markov models and each string in the observed lithic reduction: a) provides a sense of the variability and self-similarity within the observed lithic reduction. b) provides a sense of how well each simulated string approximates the observed sequences of action. This analysis was performed in the RecordLinkage package (Borg and Sariyar, 2012) in R 2.15.1.

A second edit distance metric was also generated — OM distance. OM distance calculates both the number of replacements and deletions required to transform one string of a particular class (“observed sequence, shaping phase” or “simple Markov model, thinning phase”). This provides an indication of how variable or self-similar each set of strings/sequences is. OM discrepancy was analyzed using Zapala and Schork’s (2006) extension of a Discrepancy Analysis (Studer et al. 2010, 2011) for multiple variables. This analysis was performed in R 2.15.1 using the TraMiner package. The discrepancy matrix is used as a pseudo sum of squares, allowing the application of the F statistic, reported here as pseudo F. A pseudo \( r^2 \) is also generated by the test, providing a further assessment of the strength of the effect. The effects of string/sequence class (observed x simple x prior knowledge) and phase (shaping x thinning) on discrepancy were assessed.

**TABLE 1. RESULTS OF THE QUANTITATIVE ANALYSIS OF THE OBSERVED LITHIC REDUCTION, SIMPLE MARKOV MODEL (SMM), AND PRIOR-KNOWLEDGE MARKOV MODEL (PKM).**

Procedural coherence refers to the presence of an initial shaping phase followed by a thinning phase. The SMM did not reproduce this feature. However, both the SMM and PKM produced coherent strings, meaning that they would result in the removal of a flake. Chisel-like (CPP) and Abrasion (APP) platform preparation reversal refers to the performance of abrasion before the chiseling of the edge. This reversal only occurred in the SMM. Both the observed lithic reduction and the SMM had strings/sequences in which only chiseling or only abrasion occurred, but these occurred at much higher frequencies in the SMM. These results indicate that the ‘mindless’ Markov models could simulate sequences similar to those performed by the stone knapper.
DISCUSSION

In Study 1, I was able to demonstrate that it is feasible to model the reduction of an Acheulean handaxe as a CFG. In this analysis, I was able to show that it is also possible to model the same sequences as a FSG using Markov models, though in this case the models could be significantly improved. Most of the features present in the observed strings were easily reproduced by a system lacking the higher level mechanisms required by the CFG mentioned above.

However, there are two issues with this study that should be highlighted. First, this study should be expanded and the models improved. The models were relatively crude. Note how the PKM model is structured with a shaping phase distinct from the thinning phase (see Figure 10). It could be argued that because the core has significantly changed its condition between these two phases, the states within the model are qualitatively different. To examine a core with a significant amount of cortex on it that the knapper will remove with a hard hammer may be fundamentally different from examining the core when you are removing thinning flakes with an antler billet. On the other hand, it may not be qualitatively different and all that actually changes after shifting from shaping to thinning are the relative probabilities of performing different actions as the reduction progresses. Other types of Markov models, such as time-varying Markov processes (Howard 2007), may be a more appropriate approach than that followed in this pilot study. In addition, Markov models can be derived directly from observed data or tuned based on observed data using machine-learning techniques such as the Baum-Welch algorithm.

The second issue is more theoretical. While it is possible to reproduce a sequence that mirrors the actions re-
produced during actual stoneknapping, the concept of a state is problematic in that it combines the mental state of the knapper, the physical state of his body, and the state of the core. In other words, this may be a false simplicity.

**DISCUSSION: ISSUES AND POSSIBILITIES**

These pilot studies provided three results:

1. The thinning phase of handaxe replication is as complex as many instances of spoken language.
2. Handaxe replication can be described using a Context-Free Grammar. It has a structure in which iterative units at two levels embed and concatenate actions onto a basic flake schema.
3. The observed actions of a stone knapper replicating a handaxe can be simulated by a Markov process.

As these studies found that both a CFG and Markov model fit the data well, the initial hypothesis that language and replicated Acheulean stoneknapping display a homology in terms of a shared deep grammatical structure is not supported by this analysis. As I will discuss below, this result can be attributed to methodological issues.

The ultimate goal of these three pilot studies was to explore the issues involved in modelling stoneknapping as a cognitive process, especially in relation to natural language. To compare these two behaviors, researchers must be able to place them within the same analytical and theoretical framework. The use of information theory and formal grammars can provide such a framework, but their practical application raises a number of issues.

In order to construct this framework, it is essential to determine that valid units occurring at the same organizational level are being compared. At this point, it is not clear what a valid unit of action occurring during stoneknapping actually is. To validate the units being analyzed, it may be useful to have both expert knappers and non-knappers view videos of stoneknappers at work and code it every time a new event (action) occurs. The analysis action using motion-tracking software, as in a series of studies by Bril and collaborators (Rein et al. 2013; Bril and Nonaka 2012; Bril et al. 2010; Nonaka et al. 2010; Foucart et al. 2005; Bril et al. 1996; Roux et al. 1995) could provide a more accurate and rigorous representation of the repetitive actions occurring as a knapper works.

Neuroimaging studies of stoneknapping may also help clarify this issue. Oldowan debitage activated bihemispheric parietal and cerebellar regions but not left lateralized frontal regions (Stout et al. 2011; Stout et al. 2008). Acheulean façonnage (shaping) activates these regions as well as additional right lateralized frontal regions. Stout et al. (2008) have proposed that the additional activations in the right hemisphere indicate the hierarchical cognitive organization of stoneknapping.

As these studies found that both a CFG and Markov model fit the data well, the initial hypothesis that language and replicated Acheulean stoneknapping display a homology in terms of a shared deep grammatical structure is not supported by this analysis. As I will discuss below, this result can be attributed to methodological issues.

The action grammar and Markov models used in these studies grossly simplify the cognitive processes involved in stoneknapping. Sensory and perceptual processes are simply assumed. The apparent action grammar of stoneknapping has always attracted considerable attention, perhaps distorting understanding of the cognitive foundations of stoneknapping and limiting research questions. Recent studies by researchers such as Blandine Bril and colleagues (Bril et al. 2010; Bril et al. 1996; Foucart et al. 2005; Nonaka and Bril 2012; Nonaka et al. 2010; Rein et al. 2013; Roux et al. 1995) have approached stoneknapping through radically different theoretical and methodological means, help-
ing to broaden understanding of the cognitive dimension of stone knapping considerably.

This is not to say that approaches similar to those presented in this paper are uninformative. To the contrary, they hold the promise of generating a deeper understanding of how stoneknapping is actually structured. Such an understanding is vital if evolutionary cognitive archaeologists are going to attempt to use technology as an index of cognitive evolution in the Pleistocene. The rich theoretical apparatus of formal languages and automata theory can be very useful in this regard. Consider the relationship between formal language and automata described earlier. An automata is a hypothetical computational mechanism, such as a Turing machine (Linz 2011; Turing 1936), capable of computing languages of different complexity. Finite state machines, automata instantiating Markov processes, are capable of computing FSGs. CFGs require a pushdown automaton. The crucial feature of a pushdown machine is the stack, which serves the role of storing memory. For instance, if given B, the system works on E while storing F, C, and B on the stack. This ‘working memory’ allows the system to hold the overall schema in mind while recursively embedding phrase-like platform preparation sequences. Analysis of the structure of Oldowan and Acheulean stone knapping, in relation to other lines of evidence, may indicate that Oldowan debitage does not require the neurocognitive equivalent of the pushdown stack while Acheulean façonage does.

This paper focused one dimension of the potential relationship between language and Acheulean lithic technology, the sequencing of verbal and technological actions. However, there are potential relationships along another dimension, social learning. More specifically, some researchers have asked if language is necessary to learn the skills and cultural recipes required for the production of an Acheulean handaxe. In an ethnoarchaeological study of contemporary stone knappers in Irian Jaya, Indonesia, Stout et al. (2002) found that adze makers worked in an intergenerational social context. They proposed that Acheulean knappers also may have transmitted technological knowledge in a similar way. Putt et al. (2014) have recently performed an experimental study in which novices were trained by an expert knapper using verbal and nonverbal cues. No significant difference was found between the handaxes and debitage produced by novices instructed differently. However, that study did not systematically control the content of the verbal interchanges between the novices and expert. Varying verbal strategies across groups may potentially yield different results. It is interesting, however, that this study reached ambivalent results regarding the role of language in the replication of Acheulean technologies.

The approaches piloted in these studies can provide a valuable and productive, though not unproblematic, framework for research into cognitive evolution during the Pleistocene. Future studies should address some of the issues raised here to move evolutionary cognitive archaeology forward from the analysis of surface features to deeper structural analyses.

REFERENCES


Borg, A. and Sariyar, M. 2012. RecordLinkage: Record Linkage in R, 0.4-1. CRAN Archive.


Shannon, C.E. 1951a. Prediction and entropy of printed
EXAMPLE 1:

Observed Sequence: ghabjkeop

1. Intends to remove a shaping flake with a hard hammer
2. Lift blank from thigh
3. Flip blank in hand
4. Insert flake removal sequence EFCB
5. Place blank on thigh
6. Orient lateral edge horizontally
7. Gauges blow
8. Strikes
9. Rotates lateral edge up, into view
10. Picks up debitage between fingers
11. Sweeps away debitage
12. Deletes concatenating B
13. Completed flake unit

Analysis: This is a simple flake unit without any concatenated or embedded units.

EXAMPLE 2:

Observed Sequence: ghadm′epcjkeo

1. Intends to remove a shaping flake with a hard hammer
2. Lift blank from thigh
3. Flip blank in hand
4. Insert flake removal sequence EFCB
5. Insert platform preparation sequence HIE
6. Delete chisel-like platform preparation sequence H
7. Insert abrasion sequence EGCAIO
8. Place blank on thigh
9. Orient lateral edge, raising
10. Abrasion
11. Abrasion [Repeat 2 more times]
12. Rotates lateral edge up, into view
13. Sweeps away debitage
14. Deletes AIO
15. Insert M
16. Orient lateral edge, lowering
17. Gauges blow
18. Strike
19. Strike
20. Picks up debitage between fingers
21. Deletes concatenating B
22. Completed flake unit

Analysis: This flake unit is somewhat atypical for this stoneknapper’s shaping phase in that he embedded a platform preparation sequence within it (5–13). Most of the platform preparation sequence as not performed, only those aspects of the abrasion sequence of use to the knapper.

EXAMPLE 3:

Observed Sequence: ghaejk′epckeck′ep

1. Intends to remove a shaping flake with a hard hammer
2. Lift blank from thigh
3. Rotate edge up into view after lifting
4. Flip blank in hand
5. Insert flake removal sequence EFCB
6. Place blank on thigh
7. Rotate edge up into view while blank on thigh
8. Delete M
9. Gauges blow
EXAMPLE 4:

**Observed Sequence:** ffabl^λgfgabkep

1. Intends to remove a shaping flake with a hard hammer
2. Blank already lifted, insert D
3. Rotates edge up into view after lifting
4. Rotates edge up into view after lifting
5. Insert platform preparation sequence HIE
6. Insert platform preparation sequence ENCAH
7. Insert chisel-like platform preparation sequence ENCAH
8. Place blank on thigh
9. Orient lateral edge horizontally
10. Chisel-like platform preparation
11. Chisel-like platform preparation [Repeat 3 times]
12. Delete C
13. Lifts blank
14. Rotates edge up into view after lifting
15. Deletes abrasion sequence I
16. Place blank on thigh
17. Orient lateral edge horizontally
18. Skips gauging blow, inserts J
19. Strike
20. Rotates lateral edge up, into view
21. Sweeps away debitage
22. Deletes concatenating B
23. Completed flake unit

**Analysis:** This is a simple flake unit from the shaping phase with two additional concatenated flake removals (14–19, 20–28).

**Analysis:** This is a somewhat atypical elaborated flake unit from the thinning phase with an embedded platform preparation sequence (7–13). Abrasion was skipped by the stoneknapper (14–16).
EXAMPLE 5:

Observed sequence:
gfabl^0egfabl^1l^1cl^4ghacmolqckj^4epjk^2ep

1. Intends to remove a shaping flake with a hard hammer
2. Lift blank from thigh
3. Rotates edge up into view after lifting
4. Insert flake removal sequence EFCB
5. Insert platform preparation sequence HIE
6. Insert chisel-like platform preparation sequence ENCAH
7. Place blank on thigh
8. Orient lateral edge horizontally
9. Chisel-like platform preparation
10. Chisel-like platform preparation [Repeat 18 times]
11. Orient lateral edge up into view with blank on thigh
12. Lift blank from thigh
13. Rotates edge up into view after lifting
14. Insert chisel-like platform preparation sequence
15. Place blank on thigh
16. Orient lateral edge horizontally
17. Chisel-like platform preparation
18. Chisel-like platform preparation [Repeat 21 times]
19. Delete AC
20. Insert chisel-like platform preparation sequence
21. Insert M
22. Orient lateral edge horizontally
23. Chisel-like platform preparation
24. Chisel-like platform preparation [Repeat 9 times]
25. Delete AC
26. Insert chisel-like platform preparation sequence ENCAH
27. Insert M
28. Orient lateral edge, lowering edge
29. Chisel-like platform preparation
30. Chisel-like platform preparation [Repeat 47 times]
31. Delete AC
32. Lifts blank
33. Flips blank
34. Deletes embedding H
35. Inserts abrasion sequence EGCAIO
36. Place blank on thigh
37. Orient lateral edge, lowering edge
38. Abrasion
39. Abrasion [Repeats 6 times]
40. Deletes CA and embedding I
41. Set down hammer, pickup antler baton
42. Already on thigh, insert M
43. Orient lateral edge, lowering edge
44. Gauges blow
45. Strike
46. Strike
47. Rotates lateral edge up, into view
48. Sweeps away debitage
49. Concatenates with flake removal sequence EFCB
50. Inserts M
51. Orient lateral edge horizontally

Observed sequence:
gfabl^0egfabl^1l^1cl^4ghacmolqckj^4epjk^2ep

1. T → AB
2. AB → gDB
3. gDB → gB
4. gB → gfEFCB
5. gfEFCB → gHIEFCB
6. gHIEFCB → gENCAHIEFCB
7. gENCAHIEFCB → gfaMNCAHIEFCB
8. gfaMNCAHIEFCB → gfabNCAHIEFCB
9. gfabNCAHIEFCB → gfabfNCAHIEFCB [Repeat 18 times]
11. gfabf^2NCAHIEFCB → gfabl^2eAHIEFCB
12. gfabl^2eAHIEFCB → gfabl^2egDHIEFCB
13. gfabl^2egDHIEFCB → gfabl^2eggHIEFCB
14. gfabl^2eggHIEFCB → gfabl^2egfENCAHIEFCB
15. gfabl^2egfENCAHIEFCB → gfabl^2egfaMNCNCAHIEFCB
16. gfabl^2egfaMNCNCAHIEFCB → gfabl^2egfabNCAHIEFCB
17. gfabl^2egfabNCAHIEFCB → gfabl^2egfblNCAHIEFCB
18. gfabl^2egfblNCAHIEFCB → gfabl^2egfabfNCAHIEFCB [Repeat 21 times]
19. gfabl^2egfabfNCAHIEFCB → gfabl^2egfabf^2l^2HIEFCB
20. gfabl^2egfabf^2l^2HIEFCB → gfabl^2egfabf^2l^2ENCAHIEFCB
21. gfabl^2egfabf^2l^2ENCAHIEFCB → gfabl^2egfabf^2l^2MNCAHIEFCB
22. gfabl^2egfabf^2l^2MNCAHIEFCB → gfabl^2egfabf^2l^2bNCAHIEFCB
23. gfabl^2egfabf^2l^2bNCAHIEFCB → gfabl^2egfabf^2l^2bNCAHIEFCB
24. gfabl^2egfabf^2l^2bNCAHIEFCB → gfabl^2egfabf^2l^2bNCAHIEFCB [Repeat 9 times]
25. gfabl^2egfabf^2l^2bNCAHIEFCB → gfabl^2egfabf^2l^2bNCAHIEFCB
26. gfabl^2egfabf^2l^2bNCAHIEFCB → gfabl^2egfabf^2l^2bNCAHIEFCB
27. gfabl^2egfabf^2l^2bNCAHIEFCB → gfabl^2egfabf^2l^2bNCAHIEFCB
28. gfabl^2egfabf^2l^2bNCAHIEFCB → gfabl^2egfabf^2l^2bNCAHIEFCB

70. Completed flake unit