The *SP-multiple-alignment* concept as a generalisation of six other variants of 'Information Compression via the Matching and Unification of Patterns'

J Gerard Wolff*

December 30, 2021

Abstract

This paper focusses on the powerful concept of *SP-multiple-alignment*, a key part of the SP System (SPS), meaning the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*. The SPS is outlined in an appendix.

More specifically, the paper shows with examples how the SP-multiple-alignment construct may function as a generalisation of six other variants of 'Information Compression via the Matching and Unification of Patterns' (ICMUP).

Each of those six variants is described in a separate section, and in each case there is a demonstration of how that variant may be modeled via the SP-multiple-alignment construct.

1 Introduction

As its title suggests, this article describes the concept of *SP-multiple-alignment* (SPMA) and it describes how it may be seen as an example of the concept of 'Information Compression via the Matching and Unification of Patterns' (ICMUP), a concept which is an example of ICMUP and which provides a generalisation of six other variants of ICMUP.

*Dr Gerry Wolff BA (Cantab) PhD (Wales) CEng MIEEE MBCS; ORCID ID: 0000-0002-4624-8904; CognitionResearch.org, UK; jgw@cognitionresearch.org; +44 (0) 1248 712962; +44 (0) 7746 290775; *Twitter*: @gerrywolff65; *Web*: www.cognitionresearch.org.

The SPMA concept is a key part of the *SP System* (SPS) meaning the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*. The SPS is outlined in [8, Section 3] with more detail elsewhere in that paper. It is described more fully in [6]. As will be seen, the SPS works entirely via ICMUP.

Following this introduction, Section 2 describes the origins, development and workings of the SPMA concept.

After that, Section 3 describes the ICMUP concept itself, shows how it may be seen as an example of ICMUP, and describes six other variants of ICMUP. For each of the six variants, other than the SPMA, there is a description of how it may be modeled within the SPMA framework.

2 SP-multiple-alignment

It is largely the strengths and potential of the SPMA concept that provides the strengths and potential of the SPS, outlined in Appendices **??**, A, and B.

Bearing in mind that it is just as bad to underplay the strengths and potential of a system as it is to oversell its strengths and potential, it seems fair to say that *the concept of SP-multiple-alignment may prove to be as significant for an understanding of 'intelligence' as is DNA for biological sciences. It may prove to be the 'double helix' of intelligence.*

2.1 Multiple sequence alignments

The SPMA concept has been borrowed and adapted from the concept of 'multiple sequence alignment' in bioinformatics. Figure 1 shows an example. Here, there are five DNA sequences which have been arranged alongside each other, and then, by judicious 'stretching' of one or more of the sequences in a computer, symbols that match each other across two or more sequences have been brought into line.

```
  G G A     G     C A G G G A G G A     T G     G   G G A
  | | |     |     | | | | | | | | |     | |     |   | | |
  G G | G   G C C C A G G G A G G A     | G G C G   G G A
  | | |     | | | | | | | | | | | |     | |     |   | | |
A | G A C T G C C C A G G G | G G | G C T G     G A | G A
  | | |           | | | | | | | | |   |   |     |   | | |
  G G A A         | A G G G A G G A   | A G     G   G G A
  | |   |         | | | | | | | |     |   |     |   | | |
  G G C A         C A G G G A G G     C   G     G   G G A
```

Figure 1: A 'good' multiple sequence alignment amongst five DNA sequences.

A 'good' multiple sequence alignment, like the one shown, is one with a relatively large number of matching symbols from row to row.

The process of discovering a good multiple sequence alignment is normally too complex to be done by exhaustive search, so heuristic methods are needed, building multiple sequence alignments in stages and, at each stage, selecting the best partial structures for further processing.

Some people may argue that the combinational explosion with this kind of problem, and the corresponding computational complexity, is so large that there is no practical way of dealing with it. In answer to that objection, there are several multiple sequence alignment programs used in bioinformatics—such as 'Clustal Omega', 'Kalign', and 'MAFFT'[1]—which produce results that are good enough for practical purposes.

This relative success is achieved via the use of heuristic methods that conduct the search for good structures in stages, discarding all but the best alignments at the end of each stage. With these kinds of methods, reasonably good results may be achieved but normally they cannot guarantee that the best possible result has been found.

2.2 How SP-multiple-alignments are created

Figure 2 shows an example of an SPMA, superficially similar to the one in Figure 1, except that:

- The sequences are called *SP-patterns*;
- The SP-pattern in row 0 is 'New' information, meaning information recently received from the system's environment;
- Each of the remaining SP-patterns, one per row, is an 'Old' SP-pattern, selected from a relatively large pool of such SP-patterns. Unless they are supplied by the user, the Old SP-patterns are derived via unsupervised learning (Appendix **??**) from previously received New information;
- Each SPMA is scored in terms of how well the New SP-pattern may be encoded economically in terms of the Old SP-patterns as described in [8, Section 4.1] and [6, Section 3.5].

In this example, the New SP-pattern (in row 0) is a sentence and each of the remaining SP-patterns represents a grammatical category, where 'grammatical categories' include words. The overall effect of the SPMA in this example is the parsing of a sentence ('`f o r t u n e f a v o u r s t h e b r a v e`') into its grammatical parts and sub-parts.

[1]Provided as online services by the European Bioinformatics Institute (see https://www.ebi.ac.uk/Tools/msa/).

```
0                          t w o              k i t t e n     s                    p l a y            0
                           | | |              | | | | | |     |                    | | | |
1                          | | |         Nr 5 k i t t e n #Nr |                    | | | |            1
                           | | |         |                 |  |                    | | | |
2                          | | |    N Np Nr               #Nr s #N                 | | | |            2
                           | | |    | |                         |                  | | | |
3                   D Dp 4 t w o #D | |                         |                  | | | |            3
                    | |          |  | |                         |                  | | | |
4            NP NPp D Dp         #D N Np                        #N #NP             | | | |            4
             |   |                                                  |              | | | |
5            |   |                                                  |         Vr 1 p l a y #Vr        5
             |   |                                                  |         |             |
6            |   |                                                  |  VP VPp Vr           #Vr #VP    6
             |   |                                                  |  |   |                    |
7 S Num    ; NP  |                                                 #NP VP  |                   #VP #S 7
     |     |     |                                                         |
8   Num PL ;    NPp                                                       VPp                         8
```

Figure 2: The best SPMA created by the SP Computer Model with a store of Old SP-patterns like those in rows 1 to 8 (representing grammatical structures, including words) and a New SP-pattern, '(`t w o k i t t e n s p l a y`)', shown in row 0 (representing a sentence to be parsed). Adapted from Figure 1 in [7], with permission.

As with multiple sequence alignments, it is almost always necessary to use heuristic methods to achieve useful results without undue computational demands. The use of heuristic methods helps to ensure that computational complexities in the SPS are within reasonable bounds [6, Sections A.4, 3.10.6 and 9.3.1]. Each SP-multiple-alignment is built up progressively, starting with a process of finding good alignments between pairs of SP-patterns. At the end of each stage, SP-multiple-alignments that score well in terms of IC are retained and the rest are discarded. There is more detail in [8, Section 4] and [6, Sections 3.4 and 3.5].

In the SPCM, the size of the memory available for searching may be varied, which means in effect that the scope for backtracking can be varied. When the scope for backtracking is increased, the chance of the program getting stuck on a 'local peak' (or 'local minimum') in the search space is reduced.

Contrary to the impression that may be given by Figure 2, the SPMA concept is very versatile and is largely responsible for the strengths and potential of the SPS, as described in Appendix A.

2.3 The SPMA concept and ICMUP

2.4 The process of compressing information can be as useful as the end-result of information compression

All kinds of processing in the SPS is achieved by compressing information.

With a process like the parsing of a sentence, the main interest is in the final result.

But with processes such as probabilistic reasoning or problem solving, the main interest is in what the SP System does on the way to achieving the maximum IC that the SPS can achieve.

3 Six variants of ICMUP and how each one may be modelled via the SPMA construct

For each of the six variants of ICMUP, described in this section, there is a subsection explaining how it may be modelled within the SPMA framework.

3.1 Basic ICMUP

The simplest of the techniques to be described—referred to as *Basic ICMUP*—is to find two or more patterns that match each other within a given body of information, **I**, and then merge or 'unify' them so that multiple instances are reduced to one.

This is illustrated in the upper part of Figure 3 where two instances of the pattern '`INFORMATION`' near the top of the figure has been reduced to one instance, shown just above the middle of the figure. Below it, there is the pattern '`INFORMATION`', with '`w62`' appended at the front, for reasons given in Section 3.2, below.

Here, and in subsections below, we shall assume that the single pattern which is the product of unification is placed in some kind of dictionary of patterns that is separate from **I**.

A detail that should not distract us from the main idea is that, when compression of a body of information, **I**, is to be achieved via Basic ICMUP, any repeating pattern that is to be unified should occur more often in **I** than one would expect by chance for a pattern of that size.

3.1.1 Expression of Basic ICMUP within the SPMA framework

The elements of Basic ICMUP can be seen in several parts of Figure 2. For example, '`t w o`' in row 0 matches '`t w o`' in row 3. This means that, when all the rows in the SPMA are unified, those two patterns will be unified. Likewise for '`k i t t e n`' in row 0 and '`k i t t e n`' in row 1, and so on.

3.2 Chunking-with-Codes

A point that has been glossed over in describing Basic ICMUP is that, when a body of information, **I**, is to be compressed by unifying two or more instances of a

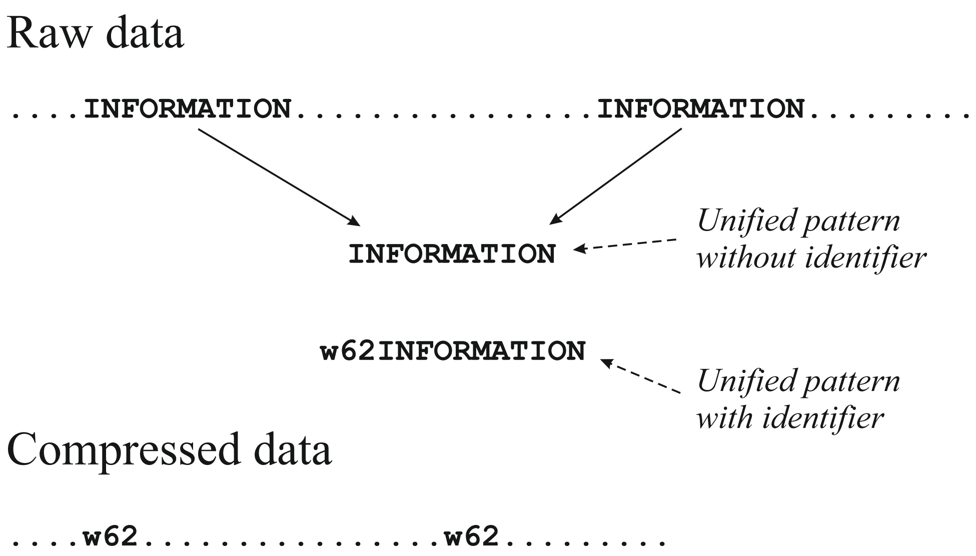

Figure 3: A schematic representation of the way two instances of the pattern '`INFORMATION`' in a body of data may be unified to form a single 'unified pattern', shown just above the middle of the figure. To achieve lossless compression, the relatively short identifier '`w62`' may be assigned to the unified pattern '`INFORMATION`', as shown below the middle of the figure. At the bottom of the figure, the original data may be compressed by replacing each instance of '`INFORMATION`' with a copy of the relatively short identifer, '`w62`'. Adapted from Figure 2.3 in [6], with permission.

pattern like '`INFORMATION`', there is a loss of information about the *location* within **I** of each instance of the pattern '`INFORMATION`'. In other words, Basic ICMUP achieves 'lossy' compression of **I**.

This problem may be overcome with the *Chunking-with-Codes* variant of ICMUP:

- A unified pattern like '`INFORMATION`', which is often referred to as a 'chunk' of information,[2] is stored in a dictionary of patterns, as mentioned in Section 3.1.
- Now, the unified chunk is given a relatively short name, identifier, or 'code', like the '`w62`' pattern appended at the front of the '`INFORMATION`' pattern, shown below the middle of Figure 3.
- Then the '`w62`' code is used as a shorthand which replaces the '`INFORMATION`' chunk of information wherever it occurs within **I**. This is shown at the bottom of Figure 3.
- Since the code '`w62`' is shorter than each instance of the pattern '`INFORMATION`' which it replaces, the overall effect is to shorten **I**. But, unlike Basic ICMUP, Chunking-with-Codes may achieve 'lossless' compression of **I** because the original information may be retrieved fully at any time.
- Details here are:
  - 1) That compression can be optimised by giving shorter codes to chunks that occur frequently and longer codes to chunks that are rare. This may be done using some such scheme as Shannon-Fano-Elias coding, described in, for example, [2]; and
  - 2) By ensuring that for any chunk, **C**, to be given this treatment, it should be more frequent in **I** than the minimum needed (for a chunk of the size of **C**) to achieve compression (Section 3.1).

3.2.1 The concept of 'chunk' and the concept of 'object'

The concept of a 'chunk' of information, which became prominent following George Miller's paper "The magical number seven, plus or minus two: some limits on our capacity for processing information" [4], is normally applied to things like words which represent relatively large amounts of redundancy in the world via a favourable combination of frequency and size.

As such, there is a clear similarity between chunks such as words and chunks such as three-dimensional objects because they can both be seen to represent

[2]There is a little more detail about the concept of 'chunk' in [16, Section 2.4.2].

redundancies in the world in the form of recurrent bodies of information. But of course 3D objects have two more dimensions compared with words.

In general, the concept of a chunk is of central importance in these variants of ICMUP because of its importance, not only in the chunking-with-codes variant but also in the schema-plus-correction variant (Section 3.3), the Run-Length Coding variant (Section 3.4), the Class-Inclusion Hierarchies variant (Section 3.5), and the Part-Whole Hierarchies variant (Section 3.6) as well.

3.2.2 Expression of Chunking-with-Codes within the SPMA framework

Within Figure 2, the Chunking-with-Codes techniques can be seen, for example, in the SP-pattern '`Nr 5 k i t t e n #Nr`', where the SP-symbols '`Nr`', '`5`', and '`#Nr`', may be seen as codes for the SP-pattern '`k i t t e n`'. The reason that there are several codes and not just one is to do with the way in which the SPMA is created. Several other examples may be seen in the same figure.

3.3 Schema-plus-Correction

A variant of the Chunking-with-Codes version of ICMUP is called *Schema-plus-Correction*. Here, the 'schema' is like a chunk of information which, as with Chunking-with-Codes, has a relatively short identifier or code that may be used to represent the chunk.

What is different about the Schema-plus-Correction idea is that the schema may be modified or 'corrected' in various ways on different occasions. This is illustrated in an example with the SPMA construct, next.

3.3.1 Expression of the Schema-plus-Correction concept within the SPMA framework

Figure 4 shows a set of SP-patterns which describes, in very simplified form, how a menu may be prepared in a cafe or restaurant. As such, it illustrates the Schema-plus-Correction variant of ICMUP.

The first SP-pattern in the figure, '`PM ST #ST MC #MC PD #PD #PM`', may be seen as a schema for preparing a meal where the two SP-symbols '`PM ... #PM`' may be seen as the identifier or code for the schema, and '`PM`' is short for 'prepare meal'.

In the same SP-pattern, the pair of SP-symbols '`ST #ST`' may serve as a slot where the starter may be specified, the pair of SP-symbols '`MC #MC`' may serve as a slot for the main course, and the pair of SP-symbols '`PD #PD`' may serve as a slot for the putting.

```
PM ST #ST MC #MC PD #PD #PM  | Prepare meal
ST 0 mussels #ST             | Starter: prepare a dish of mussels
ST 1 soup #ST                | Starter: prepare a bowl of soup
ST 2 avocado #ST             | Starter: prepare an avocado dish
MC 0 lassagne #MC            | Main course: prepare a lassagne dish
MC 1 beef #MC                | Main course: prepare a beef dish
MC 2 nut-roast #MC           | Main course: prepare a nut-roast dish
MC 3 kipper #MC              | Main course: prepare a kipper
MC 4 salad #MC               | Main course: prepare a salad
PD 0 ice cream #PD           | Pudding: prepare ice cream
PD 1 apple crumble #PD       | Pudding: prepare apple crumble
PD 2 fresh fruit #PD         | Pudding: prepare fresh fruit
PD 3 tiramisu #PD            | Pudding: prepare tiramisu
```

Figure 4: A set of SP-patterns (an SP-grammar) comprising a set of SP patterns representing, in a highly simplified form, the kinds of procedures involved in preparing a mean for a customer in a restaurant or cafe. To the right of each SP pattern is an explanatory comment, after the character '`|`'.

Each of the other SP-patterns in Figure 4 describes a dish, each one marked as a starter, or a main course, or a pudding.

We can see how this works in Figure 5. This is the best SPMA created by the SP Computer Model with the New SP-pattern '`PM 0 4 1 #PM`' and Old SP-patterns from Figure 4.

Unlike the SPMA in Figure 2, this SPMA is rotated by 90°. The two arrangements are equivalent: the choice depends largely on what fits best on the page.

Here, we can see how the SP-symbol '`0`' in column 0 has selected the starter dish 'mussels' (column 3), the SP-symbol '`4`' in column 0 has selected the main course 'salad' (column 2), and the SP-symbol '`1`' in column 0 has selected the pudding 'apple crumble' (column 4).

In short, the SP-pattern '`PM ST #ST MC #MC PD #PD #PM`' in Figure 4 may be 'corrected' by the SP-pattern '`PM 0 4 1 #PM`' to create the whole menu. In particular, the whole system allows any particular meal to be specified very economically with an SP-pattern like '`PM 0 4 1 #PM`'.

3.4 Run-Length Coding

A third variant, *Run-Length Coding*, may be used where there is a sequence of two or more copies of a pattern, each one except the first following immediately after its predecessor like this:

'`INFORMATIONINFORMATIONINFORMATIONINFORMATIONINFORMATION`'.

```
0     1     2       3         4

PM -- PM                               | Prepare meal
      ST ---------- ST                 | Prepare starter
0 ----------------- 0
                    mussels
      #ST --------- #ST
      MC -- MC                         | Prepare main course
4 --------- 4
            salad
      #MC - #MC
      PD -------------------- PD       | Prepare pudding
1 --------------------------- 1
                              apple
                              crumble
      #PD ------------------- #PD
#PM - #PM

0     1     2       3         4
```

Figure 5: The best SP-multiple-alignment created by the SP Computer Model with the New pattern, '`PM 0 4 1 #PM`', and the set of Old patterns shown in Figure 4. Comments are shown on the right, each one following the symbol '`|`'.

In this case, the multiple copies may be reduced to one, as before, something like '`INFORMATION`×5', where '×5' shows how many repetitions there are; or something like '`[INFORMATION*]`', where '`[`' and '`]`' mark the beginning and end of the pattern, and where '`*`' signifies repetition (but without anything to say when the repetition stops).

In a similar way, a sports coach might specify exercises as something like "touch toes (×15), push-ups (×10), skipping (×30), ..." or "Start running on the spot when I say 'start' and keep going until I say 'stop' ".

With the 'running' example, "start" marks the beginning of the sequence, "keep going" in the context of "running" means "keep repeating the process of putting one foot in front of the other, in the manner of running", and "stop" marks the end of the repeating process. It is clearly much more econonomical to say "keep going" than to constantly repeat the instruction to put one foot in front of the other.

3.4.1 Expression of the Run-Length Coding concept within the SPMA framework

In the SPMA framework, the Run-Length Coding concept may be expressed via recursion, as shown in Figure 6. From top to bottom, this shows the sequence '`procedure-A`', '`procedure-B`', '`procedure-C`', and '`procedure-D`'. But '`procedure-B`' is repeated three times via the self-referential SP-pattern '`ri ri1` `ri #ri b #b #ri`' which appears in columns 5, 7, and 9.

This SP-pattern is self-referential because the pair of SP-symbols '`ri ... #ri`' at the beginning and end of the SP-pattern matches the same pair of SP-symbols in the body of the SP-pattern: '`ri #ri`'.

```
0     1             2     3             4             5     6             7     8             9     10

pg ---------------- pg
                    a ----------------------------------------------------------------------------- a
a6 ------------------------------------------------------------------------------------------------ a6
                                                                                                    procedure-A
                    #a ---------------------------------------------------------------------------- #a
                    ri ------------------------------ ri
                                                      ri1
                                                      ri ---------------- ri
                                                                          ri1
                                                                          ri ---------------- ri
                                                                                              ri1
                                                                                              ri
                                                                                              #ri
                                                                                b ----------- b
b1 ---------------------------------------------------------------------------- b1
                                                                                procedure-B
                                                                                #b ---------- #b
                                                                          #ri --------------- #ri
                                                            b ----------- b
b1 -------------------------------------------------------- b1
                                                            procedure-B
                                                            #b ---------- #b
                                                      #ri --------------- #ri
                                        b ----------- b
b1 ------------------------------------ b1
                                        procedure-B
                                        #b ---------- #b
                    #ri ----------------------------- #ri
      c ----------- c
c4 -- c4
      procedure-C
      #c ---------- #c
                    d --- d
d3 ---------------------- d3
                          procedure-D
                    #d -- #d
#pg --------------- #pg

0     1             2     3             4             5     6             7     8             9     10
```

Figure 6: An SP-multiple-alignment produced by the SP model showing how, via recursion mediated by a self-referential pattern (which is '`ri ri1 ri #ri b #b #ri`' in this example), the SP system may model the repetition of a procedure or function, which in this examples is shown as '`procedure-B`'.

An important point here is that the recursive structure is only possible because the SPMA concept allows any given SP-pattern to appear two or more times in any one SPMA. In this case, the SP-patterns which, individually, appear more than once in the SPMA are the SP-patterns '`ri ri1 ri #ri b #b #ri`' and '`b b1 procedure-B #b`'.

3.5 Class-Inclusion Hierarchies

A widely-used idea in everyday thinking and elsewhere is the *Class-Inclusion Hierarchy*: the grouping of entities into classes, and the grouping of classes into higher-level classes, and so on, through as many levels as are needed.

This idea may achieve ICMUP because, at each level in the hierarchy, attributes may be recorded which apply to that level and all levels below it—so economies may be achieved because, for example, it is not necessary to record that cats have fur, dogs have fur, rabbits have fur, and so on. It is only necessary to record that mammals have fur and ensure that all lower-level classes and entities can 'inherit' that attribute.

In effect, multiple instances of the attribute 'fur' have been merged or unified to create that attribute for mammals, thus achieving compression of information.[3]

This idea may be generalised to cross-classification, where any one entity or class may belong in one or more higher-level classes that do not have the relationship superclass/subclass, one with another. For example, a given person may belong in the classes 'woman' and 'doctor' although 'woman' is not a subclass of 'doctor' and *vice versa*.

3.5.1 Expression of the Class-Inclusion Hierarchies concept within the SPMA framework

In Figure 7, the SP-multiple-alignment produced by the SP Computer Model shows how a previously unknown entity with features shown in the New SP-pattern in column 1 may be recognised at several levels of abstraction: as an animal (column 1), as a mammal (column 2), as a cat (column 3) and as the specific cat 'Tibs' (column 4). These are the kinds of classes used in ordinary systems for OOD/OOP.

From this SP-multiple-alignment, we can see how the entity that has now been recognised *inherits* unseen characteristics from each of the levels in the class hierarchy: as an animal (column 1) the creature '`breathes`' and '`has-senses`', as a mammal it is '`warm-blooded`', as a cat it has '`carnassial-teeth`' and

[3]The concept of class-inclusion hierarchies with inheritance of attributes is quite fully developed in object-oriented programming, which originated with the Simula programming language [1] and is now widely adopted in modern programming languages.

```
0           1            2              3                  4

                                                           T
                                                           Tibs
                                        C ---------------- C
                                        cat
                         M ------------ M
                         mammal
            A ---------- A
            animal
            head ---------------------- head
                                        carnassial-teeth
            #head --------------------- #head
            body ----------------------------------------- body
white-bib ------------------------------------------------ white-bib
            #body ---------------------------------------- #body
            legs ---------------------- legs
                                        retractile-claws
            #legs --------------------- #legs
eats ------ eats
            breathes
            has-senses
            ...
            #A --------- #A
furry ------------------ furry
                         warm-blooded
                         ...
                         #M ----------- #M
purrs --------------------------------- purrs
                                        ...
                                        #C --------------- #C
                                                           tabby
                                                           ...
                                                           #T

0           1            2              3                  4
```

Figure 7: The best SP-multiple-alignment found by the SP model, with the New SP-pattern '`white-bib eats furry purrs`' shown in column 1, and a set of Old SP-patterns representing different categories of animal and their attributes shown in columns 1 to 4. Reproduced with permission from Figure 6.7 in [6].

'`retractile-claws`', and as the individual cat Tibs it has a '`white-bib`' and is '`tabby`'.

3.6 Part-Whole Hierarchies

Another widely-used idea is the *Part-Whole Hierarchy* in which a given entity or class of entities is divided into parts and sub-parts through as many levels as are needed. Here, ICMUP may be achieved because two or more parts of a class such as 'car' may share the overarching structure in which they all belong. So, for example, each wheel of a car, the doors of a car, the engine or a car, and so on, all belong in the same encompassing structure, 'car', and it is not necessary to repeat that enveloping structure for each individual part.

3.6.1 Expression of the Part-Whole Hierarchies concept within the SPMA framework

Figure 8 shows how a part-whole hierarchy may be accommodated in the SP system. Here, an SP-pattern representing the concept of a car is shown in column 2, with parts such as '<engine>' '<body>', and '<gearbox>'. The SP-pattern in column 1 shows parts of an engine such as '<cylinder-block>', '<pistons>', and '<crankshaft>'. The SP-pattern in column 3 shows how the body may be divided into such things as '<steering-wheel>', '<dashboard>', and '<seats>'. The SP-pattern in column 5 divides the dashboard into parts that include '<speedometer>' and '<fuel-gauge>'. And the SP-pattern in column 4 divides the speedometer into '<speed-dial>', '<speed-pointer>', and more.

The kind of structure shown in Figure 8 exhibits a form of inheritance, much like inheritance in a class-inclusion hierarchy. In this case, recognition of something as a '<speed-dial>' suggests that it is likely to be part of a '<dashboard>', which itself is likely to be part of the '`body`' of a '<car>'. This kind of inference is the kind of thing that crime investigators will do: search for a missing body when a severed human arm has been discovered.

3.7 Generalisation of ICMUP via SP-multiple-alignment

As we have seen in preceding sections, the seventh version of ICMUP, the *SP-multiple-alignment* construct outlined in Section 2, encompasses all the preceding six versions of ICMUP.

As already mentioned, It is largely the strengths and potential of the SPMA concept that provides the strengths and potential of the SPS, outlined in Appendices **??**, A, and B. There is much more detail in [8], and even more detail in [6].

```
0                  1                  2           3                  4                 5

                                      <car>
                                      c1
                   <engine> --------- <engine>
                   e1
                   <cylinder-block>
                   <pistons>
<crankshaft> ----- <crankshaft>
                   ...
                   </engine> -------- </engine>
                                      <body> ---- <body>
                                                  b1
<steering-wheel> -------------------------------- <steering-wheel>
                                                  <dashboard> ------------------------ <dashboard>
                                                                                       d1
                                                                     <speedometer> --- <speedometer>
                                                                     sp1
<speed-dial> ------------------------------------------------------- <speed-dial>
                                                                     <speed-pointer>
                                                                     ...
                                                                     </speedometer> -- </speedometer>
                                                                                       <fuel-gauge>
                                                                                       ...
                                                  </dashboard> ----------------------- </dashboard>
                                                  <seats>
                                                  ...
                                      </body> --- </body>
<gearbox> --------------------------- <gearbox>
                                      ...
                                      </car>

0                  1                  2           3                  4                 5
```

Figure 8: The best SP-multiple-alignment found by the SP Computer Model, with the New SP-pattern '`<crankshaft> <steering-wheel> <speed-dial> <gearbox>`' shown in column 1, and a set of Old SP-patterns representing different parts and sub-parts of a car, shown in columns 1 to 5.

3.7.1 Seamless integration of class-inclusion and part-whole hierarchies

A neat feature of the SP system is that, because diverse kinds of knowledge may be represented within the framework of SP-multiple-alignments, there can be seamless integration of diverse kinds of knowledge in any combination (Appendix A.4).

This aspect of the SP system means that there can be seamless integration of class-inclusion and part-whole hierarchies, as described and illustrated in [8, Section 9.1, Figure 16].

4 Conclusion

Appendices

A Strengths and potential of the SPS in intelligence-related functions and knowledge

The strengths and potential of the SPS in intelligence-related functions and knowledge are summarised in the subsections that follow. Further information may be found in [8, Sections 5 to 12], [6, Chapters 5 to 9], [14], and in other sources referenced in the subsections that follow.

In view of the relative Simplicity of the SPS, the strengths and potential of the system summarised here mean that the system combines relative Simplicity with relatively high levels of descriptive and explanatory Power (Appendix **??**).

A.1 Versatility in aspects of intelligence

The SPS has strengths and potential in the 'unsupervised' learning of new knowledge. As noted in Appendix **??**, this is an aspect of intelligence in the SPS that is different from others because it is not a by-product of the building of multiple alignments but is, instead, achieved via the creation of *grammars*, drawing on information within SPMAs.

Other aspects of intelligence where the SPS has strengths or potential are modelled via the building of SPMAs. These other aspects of intelligence include: the analysis and production of natural language; pattern recognition that is robust in the face of errors in data; pattern recognition at multiple levels of abstraction; computer vision [9]; best-match and semantic kinds of information retrieval; several kinds of reasoning (next subsection); planning; and problem solving.

A.2 Versatility in reasoning

Kinds of reasoning exhibited by the SPS include: one-step 'deductive' reasoning; chains of reasoning; abductive reasoning; reasoning with probabilistic networks and trees; reasoning with 'rules'; nonmonotonic reasoning and reasoning with default values; Bayesian reasoning with 'explaining away'; causal reasoning; reasoning that is not supported by evidence; the inheritance of attributes in class hierarchies; and inheritance of contexts in part-whole hierarchies. Where it is appropriate, probabilities for inferences may be calculated in a straightforward manner ([6, Section 3.7], [8, Section 4.4]).

There is also potential in the system for spatial reasoning [10, Section IV-F.1], and for what-if reasoning [10, Section IV-F.2].

It seems unlikely that the features of intelligence mentioned above are the full extent of the SPS's potential to imitate what people can do. The close connection that is known to exist between IC and concepts of probability (Appendix **??**), the central role of IC in the SPMA framework, and the versatility of the SPMA framework in aspects of intelligence suggest that there are more insights to come.

As noted in Appendix **??**, the probabilistic nature of the SPS makes it relatively straightforward to calculate absolute or conditional probabilities for results from the system, as for example in its several kinds of reasoning, most of which would naturally be classed as probabilistic.

A.3 Versatility in the representation of knowledge

Although SP-patterns are not very expressive in themselves, they come to life in the SPMA framework. Within that framework, they may serve in the representation of several different kinds of knowledge, including: the syntax of natural languages; class-inclusion hierarchies (with or without cross classification); part-whole hierarchies; discrimination networks and trees; if-then rules; entity-relationship structures [7, Sections 3 and 4]; relational tuples (*ibid.*, Section 3), and concepts in mathematics, logic, and computing, such as 'function', 'variable', 'value', 'set', and 'type definition' ([6, Chapter 10], [12, Section 6.6.1], [15, Section 2]).

As previously noted, the addition of two-dimensional SP patterns to the SPCM is likely to expand the representational repertoire of the SPS to structures in two-dimensions and three-dimensions, and the representation of procedural knowledge with parallel processing.

As with the SPS's generality in aspects of intelligence, it seems likely that the SPS is not constrained to represent only the forms of knowledge that have been mentioned. The generality of IC as a means of representing knowledge in a succinct manner, the central role of IC in the SPMA framework, and the versatility of that framework in the representation of knowledge, suggest that the SPS may prove to be a means of representing *all* the kinds of knowledge that people may work with.

A.4 The seamless integration of diverse aspects of intelligence, and diverse kinds of knowledge, in any combination

An important third feature of the SPS, alongside its versatility in aspects of intelligence and its versatility in the representation of diverse kinds of knowledge, is that *there is clear potential for the SPS to provide SI.* This is because diverse aspects of intelligence and diverse kinds of knowledge all flow from a single coherent and relatively simple source: the SPMA framework.

It appears that SI is *essential* in any artificial system that aspires to the fluidity, versatility and adaptability of the human mind.

Figure 9 shows schematically how the SPS, with SPMA centre stage, exhibits versatility and integration. The figure is intended to emphasise how development of the SPS has been and is aiming for versatility and integration in the system.

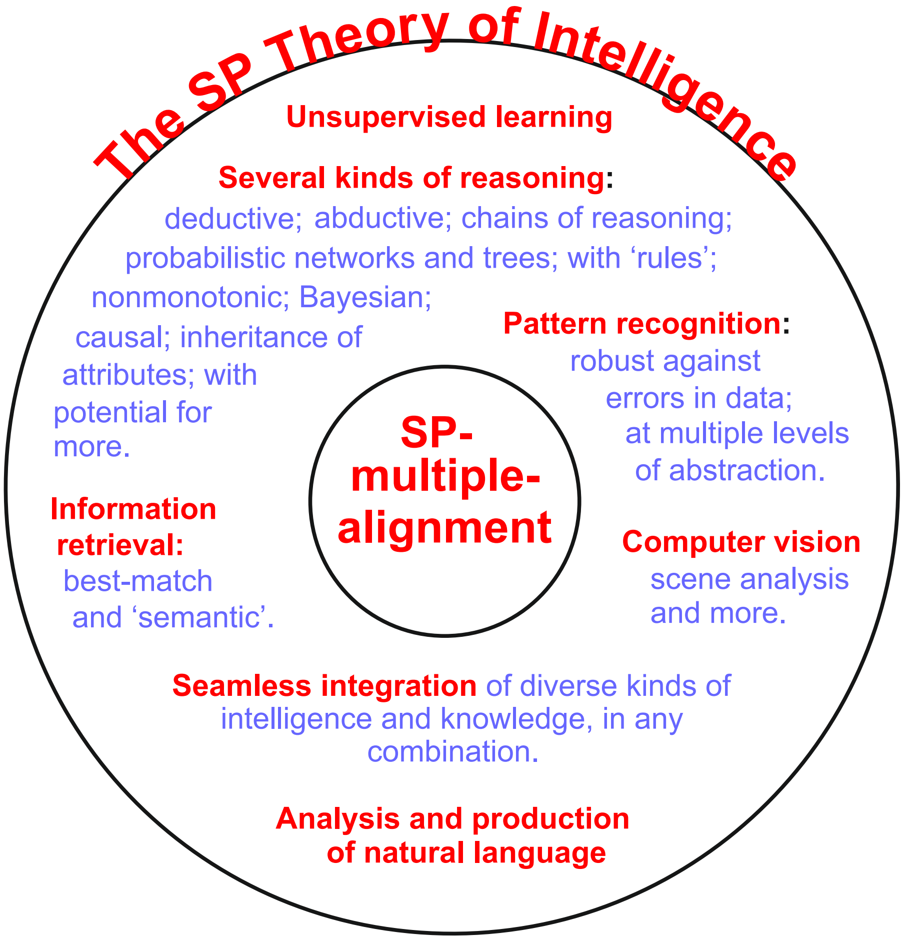

Figure 9: A schematic representation of versatility and integration in the SPS, with SPMA centre stage.

B Potential benefits and applications of the SPS

Apart from its strengths and potential in modelling AI-related functions (Appendix A), it appears that, in more humdrum terms, the SPS has several potential benefits and applications, several of them described in peer-reviewed papers. These include:

- *Big data*. Somewhat unexpectedly, it has been discovered that the SPS has potential to help solve nine significant problems associated with big data [11]. These are: overcoming the problem of variety in big data; the unsupervised learning of structures and relationships in big data; interpretation of big data via pattern recognition, natural language processing; the analysis of streaming data; compression of big data; model-based coding for the efficient transmission of big data; potential gains in computational and energy efficiency in the analysis of big data; managing errors and uncertainties in data; and visualisation of structure in big data and providing an audit trail in the processing of big data.

- *Autonomous robots.* The SPS opens up a radically new approach to the development of intelligence in autonomous robots [10];

- *An intelligent database system.* The SPS has potential in the development of an intelligent database system with several advantages compared with traditional database systems [7]. In this connection, the SPS has potential to add several kinds of reasoning and other aspects of intelligence to the 'database' represented by the World Wide Web, especially if the SP Machine were to be supercharged by replacing the search mechanisms in the foundations of the SP Machine with the high-parallel search mechanisms of any of the leading search engines.

- *Medical diagnosis.* The SPS may serve as a vehicle for medical knowledge and to assist practitioners in medical diagnosis, with potential for the automatic or semi-automatic learning of new knowledge [5];

- *Computer vision and natural vision.* The SPS opens up a new approach to the development of computer vision and its integration with other aspects of intelligence. It also throws light on several aspects of natural vision [9];

- *Neuroscience.* As outlined in Appendix **??**, abstract concepts in the SP Theory of Intelligence map quite well into concepts expressed in terms of neurons and their interconnections in a version of the theory called *SP-Neural* ([13], [6, Chapter 11]). This has potential to illuminate aspects of neuroscience and to suggest new avenues for investigation.

- *Commonsense reasoning.* In addition to the previously-described strengths of the SPS in several kinds of reasoning, the SPS has strengths in the surprisingly challenging area of "commonsense reasoning", as described by Ernest Davis and Gary Marcus [3].

- *Other areas of application.* The SPS has potential in several other areas of application including [12]: the simplification and integration of computing systems; best-match and semantic forms of information retrieval; software engineering [15]; the representation of knowledge, reasoning, and the semantic web; information compression; bioinformatics; the detection of computer viruses; and data fusion.

- *Mathematics*. The concept of ICMUP provides an entirely novel interpretation of mathematics [17]. This interpretation is quite unlike anything described in existing writings about the philosophy of mathematics or its application in science. There are potential benefits in science from this new interpretation of mathematics.

References

[1] G. M. Birtwistle, O-J Dahl, B. Myhrhaug, and K. Nygaard. *Simula Begin*. Studentlitteratur, Lund, 1973.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Hoboken NJ, Second, Kindle edition, 2006.

[3] E. Davis and G. Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015.

[4] G. A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.

[5] J. G. Wolff. Medical diagnosis as pattern recognition in a framework of information compression by multiple alignment, unification and search. *Decision Support Systems*, 42:608–625, 2006. arXiv:1409.8053 [cs.AI], bit.ly/1F366o7.

[6] J. G. Wolff. *Unifying Computing and Cognition: the SP Theory and Its Applications*. CognitionResearch.org, Menai Bridge, 2006. ISBNs: 0-9550726-0-3 (ebook edition), 0-9550726-1-1 (print edition). Distributors, including Amazon.com, are detailed on bit.ly/WmB1rs. The print version is produced via print-on-demand from "INGRAM Lightning Source" and, via that technology, is unlikely to go out of print.

[7] J. G. Wolff. Towards an intelligent database system founded on the SP Theory of Computing and Cognition. *Data & Knowledge Engineering*, 60:596–624, 2007. arXiv:cs/0311031 [cs.DB], bit.ly/1CUldR6.

[8] J. G. Wolff. The SP Theory of Intelligence: an overview. *Information*, 4(3):283–341, 2013. arXiv:1306.3888 [cs.AI], bit.ly/1NOMJ6l.

[9] J. G. Wolff. Application of the SP Theory of Intelligence to the understanding of natural vision and the development of computer vision. *SpringerPlus*, 3(1):552–570, 2014. arXiv:1303.2071 [cs.CV], bit.ly/2oIpZB6.

[10] J. G. Wolff. Autonomous robots and the SP Theory of Intelligence. *IEEE Access*, 2:1629–1651, 2014. arXiv:1409.8027 [cs.AI], bit.ly/18DxU5K.

[11] J. G. Wolff. Big data and the SP Theory of Intelligence. *IEEE Access*, 2:301–315, 2014. arXiv:1306.3890 [cs.DB], bit.ly/2qfSR3G. This paper, with minor revisions, is reproduced in Fei Hu (Ed.), *Big Data: Storage, Sharing, and Security*, Taylor & Francis LLC, CRC Press, Boca Raton, Florida, 2016, Chapter 6, pp. 143–170.

[12] J. G. Wolff. The SP Theory of Intelligence: benefits and applications. *Information*, 5(1):1–27, 2014. arXiv:1307.0845 [cs.AI], bit.ly/1FRYwew.

[13] J. G. Wolff. Information compression, multiple alignment, and the representation and processing of knowledge in the brain. *Frontiers in Psychology*, 7(1584), 2016. arXiv:1604.05535 [cs.AI], bit.ly/2esmYyt.

[14] J. G. Wolff. The SP Theory of Intelligence: its distinctive features and advantages. *IEEE Access*, 4:216–246, 2016. arXiv:1508.04087 [cs.AI], bit.ly/2qgq5QF.

[15] J. G. Wolff. Software engineering and the SP Theory of Intelligence. Technical report, CognitionResearch.org, 2017. arXiv:1708.06665 [cs.SE], bit.ly/2w99Wzq.

[16] J. G. Wolff. Information compression as a unifying principle in human learning, perception, and cognition. *Complexity*, 2019:38, February 2019. Article ID 1879746. viXra:1707.0161v3, hal-01624595 v2.

[17] J. G. Wolff. Mathematics as information compression via the matching and unification of patterns. *Complexity*, 2019:25, 2019. Article ID 6427493, Archives: vixra.org/abs/1912.0100 and hal.archives-ouvertes.fr/hal-02395680. This paper is reproduced in *New Ideas Concerning Science and Technology*, Vol. 13, 12 June 2021, Pages 132–169, doi.org/10.9734/bpi/nicst/v13/8449D.