

Article

The SP Theory of Intelligence: Benefits and Applications

J Gerard Wolff

CognitionResearch.org, Menai Bridge LL59 5LR, UK; E-Mail: jgw@cognitionresearch.org;
Tel.: +44-1248-712962; +44-7746-290775

Received: 26 May 2013; in revised form: 13 December 2013 / Accepted: 13 December 2013 /

Published: 24 December 2013

Abstract: This article describes existing and expected benefits of the *SP theory of intelligence*, and some potential applications. The theory aims to simplify and integrate ideas across artificial intelligence, mainstream computing, and human perception and cognition, with information compression as a unifying theme. It combines conceptual simplicity with descriptive and explanatory power across several areas of computing and cognition. In the *SP machine*—an expression of the SP theory which is currently realized in the form of a computer model—there is potential for an overall simplification of computing systems, including software. The SP theory promises deeper insights and better solutions in several areas of application including, most notably, unsupervised learning, natural language processing, autonomous robots, computer vision, intelligent databases, software engineering, information compression, medical diagnosis and big data. There is also potential in areas such as the semantic web, bioinformatics, structuring of documents, the detection of computer viruses, data fusion, new kinds of computer, and the development of scientific theories. The theory promises seamless integration of structures and functions within and between different areas of application. The potential value, worldwide, of these benefits and applications is at least \$190 billion each year. Further development would be facilitated by the creation of a high-parallel, open-source version of the SP machine, available to researchers everywhere.

Keywords: artificial intelligence; information compression; unsupervised learning; natural language processing; pattern recognition

1. Introduction

The *SP theory of intelligence* aims to simplify and integrate concepts across artificial intelligence, mainstream computing and human perception and cognition, with information compression as a unifying theme. This article describes existing and expected benefits of the SP theory and some of its potential applications.

The theory is described most fully in [1] and more briefly in an extended overview [2]. This article should be read in conjunction with either or both of those accounts.

In brief, the existing and expected benefits of the theory are:

- Conceptual simplicity combined with descriptive and explanatory power.
- Simplification of computing systems, including software.
- Deeper insights and better solutions in several areas of application.
- Seamless integration of structures and functions within and between different areas of application.

These points will be discussed, each in its own section, below. But first, as a background and context to what follows, there is a little philosophy, and an introduction to the SP theory with some associated ideas.

2. A Little Philosophy

Amongst alternative criteria for the success of a theory, what appears to be the most widely accepted is the principle (“Occam’s Razor”) that a good theory should combine *simplicity* with descriptive or explanatory *power* [3]. This equates with the idea—which echoes the underlying theme of the SP theory itself—that a good theory should compress empirical data via a reduction of “redundancy” in the data (thus increasing its “simplicity”), whilst retaining as much as possible of its non-redundant descriptive “power” [4]. As John Barrow has written: “Science is, at root, just the search for compression in the world.” [5] (p. 247).

These principles are prominent in most areas of science: the Copernican heliocentric theory of the Sun and the planets is rightly seen to provide a simpler and more powerful account of the facts than Ptolemy’s Earth-centered theory, with its complex epicycles; physicists have a keen interest in the possibility that quantum mechanics and relativity might be unified; biology would be greatly impoverished without modern understandings of evolution, genetics, and DNA; and so on. But in research in computer science, including artificial intelligence, there is a fragmentation of the field into a myriad of concepts and many specialisms, with little in the way of an overarching theory to pull everything together [6].

Perhaps the concept of a universal Turing machine [7,8] provides what is needed? That theory has of course been brilliantly successful—a basis for the current generation of computers and the many useful things that they can do. But it does not solve the problem of fragmentation and, although Alan Turing recognized that computers might become intelligent [9], the Turing theory, in itself, does not tell us how!

Are we getting closer to that goal? Yes, certainly. But as Hubert Dreyfus has suggested ([10] (p. 12) and [11]), we may be getting closer to achieving real intelligence in computers in the same sense that a man climbing a tree is getting closer to reaching the moon.

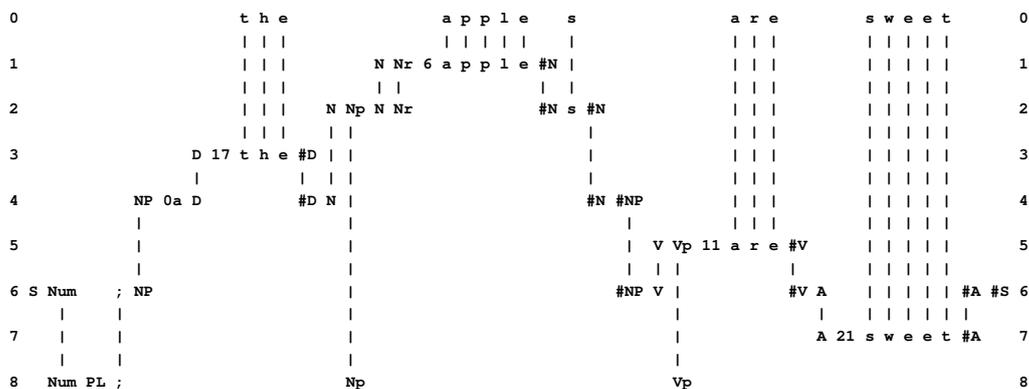
Whether or not the SP theory succeeds in getting us to the AI moon, I believe there is a pressing need, in computer science and AI, for a stronger focus on the simplification and integration of ideas. In all areas of application, including quite specialized areas, a theory that scores well in terms of simplicity and power is, compared with any weaker theory, likely to yield deeper insights, better solutions, and better integration of structures and functions, both within a given area and amongst different areas (see also Sections 6 and 7).

3. The SP Theory and the SP Machine: A Summary

In broad terms, the SP theory has three main elements:

- All kinds of knowledge are represented with *patterns*: arrays of atomic symbols in one or two dimensions.
- At the heart of the system is compression of information via the matching and unification (merging) of patterns, and the building of *multiple alignments* like the one shown in Figure 1.
- The system learns by compressing “New” patterns to create “Old” patterns like those shown in rows 1 to 8 in Figure 1.

Figure 1. The best multiple alignment created by the SP computer model with a store of Old patterns like those in rows 1 to 8 (representing grammatical structures, including words) and a New pattern (representing a sentence to be parsed) shown in row 0.



The system is realized in the form of a computer model [12]. It is envisaged that this will be developed into a high-parallel, open-source *SP machine*, an expression of the SP theory and a means for it to be applied (Section 3.2 in [2]). It is envisaged that this will be a facility for researchers everywhere to investigate what can be done with the system and create new versions of it. How things may develop is shown schematically in Figure 2.

5. Simplification of Computing Systems, Including Software

Apart from the simplification and integration of concepts in artificial intelligence, mainstream computing, and human perception and cognition (Section 4), the SP theory can help to simplify computing systems, including software.

The principle to be described here is already familiar in the way databases and expert systems are structured.

Early databases were each written as a monolithic system containing everything needed for its operation, and likewise for early expert systems. But it soon became apparent that, in the case of databases, a lot of effort could be saved by creating a generalized “database management system” (DBMS), with a user interface and mechanisms for the storage and retrieval of information, and then creating new databases by loading the DBMS with different bodies of information, according to need. In a similar way, an “expert system shell”, with a user interface and with mechanisms for inference and for the storage and retrieval of information, eliminates the need to create those facilities repeatedly in each new expert system.

The SP system takes this principle further. It aims to provide a general-purpose “intelligence”—chiefly the multiple alignment framework—and thus save the need to create those kinds of mechanisms repeatedly in different AI applications: pattern recognition, natural language processing, several kinds of reasoning, planning, problem solving, unsupervised learning, and more.

The principle extends to conventional applications, since the matching and unification of patterns and a process of searching amongst alternative matches for patterns (which are elements of the multiple alignment framework), are used in many kinds of application outside the world of artificial intelligence.

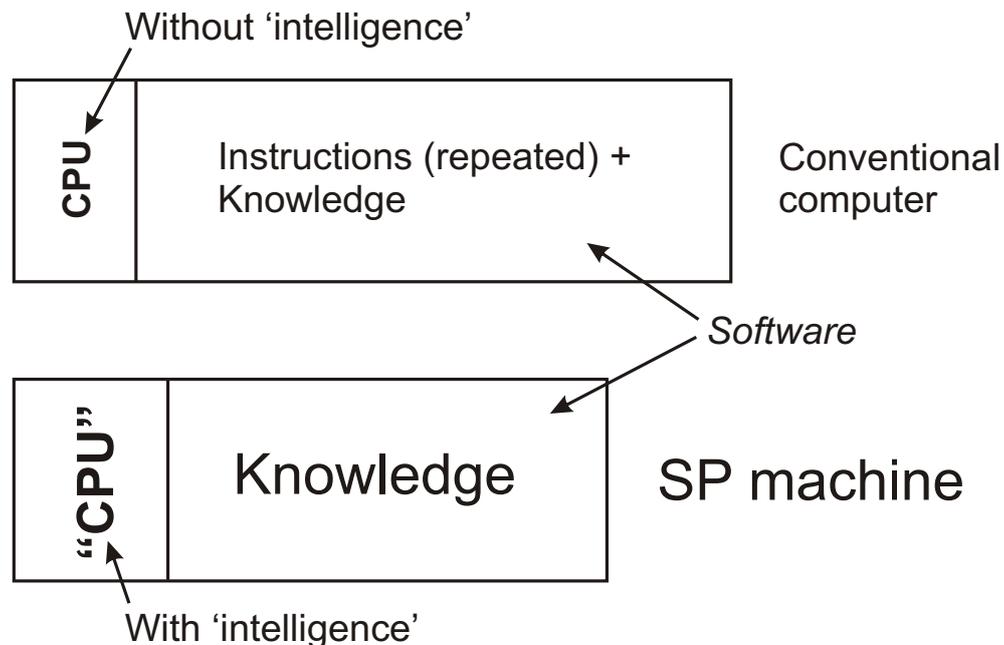
The way in which this principle may help to simplify computing systems is shown schematically in Figure 3. In a conventional computer, shown at the top of the figure, there is a central processing unit (CPU) which is relatively simple and without “intelligence”. Software in the system, shown to the right, is a combination of two things:

- Domain-specific knowledge such as knowledge of accountancy, geography, the organization and procedures of a business, and so on [13].
- Processing instructions to provide the intelligence that is missing in the CPU, chiefly processes for the matching and unification of patterns, and for searching amongst alternative matches to find one or more that are “good”. These kinds of processing instructions—let us call them MUP instructions—are used in many different kinds of application, meaning that there is a considerable amount of redundancy in conventional computing systems—where the term “system” includes both the variety of programs that may be run, as well as the hardware on which they run.

In the SP machine, shown schematically at the bottom of the figure, the “CPU” [14] is more complicated (as suggested by the larger size of “CPU” in the figure) and aims to provide much of the “intelligence” that is missing in the CPU of a conventional computer. This should mean that MUP instructions, and their many repetitions, may be largely eliminated from the software, leaving just the domain-specific knowledge. The increased complexity of the “CPU” should be more than offset by cutting out much of the redundancy in the software, meaning an overall reduction in complexity—as

suggested by the relatively small size of the figure that represents the SP machine, compared with the figure that represents a conventional computer.

Figure 3. Schematic representations of a conventional computer and an SP machine, as discussed in the text. Adapted with permission from Figure 4.7 in [1].



An overall simplification of computing systems can bring benefits like these:

- *Savings in development effort and associated costs.* With more intelligence in the CPU there should be less need for it to be encoded in applications.
- *Savings in development time.* With a reduced need for hand crafting, applications may be developed more quickly.
- *Savings in storage costs.* There may be useful economies in the storage space required for application code.
- Related benefits noted in Section 6.7.

6. Deeper Insights and Better Solutions in Several Areas of Application

This section describes some potential areas of application for the SP theory, and its potential benefits, drawing on the capabilities of the SP system (described in [1] (Chapters 4–11) and [2] (Sections 5–13)). These examples are probably just the tip of the iceberg. To mix our metaphors, there is a rich vein to be mined.

Because of its importance, I believe it is worth repeating the principle, mentioned in Section 2, that, even in specialized areas, a theory like the SP theory that has a broad base of support is likely to provide deeper insights than others, better solutions to problems, and (Section 7) better integration of

structures and functions within and between different areas of application. Putative examples from the SP theory include:

- A new method for encoding discontinuous dependencies in syntax (Chapter 5 in [1]; Section 8.1 in [2]), which is arguably simpler and more versatile than older methods.
- The multiple alignment framework, with the use of one simple format for all kinds of knowledge, is likely to facilitate the seamless integration of the syntax and semantics of natural languages, with the kinds of subtle interaction described in Section 6.2, below.
- The multiple alignment framework provides for the seamless integration of class inclusion hierarchies and part-whole hierarchies (Section 6.4 in [1]; Section 9.1 in [2]), something which is missing in most object-oriented systems.
- The use of one framework for both the analysis and production of knowledge (Section 3.8 in [1]; Section 4.5 in [2]) is a significant simplification compared with systems in which there are separate mechanisms.
- The multiple alignment framework provides for the integration of the relational, object-oriented, and network models for databases, with improvements in the relational model [15].
- The multiple alignment framework, with the use of one simple format for all kinds of knowledge, is likely to facilitate the learning of concepts that include diverse kinds of knowledge.

Theories with a narrow focus may work reasonably well in the target area but are likely to fail when the scope is widened. Even if existing technologies in a given area are doing well, there is likely to be a case for examining how the SP theory may be applied, and its potential benefits.

These kinds of benefits of a good theory have been seen repeatedly in the history of science: the germ theory of disease serves us better than the idea that diseases are caused by “bad air”; recognizing the role of oxygen in combustion is more practical than the phlogiston theory; space travel does not work unless we understand the laws of physics; and so on.

6.1. Unsupervised Learning

The first area of application to be considered—unsupervised learning—is a bit different from the others because it is less an area of application in its own right and more a contribution to others, most notably autonomous robots (Section 6.3), computer vision (Section 6.4), software engineering (Section 6.6), big data (Section 6.9), the semantic web (Section 6.10.1), bioinformatics (Section 6.10.2), and the development of scientific theories (Section 6.10.7).

Research developing computer models of language learning (reviewed in [16]; see also [17])—in particular, the importance of information compression in that area—has been one of the main sources of inspiration for the SP theory. But radical restructuring has been needed to meet the goal of simplifying and integrating ideas across artificial intelligence, mainstream computing, and human perception and cognition.

The new model, SP70 [2] (Section 3.1), has already demonstrated an ability to discover generative grammars from samples of artificial languages, including segmental structure, classes of structure, and abstract patterns. But some further work is needed to realize its full potential for learning, as outlined in [2] (Section 3.3). I believe the residual problems are soluble, and that, with their solution, the SP machine will have considerable potential in the areas mentioned above, and others.

A particular area of interest is, of course, the learning of natural languages, perhaps starting with the learning of syntactic structures. On longer time scales, there is potential for the learning of non-syntactic “semantic” structures and their integration with syntax.

6.2. Applications in the Processing of Natural Language

As noted in [2] (Section 8), the SP system supports the parsing and production of language, it provides a simple but effective means of representing discontinuous dependencies in syntax, it is robust in the face of errors in data, it facilitates the integration of syntax with semantics, and it provides a role for context in the processing of language. There are many potential applications, including such things as checking for grammatical errors in text; predictive typing that takes account of grammar as well as the spellings of individual words; natural-language front-ends to databases and expert systems; and the understanding and translation of natural languages.

It is true that some of these things are already done quite well with existing technologies but, as suggested in Sections 2, 6, and 7, there may nevertheless be advantages in exploring what can be done with SP system. For example, there are some subtle interactions between syntax and semantics which which may be difficult to handle without seamless integration of the two:

- Compare “Fafnir, who plays third base on the Little League team, is a fine boy.” with “They called their son Fafnir, which is a ridiculous name.” [18] (p. 139). The choice of “who” or “which” in these two sentences—a syntactic issue—depends on the *meaning* of “Fafnir”. In the first sentence, it means a person, while in the second sentence it means a name.
- As a contrasting example, compare “John gave me the scissors; I am using them now.” with “John gave me the two-bladed cutting instrument; I am using it now.” [18] (p. 141). In this case, the choice of “them” or “it”—also a syntactic issue—depends on the *syntactic form* of what is referred to: “the scissors” has a plural form while “the two-bladed cutting instrument” has a singular form, although the two expressions have the same meaning.

Apart from the integration of syntax with semantics, the SP system is likely to smooth the path for the integration of natural language processing with other aspects of intelligence: reasoning, pattern recognition, problem solving, and so on.

6.2.1. Towards the Understanding and Translation of Natural Languages

Given the apparent simplicity of representing syntactic and non-syntactic knowledge with patterns [2] (Sections 7 and 8), it may be feasible to use crowd-sourcing in the manner of Wikipedia to create syntactic rules for at least one natural language and a realistically large set of ontologies to provide

associated meanings. Existing grammars and existing ontologies may also prove useful in the creation of appropriate sets of SP patterns.

With this kind of knowledge on the web and with a high-parallel, on-line version of the SP machine (as sketched in [2] (Section 3.2)), it should be possible to achieve “understanding” of natural language, either in restricted domains (Sections 6.2.2, 6.2.3, and 6.2.4, below) or more generally.

With the same ontologies and with syntactic knowledge for two or more languages, it may be feasible to achieve translations between languages, with the ontologies functioning as an “interlingua” [19].

The use of an interlingua should yield two main benefits:

- When translations are required amongst several different languages, it is only necessary to create n mappings, one between each of the n languages and the interlingua. This contrasts with syntax-to-syntax systems, where, in the most comprehensive application of this approach, $n!/2(n-2)!$ pairs of languages are needed [20].
- It seems likely that the use of an interlingua will be essential if machine translation is ever to reach or exceed the standard that can be achieved by a good human translator.

6.2.2. Natural Language and Information Retrieval

The task of developing a system for understanding natural language may be considerably simplified if we narrow our focus to specific areas: those that require only a relatively restricted range of language forms. Three possibilities are outlined in this and the following two subsections.

On relatively short timescales the SP system may help to overcome some weaknesses in Internet search engines as they are now. For example, if we enter the query “telephone extensions in Anglesey” into any of the leading search engines, we get a mixture of answers: some of them—which are relevant to the query—relate to telephone extensions, while others—which are not relevant—relate to home extensions, building work and the like.

With the SP system, it should be relatively straightforward to overcome that kind of problem, at least for queries about businesses or services in a given area:

- Since a usefully-large range of natural-language questions may be asked with fairly simple language, it would only be necessary—for any given language—to create a relatively simple set of syntactic rules and lexicon, with corresponding semantic structures, all of them expressed with SP patterns.
- It should be possible to create an online tool that would make it easy for the owner or manager of a business to create an ontology for that business that describes what it is, what it does, and the kinds of products or services that it sells. It would of course be necessary for many businesses to provide that kind of information.
- With those two things in place, it should be possible, via the SP system, to create a semantic analysis for a given query (as outlined in [1] (Section 5.7)) and to use that analysis to identify businesses or services that satisfy the query.

With this approach, a query like “telephone extensions in Anglesey” would be parsed into constituents like “telephone extensions” and “in Anglesey”, each with a semantic structure. The word “extensions”, with a buildings-related meaning, would not figure in those semantic structures, so the unhelpful responses to the query would be eliminated.

6.2.3. Going Beyond FAQs

Many websites provide a list of “frequently asked questions” or FAQs which, as their name suggests, attempt to provide answers to the more commonly-asked questions about the subject of the website. This is OK as far as it goes but it would clearly be more useful if users could simply ask questions in the normal way and receive sensible answers.

To do this with the SP system means developing an ontology for the subject of the website, with an associated vocabulary and syntax, all expressed with SP patterns. This should be feasible to do, partly because of the essential simplicity of representing knowledge in the SP system, and partly because it is likely that the subject matter will be circumscribed.

6.2.4. Interactive Services

In a similar way, it should be possible to develop appropriate ontologies, vocabularies and syntaxes for such applications as booking a hotel room, booking a place on a plane or a train, booking a seat in a restaurant, and so on.

6.3. *Towards a Versatile Intelligence for Autonomous Robots*

If a robot is to survive and be effective in a challenging environment like Mars, where communication lags restrict the help that can be provided by people, there are advantages if it can be provided with human-like intelligence and versatility—as much as possible. A theory like the SP theory, which provides a unified view of several different aspects of intelligence, is a good candidate for the development of that general intelligence [21].

The SP system has the potential to give robots human-like flexibility and adaptability. Consider, for example, how top-ranking players learn to play games like pool, billiards, or snooker. This is quite different from how some existing robots play pool (see, for example, [22]), where the situation is simplified compared with what a human player faces, and where the robot is provided with a knowledge of geometry and associated mathematics.

For a person to play that kind of game well, they must spend thousands of hours on practice. Potting a ball means visual perception—from the position of the player, not an overhead camera—of the table, the cue ball, the target ball, the pocket and the cue, together with a knowledge of the motor patterns involved in striking the cue ball, and how those motor patterns are shaped by feedback from muscles and from sensory information about the impact between the cue and the cue ball.

Amongst the thousands of such configurations that a player may encounter, he or she must learn which ones lead to success. Getting beyond mere potting of a ball means learning how to impart spin to the cue ball, and judging such things as timings and forces, so that, after a shot, the configuration of balls on the table is favorable for the next shot by the given player, or difficult for his or her opponent. As before,

there are thousands of associations to be learned, in this case between visual and motor patterns when the cue ball is struck and the configuration of the balls on the table when they have all stopped moving.

At first sight, the pool-playing robot seems to have the edge over the human player because it does not need all those thousands of hours of practice. But what is missing is that afore-mentioned flexibility and adaptability: without task-specific programming, the robot cannot do such things as make a cup of tea or take the dog for a walk. By contrast, most people can learn to do those kinds of things without difficulty.

In this connection, the main strengths of the SP system are:

- Potential for the kind of visual analysis needed to assimilate the many configurations of balls, pockets, and cue (Section 6.4, next).
- The versatility of the SP framework in the representation and processing of diverse kinds of knowledge (Section 7) should facilitate the seamless integration of visual information about the table, balls, and so on, with information about actions by the player and feedback from muscles and from touch.
- Via its processes for unsupervised learning [2] (Section 5), the SP system should be able to learn associations between visual, motor and feedback patterns, as described above. It appears that similar principles have potential in the kinds of everyday learning mentioned above.

6.4. Computer Vision

The SP system has potential in the area of computer vision, as described in [23]. In brief:

- It has potential to simplify and integrate several areas in computer vision, including feature detection and alignment, segmentation, deriving structure from motion, stitching of images together, stereo correspondence, scene analysis, and object recognition (see, for example, [24]).
With regard to the last two topics:
 - Since scene analysis is, in some respects, similar to the parsing of natural language, and since the SP system performs well in parsing, it has potential in scene analysis as well. In the same way that a sentence may be parsed successfully without the need for explicit markers of the boundaries between successive words or between successive phrases or clauses, a scene may be analyzed into its component parts without the need for explicit boundaries between objects or other elements in a scene.
 - The system provides a powerful and flexible framework for pattern recognition, as outlined in [2] (Section 9), and more fully described in [1] (Chapter 6). It seems likely that this can be generalized for object recognition.
- The SP system has potential for unsupervised learning of the knowledge required for recognition. For example, discrete objects may be identified by the matching and merging of patterns within stereo images (*cf.* [25,26]) or within successive frames in a video, in much the same way that the word structure of natural language may be discovered via the matching and unification of patterns [2] (Section 5.2). The system may also learn such things as classes of entity, and associations between entities, such as the association between black clouds and rain.

- The system is likely to facilitate the seamless integration of vision with other aspects of intelligence: reasoning, planning, problem solving, natural language processing, and so on.
- As noted in [2] (Sections 8 and 9), the system is robust in the face of errors of omission, commission or substitution—an essential feature of any system that is to achieve human-like capabilities in vision.
- With regard to those problems outlined in [2] (Section 3.3), that relate to vision, there are potential solutions:
 - It is likely that the framework can be generalized to accommodate patterns in two dimensions.
 - As noted in [2] (Section 3.3), 3D structures may be modeled using 2D patterns, somewhat in the manner of architects' drawings. Knowledge of such structures may be built via the matching and unification of partially-overlapping 2D views [23] (Sections 6.1 and 6.2). Although such knowledge, as with people, may be partial and not geometrically accurate [27], it can, nevertheless, be quite serviceable for such purposes as getting around in one's environment.
 - The framework has potential to support both the discovery and recognition of low-level perceptual features.

6.5. A Versatile Model for Intelligent Databases

As described in [15] and outlined in [2] (Section 11), the SP system provides a versatile framework that accommodates several models of data (including object-oriented, relational, network and tree models), that facilitates the integration of different models, and that provides capabilities in reasoning, learning, pattern recognition, problem-solving, and other aspects of intelligence.

An industrial-strength version of the SP machine may prove useful in several areas, including, for example, crime investigations:

- The system would provide a means of storing and managing the data that are gathered in such investigations, often in large amounts.
- It may help in the recognition of features or combinations of features that link a given crime to other crimes, either current or past—and likewise for suspects.
- The system's capabilities in pattern recognition may also serve in the scanning of data to recognize indicators of criminal activity.
- It may prove useful in piecing together coherent patterns from partially-overlapping fragments of information, in much the same way that partially-overlapping digital photographs may be stitched together to create a larger picture.
- Given the capabilities of the SP system for different kinds of reasoning (Chapter 7 in [1] and Section 10 in [2]), including reasoning with information that is not complete, the system may

prove useful in suggesting avenues to be explored, perhaps highlighting aspects of an investigation that might otherwise be overlooked.

- Transparency in the representation of knowledge [2] (Section 2.4) and in reasoning via the multiple alignment framework (Chapters 7 and 10 in [1] and Section 10 in [2]) are likely to prove useful where evidence needs to be examined in a court of law, an advantage over “sub-symbolic” systems that yield results without insight.

6.6. Software Engineering

The SP system has potential in software engineering, as described in the following subsections.

6.6.1. Procedural Programming

Procedural programming of a traditional kind is not merely an artifact of the way computers have been designed. It provides an effective means of modeling the kinds of procedures in the real world where the sequence of actions is important. For example, if we are making a cake, we need to mix the ingredients before putting the cake in the oven, and we need to bake the cake before icing it.

It appears that the SP system provides much of what is needed to model real-world procedures of that kind:

- *Procedure.* A sequence of actions may be modeled with a one-dimensional SP pattern such as “do_A do_B do_C . . .”.
- *Variables, values and types.* In the SP framework, any pair of neighboring symbols may function as a slot or variable into which other information may be inserted via the alignment of patterns. For example, the pair of symbols “Num ;” in row 6 of Figure 1 is, in effect, a variable containing the value “PL” from row 8. The “type” of the variable is defined by the range of values it may take, which itself derives from the set of Old patterns provided for the SP model when the multiple alignment was created. In this case, the variable “Num ;” may take the values “PL” (meaning “plural”) or “SNG” (meaning “singular”).
- *Function or subroutine.* In the SP framework, the effect of a function or subroutine may be achieved in much the same way as with variables and values. For example, the pattern “A 21 s w e e t #A” in row 7 of Figure 1 may be seen as a function or subroutine that has been “called” from the higher-level procedure “S Num ; NP #NP V #V A #A #S” in row 6 of the figure.
- *Function with parameters.* As can be seen in Figure 4, the SP system may be run “backwards”, much as can be done with an appropriately-designed Prolog program, generating “data” from “results” (Section 3.8 in [1] and Section 4.5 in [2]). In this example, the pattern “S Num ; NP #NP V #V A #A #S” may be seen as a “function”, and symbols like “PL 0a 17 . . .” in row 0 may be seen as “values” for the “parameters” of the function.

- *Conditional statements.* The effect of the values just described is to select elements of the multiple alignment: “PL” selects the pattern “Num PL ; Np Vp”, “0a” selects “NP 0a D #D N #N #NP”, and so on. Each of these selections achieves the effect of a conditional statement in a conventional program: “If ‘PL’ then choose ‘Num PL ; Np Vp’ ”; “If ‘0a’ then choose ‘NP 0a D #D N #N #NP’ ”; and so on.

Figure 4. The best multiple alignment created by the SP model with the same Old patterns as were used in the creation of the multiple alignment shown in Figure 1, and a New pattern representing an encoding of a sentence, shown in row 0.

0	S	PL	0a	17		6		11	21	#S	0										
1	S	Num	;	NP				#NP	V		#V	A	#S	1							
2									V	Vp	11	a	r	e	#V		2				
3				NP	0a	D		#D	N		#N	#NP						3			
4					D	17	t	h	e	#D								4			
5							N	Nr	6	a	p	p	l	e	#N			5			
6							N	Np	N	Nr		#N	s	#N				6			
7													A	21	s	w	e	e	t	#A	7
8		Num	PL	;			Np				Vp										8

- *Repetition of procedures.* Conventional programs provide for the repetition of procedures with recursive functions and with statements like *repeat ... until* and *do ... while*. In the SP system, repetition may be encoded with recursive (self-referential) patterns like “X 1 a b c X 1 #X #X” which may be applied as shown in Figure 5.

Figure 5. The best multiple alignment produced by the SP model with the New pattern “a b c a b c a b c a b c” and multiple appearances of the Old pattern, “X 1 a b c X 1 #X #X”.

0		a	b	c		a	b	c		a	b	c		a	b	c				0	
1	X	1	a	b	c	X	1												#X	#X	1
2																					
3																					
4																					

- *Integration of “programs” and “data”.* Databases provide the mainstay of many software engineering projects and, as described in Chapter 6 in [1], Section 11 in [2] and [15], the SP system promises benefits in that area. In view of the versatility of SP patterns in the multiple alignment framework to represent varied kinds of knowledge [2] (Section 7), there is potential in software engineering projects for the seamless integration of “programs” with “data”.

- *Object-oriented design.* The SP system provides for object-oriented design with class hierarchies (including cross-classification), part-whole hierarchies, and inheritance of attributes (Section 6.4 in [1]; Section 9.1 in [2]). In view of the importance of object-oriented design in software engineering, this capability in the SP system is likely to prove useful for applications in that area.

To help smooth the path to working in the SP framework, there may be a case, at least in the early stages, for providing “syntactic sugar” to make things look more like conventional programming.

6.6.2. No Compiling or Interpretation

A potential advantage of the SP system compared with conventional systems is that there would not be any need for compiling or interpretation, meaning the translation of source code into machine code. Any SP program would simply be a set of SP patterns that would be processed directly by the SP machine.

It is true that the current SP computer model is compiled in the conventional way, and the same is likely to be true of any versions of the SP machine that are hosted on a conventional computing system. But the processes of building multiple alignments and creating new knowledge structures does not require compiling or interpretation as normally understood.

6.6.3. Sequential and Parallel Processing

In considering the application of sequential and parallel processing in the SP system, we need to distinguish between the workings of the SP machine and real-world processes (such as baking a cake or managing an industrial plant) to which the SP system may be applied:

- *The workings of the SP machine.* In the projected SP machine [2] (Section 3.2), it is envisaged that, while the ordering of symbols would be respected when one pattern is matched with another, the process of matching two patterns may be done left-to-right, right-to-left, or with many symbols matched in parallel. Likewise, many pairs of patterns may be matched in parallel.
- *Sequencing and parallelism in real-world processes.* Although the SP machine may be flexible in the way parallelism is applied to the matching of patterns, the system needs to respect sequencing and parallelism in the real-world processes to which the system may be applied. In broad terms, this may be achieved via the SP patterns themselves:
 - A sequence of operations may be modeled via the left-to-right sequence of symbols in a one-dimensional SP pattern, with “subroutines” if required, as outlined in Section 6.6.1.
 - Since SP patterns at the “top” level are independent of each other, they may serve to model processes that may run in parallel.

Naturally, a realistic model of any real-world project is likely to need additional information about such things as timings, dependencies between processes, and resources required.

6.6.4. Automatic Programming

With some improvements to the learning processes in the SP model (Section 9.5 in [1]; Section 5.1.4 in [2]), there is potential to by-pass at least some of the labor-intensive and costly hand-crafting of computer programs which is currently the mainstay of software engineering:

- *Learning of ontologies.* Knowledge of significant categories—people, buildings, vehicles, and so on—may be built up incrementally from information that users supply to the system. There is potential for the automatic structuring of such information in terms of class hierarchies and part-whole hierarchies.
- *Learning of procedures.* In a similar way, procedural knowledge may be built up from information supplied by users, and, via learning, such procedural knowledge may be structured automatically and generalized [28]. Likewise, for “programming by demonstration” in robotics [29].

Since knowledge in the SP system is intended to be transparent and comprehensible by people [2] (Section 2.4), it should be relatively straightforward to make adjustments and corrections to learned structures.

6.6.5. Verification and Validation

As was suggested in Section 5, the SP system may reduce or eliminate MUP instructions and their repetition in different applications. To the extent that that proves possible, we may reduce the opportunities for errors to be introduced. There is a corresponding potential to reduce or eliminate the need for “verification” in software development, and for improvements in the quality of software.

The SP machine also has potential in “validation”: helping to ensure that what is being developed is what is required. If domain-specific knowledge, including the requirements for a software system, can be loaded directly into an SP machine, without the need for the traditional kind of programming, this will help to ensure that what the system does is what the users want. Users may have tighter control than is traditional over what a system does, with more freedom to make changes when required [30].

6.6.6. Technical Debt

By reducing the complexity of the software development process and by enabling users to have more direct control over how things work, the SP machine may also help to reduce or eliminate the problem of “technical debt”, the way in which conventional systems can accumulate a backlog of “housekeeping” tasks and become unmanageably complex with the passage of time.

6.7. Information Compression

Given the central importance of information compression in the SP theory, one would naturally expect the SP machine to prove useful in that area of application. In effect, the multiple alignment framework serves, *inter alia*, to integrate three techniques for information compression: *chunking-with-codes*, *schema-plus-correction*, and *run-length coding* [1] (Section 2.2.9).

There is potential for the SP system to achieve relatively high levels of lossless compression for two main reasons:

- It is intended that, normally, the SP machine will perform a relatively thorough search of the space of alternative unifications of patterns and achieve correspondingly high levels of compression.
- If, as anticipated with some further development, the system will be able to learn discontinuous dependencies in data (see [2] (Section 3.3)), it will tap into sources of redundancy that appear to be outside the scope of traditional methods for compression of information.

In general, information compression can bring benefits in: economies in storage; speeding up transmission of information or reducing the demand for bandwidth, or some combination of the two (Section 6.7.1); providing a basis for inference (as described in [2] (Sections 2.1, 4.4, 10), and elsewhere in that article); and greater efficiency in processing with corresponding gains in energy efficiency (Section 6.7.2).

6.7.1. Economies in the Transmission of Information

A compressed version of a body of information, I , may be seen to comprise two parts:

- A “grammar” for I (which we may call G) containing patterns, at one or more levels of abstraction, that occur repeatedly in I .
- An “encoding” of I in terms of the grammar (which we may call E), including non-repeating information in I .

Where there is significant redundancy in I —which is true of most kinds of natural-language text and most kinds of images, especially moving images— E is likely to be much smaller than G . A grammar for I may also support the lossless compression of other bodies of information, provided they contain the same kinds of structures as I .

These things can provide the means of transmitting information very economically:

- A receiver (such as a TV set or a computer receiving web pages over the Internet) may be equipped with a grammar for the kind of information it is designed to receive, and some version of the SP system.
- Instead of transmitting “raw” data, or data that has been compressed in the traditional manner (containing both G and E), the encoding by itself would be sufficient.
- Using the SP system, the original data may be reconstructed fully, without any loss of information, by decoding the transmitted information (E) in terms of the stored copy of G .

A simple version of this idea, using a dictionary of words as a “grammar” for ordinary text, is already recognized (e.g., [31] and Chapter 11 in [32]). The apparent advantage of the SP system is the creation of more sophisticated grammars yielding higher levels of compression, and the application of the idea to kinds of information—such as moving images—where an ordinary dictionary would not work.

Since ordinary TVs work well without this kind of mechanism, some people may argue that there is no need for anything different. But:

- The growing popularity of video, TV and films on mobile services is putting pressure on mobile bandwidth [33], and increases in resolution will mean increasing demands for bandwidth via all transmission routes.
- It is likely that some of the bandwidth for terrestrial TV will be transferred to mobile services [34], creating an incentive to use the remaining bandwidth efficiently.
- There would be benefits in, for example, the transmission of information from a robot on Mars, or any other situation where a relatively large amount of information needs to be transmitted as quickly as possible over a relatively low-bandwidth channel.

6.7.2. Potential for Gains in Efficiency in Processing and in the Use of Energy

If, for example, we wish to search a large body of data, I , for instances of a pattern like “Treaty on the Functioning of the European Union”, and to conduct similar searches for other patterns, there can be savings in processing if:

1. I is compressed by finding recurrent patterns in I , like “Treaty on the Functioning of the European Union”, and replacing each instance of a given pattern by a relatively short code such as “TFEU”, or, more generally, compressing I via grammar discovery and encoding as described in [1] (Chapter 9 and Section 3.5) and [2] (Sections 4 and 5).
2. Searching is done with “TFEU” instead of “Treaty on the Functioning of the European Union”, or, more generally, searching is done with a compressed representation of the search pattern via the encoding processes in the SP system.

Although the first “encoding” step requires processing, that one-time cost can be more than offset by consequent savings in many searches. The savings arise because of the reduced size of I and because of the relatively small size of a code like “TFEU” compared with the pattern that it represents.

Since searching lies at the heart of how the SP system works, gains in the efficiency of searching should mean gains in the overall efficiency of processing in the SP system. There should be corresponding reductions in the use of energy.

6.8. Medical Diagnosis

The way in which the SP system may be applied in medical diagnosis is described in [35]. The expected benefits of the SP system in that area of application include:

- A format for representing diseases that is simple and intuitive.
- An ability to cope with errors and uncertainties in diagnostic information.
- The simplicity of storing statistical information as frequencies of occurrence of diseases.
- The system provides a method for evaluating alternative diagnostic hypotheses that yields true probabilities.

- It is a framework that should facilitate the unsupervised learning of medical knowledge and the integration of medical diagnosis with other AI applications.

The main emphasis in [35] is on medical diagnosis as pattern recognition. But the SP system may also be applied to causal diagnosis ([1] (Section 7.9) and [2] (Section 10.5)) which, in a medical context, may enable the system to reason that “The patient’s fatigue may be caused by anemia which may be caused by a shortage of iron in the diet”.

6.9. Managing “Big Data” and Gaining Value from It

The SP system has potential in the management and analysis of the large volumes of data that are now produced in many fields (“big data”), as discussed in [36] and summarized here.

The system may help overcoming the problem of variety in big data: the many different formats for knowledge and consequent problems in analysis. In view of the versatility of SP patterns, within the SP framework, to represent diverse kinds of knowledge [2] (Section 7), there is potential for them to serve as a universal format for knowledge and to facilitate the processing of diverse kinds of knowledge. The SP framework may also provide a mechanism for translating the original data into the universal format.

In the interpretation of data, the SP system has capabilities that include such things as the parsing and production of language, pattern recognition, and various kinds of probabilistic reasoning. But potentially the most useful facility with big data would be scanning for patterns, with recognition of family-resemblance or polythetic categories, at multiple levels of abstraction and with part-whole hierarchies, with inductive prediction and the calculation of associated probabilities, with a role for context in recognition, and robust in the face of errors of omission, commission or substitution. There are potential applications in several areas including, security, finance, meteorology, and astronomy.

With solutions to residual problems in unsupervised learning [2] (Sections 3.3 and 5.1.4), it seems likely that the SP machine, via the “DONSVIC” principle [2] (Section 5.2), will prove useful in discovering “interesting” or “useful” structures in big data, including significant entities, class-inclusion structures, part-whole structures, rules, regularities and associations, and discontinuous dependencies in data. The use of one simple format for different kinds of knowledge is likely to facilitate the learning of concepts that combine diverse kinds of knowledge.

Because of its potential for information compression (Section 6.7), the SP system may achieve useful reductions in the volume of big data and thus facilitate its storage and management. There is potential for substantial economies in the transmission of big data (Section 6.7.1), and for the efficient processing of compressed data, without the need for decompression (Section 6.7.2). And unlike conventional systems for compression of information, the SP system has capabilities in several aspects of intelligence.

6.10. Other Areas of Application

Some other areas of application where there is potential for the SP system are grouped here because, so far, that potential has not yet been much explored. But any of them may turn out to be significant.

6.10.1. Knowledge, Reasoning, and the Semantic Web

The SP framework may contribute to the development of the “semantic web”—a “web of data” to provide machine-understandable meaning for web pages [37–39]. In this connection, the main attractions of the system appear to be:

- *Simplicity, versatility and integration in the representation of knowledge.* The SP system combines simplicity in the underlying format for knowledge with the versatility to represent several different kinds of knowledge—and it facilitates the seamless integration of those different kinds of knowledge [2] (Section 7).
- *Versatility in reasoning.* The system provides for several kinds of reasoning and their inter-working (Chapter 7 in [1] and Section 10 in [2]).
- *Natural language understanding.* As suggested in Section 6.2, the SP system has potential for the understanding of natural language. If that proves possible, semantic structures may be derived from textual information in web pages, without the need, in those web pages, for the separate provision of ontologies or the like.
- *Automatic learning of ontologies.* As suggested in [2] (Section 5.2), the SP system has potential, via the DONSVIC principle, for the extraction of interesting or useful structures from data. These may include the kinds of ontologies which have been a focus of interest in the development of the semantic web.
- *Uncertainty and vagueness.* The SP system is inherently probabilistic and it is robust in the face of incomplete information and errors of commission or substitution. These capabilities appear promising as a means of coping with uncertainty and vagueness in the semantic web [40].

6.10.2. Bioinformatics

As described in [2] (Section 4), a central idea in the SP system is the concept of *multiple alignment*, borrowed from bioinformatics but with important differences from how that concept is normally understood in research in biochemistry.

Notwithstanding those differences, or perhaps because of them, the SP system has potential for the analysis of DNA sequences and amino-acid sequences in any or all of the following ways:

- The formation of alignments amongst two or more sequences, much as in multiple alignment as currently understood in bioinformatics.
- The discovery or recognition of recurrent patterns in DNA or amino-acid sequences, including discontinuous patterns.
- The discovery of correlations between genes, or combinations of genes, and any or all of: diseases; signs or symptoms of diseases; or any other feature or combination of features of people, other animals, or plants.

6.10.3. Maintaining Multiple Versions and Parts of a Document or Web Page

The multiple alignment concept provides a means of maintaining multiple versions of any document or web page, such as, for example, versions in different languages. Parts which are shared amongst different versions—such as pictures or diagrams—may be kept separate from parts that vary amongst different versions—such as text in a specific language. As with other kinds of data, the framework provides the means of maintaining hierarchies of classes (with cross classification if that is required) and of maintaining part-whole hierarchies and their integration with class hierarchies [2] (Section 9.1).

Updates to high-level classes appear directly in all lower-level classes without the need for information to be copied amongst different versions.

Of course, these kinds of things can be done in other ways. The strength of the SP framework is its versatility, reducing the need for *ad hoc* solutions in different areas.

6.10.4. Detection of Computer Viruses

The detection of already-known computer viruses and other malware can be more subtle and difficult than simply looking for exact matches for the “signatures” of viruses. The offending code may be rather similar to perfectly legitimate code and it may be contained within a compressed (“packed”) executable file.

Here, the SP system and its capabilities for pattern recognition and learning may prove useful:

- Recognition in the SP system is probabilistic, it does not depend on the presence or absence of any particular feature or combination of features, and it can cope with errors.
- By compressing known viruses into a set of SP patterns, the system can reduce the amount of information needed to specify viruses and, as a consequence, it can reduce the amount of processing needed for the detection of viruses.

6.10.5. Data Fusion

There are many situations where it can be useful to merge or integrate two or more sources of information, normally when there is some commonality amongst the different sources. For example, in ecological research there may be a need to integrate different sources of information about the movement around the world of whales or other creatures; and in the management of a website, there may be a need to consolidate a set of web pages that are saying similar things.

At first sight, the SP system appears to be tailor made for this area of application. After all, the merging of fully or partially matching patterns lies at the heart of the SP system. But, as with big data (Section 6.9), the variety of different formats for knowledge can pose problems.

A possible solution, as with big data, is to translate different kinds of knowledge into a universal format. As before, the SP framework may provide that universal format and a mechanism for the translation.

6.10.6. New Kinds of Computer

In research to develop new kinds of computer, such as optical or chemical computers, it is commonly assumed that the first step must be to create the optical or chemical equivalent of a transistor or logic gate, probably because of the significance of those kinds of structures in the evolution of electronic computers as we know them today.

But the SP theory, with its emphasis on information compression via the matching and unification of patterns, suggests new approaches to the development of new technologies for computing, perhaps by-passing some of the structures and concepts that have been prominent in the development of electronic computers.

A potential benefit is that, with that primary focus on the matching and unification of patterns, it may prove easier to take advantage of the potential for high levels of parallelism in optical or chemical computers, or new kinds of electronic computers.

6.10.7. Development of Scientific Theories

Although Occam's Razor is widely recognized as a touchstone of good science and, as noted earlier, John Barrow has written that "Science is, at root, just the search for compression in the world." [5] (p. 247), there seems to have been relatively little interest—apart from some work on principles of "minimum length encoding" (see, for example, [41–43])—in making a connection between the practice of information compression and the development of scientific theories.

A possible reason is that most compression algorithms yield an encoding of the target data which is, from a human perspective, remarkably opaque: the output is normally smaller than the input—a useful result—but the encoding does not normally contain structures or relations that are meaningful for people. As noted in [2] (Section 13), this is probably because compression algorithms are normally designed to be quick-and-dirty, aiming for speed on low-powered computers, with a low priority for finding structures that yield high levels of compression; and, in accordance with the previously-mentioned DONSVIC principle, it appears to be those structures that are most meaningful for people.

Perhaps this does not matter. In a "practicalities-not-insight" approach to artificial intelligence that appears to be favored by some researchers, results that are useful in practice are given a higher priority than "explanation" or "understanding" [44].

With the SP system, there appears to be potential for something different and ultimately more acceptable as a way of doing science: the possibility of creating a "discovery machine" [45] (p. 108ff.), deriving scientific theories from data, automatically or semi-automatically, and for obtaining analyses that people will regard as meaningful and insightful and which are also useful and practical. There is also potential for the quantitative evaluation of theories, as mentioned in Section 4.

In this area, reasons for optimism include:

- An "up market" emphasis in the SP program on finding structures that are good in terms of information compression.

- The DONSVIC principle, mentioned above—the discovery of natural structures via information compression [2] (Section 5.2). Structures that are good in terms of information compression seem generally to be structures that people regard as natural and meaningful.
- The versatility of the multiple alignment concept in the representation and processing of structures that are meaningful to people.

7. Seamless Integration of Structures and Functions within and between Different Areas of Application

This section emphasizes again the importance of how the SP system may integrate structures and functions both within and between different areas of application. Instead of what can sometimes be awkward marriages between incompatible systems, there can be smooth interworking of, for example, pattern recognition, reasoning, and learning; of different kinds of reasoning; of different kinds of knowledge; and so on.

As previously noted, this facilitation of integration arises from: the use of one simple format for different kinds of knowledge; the use of one system—the multiple alignment framework—for the processing of knowledge; and the over-arching principle of information compression by the matching and unification of patterns.

As we have seen, the system provides for the integration of syntax and semantics in natural language processing; for the same kind of processing to achieve both the analysis and the production of language; for the integration of class-inclusion hierarchies, part-whole hierarchies, and other forms of knowledge; and for the integration of different aspects of intelligence in any combination: natural language processing, pattern recognition, different kinds of reasoning, unsupervised learning, and so on.

These kinds of integration are probably necessary to achieve human-like versatility and flexibility in the workings of computers.

8. Conclusions

The SP theory of intelligence combines conceptual simplicity with descriptive and explanatory power in several areas, including concepts of “computing”, the representation of knowledge, natural language processing, pattern recognition, several kinds of reasoning, the storage and retrieval of information, planning and problem solving, unsupervised learning, information compression, neuroscience, and human perception and cognition.

In the SP machine there is potential for the simplification of computing systems, including software, with corresponding savings in the time, effort and cost in the development of applications, and other benefits related to information compression.

As a theory with a broad base of support, the SP theory promises deeper insights and better solutions in several areas of application, including unsupervised learning, natural language processing, autonomous robots, computer vision, intelligent databases, software engineering, information compression, medical diagnosis, and big data; and perhaps also in areas such as the semantic web, bioinformatics, structuring of

documents, the detection of computer viruses, data fusion, new kinds of computer, and the development of scientific theories.

The SP theory should facilitate the integration of structures and functions, both within a given area and amongst different areas—a likely pre-requisite for the achievement of human-like versatility and flexibility in the way computers work.

These potential benefits and applications are not merely of theoretical interest. If, as a conservative estimate, they were to add 5% to the value of annual worldwide IT investments [46], they would be worth \$190 billion each year, and increasing with the continuing growth of IT.

As suggested in [2] (Section 3.2) and in Section 3 of this article, further development would be assisted by the creation of a high-parallel, open source version of the SP machine, available to researchers throughout the world to examine what can be done with the system and to create new versions of it.

Acknowledgments

I am grateful to anonymous referees for many useful comments, and to Daniel Wolff for pointing out the potential of the SP system in the detection of computer viruses and in the management and analysis of big data.

Conflicts of Interest

The author declares no conflict of interest.

References and Notes

1. Wolff, J.G. *Unifying Computing and Cognition: The SP Theory and Its Applications*; CognitionResearch.org: Menai Bridge, UK, 2006.
2. Wolff, J.G. The SP theory of intelligence: An overview. *Information* **2013**, *4*, 283–341.
3. According to this criterion, a good theory in a given area lies somewhere between things which are simple but very general—Such as the laws of arithmetic—And over-specific “theories” that simply repeat what it is they are meant to explain.
4. Hence the name “SP”.
5. Barrow, J.D. *Pi in the Sky*; Penguin Books: Harmondsworth, UK, 1992.
6. Similar things were said by Allen Newell about research in human perception and cognition, in his well-known essay on why “You can’t play 20 questions with nature and win” ([47]. See also [48,49]). And writing about machine technologies to augment our senses, John Kelly and Steve Hamm say “... there’s a strong tendency [for scientists] to view each sensory field in isolation as specialists focus only on a single sensory capability. Experts in each sense don’t read journals devoted to the others, and they don’t attend one another’s conferences. ...” [45] (p. 74).
7. Turing, A.M. On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **1936**, *42*, 230–265.

8. Turing, A.M. On computable numbers, with an application to the Entscheidungsproblem. A correction. *Proc. Lond. Math. Soc.* **1936**, *42*, 544–546.
9. Turing, A.M. Computing machinery and intelligence. *Mind* **1950**, *59*, 433–460.
10. Dreyfus, H.L. *What Computers Can't Do: A Critique of Artificial Reason*; Harper and Row: New York, NY, USA, 1972; Revised in 1979.
11. Dreyfus, H.L. *What Computers Still Can't Do*; MIT Press: New York, NY, USA, 1992.
12. The latest version is SP70.
13. Even in systems such as DBMSs or expert-system shells, mentioned above, the domain-specific knowledge which is loaded into the system may be regarded as part of the software.
14. The reason for putting “CPU” in quote marks in the lower part of Figure 3 is that an SP machine may not have a discrete central processing unit like a conventional computer. The SP system is probably better suited to a “data-centric” style [45] (pp. 87–93), with a closer integration of data and processing than in conventional computers.
15. Wolff, J.G. Towards an intelligent database system founded on the SP theory of computing and cognition. *Data Knowl. Eng.* **2007**, *60*, 596–624.
16. Wolff, J.G. Learning syntax and meanings through optimization and distributional analysis. In *Categories and Processes in Language Acquisition*; Levy, Y., Schlesinger, I.M., Braine, M.D.S., Eds.; Lawrence Erlbaum: Hillsdale, NJ, USA, 1988; pp. 179–215.
17. Available online: www.cognitionresearch.org/lang_learn.html (accessed on 19 December 2013).
18. McCawley, J.D. The Role of Semantics in a Grammar. In *Universals in Linguistic Theory*; Lecture Notes in Artificial Intelligence; Bach, E., Harms, R.T., Eds.; Holt, Rinehart and Winston, Inc.: New York, NY, USA, 1968; pp. 125–169.
19. See, for example, “Interlingual machine translation”, Wikipedia, available online: <http://bit.ly/RITcHU>, retrieved 31 May 2013.
20. That number may be reduced if one or more languages are used as stepping stones to the translation of others.
21. This reference to autonomous robots does not in any way endorse or defend the unethical or illegal use of airborne drones or other autonomous robots to cause death, injury, or damage to property.
22. Archibald, C.; Altman, A.; Greenspan, M.; Shoham, Y. Computational pool: A new challenge for game theory pragmatics. *AI Mag.* **2010**, *31*, 33–41.
23. Wolff, J.G. Application of the SP theory of intelligence to the understanding of natural vision and the development of computer vision 2013. CognitionResearch.org, Menai Bridge, UK, unpublished work, 2013.
24. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer: London, UK, 2011.
25. As with information compression, a focus on the isolation of discrete objects in binocular vision is distinct from the more usual interest in the way that slight differences between the two images enables us to see the scene in depth.
26. Marr, D.; Poggio, T. A computational theory of human stereo vision. *Proc. R. Soc. Lond. B* **1979**, *204*, 301–328.

27. The Ames room illusion shows that human vision does not always yield an accurate knowledge of 3D structures (see, for example, Wikipedia, available online: http://en.wikipedia.org/wiki/Ames_room, retrieved 4 December 2013).
28. Related work includes “process mining” as in, for example, the “ProM” system, available online: <http://bit.ly/13iS2Ch>.
29. See, for example, “Programming by demonstration”, Wikipedia, available online: <http://bit.ly/121GrKJ>, retrieved 6 June 2013.
30. Notice that the distinction, just referenced, between domain-specific knowledge and MUP instructions is different from the often-made distinction between declarative programming and imperative or procedural programming. It is envisaged that the SP machine would reduce or eliminate the MUP instructions in software but that it would be able to model both the declarative “what” and (Section 6.6.1) the procedural “how” of the real world.
31. Giltner, M.; Mueller, J.C.; Fiest, R.R. Data Compression, Encryption, and in-Line Transmission System. U.S. Patent 4,386,416, 31 May 1983.
32. Storer, J.A. *Data Compression: Methods and Theory*; Computer Science Press: Rockville, MD, USA, 1988.
33. See, for example, “Data jam threat to UK mobile networks”, BBC News, 16 November 2012, available online: <http://bbc.in/T5g14s>.
34. See, for example, “TVs will need retuning again to make room for mobile services”, The Guardian, 16 November 2012, available online: <http://bit.ly/ZZ9N8C>.
35. Wolff, J.G. Medical diagnosis as pattern recognition in a framework of information compression by multiple alignment, unification and search. *Decis. Support Syst.* **2006**, *42*, 608–625.
36. Wolff, J.G. The management and analysis of big data, and the SP theory of intelligence 2013. CognitionResearch.org, Menai Bridge, UK, unpublished work, 2013.
37. Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. *Scientific American*, 17 May 2001, pp. 35–43.
38. Shadbolt, N.; Hall, W.; Berners-Lee, T. The semantic web revisited. *IEEE Intell. Systems* **2006**, *21*, 96–101.
39. See also W3C information about the semantic web at <http://bit.ly/1bX7Nv>.
40. Lukasiewicz, T.; Straccia, U. Managing uncertainty and vagueness in description logics for the Semantic Web. *Web Semant. Sci. Serv. Agents World Wide Web* **2008**, *6*, 291–308.
41. Solomonoff, R.J. A formal theory of inductive inference. Parts I and II. *Inf. Control* **1964**, *7*, 1–22, 224–254.
42. Wallace, C.S.; Boulton, D.M. An information measure for classification. *Comput. J.* **1968**, *11*, 185–195.
43. Rissanen, J. Modelling by the shortest data description. *Autom. J. IFAC* **1978**, *14*, 465–471.
44. See, for example, “Not like us: Artificial minds we can’t understand”, New Scientist, 10 August 2013, available online: <http://bit.ly/15m1cnm>. As noted in Section 6.5 of this article, transparency in the SP system, in both the representation of knowledge and in reasoning, may prove useful where evidence needs to be examined in a court of law.

45. Kelly, J.E.; Hamm, S. *Smart Machines: IBM's Watson and the Era of Cognitive Computing*; Columbia University Press: New York, NY, USA, 2013.
46. Expected to be \$3.8 trillion by 2014 (“Gartner: Big data will help drive IT spending to \$3.8 trillion in 2014”, InfoWorld, 3 January 2013, available online: <http://bit.ly/Z00SBr>).
47. Newell, A. You Can't Play 20 Questions with Nature and Win: Projective Comments on the Papers in This Symposium. In *Visual Information Processing*; Chase, W.G., Ed.; Academic Press: New York, NY, USA, 1973; pp. 283–308.
48. Newell, A. Précis of *Unified Theories of Cognition*. *Behav. Brain Sci.* **1992**, *15*, 425–437.
49. Newell, A. *Unified Theories of Cognition*; Harvard University Press: Cambridge, MA, USA, 1990.

© 2013 by the author; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).