Computational creativity seeks to understand computational mechanisms that can be characterized as creative. The creation of new concepts is a central challenge for any creative system. In this paper, we outline different approaches to computational concept creation and then review conceptual representations relevant to concept creation, and therefore to computational creativity. The conceptual representations are organized in accordance with two important perspectives on the distinctions between them. One distinction is between symbolic, spatial and connectionist representations. The other is between descriptive and procedural representations. Additionally, conceptual representations used in particular creative domains, i.e. language, music, image and emotion, are reviewed separately. For every representation reviewed, we cover the inference it affords, the computational means of building it, and its application in concept creation.

CCS Concepts:
- *General and reference → Surveys and overviews;
- *Computing methodologies → Knowledge representation and reasoning; Philosophical/theoretical foundations of artificial intelligence;
- *Applied computing → Arts and humanities;

General Terms: Algorithms, Design, Human factors, Theory

Additional Key Words and Phrases: Computational creativity, concept creation, concept, conceptual representation, procedural representation

This work has been supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733 (ConCreTe), and by the Academy of Finland under grants 276897 (CLiC) and 313973 (CACS).

Authors’ addresses: P. Xiao, H. Toivonen and O. Gross; emails: {ping.xiao, hannu.toivonen, ogross}@cs.helsinki.fi; A. Cardoso, J. Correia, P. Machado, P. Martins, H. Gonçalo Oliveira and R. Sharma; emails: {amilcar, jncor, machado, pjmm, hroliv, rahul}@dei.uc.pt; A. M. Pinto; email: amdtpinto@ciencias.ulisboa.pt; A. Díaz, V. Francisco, R. Hervás and C. León; emails: {albertodiaz, virginia, raquelhb, cleon}@fdi.ucm.es; P. Gervás; email: pgervas@sip.ucm.es; J. Forth, M. Purver and G. A. Wiggins; emails: {jamie.forth, m.purver, geraint.wiggins}@qmul.ac.uk; M. Bohanec, N. Lavrač, D. Miljković, V. Podpečan, S. Pollak, J. Kralj, T. Urbančič and M. Žnidaršič; emails: {marko.bohanec, nada.lavrac, dragana.miljkovic, vid.podpecan, senja.pollak, jan.kralj, tanja.urbananci, martin.znidarsic}@ijs.si; F. van der Velde; email: f.vandervelde@utwente.nl; S. Battersby; email: stuart@chatterbox.co.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© YYYY ACM. 0360-0300/YYYY/01-ARTA $15.00
DOI: http://dx.doi.org/10.1145/0000000.0000000

ACM Computing Surveys, Vol. V, No. N, Article A, Publication date: January YYYY.
1. INTRODUCTION

Computational Creativity [Wiggins 2006a; Colton and Wiggins 2012] is a field of research, seeking to understand how computational processes can lead to creative results. The field is gaining importance as the need for not only intelligent but also creative computational support increases in fields such as science, games, arts, music and literature. While work on automated creativity has a history of at least one thousand years (e.g., Guido d’Arezzo’s melody generation work around year 1026), the field has only recently developed an academic identity and a research community of its own\(^1\).

The computational generation of new concepts is a central challenge among the topics of computational creativity research. In this paper, we briefly outline different approaches to concept creation, and then review conceptual representations that have been used in various endeavors in this area. Our motivation stems from computational creativity, and our aim is to shed light on the emerging research area, as well as to help researchers in artificial intelligence more generally to gain an overview of conceptual representations, especially from the point of view of computational concept creation. For generic treatments of knowledge representation and reasoning, we refer the reader to other reviews [Brachman and Levesque 1985; Sowa 2000; Liao 2003; Van Harmelen et al. 2008; Jakus et al. 2013].

“Concept” has been used to refer to a bewildering range of things. In Boden’s [1990] pioneering study on creativity, a concept is an abstract idea in arts, science and everyday life. A key requirement of any computational model (creative or otherwise) that manipulates concepts, is for a representation of the concept being manipulated. Any representation describes some but not all aspects of an idea, and supports a limited number of processing options [Davis et al. 1993], which are further constrained by the scientific or technical methods available at the time. A representation is created or chosen (over competing representations) based on the extent to which it facilitates a specific task. For an interdisciplinary audience, it is important to understand that the word “representation” is used in different ways in different fields. In computer science, it means “a formalism for encoding data or meaning to be used within a computational system”. This is in contrast to the use in psychology where a representation is a mental structure that corresponds with a specific thing or class of things; thus, for a psychologist, a concept and a representation are much the same thing. For us, however, one is a thing to be encoded using a representation.

This paper reviews computational conceptual representations relevant to computational concept creation, which therefore may be useful in computational creativity. First, a brief overview of different approaches to concept creation is presented, providing a background to this review. The actual review that follows next is structured in terms of two major distinctions between different conceptual representations. One distinction is along the axis from connectionist to symbolic levels, with a spatial level

\(^1\)http://computationalcreativity.net/home/conferences/.
between them. The other distinction, especially prominent in the computer science community, is between descriptive and procedural representations.

It is perhaps worth adding explicitly that the aim of this paper is not to discuss the philosophy of cognitive concept creation (interesting though that be), but to survey relevant computational approaches in the field of computational creativity.

1.1. Approaches to Computational Concept Creation

In this section, we propose a typology of computational concept creation techniques, as background and motivation for the review of conceptual representations. Our aim here is to provide an overview of the multitude of approaches, instead of giving a detailed review of concept creation techniques.

One of the best known categorizations of different types of creativity is by Boden [1990]. Boden distinguishes combinatorial re-use of existing ideas, exploratory search for new ideas, and transformational creativity where the search space is also subject to change. While Boden does not give a computational account of how concepts or ideas could be created, her taxonomy provides a starting point for characterizing various possible approaches to concept creation. We extend it with approaches based on extraction and induction. Our taxonomy of different types of concept creation approaches is reflected in the different kinds of input they take: concept extraction transforms an existing representation of a concept to a different representation, concept induction generalizes from a set of instances of concepts, concept recycling modifies existing concepts to new ones, while concept space exploration takes a search space of concepts as input.

— **Concept Extraction** is the task of extracting and transforming a conceptual representation from an existing but different representation of the same idea. Typically, the extracted representation is more concise, explicit and centralized. Concept extraction is frequently applied to textual corpora, such as extracting semantic relations between concepts. For instance, the conceptual information that wine is a drink could be extracted from the natural language expression “wine and other drinks”. For an overview of information extraction methods see e.g., the work of Sarawagi [2008].

— **Concept Induction** is based on instances of concepts from which a concept or concepts are learned. Drawing from the field of machine learning and data mining, two major types of inductive concept creation can be identified:

— **Concept Learning** is a supervised activity, where a description of a concept is formed inductively from (positive and negative) examples of the concept (e.g., Kotsiantis [2007]). In logical terms, (a sample from) the extension of a concept is given and machine learning is then used to construct its intension. An example from the field of music is a system that is given songs by the Beatles and other bands, and the system then learns the typical chord progression patterns, as concepts, used by the Beatles.

— **Concept Discovery** is an unsupervised inductive activity where the memberships of instances in concepts are not known in advance, but methods such as clustering are used to generalize natural concepts from given examples (e.g., Jain et al. [1999]). Continuing the example above, given songs by a number of bands, the system can discover different genres of music by clustering the songs.

While inductive methods can be used to uncover existing concepts, more interesting applications arise when these methods are used to discover and formulate new possible concepts based on data.
— *Concept Recycling* is the creative reuse of existing concepts (e.g., see Aamodt and Plaza [1994]). (The original concepts usually stay in existence, too; recycling does not imply they have to be removed.) We mention two typical methods.

— *Concept Mutation*: modify some given concept by adding, removing or changing something in it. For instance, the concept of “mobile phone” has been modified to “smart phone” by allowing new applications to be used on the phones. Changing existing concepts is a well-known form of concept creation also with humans; Osborn [1953] lists a variety of techniques for such purpose. Two common mutational operations are *generalization* and *specialization* of existing concepts.

— *Concept Combination*: combine two or more existing concepts to form a new one, such as in *conceptual blending*, where “computer program” and “virus” can form a conceptual blend of “computer virus”.

Both mutation and (re-)combination of concepts are utilized heavily in evolutionary methods for concept creation. While recycling may sound an easy way to create new concepts, a key problem is to measure how meaningful and interesting the generated concepts are.

— *Concept Space Exploration* takes as input a search space of possible new concepts and locates interesting concepts in it. The space can be specified either declaratively or procedurally. A poetry writing system, for instance, can take as input a template with blanks to fill with words. This specifies a search space of possible poems, or concepts. Again, a crucial and non-trivial issue is how the quality of new concepts is measured.

The above accounts of concept creation differ in terms of their input and thus help to characterize different settings where concept creation is used. The techniques involved may, however, share methodological similarities; see the references given above for overviews. In particular, many of the above concept creation techniques can actually be described as search [Wiggins 2006a; 2006b], where the search space and the ways of traversing it depend on the specific technique. This makes it sometimes difficult to tell if a system is exploratory in nature or rather more of one of the above types of concept creation. For instance, different operations used in concept mutation can be seen as traversing a space of concepts; the space reachable by the system is defined by the initial concept to be mutated and all possible combinations of mutation operations.

Additionally, there is *transformational* creativity where the system also adjusts its own operation [Boden 1990]. Transformational or meta-creativity takes any of the above types of concept creation onto a more abstract level where additionally the data, assumptions, search methods, evaluation metrics or goals are also modified. This potentially takes a system towards higher creative autonomy [Jennings 2010].

The formalization by Wiggins [2006a] of creativity as search gives a unifying formalization over Boden's categorization (and of ours). He shows not only how both combinatorial and exploratory creativity can be seen as search at the concept level, as already mentioned above, but also how transformational creativity can be seen as search at the meta-level, i.e. on a level including also meta-information about the concept level search. This provides a powerful way of describing and comparing a wide range of creative systems.

As can be seen from the categorization above, techniques from machine learning find applications in several concept creation tasks. For a recent review on the use of machine learning in computational creativity, including the transformational case, see Toivonen and Gross [2015].

The space does not allow proper review of approaches and techniques suitable for concept creation, so that is left outside the scope of this paper. In the review that follows, we complement the more abstract works of Boden [1990] and Wiggins [2006a] by
presenting concrete conceptual representations that have been used in computational concept creation.

1.2. Organizing Conceptual Representations

Conceptual representations can be distinguished, at least, under two perspectives. One is the level of representation (symbolic, spatial vs. connectionist representations), and the other is descriptive vs. procedural representations. We give a brief introduction to these distinctions below.

Levels of Conceptual Representations. Gärdenfors [2000] proposes three levels of cognitive representations: a symbolic level; a conceptual level modeled in terms of conceptual spaces, which we term the spatial level here; and a sub-conceptual connectionist level. His theory is aimed at studying cognitive functions of humans (and other animals) as well as artificial systems capable of human cognitive functions. The idea is that all these three levels are connected, sensory input to the connectionist level feeding spatial representations of concepts, which then become symbolic at the level of language.

— At the symbolic level, information is represented by symbols. Rules are defined to manipulate symbols. Symbolic representations are often associated with Good Old Fashioned AI (GOFAI) [Haugeland 1985]. An underlying assumption of GOFAI research is that human thinking can be understood in terms of symbolic computation, in particular, computation based on formal principles of logic. However, symbolic systems have proved less successful in modeling aspects of human cognition beyond those closely related to logical thinking, such as perception. Furthermore, within a symbolic representation, meaning is internal to the representation itself; symbols have meaning only in terms of other symbols, and not directly in terms of any real world objects or phenomena they may represent. Gärdenfors proposes addressing this symbol grounding problem by linking symbols at this level to conceptual structures at the spatial conceptual level below. In linguistic terms, words (or expressions in a language of thought [Fodor 1975]) exist at this symbolic level, but ground their meaning (semantics) in the spatial level.

— At the spatial level, information is represented by points or regions in a conceptual space which is built upon quality dimensions with defined geometrical, topological or ordinal properties. Similarity between concepts is represented in terms of the distance between points or regions in a multidimensional space. This formalism offers a parsimonious account of concept combination and acquisition, both of which are closely related to conceptual similarity. Moreover, by defining dimensions with respect to perceptual qualities, spatial representations are grounded in our experience of the physical world, which provides a semantics closely aligned with a human sense of meaning.

— At the connectionist level, information is represented by activation patterns in densely connected networks of primitive units. A particular strength of connectionist networks is their ability to learn concepts from observed data by progressively changing connection weights. Nevertheless, the weights between units in the network offer limited explanatory insights into the process being modeled, which requires an understanding of the computation of each unit.

The three levels of representation outlined above differ in representational granularity, and each level has its own strengths and weaknesses in modeling cognitive and creative functions. While Gärdenfors [2000] takes the spatial level (which he calls

the “conceptual level”) as the most appropriate for modeling concepts, he stresses the
links between the levels and that different representational formalisms should be seen
as complementary, rather than competing. As such, choices of representation should
be made in accordance with scientific aims and in response to the challenges of the
particular problem at hand.

It is important to distinguish between the axis of these different levels of computa-
tional representation and the quite orthogonal axis supplied by the nature of human
conceptualization, most particularly shown in categorical perception. Here, the hierar-
chical form is of the thing being modeled, and not of the modeling formalism.

For example, a specific phenomenon that can occur in human conceptualization is
found in categorical perception, in which the process of categorization has an ampli-
fying effect on the underlying perception, by strengthening the perceptual differences
across category boundaries and weakening them within boundaries. The classical ex-
ample is speech perception, in which a varying speech sound is identified in terms
of either one or another distinctive phoneme, even though a sharp distinction is ob-
jectively not present in the underlying sound pattern. Other examples of categorical
perception are found in music (for example, musical note identification) and vision
(color categorization)—see Goldstone and Hendrickson [2010] for a recent overview).
A further complication, as for example in color categorization, is that the categories
can be hierarchical, as “red” is a super-category of “scarlet” and “vermilion”. The rep-
resentations surveyed here capture these points in various ways.

Also, in the case of human conceptualization the question arises as to why (e.g., from
an evolutionary or developmental perspective) concept formation based on perceptual
categorizations emerged. That is, why humans would tend to distinguish different cat-
egories even when the underlying perceptual domain is continuous. Perhaps this re-
results from the close (evolutionary) relation between perception and action [van der
Velde 2015]. In performing an action we (often) need to choose just one option out of
many, given the physical boundary conditions related to the action. Simply stated, we
can run in only one direction at a time, which forces a choice between the many op-
tions that could be available. The process of making such a choice could also affect the
process of classifying the underlying perceptual domain.

Descriptive vs. Procedural Representations. The other perspective to conceptual rep-
resentations is the distinction between descriptive and procedural representations.
As the name indicates, a descriptive representation describes the artifact being rep-
resented. The description may be low or high level, complete or partial. A procedural
representation, on the other hand, specifies a procedure, e.g., a program, that once ex-
ecuted produces the artifact being represented. Similar to descriptive representations,
the procedure may be low or high level, complete or partial. A procedural representa-
tion is sometimes more succinct than a descriptive representation of the same idea. An
example is the Fibonacci numbers—comparing the definition based on recurrence with
an infinite sequence of integers. Conversely, for everything we know how to produce,
there exists at least one procedural representation, although in many cases descrip-
tive representations are employed. Like the three levels of representations introduced
above, descriptive and procedural representations each have particular advantages
and deficiencies. At each of the three levels, both descriptive and procedural representa-
tions exist or can be constructed.

A noteworthy point is that the conceptual representations discussed in this review
are for the purpose of eliciting certain information or knowledge which affords certain
inferences. Any conceptual representation can be ‘represented’ in other forms for pur-
poses other than the one discussed here, e.g., any procedural representation has to be
implemented in program code to be executed.
Outline and Structure of This Paper. In this review, we bring together conceptual representations relevant to concept creation, which are either the targets of concept creation or part of creating other concepts. Some representations have general presence in computer science while others were especially created for concept creation, such as bisociation (Section 2.1), procedural representations of music and image (Section 5), and plan operator for story generation (Section 6.1.3). Part of the conceptual representations reviewed are relevant to a broad range of creative tasks (Section 2, 3, 4 and 5), while part of them are unique for certain creative domains (Section 6). For every representation reviewed, we cover the inference it supports, the computational means of building it, and its application in concept creation.

The conceptual representations included in this review are primarily organized according to the distinction between symbolic, spatial and connectionist representations, in Sections 2, 3 and 4 respectively. Symbolic and spatial representations are predominately descriptive, and connectionist representations can largely be considered procedural. Procedural representations, across the three levels, are reviewed in Section 5. Section 6 introduces conceptual representations used in four popular research domains of the computational creativity community: language, music, image and emotion. The information in the four domains may have representations at all the three levels and both descriptive and procedural representations. Discussions on the results of this review, conclusions and future work are presented in Section 7.

2. SYMBOLIC REPRESENTATIONS
Symbols are used to represent objects, properties of objects, relationships, ideas and so forth. Certain symbols might be better discussed within domains. For instance, in Section 6, we introduce symbols used in the domains of language, music, image and emotion, such as word, music note and pixel matrix. These are atomic representations. Examples of more complex symbolic representations include plan operator, SVG (Scalable Vector Graphics) file and emotional categories. In this section, we present symbolic representations that are applicable to many domains, including association, semantic relation, semantic network, ontology, information network and logic.

2.1. Association
Association means “something linked in memory or imagination with a thing or person; the process of forming mental connections or bonds between sensations, ideas, or memories”\(^3\). Association assumes a connection between concepts or symbols \(C_1\) and \(C_2\), but does not assume any specific conditions on \(C_1\) and \(C_2\). Also, the nature of the connection is not of primary focus, contrasting it with semantic relation (Section 2.2), semantic network (Section 2.3) and ontology (Section 2.4).

In the field of computational creativity, the associations spanning over two different contexts (domains/categories/classes) are of special interest, in line with Mednick’s definition of creative thinking as the ability of generating new combinations of distant associative elements [Mednick 1962]. Koestler [1964] also identified such cross-domain associations as an important element of creativity, and calls them bisociations. Figure 1 shows a schematic representation of bisociation, where concepts \(C_1\) and \(C_2\), from two different contexts \(D_1\) and \(D_2\) respectively, are bisociated. Bisociations, in various contexts, may be especially useful in discovering or creating analogies and metaphors.

According to Berthold [2012], bisociation can be informally defined as “(sets of) concepts that bridge two otherwise not—or only very sparsely—connected domains whereas an association bridges concepts within a given domain.” We argue that two concepts are bisociated if there is no direct, obvious evidence linking them, one has

\(^{3}\text{http://www.merriam-webster.com/dictionary/association.}\)
to cross contexts to find the link, and this new link provides novel insights into the domains. The bisociative connection between \( C_1 \) and \( C_2 \) may be represented together with a *bridging* concept \( B \), which has links to both \( C_1 \) and \( C_2 \) (see Figure 2). An example is the bisociation between evolution in nature and evolutionary computing bridged with the concept “optimization”. In addition to bridging concepts, Berthold [2012] introduces other types of bisociations, i.e. *bridging graphs* and *bridging by structural similarity*. The author points out that bridging concepts and bridging graphs require that the two domains have a certain type of neighborhood relation, while bridging by structural similarity allows matching on a more abstract level.

A number of bisociation discovery methods are based on graph representations of domains and finding cross-domain connections which are potentially new discoveries [Dubitzky et al. 2012]. It has also been shown by Swanson [1990] that bisociation discovery can be tackled using literature mining methods. Swanson proposed a method for finding hypotheses spanning over previously disjoint sets of literature. To find out whether phenomenon \( a \) is associated with phenomenon \( c \) although there is no direct evidence for this in the literature, he searches for intermediate concepts \( b \) connected with \( a \) in some articles, and with \( c \) in some others. Putting these connections together and looking at their meaning may provide new insights about \( a \) and \( c \).

Let us illustrate this with an example that has become well known in literature mining. In one of his studies, Swanson investigated if magnesium deficiency could cause migraine headaches. He found more than 60 pairs of articles—consisting of one article from the literature about migraine (\( c \)) and one article from the literature about magnesium (\( a \))—connecting \( a \) with \( c \) via various third terms \( b \). For example, in literature about magnesium there is a statement that magnesium is a natural calcium channel blocker, while in the literature about migraine we read that calcium channel blockers can prevent migraine attacks. In this case, calcium channel blockers are a bridge between the domains of magnesium and migraine. Closer inspection showed that 11 identified pairs of documents were, when put together, suggestive and supportive for a hypothesis that magnesium deficiency may cause migraine headaches [Swanson 1990].

Many researchers have followed and further developed Swanson’s idea of searching for linking terms between two domains in the framework of literature mining. An overview of the literature-based discovery approaches and challenges is provided by Bruza and Weeber [2008]. Furthermore, some other data mining approaches, such as co-clustering and multimode clustering [Govaert and Nadif 2014], may be suitable for identifying certain types of bisociations.
2.2. Semantic Relation

Broadly speaking, semantic relations are labeled relations between meanings, as well as between meanings and representations. In contrast to associations, semantic relations have meanings as indicated by their labels. The number of semantic relations is virtually unlimited. Some important semantic relations are synonymy, homonymy, antonymy, hyponymy-hypernymy, meronymy-holonymy, instance_of relation, causal relation, locative relation and temporal relation. Besides the general relation types, there are domain-specific relations, such as the ingredient_of relation in the food domain, or the activates relation in the biomedicine domain.

Here we briefly review how semantic relations have been extracted from various sources, mostly text (see Table I for a summary of selected research). We distinguish between approaches based on lexico-syntactic patterns and machine learning. The pioneering work of Hearst [1992] opened the era of discovering semantic relations using lexico-syntactic patterns. An example of lexico-syntactic patterns used in the automatic detection of hyponyms is “NP₁ such as NP₂”, where NP₂, a noun phrase, is potentially a hyponym of NP₁, another noun phrase. With a set of seed instances of certain relation (e.g., hyponymy), this method identifies sequences of text that occur systematically between the concepts of the instances. The patterns discovered are used in the automatic extraction of new instances. In this line of work, the relations that form the backbone of ontologies were first considered, e.g., hyponymy [Hearst 1992] and meronymy [Berland and Charniak 1999], followed by other relations, such as book author [Brin 1999], organization location [Agichtein and Gravano 2000], inventor [Ravichandran and Hovy 2002], etc.

As an alternative approach, machine learning techniques have been applied to the extraction of semantic relations from text. An overview of methods with different degrees of supervision is given by Nastase et al. [2013]. Unsupervised methods, e.g., based on clustering or co-occurrence, are mostly used to discover hypernymy and synonymy relations, and often in combination with pattern-based methods [Caraballo 1999; Pantel and Ravichandran 2004]. In supervised machine learning, models are trained by generalizing from labeled examples of expressed relations. Labeled data was provided as part of some shared tasks, such as semantic evaluation workshops (SemEVAL). SemEval-2007 Task 4 [Girju et al. 2007] provides a dataset of meronymy, causality, origin, etc., followed by a dataset of some other relations in SemEval-2010 Task 8 [Hendrickx et al. 2010]. In addition, SpaceEval 2015⁴ focuses on various spatial relations, such as path, topology and orientation. Navigli and Velardi [2010] used an annotated dataset of definitions and hypernyms for learning word class lattices. Instead of manually annotated datasets, WordNet [Fellbaum 1998; Miller et al. 1990], Wikipedia⁵ and other resources can be used for distant supervision (as large seeds for bootstrapping) [Snow et al. 2005; Mintz et al. 2009].

In contrast to extracting predefined semantic relations, the Open Information Extraction (OIE) paradigm does not depend on predefined patterns, but considers rela-

---

⁴http://alt.qcri.org/semeval2015/task8/
tions as expressed by parts of speech [Fader et al. 2011], paths in a syntactic parse tree [Ciaramita et al. 2005], or sequences of high-frequency words [Davidov and Rappoport 2006]. OIE methods are used in ReVerb [Etzioni et al. 2011], Ollie [Mausam et al. 2012] and ClausIE [Del Corro and Gemulla 2013].

Besides general semantic relations, automatically extracted semantic relations have been part of knowledge discovery in many specific domains, such as biology [Miljkovic et al. 2012], and have potential for concept creation, too. Semantic relations are also important components of semantic networks (Section 2.3) and ontologies (Section 2.4).

As well as the specific relations between symbols discussed here, other approaches take a more generally relational view of semantics, seeing text or word meanings in terms of statistical relations to other words (in the form of topic models, latent vectors, or distributional semantics, for example); we introduce such approaches in the section about spatial representations below (Section 3).

2.3. Semantic Network

Semantic networks [Sowa 1992] are a category of symbolic representations that represent collections of semantic relations between concepts. Figure 3 shows a small semantic network, which represents the sentences “The bottle contains wine. Wine is a beverage.” The concepts, i.e. “bottle”, “wine” and “beverage”, are denoted by nodes, and the relations between them, i.e. \textit{contains/contain} and \textit{is-a}, are represented by directed edges. The meaning of a concept is defined in terms of its connections with other nodes (concepts). The closely related semantic link networks [Zhuge 2012] are self-organized semantic models similar in many ways to semantic networks but emphasizing larger semantic richness and automatic link discovery.

An example of existent large semantic networks is ConceptNet [Liu and Singh 2004], a semantic network of commonsense knowledge. In ConceptNet, nodes are semi-structured English fragments, including noun, verb, adjective and prepositional phrases. Nodes are interrelated by one of the twenty four types of semantic relations, such as \textit{IsA}, \textit{PartOf}, \textit{UsedFor}, \textit{MadeOf}, \textit{Causes}, \textit{HasProperty}, \textit{DefinedAs}, and \textit{ConceptuallyRelatedTo}, represented by directed edges. The early versions of ConceptNet were built on the data of the Open Mind Common Sense Project\footnote{http://openmind.media.mit.edu.}, which collects commonsense knowledge from volunteers on the web by asking them to fill the blanks in sentences [Singh et al. 2002]. ConceptNet 5\footnote{http://conceptnet5.media.mit.edu.}, the current version, extends the previous versions with information automatically extracted from Wikipedia, Wiktionary\footnote{http://en.wiktionary.org/wiki/Wiktionary:Main\_Page.} and WordNet.

Semantic networks have wide applications, such as database management [Rousopoulos and Mylopoulos 1975], cybersecurity [AlEroud and Karabatis 2013], software engineering [Karabatis et al. 2009], etc. Semantic networks are also popular knowl-
edge sources in the computational creativity community. ConceptNet alone has been used in generating cross-domain analogies [Baydin et al. 2012], metaphor ideas for pictorial advertisements [Xiao and Blat 2013], Bengali poetry [Das and Gambèck 2014], and fictional ideas [Llano et al. 2016], as well as testing the novelty of visual blends [Martins et al. 2015]. Baydin et al. [2012] actually go beyond just using semantic networks, they also generate novel analogous semantic networks using an evolutionary algorithm.

Formally, semantic networks can in some cases be viewed as alternatives to spatial representations, the most obvious case being where nodes correspond with points in the space, and relations attached to arcs correspond with distances between them. This, however, is not a very conventional view. More common is the usage where nodes represent objects and values, and arcs represent relations between things represented by nodes, which is a much less subtle, and more logic-like approach. The geometry of spatial representations affords many implicit concepts that must be made explicit in the network formalism; this may be positive or negative depending on application. For example, in color space, red is a super-concept of scarlet and vermilion merely by virtue of geometry, and, given that the space is a good perceptual model, distances are implicit and do not need to be recorded; in a network representation the super/sub-concept relation would need to be recorded explicitly. But many concepts do not conform to the regularity of Euclidean geometry, and in these cases, a network representation may be more appropriate.

As a symbolic representation, the meanings of concepts (nodes) in a semantic network are specified purely in terms of relations to other symbolic concepts – there is no grounding [Harnad 1990]. However, mappings between symbolic networks and spatial representations, such as Gärdensfors [2000] theory of Conceptual Space (cf. Section 3.1), could be developed to leverage the strengths of each form of representation. For example, the “wine” and “beverage” concepts from the semantic network in Figure 3 could correspond to regions in a conceptual space, whereby “wine” would typically be a sub-region of “beverage” within some set of contextually appropriate dimensions. This geometrical relationship implicitly represents that “wine” is a kind of “beverage”, as opposed to the explicit “is-a” type relation used in the semantic network.

2.4. Ontology

A widely accepted definition of ontology in computer science is by Gruber [2009]:

“In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members).”

In computer science (and computational creativity) we can thus understand ontologies as resources of formalized knowledge, but the degree of formalization can vary.9

The relationship between ontologies and semantic networks is that ontologies provide representational primitives that can be used in semantic networks. The structural backbone of an ontology is a taxonomy, which defines a hierarchical classification of concepts, while an ontology represents a structured knowledge model with various kinds of relations between concepts and, possibly, rules and axioms [Navigli 2016]. For

---

9Different communities use the word “ontology” for different meanings. In philosophy, “ontology” refers to the philosophical discipline dealing with the nature and structure of “reality” and is opposed to epistemology, which concerns the understanding of reality and the nature of knowledge [Guarino and Giaretta 1995].
instance, the expression “the bottle contains wine” in a semantic network (Figure 3) obtains a much richer meaning when combined with ontologies that include “bottle”, “contains” and “wine”, as well as their properties and relations allowing one to reason with them.

We can distinguish between upper, middle and domain ontologies [Navigli 2016]. Upper ontologies encode high level concepts (e.g., concepts such as “entity”, “object” and “situation”), and are usually constructed manually. Their main function is to support interoperability between domains. They are often linked to several lower level ontologies. Examples of upper level ontologies are SUMO [Pease and Niles 2002], DOLCE [Gangemi et al. 2002] and Cyc10 [Lenat 1995]. Middle ontologies are general purpose ontologies and provide the semantics needed for attaching to domain specific concepts. For instance, WordNet [Fellbaum 1998; Miller et al. 1990] is a widely used middle ontology. Domain ontologies model the concepts, individuals and relations of the domain of interest (e.g., the Gene Ontology11). Existent ontologies often have more than one level, e.g., SUMO and WordNet contain the upper level concepts, middle ontologies and some domain ontologies.

From the perspective of the amount of conceptualization, Sowa [2010] and Biemann [2005] distinguish between formal (also called axiomatized), terminological (also called lexical) and prototype-based ontologies (Figure 4).

— Formal ontologies (e.g., SUMO) are represented in logic, using axioms and definitions. Their advantage is the inference mechanism, enabling the properties of entities to be derived (see in Figure 4 how one can derive that “chili con carne” is “non-vegetarian” food). Nevertheless, a high encoding effort is needed and there is a danger of running into inconsistencies.

— Terminological ontologies (e.g., WordNet) have concept labels (terms used to express them in natural language). The lexical relations between terms, e.g., synonymy, antonymy, hypernymy and meronymy, determine the relative positions of concepts but do not completely define them. The difference between a terminological ontology and a formal ontology concerns its degree [Sowa 2010]. If represented as a graph, a terminological ontology is a special type of semantic network (see Section 2.3).

— Prototype-based ontologies represent categories (concepts) with typical instances (rather than concept labels or axioms and definitions in logic). New instances are classified based on a selected measure of similarity. Typically, prototype-based ontologies are taxonomies, since they are limited to hierarchical (unspecified) relations and are constructed by clustering techniques. Formal and prototype-based ontologies are often combined into mixed ontologies, where some subtypes are distinguished by axioms and definitions, but other subtypes are distinguished by prototypes. Gärdenfors’ theory, which affords a kind of geometrical reasoning comparable with the logical reasoning afforded by ontologies, also affords reasoning about prototypes [Gärdenfors 2000].

Taxonomy and ontology learning can be built upon the methods of automatically extracting semantic relations (see Section 2.2). For instance, Kozareva and Hovy [2010] and Velardi et al. [2013] use a combination of pattern- and graph-based techniques. Also, clustering techniques have been used in taxonomy learning [Cimiano et al. 2005; Fortuna et al. 2006].

Ontologies have been used in many computational creativity tasks. The combination of thematically different ontologies is applied in modeling analogies (relating different symbols based on their similar axiomatisation), metaphors (blending symbols of a

10OpenCyc is a public version of Cyc (which is proprietary).
source domain into a target domain, based on an analogy, and imposing the axiomatisation of the former on the latter), pataphors (extending a metaphor by blending additional symbols and axioms from the source domain into the target, thus resulting in a new domain where the metaphor becomes reality), and conceptual blending (blending and combining two domains for the creation of new domains) [Kutz et al. 2012].

2.5. Information Network
Information networks refer to any structure with connected entities, such as social networks. Mathematically, an information network can be represented as a graph, where graph vertices represent the entities, and edges represent the connections between them. Semantic networks are a special case of information networks where vertices

Fig. 4. Examples of a) formal ontology, b) terminological ontology and c) prototype-based ontology [Biemann 2005]. Note that b) and c) illustrate only the structural backbones of the two ontologies. © Chris Biemann. Reproduced by permission.
and edges carry semantic meaning, i.e. labeled by semantic concepts. Information networks represent a broader category of knowledge representation than semantic networks. Study of information networks is less focused on the meaning encoded in the connections (which is the main focus of studying semantic networks) but more on the structure of networks.

Studies of information networks include the work of Sun and Han [2013], where an information network is defined simply as a directed graph where both the nodes and edges have types and the edge type uniquely characterizes the types of its adjacent nodes. When there are more than one type of node or edge in an information network, the network is called a heterogeneous information network; if it has only one type of node and only one type of edge, it is a homogeneous information network.

There are plenty of examples of information networks. Bibliographic information networks [Juršič et al. 2012; Sun and Han 2012] are networks connecting the authors of scientific papers with their papers. Specifically, they are heterogeneous networks with two types of nodes (authors and papers), and two types of edges (citations and authorships). Online social networks represent the communication in online social platforms. Biological networks contain biological concepts and the relations between them.

The methods of discovering new knowledge in homogeneous information networks can be split into several categories: node/edge label propagation [Zhou et al. 2003], link prediction [Barabási et al. 2002; Adamic and Adar 2003], community detection [Yang et al. 2010; Plantié and Crampes 2013], and node/edge ranking [Jeh and Widom 2002; Kondor and Lafferty 2002]. A popular set of methods are based on eigenvalues and eigenvectors (commonly referred to as spectral methods). For example, in detecting communities, the community structure is extracted from either the eigenvectors of the Laplacian matrix [Donetti and Munoz 2004] or the stochastic matrix [Capocci et al. 2005] of the network.

The methods developed for homogeneous information networks, as introduced above, can be applied to heterogeneous information networks by simply ignoring the heterogeneous information altogether. This does, however, decrease the amount of information used and can therefore decrease the performance of the algorithms [Davis et al. 2011]. Approaches that take into account the heterogeneous information are therefore preferable, such as network propositionalization [Grčar et al. 2013], authority ranking [Sun et al. 2009; Sun and Han 2012], ranking based clustering [Sun et al. 2009; Sun and Han 2012], classification through label propagation [Hwang and Kuang 2010; Ji et al. 2010], ranking based classification [Sun and Han 2012], and multi-relational link prediction [Davis et al. 2011].

2.6. Logic

Many kinds of logics have been used to represent concepts and complex knowledge, such as classical First Order Logic, Modal Logics—including Linear Temporal Logic and Deontic Logic—and other non-classical logics, e.g., Default and Non-Monotonic Logics.

A declarative representation of a concept can be a single symbol. More complex concepts can be represented by the composition of simpler formulas (corresponding to simpler concepts). These compositions are built by establishing some relations (e.g., conjunction, disjunction, negation, implication, etc.) between concepts. Ontologies (see Section 2.4) are built with a specific sub-family of logic languages, i.e. the description logics—the veg_food(x)—concept in Figure 4 is one such example resorting to first order logic.

Logic-based symbolic approaches can be used to represent and reason with both time-independent and temporal concepts [Bouzid et al. 2006]. In addition to descriptive representations of concepts, logic-based approaches, in particular Logic Programs,
also be used for symbolic procedural representations via inductive definitions [Hou et al. 2010]—e.g., the following Logic Program (written in Prolog) defines the concepts of even and odd natural numbers, assuming $\text{suc}(X)$ stands for the successor of the natural number $X$:

\[
\begin{align*}
\text{even}(0). \\
\text{even}(\text{suc}(X)) & : - \text{odd}(X). \\
\text{odd}(\text{suc}(X)) & : - \text{even}(X).
\end{align*}
\]

The use of logical representations and tools in computational creativity tasks has just started. Two of such work concern the computational modeling of conceptual blending, a cognitive process which, by selectively combining two distinct concepts, leads to new concepts, called “blends” [Fauconnier and Turner 1998]. Besold and Plaza [2015] constructed a conceptual blending engine based on generalization and amalgams; and Confalonieri et al. [2015] used argumentation in order to evaluate conceptual blends.

3. SPATIAL REPRESENTATIONS

In comparison to symbolic and connectionist representations, the importance of spatial representations was raised by Gärdenfors [2000] with the theory of Conceptual Spaces. Since before Gärdenfors’ proposal, in the computing community, a spatial representation called Vector Space Model (VSM) has been a popular tool for modeling many different domains and applications [Dubin 2004], with topic model being a prominent example of concept creation. In this section, we introduce these three kinds of spatial representations and their relevance to concept creation.

3.1. Gärdenfors’ Conceptual Spaces

Gärdenfors [2000] proposes a geometrical representation of concepts, named conceptual spaces. A conceptual space is formed by quality dimensions, which “correspond to the different ways stimuli are judged to be similar or different” [Gärdenfors 2000, p. 6]. An archetypal example is a color space with the dimensions hue, saturation (or chromaticism) and brightness. Each quality dimension has a particular geometrical structure. For instance, hue is circular, whereas brightness and saturation have finite linear scales (Figure 5). It is important to note that the values on a dimension need not be numbers.

A domain is a set of integral (as opposed to separable) dimensions, meaning that no dimension can take a value without every other dimension in the domain also taking a value. Therefore, hue, saturation and brightness in the above color model form a single domain. A conceptual space is simply “a collection of one or more domains” [Gärdenfors 2000, p. 26]. For example, a conceptual space of elementary colored shapes could be

Fig. 5. The color space in terms of hue (radial angle in the horizontal plane), brightness (vertical axis), and saturation (horizontal radial magnitude), which are integral dimensions and therefore form a domain.
defined as a space comprising the above domain of color and a domain representing the perceptually salient features of a given set of shapes.

A property corresponds to a region of a domain in a conceptual space (and more specifically, a natural property corresponds to a convex region). A concept in Gärdenfors' formulation is represented in terms of its properties, normally including multiple domains. Interestingly, this means that property is a special, single-domain case of concept. For instance, the concept “red” is a region in the color space. It is also a property of anything which is red.

An object is a point in a space (i.e. a point in a certain region (property) of each of one or more domains). The spatial location of an object in a conceptual space allows the calculation of distance between objects, which gives rise to a natural way of representing similarities. The distance measure may be a true metric (for example, Gärdenfors [2000] suggests that Euclidean distance is often suitable with integral dimensions, and cityblock distance with separable dimensions); or non-metric, such as a measure based on an ordinal relationship or the length of a path between vertices in a graph. When calculating distance, salience weights associated with each of the dimensions can be varied. It is the context in which a concept is used that determines which dimensions are the most prominent, and hence, have bigger weights.

Such spatial representations naturally afford reasoning in terms of spatial regions. Boundaries between regions are fluid, an aspect of the representation that may be usefully exploited by creative systems searching for new interpretations of familiar concepts. A further consequence of the geometrical nature of the representation is that conceptual spaces are particularly powerful in dealing with concept learning and concept combination. Learning can be modeled via supervised classification in terms of distance from prototypical centroids of regions, or via unsupervised clustering based on spatial distances. Combination can be understood in terms of intersection of spatial regions, or in terms of replacing values of one concept’s regions by another for more complex cases where logical intersection fails—see Gärdenfors [2000, Sections 4.4, 4.5] for discussion.

While Gärdenfors’ theory has yet to be fully formalized in mathematical terms, several approaches to formalization of some aspects of it have appeared in the literature. Two approaches build on an initial formalization by Aisbett and Gibbon [2001]. One, based on fuzzy set theory, is presented in detail by Rickard et al. [2007b], drawing on their previous work [Rickard 2006; Rickard et al. 2007a]. The other, employing vector spaces, is presented by Raubal [2004], with subsequent related work by Schwering and Raubal [2005] and Raubal [2008]. Moreover, Chella and colleagues [2004; 2007; 2008; 2015] have done substantial work in formalizing Conceptual Space theory and applying it to robotics. There is empirical evidence to support the theory as a cognitive model [Jäger 2010].

3.2. Vector Space Model (VSM)

As Gärdenfors [2000] points out, an appropriate approach to computation with geometric representations is the use of Vector Space Models (VSMs): they provide algorithms and frameworks for classification, clustering and similarity calculation which lend themselves directly to some of the key questions in conceptual modeling. In text modeling, the use of VSMs is long established, having been introduced by Salton when building the SMART information retrieval system [Salton 1971], taking the terms in a document collection as dimensions. Every document is represented by a vector of terms, where the value of each element is the (scaled) frequency of the corresponding term in the document. Each term in a document has a different level of importance, which can be represented by additional term weights in a document vector. A popular weighting schema is TF-IDF (Term Frequency - Inverse Document Frequency) [Spärck
Conceptual Representations for Computational Concept Creation

Jones 1972], which is based on the idea that terms that appear in many documents are less important for a single document. To see how similar two documents are, their vectors can be compared, with the most commonly used similarity measure being the cosine value of the angle between the document vectors [Salton 1989]. Such a VSM assumes pairwise orthogonality between term vectors (the columns), which generally does not hold due to correlation between terms. The Generalized Vector Space Model (GVSM) provides a solution for this problem [Wong et al. 1985; Raghavan and Wong 1986].

On the other hand, the same matrix can be used to calculate the similarity between two terms (words), by taking a different perspective: a term can be represented by a vector over the documents where they appear (i.e., if document vectors are rows in the document-term matrix, term vectors are the columns). This approach has been exploited by Turney and Pantel [2010] to build models of lexical meaning which reflect human similarity judgments, by reducing the co-occurrence context from the scale of whole documents to windows of a few words. In a contrasting approach, word vectors with similar properties are learned by neural networks (see e.g., Mikolov et al. [2013]), and these are good at capturing syntactic and semantic regularities, often remaining computationally efficient and retaining low dimensionality—see Baroni et al. [2014] for a review and comparison. A range of distance measures can also be used with these models; although cosine distance is the most commonly used, others can be more suited to different domains and tasks, and Kiela and Clark [2014] show a mean-weighted cosine distance variant to be most accurate in reflecting human judgments.

Extending the above VSMs to model the compositional meaning of phrases and sentences (rather than individual words) is the subject of much current research, with a range of methods including hierarchical compression using neural auto-encoders (e.g., Socher et al. [2013]), sequence modeling using convolutional networks (e.g., Kalchbrenner et al. [2014]), and categorical combination using tensor operations (e.g., Coecke et al. [2011]). Extension beyond the sentence to models of discourse meaning is also being investigated (e.g., Kalchbrenner and Blunsom [2013]).

In addition to word-context matrices, pair-pattern matrices have been used in VSMs, where rows correspond to pairs of terms and columns correspond to the patterns in which the pairs occur. They are used to measure the similarity of semantic relations in word pairs [Lin and Pantel 2001; Turney and Littman 2003]. Higher-order tensors (matrices are second-order tensors), such as a word-word-pattern tensor, has also been found useful in measuring the similarity of words [Turney 2007].

VSM-based models have been used in generating spoken dialogues [Wen et al. 2016] and modern haikus [Wong and Chun 2008]. Venour et al. [2010] constructed a novel semantic space, where the distance between words reflects the difference in their styles or tones, as part of generating linguistic humors. Juršič et al. [2012] took advantage of a document-term matrix and centroid vectors in order to find bridging terms of two literatures. Besides constructing VSMs from text, vectors of RGB color values were used by de Melo and Gratch [2010] to evolve emotional expressions of virtual humans. Thorogood and Pasquier [2013] used vectors of low-level audio features in generating audio metaphors. Maher et al. [2013] used vectors of attributes (e.g., display area, amount of memory, and CPU speed) to measure surprise. Furthermore, the future applications of VSMs in computational creativity tasks were discussed by McGregor et al. [2014].

3.3. Topic Model

Topic modeling is a general approach to modeling text using VSMs. It assumes that documents (or pieces of text) are composed of, or generated from, some underlying latent concepts or topics. One of the earliest variants, Latent Semantic Analysis
LSA [Landauer et al. 1998] has become a widely used technique for measuring similarity of words and text passages. LSA applies singular value decomposition (SVD) to a standard word-document (or word-context, where “context” refers to some window of words around a term) matrix; the resulting eigenvectors can be seen as latent concepts, as they provide a set of vectors that characterize the data but abstract away from the surface words, while relating words used in similar contexts to each other. By using these concept vectors as the bases, we obtain a new latent semantic space, and by limiting them to the eigenvectors with the largest values, this space can have a drastically smaller number of dimensions, while still closely approximating the original. LSA is not limited to words and their contexts. It can be generalized to unitary event types and the contexts in which instances of the event types appear (e.g., bag-of-features in computer vision problems [Sivic et al. 2005]); and it has been successfully applied in many tasks including topic segmentation [Olney and Cai 2005]. However, while the number of dimensions chosen for this latent semantic space is critical for performance, there is no principled way of doing it. Moreover, the dimensions in the new space do not have obvious interpretations.

An approach which solves some of these problems is Probabilistic Latent Semantic Indexing (PLSI) [Hofmann 1999]. PLSI can be seen as a probabilistic variant of LSA; rather than applying SVD to derive latent vectors by factorization, it fits a statistical latent class model on a word-context matrix using Tempered Expectation Maximization (TEM). This process still generates a low-dimensional latent semantic space in which dimensions are topics, but now these topics are probability distributions over words—i.e. sets of words with a varied degree of membership to the topic, which we can see as latent concepts here—and documents are probabilistic mixtures of these topics. The number of dimensions in the new space is determined according to the statistical theory for model selection and complexity control. This can be used directly to model document content and similarity, or e.g., embedded within an aspect Markov model to segment and track topics [Blei and Moreno 2001].

A shortcoming of PLSI, however, is that it lacks a generative model of document-topic probabilities: it therefore must estimate these from topic-segmented training data, and is not directly suitable for assigning probability to a previously unseen document, instead requiring an additional estimation process during decoding [Blei and Moreno 2001]. These are addressed by models such as Latent Dirichlet Allocation (LDA) [Blei et al. 2003], which take a similar latent-variable approach but make it fully Bayesian, allowing topics to be inferred without prior knowledge of their distribution. LDA has been used very successfully for fully unsupervised induction of topics from text in many domains—see e.g., Griffiths et al. [2005] and Hong and Davison [2010]. It requires the number of topics and certain hyper-parameters to be specified; but even these can be estimated by hierarchical Bayesian variants—see e.g., Blei et al. [2004].

Variants of LDA which incorporate data from outside the text can then go even further towards full concept discovery by inducing topics with relations between text and author, social network properties, and so on (see Blei [2012] for an overview). This has been particularly important in social media modeling, where texts themselves are very short. Here, extended variants have been developed and successfully applied in many ways, with the notion of topic (or concept) depending on objective: for example, to discover and model health-related topics [Paul and Dredze 2014]; to model topic-specific influences by incorporating information about network structures [Weng et al. 2010]; to detect and profile breadth of interest in audiences using timeline-aggregated data [Concannon and Purver 2014]; to build predictive models for review sites by including user- and location-specific information [Lu et al. 2016]; and to help predict
stock market movements via incorporating sentiment aspects [Nguyen and Shirai 2015].

Topic models have been used in various ways in the computational creativity community. Strapparava et al. [2007] used LSA to compute 

**lexical affective semantic similarity** in order to generate animated advertising messages. Topic vectors are used for conceptual blending by Veale [2012]. Xiao and Blat [2013] used a LSA space built from Wikipedia to generate pictorial metaphors.

4. CONNECTIONIST REPRESENTATIONS

Connectionist representations are composed of interconnected simple units, featuring parallel distributed processing. Hebb [1949] proposes that concepts are represented in the brain in terms of neural assemblies. The neural blackboard architecture [van der Velde and de Kamps 2006] suggests a way of combining neural assemblies in order to account for higher-level human cognition. The most commonly used family of connectionist models are artificial neural networks (ANNs). A more complex version of ANNs, deep neural networks, provides representations at a series of abstraction levels. In this section, we introduce these four conceptual representations and their relevance to concept creation.

Given their ‘network-alike’ look, these connectionist representations may resemble semantic networks (see Section 2.3), ontologies (see Section 2.4) and information networks (see Section 2.5). However, in each of the conceptual representations, the relations (and the way of interaction) between the units of a ‘network’ are fundamentally different. In particular, most connectionist representations are procedural. They have no explicit representations of concepts: the representations are rather distributed in the network and made explicit only when the connectionist representation is executed. For more detailed discussions on ANN frameworks for distributed representations, please refer to Kröse and van der Smagt [1993] and Hinton et al. [1986].

4.1. Neural Assembly and Neural Blackboard

Hebb [1949] proposes that concepts are represented in the brain in terms of neural assemblies. That is, in modern terminology, one can say that Hebb spoke about concepts in the brain (e.g., Abeles [2011]). This is also in agreement with Hebb’s hypothesis that thinking consists of sequential activations of assemblies, or “phase sequences”, as he called it (e.g., see Harris [2005] for a recent analysis). A neural assembly is a group of neurons that are strongly interconnected. As a consequence, when a part of an assembly is activated, e.g., by perception, it can reactivate the other parts of the assembly as well. For example, when we see an animal, certain neurons in the visual cortex will be active. But when the animal makes a sound, certain neurons in the auditory cortex will be active as well. Neural assemblies arise over time, based on learning processes such as Hebbian learning [Hebb 1949], i.e. neurons that fire together wire together. Over time, other neurons could become a part of the assembly as well, in particular when they are consistently active together with the assembly. Examples are the neurons that represent the word we use to name the animal, or neurons involved in our actions or emotions when we encounter it [van der Velde 2015].

Figure 6 illustrates (parts of) a neural assembly that could represent the concept “dog”. It would consist of the neurons involved in perceiving the animal, but also of neurons representing the relations is dog, can bark or has fur, and neurons that represent other aspects of our (e.g., emotional) experiences with dogs. Figure 6 does not imply that the ovals representing concepts are semantically meaningful on their own. Each ‘concept node’ in the figure represents a neural assembly, consisting of a network structure that integrates all aspects of the concept. In particular, it represents the interconnection of perception and action components of the concept, which represents
A fundamental issue with neural assemblies is how they can account for the non-associative aspects of human cognition [van der Velde 1993]. Of course, associations are crucial because without them we could not survive in any given environment. But for high-level cognition (e.g., language, reasoning), associations are not enough. Instead, relations are necessary, because they provide the basis for systematic knowledge. For example, we can apply the relation greater than to any pair of objects, not just the ones we happen to be acquainted with. In contrast, associations are always coincidental. For example, in the classical Pavlov experiment, the sound of the bell was associated with the food, but it could have been any other stimulus (as was indeed tested by Pavlov). Thus, relational knowledge cannot be established on the basis of associations alone.

The fact that in human high-level cognition relations are implemented in neural networks can result in a mixed representation, in which relations and associations are combined. For example, frequently occurring relations (e.g., dogs chase cats) could result in a more directly associative link between the concepts involved (“dogs” and “cats”). Such more direct associations are sometimes used in idioms like “They fight like cats and dogs”.

As noted, associations are direct links (connections) between neural entities (e.g., neurons or neural populations). Associative links, when they are available, result in very fast activations. In contrast, the more elaborate links provided by relations, which also require the specific type of relation to be expressed, operate more slowly. In this way, forming associations on top of relations can provide for a double response; a fast one based on the associations and a slower one based on the relations. This combination can help in, say, hazardous circumstances, where speed is essential. It could also reduce the processing time in specific cases. But, on the reverse side, it could also lead one astray from the correct analysis.

Van der Velde and de Kamps [2006] present a neural architecture in which (temporal or more permanent) connections between conceptual representations based on neural assemblies could be formed. Figure 6 illustrates the relation dog likes black cats, where the neural assemblies representing the concepts “dog”, “likes”, “black” and “cats” are interconnected in a neural blackboard. The blackboard consists of neural populations.
that represent the specific types of concepts and the relations between them. Thus, \( N_1 \) and \( N_2 \) represent a noun, \( V \) a verb, \( Adj \) an adjective and \( S \) a clause. The connections in the blackboard are gated connections (consisting of neural circuits). Gated connections provide control of activation, which allows the representation of relations and hierarchical structures, as found in higher-level human cognition. In this way, a temporal connection can be formed between the conceptual assemblies and populations of the same type in the blackboard (“dog” with \( N_1 \), “cats” with \( N_2 \), “like” with \( V \), “black” with \( Adj \)) and between the type populations themselves, which results in the representation for the clause (relation) _dog likes black cats_.

4.2. Artificial Neural Network (ANN)

ANNs draw inspiration from biological neurons, though their usage is usually not to model human brains [Sun 2008]. The building blocks of ANNs are simple computational units that are highly interconnected. The connections between the units determine the function of a network. In ANNs, concepts are implicitly represented by four parts: the network-wide algorithm, the network topology, the computational procedures in each individual unit, and the weights of their connections. Executing such algorithms produces explicit representations of concepts in the form of activation patterns, although individual nodes, e.g., in the output layer of an ANN classifier, can also be seen as outwards facing representations of concepts (e.g., the activation of an output node is regarded as a representation of the corresponding class/concept). It is, however, important to highlight that most individual nodes of neural networks carry no recognizable semantic value.

ANNs can be viewed as directed weighted graphs in which artificial neurons are nodes and edges are connections between neuron outputs and neuron inputs. Based on the connection patterns (architecture), ANNs can be grouped into two categories: feed-forward neural networks, in which graphs have no loops; and recurrent (or feedback) neural networks [Medsker and Jain 1999], in which loops occur due to feedback connections. The most common family of feed-forward neural networks is multilayer perceptron [Haykin 1994]. Some of the well-known recurrent neural networks are Elman Network [Cruse 1996], Hopfield Network [Gurney 1997], and Boltzmann Machine [Hinton and Salakhutdinov 2006].

After the learning phase, standard feed-forward networks usually produce only one set of output values, rather than a sequence of values, for a given input. They are also memory-less in the sense that their responses to inputs are independent of the previous network states. Feed-forward neural networks are usually used as classifiers, by learning nonlinear relationships between inputs and outputs. Typically, the nodes in the output layer of a feed-forward ANN correspond to regions in the input feature space, and thus can be seen as representing centroids, or prototypes, of such conceptual regions. Recurrent, or feedback, neural networks, on the other hand, are dynamic systems. Because of the feedback paths, the inputs to neurons are then modified, which leads the network to enter a new state. Regarding the representation of temporal concepts, recurrent neural networks can be trained to learn and predict each successive symbol of any sequence in a particular language.

ANNs learn by iteratively updating the connection weights in a network, toward better performance on a certain specific task. There are three main learning paradigms: supervised, unsupervised and hybrid. Since ANNs can learn patterns of neuron activation (both simultaneous and sequential activations), they can be used to simulate creative processes, e.g., via combining two or more patterns into a single one, or creating a random variation of a learned pattern.

Various types of ANNs have been used, in the music domain, for melody generation [Todd 1992; Eck and Schmidhuber 2002; Hoover et al. 2012], and improvisation.
4.3. Deep Neural Network

Deep networks are a recent extension of the family of connectionist representations, which attempt to model high-level abstractions of data using deep architectures [Bengio et al. 2013]. Deep architectures are composed of multiple levels of nonlinear operations, such as neural nets with many hidden layers. This usually results in a stack of “local” networks whose types need not be the same across the entire deep representation.

The human brain is organized in a deep architecture [Serre et al. 2007]. An input percept is represented at multiple levels of abstraction, each level corresponding to a different area of the cortex. The brain also appears to process information through multiple stages of transformation. This is particularly evident in the primate visual system, with its sequence of processing stages, from detecting edges, primitive shapes, to gradually more complex shapes.

Deep representations are built with deep learning techniques [Bengio et al. 2013; LeCun et al. 2015]. Deep learning algorithms typically operate on networks with fixed topology and solely adjust the weights of the connections. Each type of deep architectures is amenable to specific learning algorithms: for example, deep convolutional networks are usually trained with backpropagation (see e.g., Kalchbrenner et al. [2014]), while deep belief networks [Hinton et al. 2006; Hinton 2009] are obtained by stacking several Restricted Boltzmann Machines [Hinton and Salakhutdinov 2006], each of which is typically trained with the Contrastive Divergence algorithm. Deep belief networks are based upon probabilistic approaches, whereas other approaches exist, such as auto-encoders which are based upon reconstruction-based algorithms, and manifold learning which has roots in geometrical approaches. Although the stacked layers may allow the network to effectively learn the intricacies of the input, the fact that they usually have a fixed topology imposes representation and learning limits a priori. In contrast, a deep learning algorithm for dynamic topologies, allowing the creation of new nodes or layers of nodes, would enable the creation of new concepts and new dimensions in a conceptual space.

Deep representations and learning have been used in modeling and generating language (see Section 3.2), producing jazz melodies [Bickerman et al. 2010], creating spaceships for 2D arcade-style computer games [Liapis et al. 2013], generating images [Goodfellow et al. 2014; Gregor et al. 2015], transferring visual styles [Gatys et al. 2015], and as part of a computational framework of imagination [Heath et al. 2015].
quence of pitches by performing a set of operations on sub-sequences. An example is the GP-Music system [Johanson and Poli 1998].

“The item returned by the program tree in the GP-Music System is not a simple value but a note sequence. Each node in the tree propagates up a musical note string, which is then modified by the next higher node. In this way a complete sequence of notes is built up, and the final string is returned by the root node. Note also that there is no input to the program; the tree itself specifies a complete musical sequence.”

It is straightforward to conceive a naive descriptive representation, since we can always resort to the enumeration of the elements of the artifact. Therefore one should ponder about the motivation for using procedural representations. A program can take advantage of the structure of an artifact—e.g., repetition of note sequences, relations between sequences, cycles, etc.—and as such the size of the procedural representation of an artifact that has structure can be significantly smaller than the size of its descriptive representation. Additionally, it is also easier to induce structural changes, in the case of creating new concepts.

Procedural representations are particularly popular for image creation in Evolutionary Computation (EC). Many of them are expression-based, such as the example in Figure 7 showing both a symbolic expression and the corresponding image produced by this expression. Some notable examples of expression-based evolutionary computation are by Sims [1991], Rooke [1996], Unemi [1999], Saunders and Gero [2001], Machado and Cardoso [2002] and Hart [2007].

Machado et al. [2010] evolved non-deterministic context free grammars. The grammars are represented by means of a hierarchic graph, which is manipulated by graph-based crossover and mutation operators. The context free grammar constitutes a set of program instructions that are executed in order to generate the visual artifacts; so while the grammar has a symbolic representation, the representation of the image is procedural. One of the novel aspects of this approach is that each grammar has the potential to represent, and generate, a family of akin shapes (Figure 8).

Zhu and Mumford [2007] used stochastic context sensitive grammars embedded in an And-Or graph to represent large scale visual knowledge, using raster images as input, for modeling and learning. In their preliminary works, they show that the grammars enable them to parse images and construct descriptive models of images. This allows the production of alternative artifacts and the learning of new models.

Byrne et al. [2012] evolved architectural models using grammatical evolution. Grammatical evolution is a grammar based form of Genetic Programming (GP), replacing the parse-tree based structure of GP with a linear genome. It generates programs by evolving an integer string to select rules from a user-defined grammar. The rule selections build a derivation tree that represents a program. Any mutation or crossover operators are applied to the linear genome instead of the tree itself. McDermott [2013] also used grammatical evolution to evolve graph grammars in the context of evolutionary 3D design. Greenfield [2012] used GP to evolve controllers for drawing robots. The author resorted to an assembly language where each statement is represented as a triple. The programs assume the form of a tree.

Music, or more specifically composition as a creative process, has been another common application for procedural representations. One of the first, if not the first, evolutionary approaches to music composition resorts to a procedural representation. Horner and Goldberg [1991] used Genetic Algorithm (GA) for evolving sequences of operations that transform an initial note-sequence into a final desired sequence within a certain number of steps. Putnam [1994] was one of the first to use GP for music generation purposes. He used the traditional GP tree-structures to interactively evolve
sounds. Spector and Alpern [1994] used GP to evolve programs that transform an input melody by applying several transformation functions (e.g., invert, augment and transpose). The work of Johanson and Poli [1998] constitutes another early application of GP in the music domain. Hornel and Ragg [1996] evolved the weights of neural networks that perform harmonization of melodies in different musical styles. McCormack [1996] explored stochastic methods for music composition and proposed evolving the transition probability matrix for Markov models. Monmarché et al. [2007] used artificial ants to build a melody according to transition probabilities while also taking advantage of the collective behavior of the ants marking paths with pheromones. They evolved graph-like structures, the vertices are MIDI events and a melody corresponds to a path through several vertices. McCormack [1996] focused on grammar-based approaches for music composition, exploring the use of L-systems. In an earlier work, McCormack [1993] used L-systems for evolving 3D shapes.

6. APPLICATION-SPECIFIC REPRESENTATIONS

In the above sections, we introduced conceptual representations at each of the symbolic, spatial and connectionist levels, as well as both descriptive and procedural representations. The representations reviewed so far are domain-generic. In this section, we review conceptual representations that have been used in three popular applications domains of computational creativity research: language, music and images. In addition we review representations of emotion, an important factor for artifacts in all these domains, and therefore interesting for computational creativity. The information in these four domains may have representations at all the three levels and both descriptive and procedural representations.

6.1. Language

In the language domain, one of the atomic conceptual representations is word. A single word is ambiguous. Word cluster and fuzzy synset are representations aimed at expressing meanings more precisely. Word association is association (see Section 2.1) between words, which is the building block of a more complex representation, word
association graph. The above are conceptual representations used in a broad range of
language-related tasks. In addition, we review two conceptual representations used in
narrative generation, plan operator and narratological category.

6.1.1. Word, Word Cluster and Fuzzy Synset. Natural language is typically used to re-
fer to a concept, which can be denoted by a word. Traditionally, words are collected
in dictionaries. Broad-coverage lexical knowledge bases (LKBs) are computational re-
sources. WordNet [Fellbaum 1998; Miller et al. 1990] is a manually constructed LKB,
designed with the inspiration of current psycholinguistic theories of human lexical
memory. The most ambitious feature of WordNet is its attempt to organize lexical in-
formation in terms of word meanings, rather than word forms. English nouns, verbs,
adjectives and adverbs are organized into synonym sets (synsets), each representing
one underlying lexical concept (word sense). Another prominent feature of WordNet
is that synsets are linked by conceptual-semantic and lexical relations, such as syn-
onymy, antonymy, hypernymy, hyponymy, holonymy, meronymy, attribute, cause and
domain. LKBs have been very popular in text-based creative systems, such as poetry
generation [Gonçalo Oliveira 2012; Agirrezabal et al. 2013], narrative [Gervás 2009; Riedl and Young 2010], computational humor [Ritchie 2001], etc.

As natural language is ambiguous, in opposition to formal languages, one word is
often not sufficient for referring to a specific concept. Whether with explicit or implicit
relations, a group of words is a common alternative for describing a concept. A no-
table resource of word clusters is Roget’s Thesaurus [Roget 1992], where semantically
related words and phrases are organized in groups led by head words. Most of the
computational work on harvesting word clusters relies on Harris’ [1968] distributional
hypothesis, which assumes that similar words tend to occur in similar contexts. After
defining the contexts of words, these works generally follow a procedure of clustering
words according to their distributional similarity (see Section 3.2).

A special case of word cluster is synset. New synonyms can be discovered from raw
text using semantic relatedness measures (see Section 6.1.2). Apart from raw text,
synonyms can be extracted from dictionaries [Gonçalo Oliveira and Gomes 2013], es-
pecially from definitions having only one word or using the “same as” pattern.

Furthermore, from a linguistic point of view, word senses are not discrete and cannot
be separated with clear boundaries [Kilgarriff 1996; Hirst 2004]. This has some paral-
lelism with the psychological notion of categorical perception (see Section 1.2). Sense
division in dictionaries and ontologies is thus often artificial. This also applies to con-
cepts, and representing them as crisp objects does not reflect the way humans organize
knowledge. A more realistic approach would be adopting models of uncertainty, such as
fuzzy logic. Velldal [2005] represents word sense classes as fuzzy clusters, where each
word has an associated membership degree. Other works represent concepts as fuzzy
synsets. The fuzzy membership of a word in a synset can be interpreted as the confi-
dence level of using this word to indicate the meaning of the synset. In the Swedish
WordNet [Borin and Forsberg 2010], words have fuzzy memberships to synsets, based
on the opinion of users on the degree of synonymy of synonym pairs. Fuzzy synsets
were automatically discovered in synonym networks extracted from Portuguese dictio-
naries [Gonçalo Oliveira and Gomes 2011] and from the redundancy across different
lexical-semantic resources [Gonçalo Oliveira and Santos 2016].

6.1.2. Word Association, Word Association Graph. Word association is a pair of words that
are related in some way. Word associations have been collected in psychological ex-
periments, where a word (stimulus) is presented to a person who is asked to say or
write down the word that first comes to his mind. For instance, some of the most fre-
quent responses to the stimulus word “apple” are: “pie”, “pear”, “orange”, “tree” and
“core”. The responses are in various relations with the stimulus, such as synonymy,
antonymy, meronymy, hierarchic relations (category-exemplar, exemplar-category and
category coordinates), idiomatic and functional relations [Cruse 1986]. There exist two
large collections of word associations, the Edinburgh Associative Thesaurus (EAT)\textsuperscript{12}
and the University of South Florida Free Association Norms\textsuperscript{13}.

The computational work on harvesting word associations is closely related to cal-
culating \textit{semantic relatedness/similarity}. In general, semantic relatedness/similarity
is measured by using distance measures in certain materialized conceptual space,
mainly knowledge bases (KBs) and raw text. KBs include dictionary, thesaurus and
ontologies, which are represented as graphs or networks. Hence, the semantic related-
ness measures using KBs are path related calculations. With raw text, there are mea-
sures based on the frequency of word co-occurrence, such as log-likelihood ratio [Dun-
nings 1993], Pointwise Mutual Information and Information Retrieval (PMI-IR) [Tur-
ney 2001], Normalized Google Distance (NGD) [Cilibrasi and Vitanyi 2007], and VSMs
(see Section 3.2).

Words can be connected according to their associations, which becomes a graph. In
this graph, a word, or the concept it denotes, is defined by the connections it has with
other words, e.g., “car” is defined by its associations to “drive”, “road”, “vehicle”, “traf-
fic”, “personal”, etc.

Word associations and word association graphs have broad usage in NLP tasks, such
as word sense disambiguation [Navigli 2009]. In the computational creativity field,
Toivanen et al. [2014] used document specific word associations in poetry generation.
PMI scores computed on Wikipedia are used to evaluate the semantic proximity of
automatically generated song lyrics to their seeds [Gonçalo Oliveira 2015]. Xiao et al.
[2016] used word associations in building a metaphor interpreter. Word association
graphs were used to solve Remote Associate Test [Gross et al. 2012]. Nevertheless, not
much work has been done for creating novel concepts by exploiting the structures of
word association graphs, which is a promising direction for future research.

6.1.3. \textit{Plan Operator}. A different type of concept (related to the implementation of nar-
rative systems as well as the much broader field of planning [LaValle 2004]) are actions
as operators that change the world. In the field of planning, such actions are defined
as \textit{plan operators}. Actions in a story are applicable if certain conditions hold in the
state of the world before they happen, and after they happen they change the state of
the world. This idea has been represented by defining actions with an associated set of
preconditions and another set of postconditions or effects. Table II shows examples of
story actions linked by preconditions.

Specific planners [Fikes and Nilsson 1971; Pednault 1987] may represent planning
operators in different ways. Attempts have been made to standardize AI planning lan-
guages, with the Planning Domain Definition Language (PDDL) [McDermott et al.
1998] being a significant reference in this effort. The problem of constructing plan
operators for specific applications is an open research question, with ongoing efforts
considering constructing them as a composition of general components [Clark et al.
1996]. Concept creation technologies could be applied in this case with considerable
advantage.

6.1.4. \textit{Narratological Category}. A more elaborate concept associated with language is
that of a \textit{narratological category}. These arise within the expanding area of research
called \textit{computational narratology}, which involves computational representation and
treatment of the fundamental ingredients of narratives. This type of representation
would be an important stepping-stone towards achieving automatic processing of nar-

\textsuperscript{12}http://www.eat.rl.ac.uk.
\textsuperscript{13}http://w3.usf.edu/FreeAssociation/.
rative, and it is an integral part of ongoing efforts for the automatic generation of narrative [Kybartas and Bidarra 2017]. Existing efforts to represent fundamental ingredients of narratives have been based on analyses of narrative ingredients by literary scholars, and they have led to various proposals of their computational representations. Of the many theories of narrative developed in the Humanities, only a few have bridged the gap to become tools in the hands of AI researchers. Propp's [1968] Morphology of the Folktales is one of these, having been applied in several AI systems for story generation. The two cornerstones of Propp's analysis of Russian folktales are a set of character roles (which the author refers to as *dramatis personae*) and a set of character functions (acts of characters). These have been used in several systems [Turner 1993; Grasbon and Braun 2001; Fairclough and Cunningham 2004; Gervás et al. 2005; Wama and Nakatsu 2008; Imabuchi and Ogata 2012], represented in slightly different ways. Of all these, the most explicit conceptual representation of Propp's set of narratological categories is the description logic formulation developed by Peinado [2008].

Another popular narrative theory in the computing community is the *three-act restorative structure*, though at a different level of detail from Propp's. This model, derived from Joseph Campbell's [1968] analysis of the structure of myths, is a dominant formula for structuring narrative in commercial cinema [Vogler 1998].

Another source that is also being considered in AI is the work of Chatman [1978]. This model constitutes a step up from the models of Propp and Campbell in the sense that it considers a wider range of media, from literature to film. For an AI researcher in search of a model, the greatest advantage of Chatman's approach is his effort to identify a common core of elementary artifacts involved in several approaches to narrative theory. Chatman studied the distinction between *story* and *discourse*, and proposed ways of decomposing each of these domains into elementary units. His idea of structuring story and discourse in terms of *nuclei* and attached *satellites* provides a way of organising internally the knowledge entities on which computational systems rely for representing stories.

Furthermore, there is ongoing research on automatically learning the equivalent of Propp's morphology from a set of annotated texts [Finlayson 2012]. This process involves the automatic creation of character functions. It is achieved by Analogical Story Merging (ASM), a novel machine learning technique which provides computational purchase on the problem of identifying a set of plot patterns from a given set of stories. Propp's manner of abstracting narrative structure from a set of stories is far from being the only possible one. Concept creation technologies should consider possible alternative abstractions which might be automatically generated from a corpus of sample stories.

### 6.2. Music

Honing [1993] notes that representations for music tend to be motivated either to address predominantly technical issues, or to capture more perceptual or cognitively-salient musical qualities. The former category emphasizes observable and measurable musical attributes, such as the velocity of key presses on a piano keyboard, the position of note symbols on a musical score, or the propagation of sound waves during a musical...
performance. The latter category seeks to capture the cognitive aspects of musical thought and behavior—the concepts that predominate when listening to, performing, composing, or otherwise engaging in music.

In an early discussion of the application of computer technology to music research, Babbitt [1965] employs the terms acoustic, graphemic and auditory to distinguish three related domains of musical information. The acoustic domain encompasses the physical manifestations of music as sound. Representations of auditory information include analogue recordings on electromagnetic tape, or streams of binary digits resulting from analogue-to-digital conversion. The representation of acoustic information most naturally falls within Honing's category of technical representations.

The graphemic domain pertains to the graphical notation of music, such as conventional musical scores and tablature. Graphical notations are themselves representations, serving primarily as musical aide-mémoires and means of communicating musical ideas. From the computational perspective, there is scope for both technical and cognitive representational approaches. For example, where the aim is simply to represent the exact layout of notation symbols on a score, a purely technical representation is adequate. However, if the aim is to also represent associated music-theoretical meaning, or possible performance interpretations, then the representation language must necessarily express, at least in part, the musical knowledge assumed by each notation system. Such information could also be described declaratively or procedurally. For example, a representation could describe a trill declaratively as an object of ornamentation, or alternatively, as a procedure representing how a trill is produced [Honing 1993].

The auditory domain covers information about music as perceived by the listener, aligning with Honing’s category of conceptual and mental representations. The characterization of musical information into the domains of the acoustic, graphemic, and auditory is not exhaustive; for example, gestures made by performers would be another potentially relevant domain of information [Selfridge-Field 1997]. However, the distinctions are nonetheless important categories of musical information. The phenomenon of music itself cannot be said to exist in any one domain exclusively, but instead can be understood as something that exists between the domains, with each one offering a particular perspective from which to study music [Wiggins 2008], making music a rich and challenging area of application within the field of knowledge representation.

A simple, yet powerful approach to a general representation of music is proposed by Wiggins et al. [1989], Harris et al. [1991], and Smaill et al. [1993]. The Common Hierarchical Abstract Representation for Music (Charm) aims to support a high degree of generality, enabling interoperability and extensibility. Charm is defined initially as a representation of music at the symbolic level, in which readily identifiable musical objects are represented by discrete symbols. As such, Charm is particularly appropriate for representing a wide range of graphemic information, but can also be extended to represent lower-level acoustic or other continuously-valued musical information. Furthermore, symbolic representations are particularly appropriate for the high-level description of a range of perceptual attributes and concepts, such as discrete musical events, groupings of events, and for expressing the formal properties of relationships between such structures.

Charm is based on the computer science concept of abstract data typing. Despite the direct incompatibility of many music representation schemes, a considerable degree of commonality exists at an abstract level. For example, most schemes define some way of representing pitch, whether in terms of MIDI note numbers, scale degree, microtonal divisions of the octave, or frequency. However, at an abstract level, common patterns of operations can be observed, which are irrespective of the underlying implementation.
Therefore, the authors proposed an abstract representation, in which musically meaningful operations can be defined in terms of abstract data types. Harris et al. [1991] defined basic data types for pitch (and pitch interval), time (and duration), amplitude (and relative amplitude) and timbre. Therefore, the abstract event representation is the Cartesian product:

\[ \text{Pitch} \times \text{Time} \times \text{Duration} \times \text{Amplitude} \times \text{Timbre} \]

In the case of time, the following functions can be defined where the arguments \( \{t, d\} \) denote Time or Duration data types respectively.

\[
\begin{align*}
add_{td} &: \text{Duration} \times \text{Duration} \rightarrow \text{Duration} \\
add_{tt} &: \text{Time} \times \text{Duration} \rightarrow \text{Time} \\
sub_{tt} &: \text{Time} \times \text{Time} \rightarrow \text{Duration} \\
sub_{dd} &: \text{Duration} \times \text{Duration} \rightarrow \text{Duration}
\end{align*}
\]

Typed equivalents of arithmetic relational operators (e.g., \( \leq, \geq, =, \neq \)) are also defined, permitting ordering and equality relations to be determined. With the exception of timbre, the internal structure of each basic data type is the same, allowing comparable functions to be defined modulo renaming [Harris et al. 1991].

The abstract data type approach to representing music extends beyond the representation of surface level events. Charm formally defines the concept of the constituent, which allows arbitrary hierarchical structures to be specified [Harris et al. 1991]. At the abstract level, a constituent is defined as the tuple:

\[ \langle \text{Properties/Definition, Particles} \rangle \]

Particles is a set whose elements, called particles, are either events or other constituents. No constituent can be a particle of itself, defining a structure of constituents as a directed acyclic graph. Properties/Definition is the "logical specification of the relationship between the particles of this constituent in terms of the membership of some class" [Harris et al. 1991]. The distinction between Properties and Definition is made explicit in a concrete implementation. However, at the abstract level, they both logically describe the structure of the constituent. Properties refer to "propositions which are derivably true of a constituent" [Harris et al. 1991]; for example, that no particle starts between the beginning and end of any other particle, defined as a stream:

\[
\text{stream} \equiv \forall p_1 \in \text{particles}, \neg \exists p_2 \in \text{particles}, \ p_1 \neq p_2 \wedge \text{GetTime}(p_1) \leq \text{GetTime}(p_2) \wedge \text{GetTime}(p_2) < \text{add}_{td}(\text{GetTime}(p_1), \text{GetDuration}(p_1))
\]

where \( \text{GetTime} \) and \( \text{GetDuration} \) are selector functions returning the timepoint and duration respectively of a given particle. Definitions are propositions that are true by definition; for example, that a set of particles contains all the events notated in a score of a particular piece of music.

An implementation of a Charm-compliant representation requires some additional properties, both for computational efficiency and user convenience. The following is an example of a simple ‘motif’ constituent [Smaill et al. 1993]:

\[
\text{constituent}(c_0, \text{stream}(0, t_1), \text{motif}, [e_1, e_2, e_3, e_4])
\]

Every event and constituent defined within the system must be associated with a unique identifier, shown as \( c_0, e_1, e_2 \) and so forth in the above example. The constituent is a stream, with a start time and a duration, denoted by the property \( \text{stream}(0, t_1) \), which is derivably true from the events it contains. In contrast, the
A constituent is defined as a motif, and the user is free to provide such definitions for their own purposes. A wider benefit of adopting an abstract data type approach to music representation is that it provides the basis for developing a common platform for sharing data and software tools. Smaill et al. [1993] demonstrate that both the implementation language and concrete data representation are immaterial providing that the correct behavior of the abstract data types is observed. From a formal perspective, many issues of representation discussed in the field can be seen as concerning merely arbitrary matters of encoding or data serialization. Although encoding schemes may well be designed to meet particular needs, such as to facilitate efficient human data entry or to be space efficient, ambiguity can arise when implicit ontological commitments are left unstated, ultimately limiting potential usefulness.

6.3. Image

Images are commonly represented on computers by matrices of pixels. For instance, one byte can be used to encode the color value for each pixel, or three bytes per pixel in RGB images.

den Heijer and Eiben [2011] used Genetic Algorithms (GAs) to evolve Scalable Vector Graphics (SVGs), manipulating directly SVG files through a set of specifically designed mutation and recombination operators. In a more recent approach, den Heijer [2013] manipulates directly BMP, GIF, PNG and JPG files to produce glitch art effects.

We consider parametric representations as a particular type of descriptive representation where the representation encodes a set of parameters that determine the behavior of a generative system. The work of Draves [2007] is a prototypical example of such an approach, where the genotype encodes the parameter set of fractal flames, i.e. a set of up to several hundred floating-point numbers. In the words of Draves:

“The language is intended to be abstract, expressive, and robust. Abstract means that the codes are small relative to the images. Expressive means that a variety of images can be drawn. And robust means that useful codes are easy to find.”

Other recent examples include the work of Reed [2013] who evolved a Bézier curve that is then rotated around an axis to create a vase. The representation consists of five coordinates and three integers, which determine the angles within the curve. Machado and Amaro [2013] used a string of floating-point numbers to encode and evolve a set of parameters that specify the sensory organs and behavior of artificial ants, who are used to create non-photorealistic renderings of input images.

6.4. Emotion

In recent times, emotion has become a focus of interest in computational applications. Among various conceptual representations of emotions [Cowie and Cornelius 2003], the most popular two are emotional categories and emotional dimensions. Connectionist representations of emotions also exist, such as [Norton et al. 2010]. We describe the two most common representations below.

**Emotional Categories:** Natural languages provide assorted words with varying degrees of expressiveness for describing emotional states. Several approaches have been proposed to reduce the number of words used to identify emotions, such as basic emotions, super-ordinate emotional categories, and essential everyday emotion terms. Basic emotions refer to those that are more well-known and understandable to everybody than others [Cowie and Cornelius 2003]. In the super-ordinate emotional categories, some emotional categories are proposed as more fundamental, with the argument that they subsume the others [Scherer 1984]. Finally, the essential everyday emotion terms
focus on emotional words that play an important role in everyday life [Cowie et al. 1999].

Emotional Dimensions: These spatial representations model the essential aspects of emotions numerically. They deal with artificially imposed scales of identified characteristics of emotions. Although there are different dimensional models with different dimensions and numerical scales [Fontaine et al. 2007], most of them agree on three basic dimensions called evaluation, activation and power [Osgood et al. 1957].

— Evaluation represents how positive or negative an emotion is. At one extreme, we have emotions such as happiness, satisfaction and hope, while at the other we find emotions such as unhappiness, dissatisfaction and despair.

— Activation represents an activity versus passivity scale of emotions, with emotions such as excitation at one extreme, and emotions such as calmness and relaxation at the other end.

— Power represents the sense of control the emotion exerts on the subject. At one end of the scale we have emotions that are completely controlled, such as fear and submission, and at the other end we find emotions such as dominance and contempt.

Emotional dimensions describe a continuous space as opposed to the discrete space of emotional categories.

There exist collections of evaluation ratings estimated by computational methods, called affect lexicons, such as General Inquirer14, WordNet-AFFECT [Strapparava and Valitutti 2004], SentiWordNet [Esuli and Sebastiani 2006; Baccianella et al. 2010], Macquarie Semantic Orientation Lexicon (MSOL) [Mohammad et al. 2009], and SenticNet15.

7. CONCLUSION

We have structured this review according to three levels of representation (symbolic, spatial, connectionist), inspired by Gärdenfors [2000], and separately considered additional procedural representations and domain-specific representations in four popular application areas of computational creativity (language, music, image and emotion). We hope that this organization will act as a map, helping researchers navigate in the forest of conceptual representations used in computational concept creation.

In the Introduction, we also gave a taxonomy of concept creation approaches, where the main classes of methods are concept extraction, concept induction, concept recycling, and concept space exploration. We next summarize the results of this review by considering relations between the different levels of conceptual representations, the application domains, and the types of concept creation methods.

Consider the application domains first. Above in Section 6 we considered representations in four major application domains of concept creation—language, music, image and emotion. In addition to the domain-specific concept representations, the generic representations of Sections 2–5 can also be used in these domains. Domain-generic representations are especially useful in applications that process information from multiple domains. It actually turns out that in all the domains that we have considered, conceptual representations have been used from all the levels and categories used to structure this review (symbolic, spatial, connectionist; declarative, procedural).

Take for instance text documents. At the symbolic level, a document as a sequence of words is ready for people to read. At the spatial level, documents are routinely represented as vectors, allowing e.g., comparison of semantic similarity between documents.

14http://www.wjh.harvard.edu/~inquirer/.
15http://sentic.net.
Table III. Concept creation methods applied at different levels and types of conceptual representations

<table>
<thead>
<tr>
<th>Level/type of representation</th>
<th>Extraction</th>
<th>Induction</th>
<th>Recycling</th>
<th>Exploration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbolic level</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Spatial level</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Connectionist level</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Descriptive</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Procedural</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

At the connectionist level, text can be generated by activating an artificial neural network which captures characteristics of a collection of documents. The connectionist representation is a procedural one, while the spatial and symbolic representations used in the example are declarative ones.

In Table III we summarize, based on the above review, how concept creation methods relate to the different levels and types of conceptual representations. Two interesting observations can be made. First, concept extraction has mainly been applied at one of the three levels only, the symbolic level, between different symbolic representations. This is for an obvious reason: symbolic representations are often designed to be manipulated and translated, and at the very least, by definition, have meanings that can be processed as symbols. Spatial and especially connectionist representations lend them much less for such translation, if at all. According to Gärdenfors, the three levels are connected, so that the connectionist level feeds spatial representations, which in turn become symbolic in language. In the same way, it seems plausible that concept extraction methods operating between levels could be developed. For instance, McGregor et al. [2015] take steps towards establishing such a mapping between spatial and symbolic levels. The second observation is that concept induction, concept recycling and concept space exploration have all been used for almost all the levels and types of conceptual representations, with the exception that we are not aware of applications of concept recycling to spatial representations. This seems a promising area for research in concept creation: spatial representations lend themselves for mutation and combination, the question is more in how to utilize this capability in concept creation in a useful way.

This review demonstrates that conceptual representations at each of the symbolic, spatial and connectionist levels, as well as both descriptive and procedural representations, are abundant. Still, promising new representations are emerging at all levels, such as bisociation (Section 2.1), heterogeneous information network (Section 2.5), conceptual spaces (Section 3.1), neural blackboard (Section 4.1), and deep neural network (Section 4.3).

Furthermore, numerous avenues exist for research into computational concept creation and conceptual representations, we here mention some of them. For instance, in the field of concept extraction, an interesting possibility for future work is automatically building plan operators. Concerning concept induction, a particularly interesting line of future work is learning new narratological categories. In terms of concept recycling, the combination of thematically different ontologies can be a new approach for dealing with analogies, metaphors, pataphors and conceptual blending. Regarding concept space exploration, an interesting opportunity is discovering novel concepts in word association graphs by exploiting graph structure measures. Finally, with respect to transformational creativity, finding new interpretations of familiar concepts in a conceptual space model would offer ways to be creative beyond the original limits.
REFERENCES


Tarek R. Besold and Enric Plaza. 2015. Generalize and Blend: Concept Blending Based on Generalization, Analogy, and Amalgams. In *Proceedings of the Sixth International Conference on Computational Creativity (ICCC ’15)*. Park City, Utah.


ACM Computing Surveys, Vol. V, No. N, Article A, Publication date: January YYYY.


Peter Bruza and Marc Weeber. 2008. Literature-Based Discovery. Springer Berlin Heidelberg.


Conceptual Representations for Computational Concept Creation


Oliver Kutz, Till Mossakowski, Joana Hois, Mehul Bhatt, and John Bateman. 2012. Ontological Blending in DOL. In *Proceedings of the First International ECAI Workshop on Computational Creativity, Concept Invention, and General Intelligence* (C3GI’12). Montpellier, France.


Dekang Lin and Patrick Pantel. 2001. DIRT @SBT@Discovery of Inference Rules from Text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’01)*. San Francisco, CA, 323–328.


Conceptual Representations for Computational Concept Creation


A:42 P. Xiao et al.


ACM Computing Surveys, Vol. V, No. N, Article A, Publication date: January YYYY.


Received ; revised ; accepted