Dynamic Symbol Systems:
An Introduction to the Local Model

Herbert Jaeger
Faculty of Technology, University of Bielefeld
POB 100131, D-33501 Bielefeld, Germany
email: herbert@techfak.uni-bielefeld.de

Abstract

Dynamic symbol systems (DSS) are a class of dynamic systems especially developed for modeling situated agents. They combine a symbolic format of representation with a self-organizing dynamics. The theory can be used for theoretical purposes, as a discrete, approximative reconstruction of continuous dynamic systems. It can also be used for the practical design of multi-granular information processing systems, since it is formulated in algorithmic terms. Furthermore, as processing on a conceptual level is concerned, the dynamic system perspective suggests answers to some questions that are hard for logic-oriented approaches. All in all, DSS contributes to bridging the gap between dynamics-oriented, bottom-up techniques, and representation-oriented, top-down perspectives in AI. The paper gives an informal introduction to the part of DSS theory concerned with local phenomena, while the global part of the theory is very briefly sketched.

October 1994 draft version submitted to AISB-95

1 Introduction

Dynamic system techniques are traditionally used in connectionism (e.g., Smolensky 1986, Yao & Freeman 1990). More recently, they have started to prosper in psychology (Smith & Thelen 1993), cognitive science (Clancey 1993, van Gelder & Port 1994), behavior-oriented robotics (Smithers 1992, Steels 1993), and in agent-oriented AI (Kiss 1993).

Dynamic system models offer universal mechanisms for explaining several classes of phenomena: continuous shift vs. sudden qualitative reorganisation of systems, reduction of degrees of freedom in control problems, quasi-periodic activity, emergence of macro-level patterns on a continuous or fine-grained, dynamic microlevel, and others.

However, currently used, calculus-based dynamic system techniques have some drawbacks. Empirical state space analysis requires a large amount of high-quality numerical data (e.g., Robertson et al. 1993). The mathematical distinction between "slow" control parameters and "fast" state variables bars the way to a comprehensive account of dynamically changing time scales, which is a crucial and common phenomenon in agents of sufficient complexity (Jaeger 1994b). Standard dynamic system models do not directly yield executable computer programs (a step in this direction is the PDL computer language, cf. Steels 1994).

These difficulties root in the particular mathematical nature of the dynamic system models currently in use, i.e., continuous systems which are ultimately specified by differential equations. In order to circumvent them, I suggest to consider simpler, approximative dynamic system models as a supplement to the standard techniques.

I have defined and investigated *dynamic symbol systems* (Jaeger 1994a), a class of discrete dynamic systems tailored to the modelling of multi-granular agent architectures. Dynamic symbol systems (DSS) retain the explanatory mechanisms listed above while shunning the problems of high-precision, continuous models. The price to be paid is the loss of numerical accuracy.

DSS theory is a mathematical formalism. It can be employed in several ways. First, it can be used as an approximative, qualitative, transparent model of the information processing achieved in continuous systems like, for instance, neural networks. Second, since the theory is formulated in algorithmic terms, DSS can be used to design information processing systems that are directly runnable. Third, DSS can be seen as a formalism for the representation and processing of conceptual knowledge. In fact, DSS theory has evolved from attempts to model conceptual reasoning as a self-organizing process (Jaeger 1991, 1992).

This paper provides a largely informal introduction to the basic theory of DSS, which is concerned with local phenomena. A rigorous presentation of the entire approach can be found in Jaeger (1994a). Section 2 is an epistemological prelude, which explains how the notions of a *symbol* and its *meaning* are meant in DSS. In the main section 3, the local theory of DSS is sketched. Section 4 provides a brief glance at the global theory. Section 5 briefly points out related work. Finally, section 6 justifies the entry of DSS into the game of AI.

2 Epistemological prelude: the nature of dynamic symbols

Dynamic symbol systems are discrete systems made from *dynamic symbols*. Dynamic symbols share many properties with classical symbols in the sense of the physical symbol system hypothesis (Vera & Simon 1993). However, their semantics is quite different from the standard, logic-oriented view. In this section, I describe what dynamic symbols *are* and what they *mean*.

Dynamic symbols *are* physically observable, repeatedly identifiable, dynamic regularities in a physical information processing system. They are physical events of some duration, like any other events researched in the natural sciences, e.g., the collision of two elementary particles. A dynamic symbol derives its identity from some empirical observation procedure that reliably indicates the presence of the regularity. A dynamic symbol is *not* a "platonic" entity of any sort. Typical examples are attractor states in neural networks, but transient regularities other than classical attractors make perfect dynamic symbols, too (Jaeger 1994b).

In order to think or talk about dynamic symbols, or in order to represent dynamic symbols in formalisms such as DSS theory, researchers use *formal symbols*. They denote dynamic symbols in the same way like, for instance, the formal symbol "m_7" denotes the earth in a formula describing the solar system.

The *meaning* of a dynamic symbol is defined in terms of the internal dynamics of the information processing system in which the dynamic symbol occurs. Roughly, the meaning of a dynamic symbol *s* consists in a "halo" of other dynamic symbols which are "accessible" from *s*. A precise definition will be given in the next subsection. The important thing to note here is that the DSS account of symbol meaning is not externally referential, but system-internal and "associative".

An example will help to clarify these issues. Consider a robot capable of grasping apples. Being built with connectionist techniques, and having a long learning history, the robot's information processing is opaque at first sight. The experimenter wants to construct a transparent, telltale DSS model of what is going on in the robot. He monitors the computational activity at some place in the "brain". He finds, among others, a particular activity pattern whose appearance is highly correlated with the robot's successfully grasping an apple. This activity pattern is a dynamic symbol. In order to introduce it into his DSS model, the experimenter writes down in his (mental or physical) notebook a formal symbol: `apple`. Fig. 1 depicts this scene.

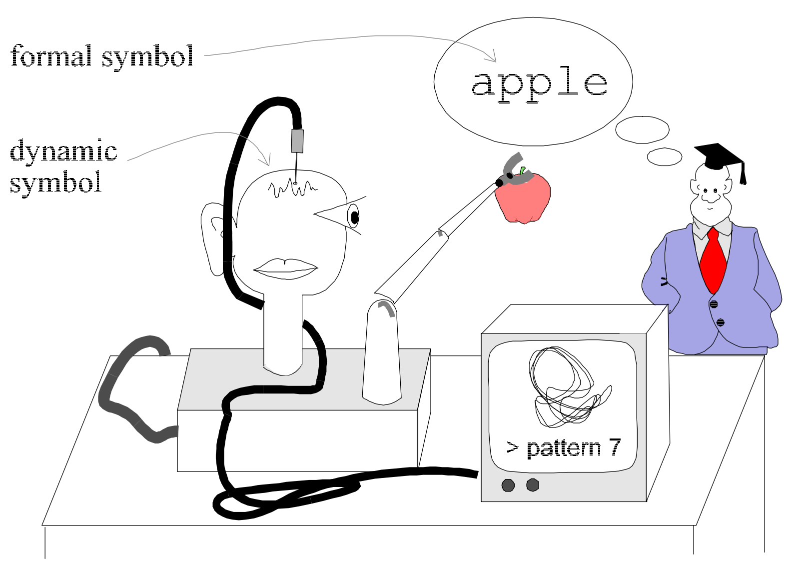

Fig. 1: The occurence of a dynamic symbol, and of a formal symbol denoting it.

In this scene, the researcher uses the formal symbol `apple` to denote an activity pattern. This is nothing but a mnemotechnic aid, which helps one to remind that this particular activity pattern is empirically correlated with the robot's grasping an apple. It is dangerous, however, to use such telling names. The dynamic symbol denoted by the formal symbol `apple` does *not* have a particular physical apple, or the class of all physical apples, for its external reference. To reiterate, an external meaning is not defined for a dynamic symbol at all.

Upon closer inspection one finds that the empirical correlation between the neural activity pattern and grasping events does not justify ascribing to the former an extension of physical apples. First, the empirical correlation is unlikely to be perfect. This destroys any "platonic" certainties. Second, why should the extension be "apples" at all? Other extensions would be equally good candidates, for instance, "grasp-thing-making-experimenter-smile". In fact, one must not assume that the robot possesses object concepts at all.

Thus, DSS takes the epistemological stance of radical constructivism (Maturana & Varela 1984). However, the relation between a dynamic symbol and the corresponding formal symbol *is* a classical denotation. But, this relation is not part of DSS theory proper; it would be part of a meta-theory in which some logic-oriented philosopher might wish to describe what DSS-oriented researchers do.

3 Local dynamics: coherent languages and generators

DSS describes information processing systems on both a local and a global level. This section sketches local DSS theory. Unless noted otherwise, precise definitions and proofs can be found in Jaeger (1994a).

DSS theory assumes that dynamic symbols can be observed "locally" within an agent. I.e., when the agent's internal processing dynamics is monitored at some particular "locus", a particular collection of dynamic symbols can be observed. At another locus, possibly other dynamic symbols occur. DSS does not specify the size or the physical nature of loci. In a neural architecture, for instance, a single unit, or some cell assembly, or even the entire network are potential loci for observations of dynamic symbols.

When an agent's internal processing dynamics is monitored locally for some time, DSS theory assumes that one can observe temporal *sequences* of dynamic symbols. Such sequences are called *associations*. Mathematically, the set of associations that can potentially be observed at a given locus is a formal language. Given some plausible assumptions (recurrence, finite number of distinguishable dynamic symbols, finite memory), the type of such languages can be specified. One arrives at the notion of *coherent* languages.

(A last aside. An association is a physical sequence of physical units, namely, dynamic symbols. A formal language, on the other hand, is a set of formal words, i.e. a set of sequences of formal symbols. Thus, one should actually distinguish between physical associations and formal associations. However, I will always talk about dynamic rather than formal symbols. In doing so, I act like the physicist who reads the formal representation "$E = mc^2$" as "energy equals ... " rather than as something like "the formal symbol for energy ...")

Coherent languages are a subclass of the regular languages. A convenient tool to describe them are *generators*. Fig. 2 presents an example. A generator is a finite, cyclic, directed graph, whose edges are labelled by dynamic symbols. Each finite path in a generator yields an association of the corresponding coherent language, in an obvious fashion.

More formally, let Σ be a finite alphabet (of dynamic symbols). A generator is a pair G = (S, *trans*), where S is the set of nodes (called *local states*), and *trans* $\subseteq S \times \Sigma \times S$ is the set of labelled edges (called *transitions*). G is required to be finite and cyclic. Then,

$$C = \{s_1 \ldots s_n \in \Sigma^n \mid n \geq 0, \text{ there exists a sequence } (x_0, s_1, x_1), (x_1, s_2, x_2), \ldots, (x_{n-1}, s_n, x_n) \text{ of transitions in } \mathit{trans}\}$$

is the language generated by G.

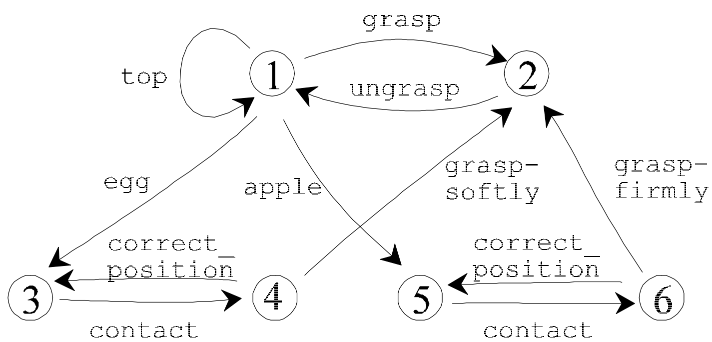

Fig. 2: A generator.

Among many others, the associations `egg contact grasp-smoothly` and `apple contact grasp-firmly` belong to the coherent language specified by the generator from fig. 2. In DSS terminology, `contact` appears in the *context* `egg` in the first association, and in the context `apple` in the second. In the context of `egg`, `contact` can be *continuated* by `grasp-smoothly` but not by `grasp-tightly`. The context `egg` *filters* out the continuation `grasp-tightly`. Context-induced filtering is a basic phenomenon in the local theory of DSS.

Contexts can be extended to the left. For instance, the context `egg` can be extended to `grasp ungrasp egg`. Interpreting associations temporally, this means taking into account a larger portion of the past. Generally, such extensions enhance the filtering effects of contexts: knowing more about the past yields more information about the future. For instance, after observing `contact`, one only knows that the possible continuations begin with `grasp-tightly` or `grasp-smoothly` or `correct_position`, but after observing the more extended context `egg contact`, one can exclude the continuation `grasp-tightly`.

However, certain contexts are "saturated" in that no further extension to the left can properly enhance their filtering effects. Such contexts are called *phase-fixing*. In fact, every context can be extended to the left to become phase-fixing. For instance, in fig. 2, the context `contact` is not phase-fixing, but its extension `egg contact` is. Two phase-fixing contexts ***p***, ***q*** are *equivalent* when they possess the same continuations. In fig. 2, the contexts `grasp-tightly` and `grasp-smoothly` are phase-fixing and equivalent. The equivalence classes of phase-fixing contexts are called the *phases* of a coherent language C. The term is motivated by viewing a generator as a stochastic oscillator. Phases, then, correspond to maximally informative (i.e., predictive) internal states of this oscillator.

Phases are a central notion in DSS. The phases of a coherent language C can be interpreted as the nodes of a particular generator for C, its unique *phase generator* $G_\varphi(C)$. $G_\varphi(C)$ can be effectively computed from any other generator G of C. The generator shown in fig. 2 is actually such a phase generator. Phase generators have quite pleasant formal properties.

Given a collection of associations belonging to a coherent language, the corresponding phase generator can be approximatively reconstructed with an inductive learning procedure (unpublished result). The procedure uses a "horizon" control parameter h, which specifies the maximal length of associations that are precisely reconstructed from the teaching set. If one selects $h = \max \{n_\varphi \mid \varphi \text{ is a phase}, n_\varphi = \min \{m \mid m = \text{length}(\boldsymbol{p}), \boldsymbol{p} \in \varphi\}\}$, and if the teaching material contains all the language's associations of length $\leq 2h$, the original phase generator is perfectly reconstructed.

As an additional benefit, this procedure yields transition probabilities for the labelled edges in the reconstructed phase generator. A generator equipped with transition probabilities can canonically be regarded as a description of a sequence-generating stochastic process. At first sight, it resembles a Markov chain. However, although every (ergodic) Markov chain can be captured by a probabilistic phase generator, the converse does not hold. In a Markov chain, the outcome of a time step is influenced by a universally bounded number of preceding steps. By contrast, in coherent languages, transitions steps can be properly influenced by events that lie arbitrarily far in the past.

For an example, assume that the phase generator in fig. 2 were augmented by probabilities. Then, after issuing a periodic subsequence (`contact correct_position`)^n of arbitrary length 2n, the probabilities of whether the next dynamic symbol is `grasp-firmly` or `grasp-softly` depend on whether this subsequence was preceded, 2n steps back in the past, by `egg` or by `apple`.

The canonical morphisms between generators are *simulations*. Intuitively, when a generator G_1 is simulated in some other generator G_2, the graph structure of G_1 is "spooled" into the graph structure of G_2, preserving edge labels. Formally, a simulation of $G_1 = (S_1, \mathit{trans}_1)$ in $G_2 = (S_2, \mathit{trans}_2)$ is a mapping $\sigma: S_1 \to \boldsymbol{P}(S_2)$ from S_1 to the power set of S_2, which satisfies

(i) $\sigma(x)$ is nonempty for all $x \in S_1$,
(ii) for all $(x_1, r, x_1') \in \mathit{trans}_1$, for all $x_2 \in \sigma(x_1)$ there exists $x_2' \in \sigma(x_1')$ such that $(x_2, r, x_2') \in \mathit{trans}_2$.

This is written $\sigma: G_1 \to G_2$. A fundamental *simulation theorem* states that C_1 is a coherent sublanguage of a coherent language C_2 iff there exists a simulation $\sigma: G_\varphi(C_1) \to G_\varphi(C_2)$ between the corresponding phase generators.

The notion of phases is crucial for an information-theoretic interpretation of associations. The *information* $H(\boldsymbol{s})$, which is gained by the observation of an association $\boldsymbol{s}$, is specified as the knowledge provided by $\boldsymbol{s}$ about the phase of the stochastic oscillator $G_\varphi(C)$:

$$H(\boldsymbol{s}) = -\log_2 |\Phi(\boldsymbol{s})| / |\Phi(C)|,$$

where $\Phi(C)$ is the set of all phases of C, and $\Phi(\boldsymbol{s})$ is the set of phases compatible with the observation of $\boldsymbol{s}$ (i.e., $\Phi(\boldsymbol{s})$ contains those phases which can be reached, in the phase generator $G_\varphi(C)$, on a derivation path for $\boldsymbol{s}$). In global DSS theory, $H(\boldsymbol{s})$ is important for the account of self-organization in self-organizing streams.

$\Phi(\boldsymbol{s})$ is called the *meaning* of $\boldsymbol{s}$. In the example from fig. 2, the set of local states {4, 6} is the meaning of the association `contact`. This definition of an association's meaning amounts to a specification of its possible future continuations; i.e., it is a dynamic, system-internal account of meaning. If $\boldsymbol{s} = s_1...s_n$, then $\Phi(\boldsymbol{s})$ can also be interpreted as the meaning of the single dynamic symbol s_n in the context $s_1...s_{n-1}$.

An important phenomenon in continuous dynamic systems is bifurcations. In a particular variety (pitchfork bifurcations), an attractor splits into two sibling attractors, due to a shift in some control parameter(s). In situated agents, this is a possible mechanism for adaptive *differentiation* of behaviors or concepts.

DSS is too coarse an approximation of continuous dynamic systems for capturing bifurcations in detail. Still, something can be done in order to get hold of this important mechanism. DSS theory assumes that among the dynamic symbols occuring at some locus, some are bifurcation siblings of others. Formally, this "sibling relation" is cast into a *differentiation ordering* $\leq$ on the dynamic symbols of a coherent language. Fig. 3 shows a differentiation ordering for the symbols appearing in fig. 2.

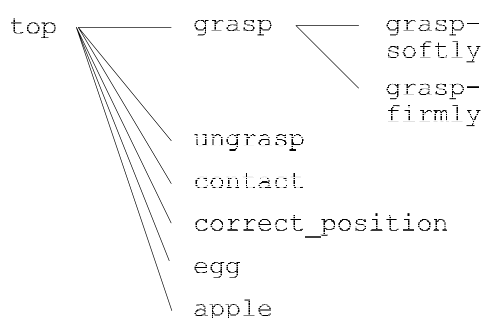

Fig. 3: A differentiation ordering for the symbols appearing in fig. 2.

A differentiation ordering must satisfy certain constraints. The basic idea is that associations made from more differentiated dynamic symbols must be "subsumed", within a coherent language, by less differentiated associations. An association $s_1...s_n$ is said to be *subsumed* by $t_1...t_n$, if $t_i \geq s_i$ (i = 1, ..., n). For instance, referring to figs. 2 and 3, `top top` subsumes `grasp ungrasp`, which further subsumes `grasp-firmly ungrasp`.

Technically, this subsumption constraint is expressed via a *generalized simulation* $\sigma_\leq$. The definition of generalized simulations $\sigma_\leq$ is derived from the definition of ordinary simulations by replacing condition (ii) with

(ii') for all $(x_1, r_1, x_1') \in trans_1$, for all $x_2 \in \sigma(x_1)$ there exists $x_2' \in \sigma(x_1')$, $r_2 \geq r_1$, such that $(x_2, r_2, x_2') \in trans_2$.

The simulation theorem carries over to generalized simulations (unpublished result): $C_1 \leq C_2$ iff there exists a generalized simulation $\sigma_{\leq}$: $G_\varphi(C_1) \rightarrow G_\varphi(C_2)$. $C_1 \leq C_2$ means that every association in C_1 is subsumed by some association in C_2.

I return to the issue of constraining differentiation orderings. Such an ordering is required to satisfy that there exist a generalized autosimulation $\sigma_{\leq}$: $G_\varphi(C) \rightarrow G_\varphi(C)$ of the phase generator, such that every transition $(x_1, r_1, x_1') \in$ *trans*, where r_1 has a subsumer $r_2 \geq r_1$, is mapped on some $(x_2, r_2, x_2') \in$ *trans*. In the running example, such a generalized simulation is provided by $\sigma_{\leq}(1) = \{1\}$, $\sigma_{\leq}(2) = \{1, 2\}$, $\sigma_{\leq}(3) = \{1, 3\}$, $\sigma_{\leq}(4) = \{1, 4\}$, $\sigma_{\leq}(5) = \{1, 5\}$, $\sigma_{\leq}(6) = \{1, 6\}$.

Generalized autosimulations of the kind just used reveal a differentiation hierarchy of cyclic substructures in phase generators. Intuitively, more differentiated substructures "unfold" from less differentiated ones by the combined effects of dynamic symbol differentiation and an elaboration of the graph substructure. In the example, the least differentiated substructure is the single `top` loop on the carrier {1}, which differentiates to the substructure on the carrier {1, 2}, which in turn differentiates to the substructures on {1, 2, 3, 4} and {1, 2, 5, 6}. In terms of continuous attractor states of a neural assembly, the `top` loop might be interpreted as a highly chaotic "ground state", whereas the more differentiated subgenerators would correspond to less chaotic states featuring several lower-dimensional, or even near-limit-cycle, substates (beautiful examples in Yao & Freeman 1990, caveats discussed in Jaeger 1994b).

A hierarchically differentiated generator somewhat resembles an inheritance network of the kind used in symbolic AI for the representation of conceptual knowledge. The logical ramifications of this view on generators are examined in Jaeger (1992, 1994a), where the reader can also find a discussion of how a dynamic system perspective might shed new light on old questions, e.g., the problem of conceptual cycles, context sensitivity of concept interpretation, and others.

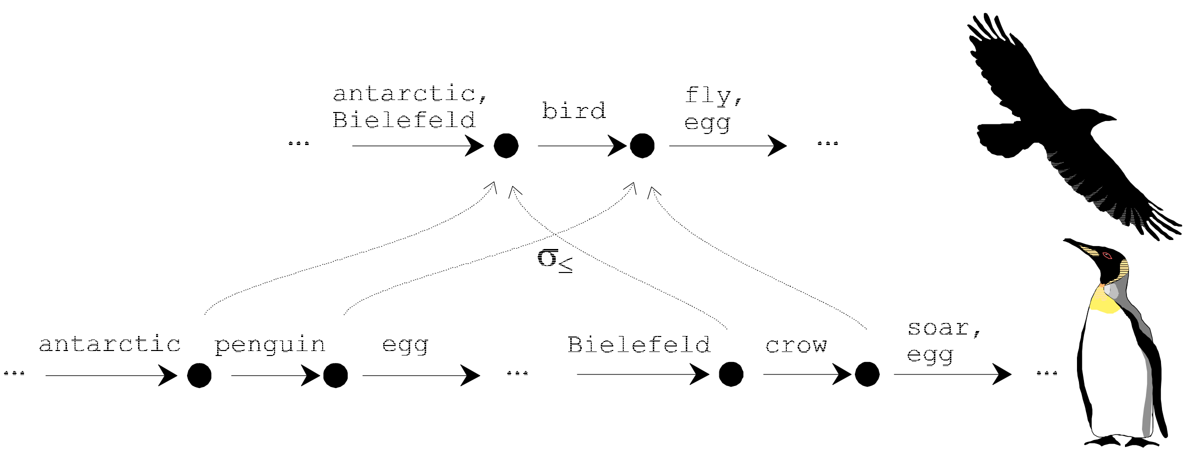

Fig. 4: To fly and fly not.

At the present occasion, I restrict myself to a phenomenon in generators which corresponds to nonmonotonic inheritance. Fig. 4 shows three portions from a hierarchically differentiated generator, which is presumably reconstructed from an ornithologist's left temporal lobe. The two lower portions belong to relatively differentiated substructures of the presumed generator;

the upper portion belongs to a less differentiated substructure. The former are mapped on the latter via a generalized autosimulation. The "unfolding" of the upper structure to the lower ones has several effect: the contexts `Bielefeld` and `antarctic` are separated from each other; `bird`s become `crow`s in `Bielefeld` and `penguin`s in the `antarctic`; and last but not least, `crow`s `soar` and `penguin`s stay grounded.

The basic model presented in this section can be augmented in order to capture different noise levels, and different levels of observational precision. This is achieved by using as a local phase space model not a single generator, but a two-dimensional array of generators, which are derived from each other via simulations. A detailed account (which in some aspects, however, reflects an earlier state of the theory) can be found in Jaeger (1994b).

In sum, generators with internal differentiation relations are the basic DSS "local phase space" model. On one hand, they can be considered as an approximative description of temporal and bifurcation relations of local regularities in continuous dynamic systems. On the other hand, they can be regarded as a peculiar version of inheritance networks for the representation of conceptual knowledge.

4 A glance at global DSS theory: self-organizing streams and associeties

Realistic information processing systems, in particular neural networks, are spatially extended and functionally structured. DSS offers, first, a model of spatially extended subsystems with locally homogeneous dynamics: *self-organizing streams*. Second, DSS describes how several such subsystems can be coupled together, achieving complex, functionally structured, multi-granular information processing systems: *associeties*. This subsection briefly points out the basic characteristics of self-organizing streams and associeties.

Self-organizing streams are, in some aspects, reminiscent of cellular automata. At a given point in time, a self-organizing stream appears as a spatial pattern of dynamic symbols, a *configuration*. Like configurations in cellular automata, DSS configurations change in time due to an operation which locally changes the pattern. These operations are called *microchanges*. Microchanges reflect the local dynamics which is described by a coherent language, as outlined in the previous section. A self-organizing stream, thus, always comes with a coherent language C which determines the local dynamics in a homogeneous fashion; this language is the analogue of the lookup table for the local transition rule in cellular automata. C can be considered as the self-organizing stream's long-term memory.

More precisely, a configuration is a finite, directed graph whose edges are labeled by dynamic symbols from the underlying dynamic symbol space. A microchange locally alters the edge labels and the very graph structure (which puts an end to the analogies between cellular automata and self-organizing streams, since in the former, the topological structure remains unchanged). Microchanges are applied asynchronously and in parallel at different loci in configurations. Input and output is achieved by deleting and adding labelled edges that lead to/from the configuration. These I/O operations, being themselves variants of microchanges, can be executed independently from "proper" microchanges at any time. Input arrives in the form of an arbitrary, directed, cycle-free, edge-labled graph, which is successively fed into the self-organizing stream. Output is generated in an inverse fashion. In the simplest case, this I/O format boils down to a sequence of dynamic symbols (example: language processing). In other cases, the format can be a broad "band" where many interconnected dynamic symbols pass through the stream simultaneously (as in visual information processing). Taken all together, a self-organizing stream is an anytime algorithm for processing symbolic streams of a quite general format.

A configuration is essentially (i.e., ignoring I/O transitions) a generator. Thus, a configuration yields a coherent language. This language models a "mental state" (of a homogeneous processing subsystem) in terms of a set of associations that are presently "active". The history of a self-organizing stream can, therefore, be described as a temporal development C_1, C_2, C_3, ... of languages. The self-organizing properties of a stream concern this language development. Remember that a self-organizing stream is governed in its local behavior by a generator describing a particular language C. The microchange mechanism is devised in a way such that "disordered" fragments of the configuration languages C_i are *attracted* by sublanguages of the "ruling" language C. This is achieved by various aspects of microchanges, which include concatenation of associations, differentiating dynamic symbols to make them fit better with C (these two effects amount to pattern completion), and deletion of unfitting edges (an aspect of noise filtering). The net effect is that in the absence of input, the configuration languages C_i are likely to converge to a sublanguage of C. When the process is perturbed by input, it still can be understood as continually "trying" to reach C.

A crucial phenomenon in a self-organizing stream is the formation of cyclic substructures that yield a sublanguage of C. Such structures are called *resonances*. Once formed, a resonance is likely to persist for some time, which amounts to a short-time memory effect. Furthermore, a resonance is likely to attract, modify and incorporate symbolic material from its vicinity. Resonances are the DSS model of gestalt formation. They are attractors in the global, parallel dynamic system presented by a self-organizing stream.

Self-organizing streams can be coupled together in *associeties*. Two coupling mechanisms exist. First, different streams can communicate via their output and input. Second, a higher-level, coarse-grained stream can *ground* in a lower-level, fine-grained stream. Intuitively, resonances in the lower-level stream are replayed by single dynamic symbols in the higher-level stream, such that the connectivity topology between finer-grained resonances is mirrored in coarser-grained associations. For instance, a conceptual-level stream can ground in a stream processing visual features. In that case, the finer-grained resonances made from visual features yield analogical representations for the concept-level symbols.

Between a higher-level and a lower-level stream, simultaneous top-down and bottom-up interactions are mediated by grounding relations, such that self-organization in each stream fosters self-organization in the other.

Fig. 5 gives an impression of an associety. Large ovals correspond to self-organizing streams, x's of various size to dynamic symbols of various granularity. Interstream band communication and communication with the sensomotoric interface are rendered by bold shaded bands. Resonances are indicated by cyclic dotted arrows. A grounding link is shown enlarged, which couples a fine-grained resonance from a visual feature processing stream with a dynamic symbol from conceptual-level stream.

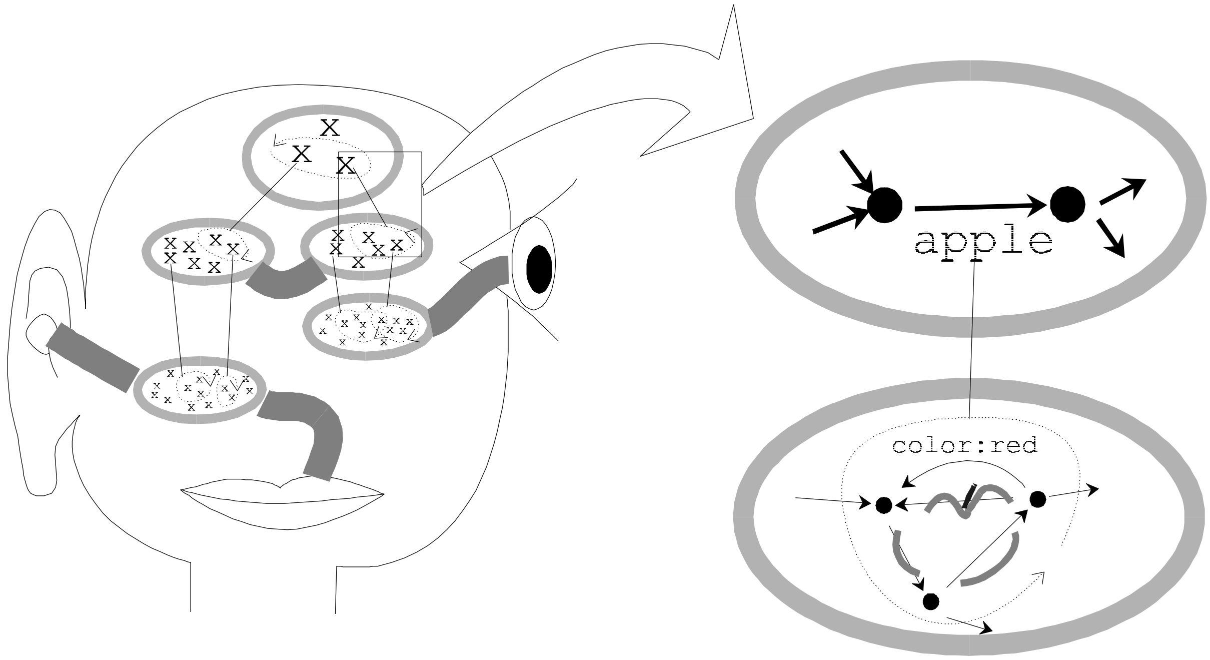

Fig. 5: An associety.

5 Related work

Combining a symbolic format of representation with a self-organizing dynamics is not a unique feature of DSS. Classifier systems, localist neural networks, and some singular approaches feature this combination, too.

Classifier systems (Holland 1975) have been applied in a few cases to modeling reasoning in agents (e.g., Patel & Schnepf 1991). Classifiers roughly compare to associations, and cyclic chains of them to resonances. However, classifier systems are typically used for modeling long-term adaptation processes with genetic algorithms. DSS, by contrast, focusses on short-term, situated activity. Thus, classifier systems and DSS have complementary recommendations.

Localist neural networks are characterized by their neurons carrying symbolic labels. Composite symbolic structures arise from simultaneous unit activations (e.g., Waltz & Pollack 1985, Smolensky 1986) or from spike train correlation (e.g., Mani & Shastri 1993). Recurrent localist networks could in principle realize stream processing in comprehensive multi-granular, multi-module architectures, although presently no such approach features that combination. Thus, this class of connectionist techniques is in principle comparable to DSS. When one takes a closer look at these techniques and DSS, one finds a number of shared properties:

- local interactions in a collective of informational units,
- formation of "coherent" composites,
- in logical terms, a type-free nature of informational units, i.e., a coincidence of what classical AI would call individuals, classes, and relations,
- feedback and cyclicity, and
- thermodynamic control parameters, in particular, computational temperature.

In (Jaeger 1994), I argue that these characteristics are due to turn up universally in architectures that combine a self-organizing dynamics with symbolic representation. The Copycat architecture (Hofstadter & Mitchell 1992) is another instance that I would include in this class. It achieves an analogy-based discovery of novel concepts, featuring an intricate combination of custom-built techniques that cannot be readily classified in traditional terms.

6 Conclusion

Dynamic symbol systems combine a symbolic format of "representation" (without an extensional semantics being intended) with a self-organizing dynamics. The theory is mathematically rigorous, essentially simple, and rests in a worked-out epistemological frame. Thus, DSS contributes to bridging the gap between dynamics-oriented, bottom-up techniques, and representation-oriented, top-down perspectives in AI. All this can be said of some related approaches as well. The entry of DSS into the game can nonetheless be justified, I believe, by the following points:

- DSS can serve as an approximative description of continuous dynamic systems.
- DSS is unique in its combination of stream processing with a coverage of the multi-granular periphery-centre axis of an agent.
- DSS suggests answers to some hard problems of logic-oriented AI, for instance concerning terminological cycles, analogical representations, and nonmonotonic inheritance.

Of course, DSS has its limits. It is a very crude, non-quantitative approximation to continuous dynamic systems. Its use for modeling conceptual-level reasoning is restricted by the fact that DSS-style information processing cannot capture numerical, logical, or otherwise "exact" modes of reasoning. Self-organization, at least as - imperfectly - understood today, is possibly necessary, but certainly not sufficient to explain intelligent thinking and acting.

References

Clancey, W.J. (1993): Situated Action: A Neuropsychological Interpretation. Cognitive Science 17, No 1, 1993, 87-116

Hofstadter, D.R., Mitchell, M. (1992): The Copycat Project: A Model of Mental Fluidity and Analogy-Making. To appear in: Holyoak, K., Barnden, J. (eds.): Advances in Connectionist and Neural Computation Theory, Vol. II: Analogical Connections. Ablex, Norwood, N.J.

Holland, J.H. (1975): Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, 1975

Jaeger, H. (1991): An Introduction to Dynamic Concept Systems. In: Boley, H., Richter, M.M. (eds.), Processing Declarative Knowledge. Proceedings of the PDK-91 at Kaiserslautern, Springer Verlag, Berlin (Lecture Notes in Artificial Intelligence 567), 87-106

Jaeger, H. (1992): A Type-free Semantics for Concepts in Contexts. In: Brézillon, P. (ed.): Proceedings of the IJCAI-93 Workshop on Using Knowledge in its Context. Technical Report LAFORIA 93/13, Institut Blaise Pascal, Université Paris VI et VII, 1993, 51-61

Jaeger, H. (1994a): Dynamic Symbol Systems. Ph.D. thesis, Faculty of Technology, University of Bielefeld 1994

Jaeger, H. (1994b): On Modelling Behaviors and Concepts as Attractors. Submitted to the DRABC-94 at San Sebastian, December 1994

Kiss, G. (1993): Autonomous Agents, AI and Chaos Theory. In: Meyer, J.A., Roitblatt, H.L., Wilson, S.W. (eds.): From Animals to Animats 2. Proceedings of the Second Int. Conf. on Simulation of Adaptive Behavior, 518-524

Mani, D.R., Shastri, L. (1993): Reflexive Reasoning with Multiple Instantiation in a Connectionist Reasoning System with a Type Hierarchy. Connection Science 5, No 3/4 (1993), 205-242

Maturana, H.R., Varela, F.J. (1984): El árbol del concocimiento. English: The Tree of Knowledge: the Biological Roots of Human Understanding. Shamhala Press, Boston 1987

Patel, M.J., und Schnepf, U. (1991): Concept Formation as Emergent Phenomena. Arbeitspapiere der GMD 602, GMD, St. Augustin 1991

Robertson, S.S., Cohen, A.H., Mayer-Kress, G. (1993): Behavioral Chaos: Beyond the Metaphor. In: Smith, L.B., Thelen, E. (eds.) (1993): A Dynamic Systems Approach to Development: Applications. Bradford/MIT Press, Cambridge, Mass., 119-150

Smith, L.B., Thelen, E. (eds.) (1993): A Dynamic Systems Approach to Development: Applications. Bradford/MIT Press, Cambridge, Mass.

Smithers, T. (1992): Taking Eliminative Materialism Seriously: A Methodology for Autonomous Systems Research. In: Varela, F.J., Bourgine, P. (eds.) (1992): Toward a Practice of Autonomous Systems. Proceedings of the 1st European Conf. on Artificial Life. MIT press/Bradford Books, Cambridge, MA., 31-40

Smolensky, P.(1986): Information Processing in Dynamical Systems: Foundations of Harmony Theory. In: Rumelhart, D.E., McClelland, J.L. (eds.): Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, MIT Press, Cambridge, Mass. 1986, 194-281

Steels, L. (1993): A Mathematical Framework for Autonomous Robots. Proceedings of the EWAIC '93, Moscow 1993, 333-334

Steels, L. (1994): Mathematical Analysis of Behavior Systems. In: Gaussier, P., Nicoud, J.-D. (eds.): Proceedings of the "From Perception to Action Conference", Lausanne, Sept. 1994, IEEE Computer Society Press, Los Alamitos, CA, 88-95

Van Gelder, T., Port, R. (1994): It's About Time: An Overview of the Dynamical Approach to Cognition. Research Report 116, Cognitive Science Programme, Indiana University, Bloomington, Indiana

Vera A.H., Simon, H.A. (1993): Situated Action: A Symbolic Interpretation. Cognitive Science 17 No. 1, 1993, 7-48

Waltz, D. L., Pollack, J.B. (1985): Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation. Cognitive Sience 9, 51-74.

Yao, Y., Freeman, W.J. (1990): A Model of Biological Pattern Recognition with Spatially Chaotic Dynamics. Neural Networks 3, No. 2, 153-170