The Illusion of Agency: Two Engineering Approaches to Compromise Autonomy and Reactivity in an Artificial System

Jean-Julien Aucouturier and Takashi Ikegami
Adaptive Behavior 2009 17: 402
DOI: 10.1177/1059712309344420

The online version of this article can be found at:
http://adb.sagepub.com/content/17/5/402

Published by:
©SAGE
http://www.sagepublications.com

On behalf of:
ISAB
International Society of Adaptive Behavior

Additional services and information for Adaptive Behavior can be found at:

Email Alerts: http://adb.sagepub.com/cgi/alerts
Subscriptions: http://adb.sagepub.com/subscriptions
Reprints: http://www.sagepub.com/journalsReprints.nav
Permissions: http://www.sagepub.com/journalsPermissions.nav
Citations: http://adb.sagepub.com/content/17/5/402.refs.html

>> Version of Record - Sep 23, 2009

What is This?
The Illusion of Agency: Two Engineering Approaches to Compromise Autonomy and Reactivity in an Artificial System

Jean-Julien Aucouturier1, Takashi Ikegami2
1 Temple University, Japan Campus, Riken/ERATO Okanoya project, Tokyo, Japan
2 Graduate School of Arts and Science, The University of Tokyo, Tokyo, Japan

This article describes and compares two approaches that can be used to build artificial systems that users tend to recognize as agents, based on a review of two systems previously built by the authors. One system, an interactive musical instrument, is a typical artificial intelligence (AI) implementation, based on the technique of constraint programming. The other, a dancing robot, is a typical artificial life (AL) endeavor, specified as a non-linear dynamical system. Although very different in their design, we found that both systems have the same goal of compromising autonomy and reactivity in their user interaction. They elicit interaction and—we argue here—an elusive feeling of agency, because they are neither too predictable nor too random. Creating and controlling such a compromise in a programmatic way is not a trivial problem: we find that both approaches (AI and AL) raise similar pragmatic problems that are in fact rooted in human perception science. Psychological experimentation is needed to clarify the relation between the internal dynamics of the artificial systems and the ongoing feeling of agency (or absence thereof) imparted in their human user. As a first step toward such experimentation, we derive a minimal mathematical model which subsumes both implementations and abstracts them from their respective contexts of music and dance. This model is similar to a Van der Pol oscillator, forced by an input signal coupled to its output in a non-programmatic way. It isolates two critical variables controlling the illusion of agency: the model’s sampling rate and the polynomial order of its reactive term.

Keywords agency · interaction · constraint programming · autonomy · chaotic itinerancy

1 Introduction

The concept of agency is a central concern for artificial intelligence (AI) and artificial life (AL). Whether one tries to mimic human intelligence in a piece of software or to let life-like behaviors emerge in an autonomous robot, the goal is to create some sort of agency in the object. Yet, attempts to define the term in an operational way (“what do you need to do to build an agent?”) tend to rely circularly on other concepts such as autonomy, sense-making, or consciousness—none of which has a consensual engineering definition (“what do you need to do to make something autonomous?”). While it elicits an intuitive agreement, what makes something an agent seems to resist analytical definitions. Many categories are like this: wide-
spread, yet undefined. To take a musical metaphor, agency resembles the concept of musical genre: everybody would agree on some prototypical example of, say, jazz (e.g., John Coltrane, Ballads, 1962) or an agent (e.g., my co-author); yet, nobody can ascertain when a piece of music stops being jazz to become, say, rock (e.g., Weather Report, Black Market, 1976; see Gjerdingen & Perrott, 2008) or what makes an agent stop being one (e.g., a patient suffering from an epileptic stroke; see Barandiaran, Di Paolo, & Rohde, 2009).

In this article, we try to avoid philosophical discussion of what properties of a system are needed to make it an agent. Rather, we take a pragmatic approach and propose to examine the technical design of two artificial interactive systems that we have found to be recognized as agents by their human users. Both systems (namely, a piece of musical software and a dancing robot) are previous work, described in detail elsewhere: we only review them here and bring them together in the new light of the problem of agency. This article does not attempt to make a strong claim.

Figure 1  Excerpt from The Argument Sketch, from BBC comedy TV show Monty Python's Flying Circus. Michael Palin (A, left) and John Cleese (B, right) fail to interact properly as B automatically gainsays any statement A makes. Images extracted by the authors from a video. ©BBC 1972–2009.
that both systems are agents: we only take them as starting points, based on anecdotal evidence of their success from our own experience interacting with them. Although very different in their design, we found that both systems owe their interest to a common rationale: they were designed explicitly to compromise autonomy and reactivity in the interaction with their human user.

As human observers, we have an intuitive feeling of what type of system we would readily call an agent: something we can interact with in an interesting way; something in which we can sense a complexity, a motivation; something worthy of our time; something like us. Daily experience and popular culture abounds with examples (or counter-examples) of agency. Take two well-known sketches from the classic BBC comedy program *Monty Python’s Flying Circus*. Both feature a discussion between two interlocutors (call them A and B), one of whom (B in our example) is responding with abnormal behavior. In the first (The Argument Clinic Sketch; Monty Python, 1972), A (Michael Palin) and B (John Cleese) are discussing what a proper rhetorical argument should be. To each of the elaborated points interlocutor A is bringing up, B simply responds with reflex-like contradictory arguments (“no it isn’t”) in a semi-automatic way (Figure 1). B is over-reactive: he responds to each of A’s actions, instantly, reliably; but always in the same syntactic and semantic way. B is not what A would recognize as an agent—the interaction falls short and B’s behavior eventually results in irritation and aggression.

Implementing such automatic reactivity in an artificial system is a standard part of the human–computer interaction toolbox. Weizenbaum’s (1966) ELIZA program is a software agent responding to a user’s natural language assertions, in a way mimicking a human conversation (a so-called chatterbot). In some cases, the software is able to sustain the illusion of human conversation for several interchanges:

(Human) Are you a robot?
(Eliza) Why are you interested in whether I am a robot or not?
(H) Because it’s my problem
(E) Your problem?
(H) Yes. I want to build a robot.
(E) You seem to be quite positive.

Yet, Eliza is implemented using very simple pattern matching techniques: it parses the human input, looks for keywords (e.g., “no,” “I’m a …,” “Are you…”) and responds using predetermined sentence prototypes, such as (in Common-Lisp syntax):

```
{((?* ?x) are you (?* ?y))
(Why are you interested in whether
 I am ?y or not?)}
```

Similarly, a music interactive system such as Rowe’s Acturus (2005; Rowe, 1993) responds to a human trombone performance in real-time, by detecting playing parameters such as tempo or accentuation and triggering predetermined samples or musical phrases according to these parameters. Complete programming environments, such as Cycling74 MAX/MSP, are based on this direct mapping paradigm between input and output to quickly prototype real-time interactive systems.

While complete reactivity is not what a human observer would expect of an agent, the opposite attitude will not do either. In a second Monty Python sketch (Interesting People; Monty Python, 1969), A (Michael Palin) interviews B (Eric Idle) for a fake TV show. While A’s tone is dynamic and to-the-point, B starts answering questions in a slow and unreactive manner, lingering through long sentences while ignoring his interlocutor’s interruptions and attempts to initiate turn-taking (Figure 2). B shows nearly complete autonomy: once A initiates the interaction (“Well, Mr Walters, what’s it like… ?”), B launches into a self-consistent but autistic monologue, using a disengaged oscillating body language, in which A’s input is not needed any longer. As in the “Argument” sketch, the interaction soon breaks down: A shows signs of boredom, tries to change topic and interrupt the speaker, and eventually turns his back to him and carries on with another interviewee. B is another example of a non-agent, failing to be recognized as such not because of too much reactivity, but rather because of too much autonomy.

As with reactivity, many techniques are known to implement such types of autonomy and self-consistency in an artificial system. Statistical models such as Markov models are able to capture regularities in data, and then synthesize infinite streams of original data with similar statistical properties. The result is often surprisingly convincing, especially in contexts where only local consistency is important, such as song lyrics:
A: Well, Mr Walters, what's it like being invisible?
B: (slowly and boringly) Well, for a start, at the office where I work I can be sitting at my desk all day and the others totally ignore me. At home, even though we are in the same room,...

A: Well, whilst we've got interesting people, we...
B: (droning on) ... my wife does not speak to me for hours, people pass me by in the street without a glance in my direction, and I can walk into a room without...

A: ... we met Mr Oliver Cavendish who...
B: (and on) ... Even now you yourself, you do hardly notice me...

Figure 2  Excerpt from “Interesting People,” a TV show parody from BBC comedy program Monty Python's Flying Circus. Michael Palin (A, left) and Eric Idle (B, right) fail to interact properly as B launches in an autonomous monologue, ignoring every attempt A makes to engage and eventually interrupt him. Images extracted by the authors from a video. ©BBC 1969–2009.

She’s standing right in times of trouble / Why she had to be / There is a chance that they may be an answer / Let it be / Yesterday love was such an answer / Let it be

©Sony/ATV Music Publishing

is the typical output of a first-order Markov chain trained on the lyrics of the two Beatles songs “Let it Be” and “Yesterday.” Departing from lingual communication, similar techniques can be used in such domains as musical composition, visual, or bodily interaction. For instance, n-grams are used in David
Cope’s (1996) EMI system to generate musical scores in the style of Bach or Chopin. With only limited human editing, scores produced by the machine managed to dupe even expert musicologists. Another approach to simulate consistent autonomous behavior uses non-linear dynamical systems. Bradley proposes to map key poses of pre-existing choreographic sequences (ballet steps) on a Rössler attractor, and use the chaotic behavior of the system to generate an infinity of variations on the same style, constrained by the fixed dynamics of the system, but ever-changing because of its sensitivity to initial conditions (Bradley & Stuart, 1998). But none of these systems have any built-in reactivity: there are purely autonomous, each a computational Mr Walters (Figure 2).

We see from these examples that an artificial system is not recognized as an agent if it cannot somehow compromise these two extreme behaviors of reactivity and autonomy. Too much of the former is boring, as is too much of the latter. However, while many techniques are know to simulate one or the other, combining the two is more challenging and has not received much attention. One common approach is to build a purely reactive mapping mechanism, then change its parameters dynamically. This is how, for instance, authors have proposed to map audio-visual input to the movement of a dancing robot interacting with its environment (Michalowski, Sabanovic, & Kozima, 2007; Tanaka & Suzuki, 2004). However, there seems to be no principled and satisfactory way to model the added dose of creativity, “randomness” being the default solution found: Michalowski et al. (2007, p. 91) propose to select parameters randomly, and to modify them “at random intervals in order to keep the dance interesting.” Similarly, the dynamic evolution of mappings in Tanaka and Suzuki (2004, p. 423) is “predesigned or just random.” Sometimes, creativity, or rather “complex and unpredictable movement” (Tanaka & Suzuki, 2004, p. 423), appears as a side effect of the lack of robustness in, for example, vision algorithms (which indeed are notoriously difficult in unconstrained environments), and this is viewed as a desirable feature (“exploiting environmental complexity in a good way for the application”).

Another approach to combine reactivity and autonomy is to build interactivity on top of an autonomous behavior. Pachet’s Continuator system is a musical instrument continuously building a Markov model of the style of the musician playing with it (Pachet, 2004). When the musician stops after playing a phrase, the machine responds by playing the optimal Markov continuation of the phrase according to the statistical model it has been building. The dialog that follows, a strange example of “jamming with oneself,” proved highly enjoyable for professional musicians and children alike. It makes a strong case for investigating further frameworks to compromise reactivity and autonomy.

In the following, we review and contrast two technical solutions (previous work by the authors), which were explicitly designed to achieve such a compromise. One of them is based on a symbolic AI technique (constraint satisfaction programming) and the other on a non-linear dynamical system typical of AL (namely, chaotic itinerancy). They were implemented for different applications: the former to create a musical interactive system (in the tradition of the Continuator), the latter to make a robot dance to music. Both have design pros and cons, and contrasting them sheds interesting light on the problem of agency.

2 An AI Approach: Constraint Satisfaction Programming

Our first system is a computer-based music interactive system, named Ringomatic (Aucouturier & Pachet, 2006), which produces an audio signal by concatenating short audio recordings extracted from existing musical recordings. For instance, it can typically produce a continuous stream of drum sounds, extracted from all the parts of the Beatles songs where Ringo Star can be heard playing (hence the name, an automatic Ringo Star). The output is produced in real-time, by automatically selecting the fragments to be concatenated from a large database of sounds, according to a user-generated musical input, for example on a piano connected to the system. For instance, the system may select and concatenate calm and slow drum sounds when the user plays the piano in a calm and slow manner, and more energetic drum sounds when the user plays with more amplitude. Note that the system does not need to be restrained to drum sounds, or Beatles recordings—it can be made to generate accompaniment made of vocal harmonies from Simon and Garfunkel, guitar chords from Gilberto Gil, or whatever fits the musical situation. Musical examples of the system can be found online.2
In order to become an interesting musical agent, the system clearly has to compromise reactivity and autonomy: if it is only reactive, for example, playing loud when the user plays loud, it would be a musical instrument, not an agent. If it is purely autonomous, for example, playing a predetermined sequence of sounds regardless of the user’s interaction, it would be a record player, and again not an agent. To formalize these two extremes, as well as all the intermediate stages, we use the paradigm of constraint satisfaction programming (CSP).

2.1 Local and Global Constraints
Constraint satisfaction programming (CSP) is a paradigm for solving difficult combinatorial problems, particularly in the finite domain. In this paradigm, problems are represented by variables which have a finite set of possible values, and constraints represent properties that the values of variables should have (i.e., satisfying the constraint) in solutions. CSP is a powerful paradigm because it lets the user state problems declaratively by describing a priori the properties of its solutions and use general-purpose algorithms to find them. There have been numerous applications of CSP to music, for example for the automatic harmonization of melodies (Pachet & Roy, 2001).

More specifically, we describe reactive behavior as possible “local constraints” on the next sound to be concatenated. Local constraints are posed dynamically in response to the user’s input. For instance, the programmer may decide to set a local constraint that forces the next selected sound to have the same amplitude as the user’s input (i.e., the drummer playing loud when the user plays loud). A local constraint only affects the present (i.e., the next sound to be played), not the past. Its satisfaction does not depend on the sounds that were chosen and concatenated in the past.

Similarly, autonomous behavior is formalized as “global constraints” on the next sound to be played, as well as on past sounds. For instance, the programmer may decide to pose a global constraint that forces the sequence of selected sounds to have constant amplitude, regardless of the user’s input. The satisfaction of a global constraint does not depend on the user’s input: it constrains the choice of the next sound to be played based on past output sounds only.

Both types of constraints are implemented as cost functions. For instance, a local constraint matching the current amplitude of the user input (say, a floating-point number between 0, silence, and 1, very loud) will assign a cost to every possible choice of sound to be played next, according to:

$$cost_{local}(x_t) = |x_t - I(t)|$$  \hspace{1cm} (1)

where $x_t$ is the amplitude of the sound considered for concatenation, $I(t)$ is the target amplitude of the user input at the same time $t$. Similarly, a global constraint forcing the sequence to have continuous amplitude will assign a cost to every possible choice of sound to be played next, according to:

$$cost_{global}(x_t) = |x_t - x_{t-1}| = \dot{x}$$ \hspace{1cm} (2)

where $x_t$ is the amplitude of the sound considered for concatenation and $x_{t-1}$ is the amplitude of the previous sound played in the past. The optimal sound to be concatenated next is simply selected as the one that minimizes the cost function

$$cost(x_t) = \alpha \cdot cost_{local}(x_t) + \beta \cdot cost_{global}(x_t)$$  \hspace{1cm} (3)

(with $\alpha$ and $\beta$ two arbitrary weighting coefficients) among all possible choices of sound. In the example above, $cost_{local}$ penalizes (high cost) the candidates with an amplitude much different from the user input (i.e., penalizes for lack of reactivity), and $cost_{global}$ penalizes the candidates with an amplitude much different from the previous output (i.e., penalizes for lack of autonomy). When both costs are cumulated, the system selects sounds that conciliate both types of behavior (Figure 3). The field of CSP is rich with algorithms to find such optimal solutions even in very large domains using, for example, local search techniques (see Aucouturier & Pachet, 2006, for more details).

2.2 Design Aspects
2.2.1 Explicit Behavior Primitives One important aspect of the CSP formulation is that constraints, that is, cost functions, have to be programmed. It is the responsibility of the application designer to decide what are the behavior primitives corresponding to the reactive aspects of the system on the one hand (i.e., what are the local constraints) and its autonomous aspects on the other hand (i.e., what are the global
Adaptive Behavior 17(5)

2.2.2 Explicit Control of Trade-Off  The application programmer can manipulate the trade-off between the satisfaction of reactivity and autonomy by putting weights \( \alpha \) and \( \beta \) on the local and global constraints respectively (see Equation 3). A variety of behaviors can be achieved ranging between complete reactivity (\( \beta = 0 \), no continuity of output energy) and complete autonomy (\( \alpha = 0 \), no matching of input energy; see Figure 4 and experiment below).

2.2.3 Indirect Control of Time Course  Under two contradictory constraints such as above, one may intuitively wish for an output sequence that will select sounds with energies that gradually converge to match the input, at a speed depending on the constraints weights (the more reactive, the faster the match). However, careful analysis of the system shows that this time course depends closely on the mathematical formulation of the cost functions. If the cost functions in Equations 1 and 2 are summed, the result is a linear function whose slope depends on \( \alpha \) and \( \beta \). Its minimum corresponds either to \( I(t) \) (perfect reactivity) or to \( x_{t-1} \) (perfect autonomy), and intermediate values are never reached, even if they exist (Aucouturier & Pachet, 2006). More gradual time courses can be achieved using non-monotonic cost functions such as:

\[
\text{cost}_{\text{local}}(x_t) = |x_t - I(t)|^2
\]

and
More generally, the dynamics of the system depend on the gradient of the costs in the neighborhood of the constrained values, which can be counterintuitive in the frequent case of several simultaneous constraints. Minor changes in the mathematical formulation of the cost functions can alter the system’s dynamics in a very visible way.

2.3 Experiments

We validated the behavior of the Ringomatic system by numerous laboratory experiments (Aucouturier & Pachet, 2006). Figure 4 shows the behavior of the system subjected to a continuous piano input, played by a human experimenter on a MIDI keyboard. The input consists of five repetitions of a short 1 s improvised motive (built on a blues scale), separated by a 1 s pause. The system records the input’s amplitude, obtained as the velocity parameter of the MIDI note-on messages from the keyboard. The CSP system implements the two constraints formalized above: a local constraint forcing the output to match the input’s amplitude (weighted by a parameter $\alpha$) and a global constraint forcing the amplitude of the output to be constant (weighted by $\beta$). The system selects segments to be concatenated in a dataset of 500 drum sounds, each indexed by amplitude (computed as root mean square of the audio signal). Between each repetition of the input played by the human, the ratio $r = \alpha/\beta$ between the constraint weights of the CSP system is changed: first $r = \infty$ (i.e., $\beta = 0$), then $r = 2$, $r = 1$, $r = 1/2$, and $r = 0$ ($\alpha = 0$). Figure 4 shows the amplitude of the system’s real-time output, aligned in time with the amplitude of the input. For $\beta = 0$, the system simply matches the user’s input (with a slight phase delay). For $\alpha = 0$, the system does not take any account of the interaction and only obeys to the continuity constraint. With intermediate settings, the drumming machine follows the input amplitude while still preserving continuity, thus yielding a more musical output.

![Figure 4](image_url)

**Figure 4** System behavior with different ratio $r = \alpha/\beta$ of local to global weight. Solid line: amplitude of the user’s input to the system, as a function of time. Circled-line: amplitude of the system’s output, as a function of time. An ideally reactive system should have both lines superimposed (with a short phase delay). A: $r = \infty$, the drumming machine reacts immediately; B: $r = 2$; C: $r = 1$; D: $r = 1/2$; E: $r = 0$, complete autonomy of the system. The most natural interaction is obtained for intermediate ratios such as B and C.
Outside of the laboratory, Ringomatic was extensively performed in a musical context, using a real-time audio input played on an accordion and a database of sound extracts from popular music recordings, both in academic meetings (e.g., International Conference on Music Information Retrieval, London, UK, 2005) and at music festivals.

3 An AL Approach: Chaotic Itinerancy

Our second implementation addresses the problem of controlling a robot to execute free and solitary dance movements to music, with the similar goal of being neither simply reactive to the music, nor too oblivious of it. In contrast to the previous approach, we pre-program neither the dance patterns nor their alternation, but rather implement basic dynamics in the robot and let the behavior emerge (Aucouturier, Ogai, & Ikegami, 2008).

Our approach is based on the work of Ikegami (2007) which simulated a mobile robot using a network of neurons controlled by a biologically inspired model, the FitzHugh–Nagumo model (FHN). The system moved around by sensing environmental patterns through its input neurons and computing the motor outputs via the FHN network. After investigating the spatio-temporal dynamics of the system, Ikegami discovered spontaneous switches between different modes of neural activities: internal neural states could have a variety of local and global attractors because of the interference between the internal neuronal network dynamics and input signals. On a macro-level, this translated into a succession of “navigation styles,” with seemingly spontaneous switches from one to the next. When a given navigation style activates the system’s input neurons with a compatible spatial pattern, the motion is quasi-stable. When the entrainment breaks down because of the environment or the internal dynamics, the navigation style changes, entering a different pattern. This interesting switching behavior is the result of the complex dynamic properties of the FHN neuronal model.

3.1 FitzHugh–Nagumo and Chaotic Itinerancy

A FitzHugh–Nagumo neuron is a coupled system of a fast variable $u$ modeling the excitation of real neuron’s membrane potential and a slow variable $\omega$ controlling its refractory state:

$$\frac{du}{dt} = c\left(u - \frac{u^3}{3} - \omega + I(t)\right) \tag{6}$$

$$\frac{d\omega}{dt} = a + u - b\omega \tag{7}$$

where $I(t)$ is an input signal (a pulse train), and we take $a = 0.7$, $b = 0.8$, and $c = 10$ (FitzHugh, 1961; Nagumo, Arimoto, & Yoshizawa, 1962). The neuron generates a spike when its output $u$ reaches above a threshold $T_u$ (here, $T_u = 0$). The dynamic properties (attractor, bifurcations) of the FHN equations are well known: depending on the period of the input spike train, the output is generally periodic, but sometimes also chaotic (Kostova, Ravindran, & Schonbek, 2004). When many FHN neurons are connected to one another with time delays or different signaling speeds, the overall system has capacity to generate a specific type of chaotic dynamics, called chaotic itinerancy (CI).

Chaotic itinerancy is a relatively common feature in high-dimensional chaotic systems: it consists of itinerant behavior among low-dimensional local attractors through higher dimensional chaos (Ikeda, Otsuka, & Matsumoto, 1989; Kaneko & Tsuda, 2003). While CI is generally defined in a closed system, we are interested here in placing it in open interaction with a spatial or temporal environment, something we call “embodied Chaotic Itinerancy” (ECI). A system with ECI dynamics does not have constant interaction with its environment: when it orbits in local periodic attractors, it is robust against changes on its input signals; however, chaotic transients between attractors are ready for receiving incoming signals. Namely, a robot receives the external signal when its internal states are chaotic ones, otherwise it neglects the input signals. This attachment–detachment behavior makes ECI a minimal but seductive model of the neural dynamics behind conscious states (Ikegami, 2007) and active perception (Kay, 2003). Most interestingly for our purpose, ECI was also proposed as a model of turn-taking interaction (Ikegami & Iizuka, 2007), precisely what is lacking in the Monty Python exchanges of Figures 1 and 2. In a common dynamical framework, ECI allows reactivity to the environment, autonomy therefrom, and the alternation between the two.

For our dancing robot, we generate chaotic itinerancy by inputting to the robot neural network a sequence of pulses corresponding to the beats of the music being danced to, and propagating the pulses in
the networks by integrating Equations 6 and 7 with 4th-order Runge–Kutta (Press, Flannery, Teukolsky, & Vetterling, 1986). After some propagation in the network, neurons output sequences of pulses which regularly enter quasi-periodic states proportional to the input, followed by less predictable sequences. The whole behavior of the robot is driven by such output sequences: our test robotic platform being a simple vehicle with two wheels (Figure 5), we use the output of four neurons of the network to generate the left and right wheel velocity $V_L$ and $V_R$. Each output spike of a neuron triggers a square pulse with a length $W_p$, which we integrate using:

$$V_L(t) = \tanh(h_1(t) + h_2(t))$$  \hspace{1cm} (8)

$$V_R(t) = \tanh(h_3(t) + h_4(t))$$  \hspace{1cm} (9)

where $h_i(t)$ is a test function holding on the output pulse train of the $i$th “motor” neuron, returning 1 if a spike is active at time $t$ (i.e., was generated within $W_p$ time steps in the past), else 0.

Periodic spike trains generate constant or periodic values of velocities for both wheels. Chaotic spike trains generate non-predictable sequences of velocities for both wheels. As a result, in a typical ECI manner, the robot exhibits a variety of motion styles, some strongly coupled to the musical rhythm and others more independent from the input—as well as spontaneous jumps from one style of motion to the next. Importantly, this behavior is not random, since it is the deterministic solution of a non-linear dynamical system. Videos and computer simulations of the robot motion can be found online.5

### 3.2 Design Aspects

#### 3.2.1 No Control Over Behavior Primitives

In the artificial intelligence approach of constraint programming, each of the behavior patterns of the agent is explicitly programmed. The application designer specifies, for example, that when the agent is reactive, it will try to match the user’s amplitude and when it is autonomous, it will try to keep its amplitude continuous. Even if the global behavior is unspecified, since it is decided dynamically as the optimal solution at the current time step, each of its building blocks, or “primitives,” is well known. In the artificial life approach, this is not possible. It is the same abstract output neuron sequence (the solution of a set of non-linear differential Equations 6 and 7) which alternates between reactivity and autonomy. The mapping to behavior stays analytically the same (Equations 8 and 9), and the different behaviors only emerge by virtue of the dynamics of the system, its past state and the specific musical input that it receives. Especially, the shape of the quasi-periodic patterns depends on the attractors through which the chaotic dynamics
itinerates—none of these can be programmed explicitly.

3.2.2 No Control of Time Course  Similarly to constraint programming, the time course of the CI system depends on the mathematical formulations hidden “under the hood” (cost function in the former, initial conditions and network parameters in the latter) and cannot be controlled easily. Because of its chaotic nature, alternations between reactive and autonomous phases are deterministic but unpredictable. In particular, periodic patterns may reappear several times (as the orbit re-enters the same attractor), but not necessarily. A FHN neuron behaves chaotically depending on its periodic input. In the case of music, this periodicity cannot be controlled by the application designer. In the case of ECI modeling too (Ikegami, 2007), the temporal periodicity of the input is determined by both spatial pattern and the navigation style of the agent, so that it cannot be controlled externally.

3.2.3 The Need for Timescale Adaptation  The chaotic dynamics of the neural network evolve at a very fast rate: rhythm beats in a typical musical input occur at a timescale of 100 ms (e.g., 500 ms for a series of quarter notes at 120 BPM); several hundreds of Runge–Kutta updates are needed between each input spike; hence, output pulse trains are updated at a timescale of 1 ms. However, motor commands in the robot cannot be updated faster than 100 ms: first, because of typical hardware limitations of the platform (actuators do not react fast enough); second, and most importantly, because of the timescale of the observer’s perception of the robot motion: chaos at the millisecond timescale is beyond our time resolution. Therefore, we had to introduce a third intermediate timescale (roughly, 30 ms), at which the network’s (fast, 5 ms) pulse trains are sampled into intermediate values. These values are then (slowly, 100 ms) interpolated, converted to speeds, and sent to the robot. Using this intermediate timescale, the robot’s (fast) internal dynamics can translate into (slow) correlations between the music and the dance—and thus be detected by an external observer. This third timescale, artificially brought into the system, can thus be considered as an important ingredient to design agency.

Experiments with different sampling rates (reported online) showed that the dancing behavior is highly sensitive to this timescale.

3.2.4 Automatic Fluctuations of Trade-Off Ratio  While the level of compromise in the CSP approach (i.e., weight ratio between local and global constraints) is fixed for a given implementation, ECI-generated behavior automatically fluctuates between reactivity and autonomy with the passing of time. It is possible to quantify this behavior with information-theoretical measures between the system’s input and output. For instance, the information circulation \( T_{X,Y} \) between input \( X \) and output \( Y \) of the network measures how much the knowledge of the input contributes to the prediction of the future output, compared with the single knowledge of the output’s previous state (Palus, Komarek, Hrncir, & Sterbova, 2001). It is given by:

\[
T_{X,Y} = T_{X \rightarrow Y} - T_{Y \rightarrow X} \tag{10}
\]

where

\[
T_{X \rightarrow Y} = \frac{1}{\tau_Y} \sum_{\tau_Y = 0}^{\tau_Y^*} I(Y, Y_\tau/X) - \frac{1}{\tau_Y} \sum_{\tau_Y = 0}^{\tau_Y^*} I(Y, Y_\tau) \tag{11}
\]

is the information transfer rate from \( X \) to \( Y \). \( I(Y, Y_\tau) \) is the mutual information between output sequence \( Y \) and the time-shifted sequence \( Y_\tau = \{y(t - \tau)\} \); \( I(Y, Y_\tau/X) \) is the conditional mutual information between \( Y \) and \( \tau \) given the input sequence \( X \); and \( \tau_Y \) is the embedded delay of sequence \( Y \), computed as the first local minimum of \( I(Y, Y_\tau) \) for \( \tau = \{0, 1, 2, \ldots\} \).

Figure 6 shows a typical information circulation profile over time for a robot trajectory. The system oscillates between maximum \( T_{X,Y} \) (reactivity to the input) and null \( T_{X,Y} \) (autonomy from the input). In contrast, the same measure applied on the input and output of the CSP system yields a constant circulation value, depending on the chosen constraint weights.

3.2.5 No Direct Control of Trade-Off Ratio  ECI does not offer an explicit control of the balance between reactivity and autonomy such as the constraint weights available in the CSP approach. The properties of the
fluctuating $T_X, Y$ depend on the system’s dynamics, that is, its networks parameters such as time delays of the signal transfer from a FHN neuron to other FHN neurons and the basic FHN constants. It is unclear yet whether this parameter space offers any regularities that can be exploited with, for example, genetic algorithms in order to tune the network for a given behavior (e.g., “fluctuations every 5 s”).

3.3 Experiments

We tested the behavior of the dancing robot via numerous computer simulations in laboratory conditions. Figure 7 shows different stages of the robot trajectory, when the system is subjected to a 3 min piece of electronic music (Keichiro Shibuya, Atak001). The music is analyzed in real-time, which generates an aperiodic pulse-train of musical beats, with a mean inter-beat interval of about 500 ms (about 120 BPM). The trajectory of the robot is simulated in the $(x, y)$ plane using the following approximation (no friction, no mass)

$$
\frac{dx}{dt} = g_1(V_L(t) + V_R(t))\cos\theta(t) \tag{12}
$$

$$
\frac{dy}{dt} = g_1(V_L(t) + V_R(t))\sin\theta(t) \tag{13}
$$

$$
\frac{d\theta}{dt} = g_2(V_L(t) - V_R(t)) \tag{14}
$$

where $x, y$ is the space displacement vector and $\theta$ the heading direction. We use $g_1 = 50$ and $g_2 = 10$. Timescales were chosen as 5 ms, 30 ms, and 100 ms. Each
Adaptive Behavior 17(5)

Figure 7 Simulation of the robot trajectory in the \((x, y)\) plane for a given music piece. Each figure is an overlay of 100 successive robot time steps. Successive figures correspond to different stages of the simulation (every 25 s). None of these patterns can be programmed explicitly.

Figure in Figure 7 is an overlay of 100 successive robot time steps. Successive figures correspond to different stages of the simulation (every 25 s). The orbit shows typical chaotic itinerancy behavior, with locally quasi-periodic trails in attractors of various shapes and abrupt transitions from one attractor to the next through higher-dimensional chaos. We observe that different songs generate different types of orbits and styles of motion.

Outside of the laboratory, our system was demonstrated and compared with (a) a random-walk implementation and (b) a non-chaotic deterministic implementation in a public event in Apple Store Ginza, Tokyo, Japan on May 31st, 2007. (Video and press material available online.)

4 Discussion: Agency as a Perception Problem

Both approaches aim to elicit a feeling of agency in the human user interacting with the artificial system. However, implicitly, they make different assumptions about what matters for this user to detect agency.

4.1 AI's Agency: The Sense of Purposefulness

The AI approach of constraint programming postulates that agency is evoked by sensing different atomic, symbolic strategies in the observed system. For instance, by interacting with the system, a musician user should get a feeling that the system is aiming to follow the input, while also trying to make its output continuous. The recognition of such strategies requires continuous observation, and typically an active interaction with the system: playing louder to see if the system follows, exploring different playing styles to see what matters to the system, and so forth. Progressively, the human user will form hypotheses and test them, trying to fail the agent: “is the system simply reactive?” (play the same thing twice, and see if the output is the same), “is the system random?” (stop playing abruptly, and see if the system reacts), and so on. Such hypothesis-testing is reminiscent of a Turing test, especially when judges of the Loebner prize attempt to discriminate chatterbots, such as Eliza, against (unknown) human participants. Judges with some knowledge of the typical pattern-matching implementation of such wannabe-agents will try to push the system into syntactic areas where these patterns generate awkward linguistic output. For instance, a judge may be aware that a naive application of pronoun transposition interacts poorly with tag questions like “aren’t you,” and explicitly aim to provoke such mistakes (Shieber, 1994):

01:17:52 Judge 10
You’re trying to make this difficult for me aren’t you?

01:18:53 Terminal 5
What makes you think I am trying to make this difficult for you aren’t i?

However, besides simple sanity-checks, the cognitive strategy to decide whether a system is “enough of an agent” is unclear, especially because humans themselves may fail some of these tests: sometimes, we may repeat the same answer several times, in a mock automatical way; sometimes, we may choose to change topic and simply ignore an interlocutor’s question; all of this variability makes for natural and interesting interaction. In the following exchange, a Loebner prize judge mistakes a human participant for a computer pro-
gram because he/she lacks what the judges deems common knowledge (Pavia, 2008):

The other correspondent was undoubtedly a robot. I asked it for its opinion on Sarah Palin, and it replied: “Sorry, don’t know her.” No sentient being could possibly answer in this way.

The decision sometimes involves the feeling that a given behavior, however erratic, would be too difficult to implement in a programmatic way in a machine—which obviously depends on the technical expertise of the judge. For instance, in the following exchange, the same judge mistakes another machine’s answer produced by simply matching lexicon (“computer” → “programmers”) for a witty human comment—certainly a computer could not be programmed to be humorous (Pavia, 2008):

“Let’s cut straight to the point,” I wrote. “Are you the human or the computer?” One replied: “What do you think?” The other wrote: “Some of my friends are programmers.”

The first was the sort of thing I had been told to expect from a conversation program. The second respondent was playful, implying in his answer that he might well be a computer program whose only friends were programmers.

The idea that agency is in the eye of its beholder, and notably depends on the technical/analytical advancement of the observer’s culture, is reminiscent of religious myths where agency and purposefulness is granted to yet-unexplained natural phenomena (e.g., thunder, epidemic)—which since then have been explained and consequently have failed “the Turing test of modern science” (Kelemen, 1999). Similarly, experiments show that 9-month-olds, but not adults, tend to interpret the random movement of dots on a computer screen as intentional and goal-oriented (Csibraa, Gergely, Böaroa, Kooasc, & Brockbank, 1999). We detect agency as a side effect of our ability to recognize, for example, an ill-intentioned predator in our environment.

Hence, for a symbolic rule-based system such as Ringomatic or Eliza, giving the illusion of agency reduces to an attempt to trigger a false positive of the human’s “purposefulness detector.”

4.2 AL’s Agency: The Sense of Ownership

In contrast, the AL approach of chaotic itinerancy assumes that agency is spontaneously generated and detected, not with a cognitive or symbolic strategy of hypothesis testing, but rather in a statistical, sensorimotor way. Deciding that our dancing two-wheeled robot is an agent involves sensing long-term correlations between the input and the output of the system, which cannot be broken down into behavior rules such as “turn left on the music’s downbeats.” Experiments by Gergely and Watson (1999) show that 3-year-old children have a preference for “imperfect contingency.” A baby playing with a mobile gets most interested in its motion when it is unpredictable to some degree—which is reminiscent of our playful robot. Further experiments with sets of pictures of varying heterogeneity (different cars, different birds) show that human, baboon, and even pigeon behaviors correlate perfectly with the mathematical measure of the entropy of data (Wasserman, Young, & Cook, 2004)—we could go as far as saying that biological beings are potentially equipped with all the computational artillery needed to estimate constructs such as the information circulation in Equation 10.

While we are adept at measuring statistical correlations and independence in our environment, we also routinely assess such relations in our own behavior. When we feel that we own our hand, that is, that we are “the initiator or source of our movements, actions or thoughts” (Jackson & Decety, 2004), we are in effect detecting “our own agency.” Although we do not normally reflect on this in everyday life, this is a non-trivial sensorimotor ability, which involves active movement, proprioception, and visual feedback (Longo & Haggard, 2009). It develops early and seems to be a prerequisite to making sense of the world (Piaget, 1954). Further, sensing agency from a third-person point of view likely involves the same neural systems (e.g., mirror neurons) as sensing our own agency (or broadly defined, our own sensorimotor coherence), see, for example, research on facial gestures (Carr, Iacoboni, Dubeau, Mazziotta, & Lenzi, 2003). In other words, one may decide that something is an agent if its (observed) sensorimotor dynamics entrain the recognition processes of one’s own agency: “if it was me, I would own it.” We detect agency as a side effect of recognizing our ownership over our actions and thoughts.
Hence, for a non-linear dynamical system such as Miuro, giving the illusion of agency reduces to its demonstrating sensorimotor abilities that elicit a feeling of ownership in its human observer.

4.3 A Common Mathematical Framework

Interestingly, the two explanations of AI/CSP and AL/ECI can be reconciled. If the cost functions are specified with a quadratic term (a lesson learned from Equations 4 and 5), the CSP cost Equation 3 can be rewritten in continuous form as

\[ f(x) = \alpha(x - I(t))^2 + \beta \dot{x}^2 \]  

which reads as the sum of potential and kinetic energy of the system. The force behind the system’s reactivity then translates as potential energy, which is mainly derived from the environment \( I(t) \), while autonomy translates as the system’s own kinetic energy. Neglecting \( \alpha \) leaves the system at the sole mercy of its kinetic energy (i.e., autonomous), while putting \( \beta = 0 \) produces a system which simply minimizes its potential energy from the environment (i.e., reactive).

This formulation provides a way to grant chaotic behavior to the basic CSP system, bridging the gap to the ECI system: with non-null \( \alpha \) and \( \beta \) terms, the system can be forced into chaotic regime by using a non-linear (e.g., 4th order) function in addition to the quadratic form in the reactive term:

\[ \text{cost}_{local}(x) = \gamma(x)^4 + (x - I(t))^2 \]  

which introduces a non-linear term in the differential equation:

\[ \ddot{x} - 4\gamma(x)^3 - 2x + I(t) = 0 \]  

Similarly, the FHN Equations 6 and 7 can be reduced (if we take \( b = 0 \) in Equation 7) into the well-studied forced Van der Pol oscillator (Van der Pol & van der Mark, 1927):

\[ \ddot{u} - \mu(u^2 - 1)\dot{u} + u - I(t) = 0 \]  

Both Equations 17 and 18 have equivalent dynamics, in the same drive toward (embodied) chaotic itinerancy. The important difference between this model and a simple Van der Pol model is the same as between chaotic itinerancy and embodied chaotic itinerancy: the excitation term \( I(t) \) is not independent from the output of the system. It is coupled with unprogrammable factors. In the case of the AI approach, a human player is required—his (musical) input to the system is the indirect result of the system’s own output. In the case of the AL approach, real space and time is required: moving in space changes the timing and patterns of the robot’s input. Our agency model is not fully programmed. Rather, it is defined in interaction. Remove the coupling with user and/or environment and both are gone: the dynamic trade-off between reactivity and autonomy, and the resulting evoked feeling of agency.

It is important to note that all implementations of this common mathematical formula do not always induce agency for its external observers. Two critical variables need to be controlled:

- The amount of non-linearity in the reactive term \( \gamma(x) \) in Equation 17 (i.e., its polynomial order). While experiments with the two implementations described in this article show that fluctuations of autonomy/reactivity are an important ability to elicit agency, it is yet unclear which type of chaotic dynamics are best suited (controlled by the FHN network topology in the case of Miuro, or \( \gamma(x) \) in the case of Equation 17).
- The timescale at which the system is sampled. For CSP, the formulation of the cost functions (e.g., Equation 4 instead of 2) control how the compromise between local and global costs unfolds with time—too fast, and the system reduces to one extreme or the other. For ECI, the fast internal dynamics have to be sampled and interpolated to translate into correlations that are observable for the user, and usable in the system’s own sensorimotor loop.

4.4 Toward a Continuous Turing Test

Understanding how these variables control the elicited feeling of agency (or absence thereof) in a human user is as much a biological problem as an engineering design concern. For human cognition too, both self and third-person agency have complex dependencies on, for example, timescales: agency easily break down when
time delays are introduced. Infants do not self-attribute their reflection in a mirror (Povinelli, Landau, & Perilloux, 1996); similarly, infants do not mistake a video recording for their mother (Murray & Trevarthen, 1986). Adults, too, were found to (mistakenly) self-attribute a rubber hand (Botvinick & Cohen, 1998); time delays compel us to experience a disconnection between our visual and proprioceptive feedbacks (Shimada, Hiraki, & Oda, 2005). Timescale adaptation and integration is arguably one of the greatest challenges facing the human central nervous system, which has to compromise, for example, fast neuronal activation and slow motor signaling to achieve motor control (Smith, Ghazizadeh, & Shadmehr, 2006), just as our robot controller has to adapt the fast Runge–Kutta updates of the dynamical system and the slow updates of its motor hardware.

Hence, the next step of research aiming at building artificial agent systems is to evaluate such systems in interaction with a human user. More precisely, one needs to correlate the two time processes of (a) the internal dynamics of the artificial system (possibly in the form illustrated in Figure 6) and (b) the ongoing feeling of agency (or absence thereof) elicited in their human user. A possible experimental methodology for this would be a continuous measurement of a Turing-like decision, rating a human user’s perception of the system’s agency as it unfolds, builds up or dissolves with the passage of time. Continuous ratings are often used to study affective responses to continuous stimuli, such as a music performance during a concert (McAdams, Vines, Vieillard, Smith, & Reynolds, 2004).

We believe the simplified mathematical model above provides a first step toward such experimentation. First, it subsumes both the CSP and ECI approaches, hence avoiding the difficult problem of comparing two different classes of architecture (Rohde & Stewart, in press). Second, it abstracts both implementations from their respective contexts of music and dance, both high-level human activities involving a multitude of appreciation factors (taste, expertise, culture, etc.) which are difficult to control for if one aims to collect ratings of a perceived agency. Finally, the model lends itself well to the control of a low-level, psychophysical set-up. As future work, we envision a variation on Heider and Simmel’s (1944) classic film experiment (“Mr Triangle and Mrs Circle,” see Figure 8), where the movement of one shape is controlled by a human subject and the reaction of another shape is simulated using the above dynamical system. While carefully human-designed movement evokes a systematic anthropomorphic interpretation of the Heider and Simmel’s scene, surely this would not be the case for completely random movement: by varying the control algorithm, from random to chaotic with different reactive orders and timescales, we hope to elucidate some of the low-level analytic aspects behind the attribution of agency to an artificial system.

5 Conclusion

This article proposes that agency is assigned by an observer based on interaction, somewhere between reactivity and autonomy. We have described two approaches to explicitly engineer such a compromise into an artificial system. The first, based on symbolic artificial intelligence, is related to an understanding of agency as “being purposeful.” The second, using artificial life and dynamical systems, views the detection of agency as a sense of “ownership.” Although very different in their design, both approaches reveal that the feeling of agency (or the absence thereof) elicited in a human
user depends on the control of two critical aspects: the type of (chaotic) dynamics involved in the system and the timescale at which the system is sampled. Understanding these two aspects with more than anecdotal evidence requires systematic human evaluation, in the form of a continuous-time Turing test. As a first step toward such experimentation, we have proposed a unified mathematical model (similar to a Van der Pol oscillator) which subsumes both approaches and make these two aspects easier to manipulate.

Notes

1 State-of-art audio processing techniques allow a large degree of content-based analysis of sound, that is, automatically computing the amplitude or tempo of a musical extract.
2 http://www.jj-aucouturier.info/projects/ringo/
3 Time delays are needed to break down periodicities in the input’s inter-spike-intervals, which brings the otherwise periodic FHN model into chaotic regime.
4 As before, using signal processing techniques to extract this information from the audio signal in real-time.
5 http://www.jj-aucouturier.info/projects/miuro/
6 The important point raised by information circulation is that it expresses the “attachment/detachment” behavior between a robot and the environment. Agency cannot be attributed to the motion behavior of pure purposeful (attached) behavior. It is only possible when we have both coupling and decoupling states.

Acknowledgments

The first author wishes to thank François Pachet for his insightful work on reactive/autonomous musical interaction, for example with the Continuator system. Work on dancing robot Miuro involved the collaboration of Dr. Yuta Ogai (University of Tokyo) and members of ZMP Inc. This article is based on a discussion started in the Workshop on Agency in Natural and Artificial Systems, SAB 2008, Kyoto, Japan.

References


About the Authors

Jean-Julien Aucouturier is an assistant professor of computer science in Temple University (Japan Campus). He holds a Ph.D. in computer science from the University of Paris 6, France. He is interested in audio technology and artificial intelligence, and is currently doing research about the future of the music industry. See more on http://www.jjaucouturier.info

Takashi Ikegami is Full Professor in the Department of System Studies, Graduate School of Arts and Sciences, The University of Tokyo, Japan. He obtained his Ph.D. in physics at the University of Tokyo in 1989. Since then he has been working in the field of artificial life and organized the 9th International Conference on ALIFE in Boston in 2004. Takashi’s interests are investigating the gaps and continuity between life and mind and understanding biological autonomy. Thus he has been collaborating with researchers from other fields to study self-moving oil droplets, flies’ spontaneous flying, and new robot design. E-mail: ikeg@sacral.c.u-tokyo.ac.jp