Liminalities: A Journal of Performance Studies
Vol. 14, No. 1 (2018)

Can Robots Improvise?

Gunter Lösel

The idea of improvising robots might seem a strange subject, a bit of a joke. Isn't robot-like behavior the exact opposite of improvisation? Improvisers obviously tend to think so, and being an improviser myself, I certainly don't want improvisers to be replaced by robots – a notion that would fit the tradition of Edward Gordon Craig, who claimed the stage for the 'Über-Marionette' (Craig, 1911). But I do think that we can learn something about improvisation by trying to simulate it within machines. In what follows I will apply a paradigm of cognitive psychology: If we can simulate a mental process in a computer, we might get insights into cognitive processes that are hard to get another way because they are happening somewhere in the 'black box' of our minds. In this paradigm, computation is a kind of research, not so much in terms of technical innovation, but in terms of modeling. The question is: If we succeed in constructing improvising robots, what does this teach us about the cognitive processes of improvisation in humans?

Methodologically three steps have to be taken. Firstly the phenomena and domain knowledge of improvisation must be described and systematized, beginning with the description of phenomena and the specific language of improvisers. In a second step one must translate the findings into the language of cognitive psychology, which might enable us, in a third step, to translate the rules into algorithms; a formalized, mathematical language that can generate computer- or robot-behavior. The approach can be sketched like this:

Gunter Lösel heads the Research Focus Performative Practice at the Zurich University of the Arts (since 2014) and has been publishing on the theme of improvisation since 2004. He is working as an improvisational actor with his own theater group and holds a Ph.D in Theatre Studies as well as a diploma in psychology.

ISSN: 1557-2935 <http://liminalities.net/14-1/robots.pdf>

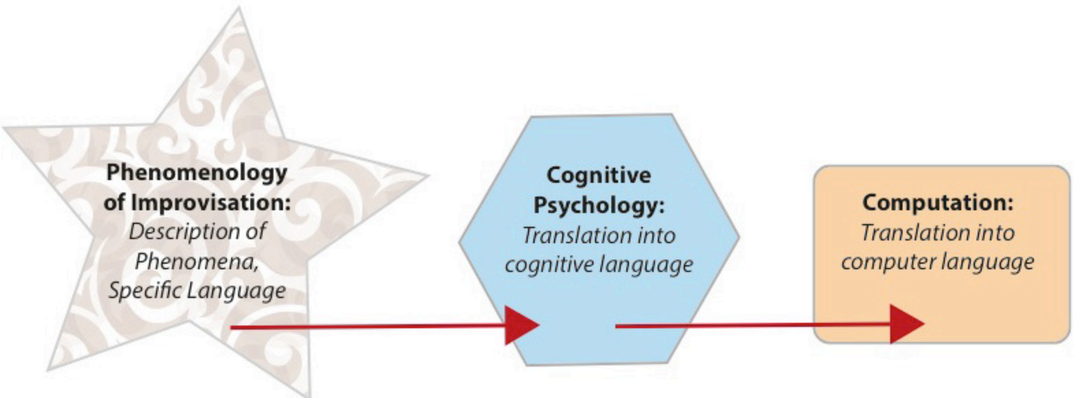

Fig. 1: Translation into a formalized language

Eventually, if we succeed in generating improvising agents, we can compare their 'improvised' behavior to improvisations performed by humans. This can be seen as a way to *validate* the model, but also to specify differences, that will inform us about human and robotic possibilities and limitations.

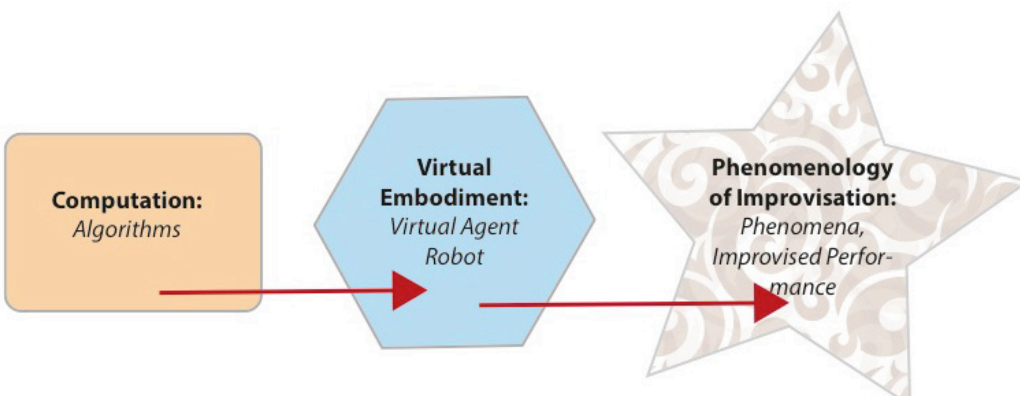

Fig. 2: Validation of Concepts

It is important to note that this kind of research *can lead to false positives*: A simulation might be able to produce phenomena that look like improvisation but, when analyzed more closely, are not. Still, this seems a promising way of gaining insight into the process of improvisation. There are a couple of existing explorations in the field, including the actual construction of an improvising robot. In the past decades musicians and computer developers have designed and constructed improvising musical automata, starting with *Voyager* in 1977 (Lewis, 1999) up to *Shimon* (Weinberg, Godfrey, Rae, & Rhodes, 2007), an improvising robot marimba player, that I will describe further down. In improvised dance and improvisational theater corresponding research has started much later and with less impressive results, presumably due to the fact that (1) dance involves a body that requires advanced technological effort and (2) theatrical signs such as language, social schemas, bodily expression and so on are much harder to translate into a code than musical signs. Up to this point there are no improvising robot actors and dancers in sight.

To teach improvisation to machines is challenging, given the fact that simply applying rules will not guarantee a successful improvisation – neither in a human nor in an automatic player. The machine has to have some degree of autonomy, so apart from implementing rules for improvisation the digital agent or robot must have some built-in module that facilitates non-deterministic, surprising, maybe creative, maybe aesthetic decisions. In what follows I will describe and discuss these contributions, speculating about the question of how far we are away from improvising robots, what features are necessary to build them and what this tells us about the human capacity to improvise.

1. Cognition and Improvisation

There have been several approaches to cognition in improvisation, using linguistic analysis (Sawyer, 2003) phenomenological explorations (Benson, 2003), analysis of domain knowledge (Lösel, 2013), systems theory (Borgo & Goguen, 2007) and Daniel Kahnemann's categories of system 1 and system 2 cognition (Drinko, 2013). Here I am using Magerko et al.'s approach as a starting point because it connects directly to computing. Brian Magerko, Waleed Manzoul, Mark Riedl, Allan Baumer, Daniel Fuller, Kurt Luther, and Celia Pearce (2009) conducted a study on the cognition of improvising actors at the Georgia Institute of Technology. The paper, *An Empirical Study of Cognition and Theatrical Improvisation*, presents the authors' view of improvisation as a process of problem-solving. They explore what improvisers conceive as the 'problem of the scene' and what they do to solve it by using video-cued-recall: Experienced improvisers were asked to improvise in a laboratory-situation, while the material was video-recorded. After the performances, the participants were shown their scenes again and asked to make comments about what they were thinking at specific points in the performance. This method has limitations; it can only uncover what the improvisers consciously experience and what they can verbalize. Nevertheless, it is probably one of the best current methods of "looking inside the heads" of improvisers.

The authors identified 4 categories of cognitive processing in improvisation:

(1) Basic Cognition,
(2) Shared Mental Models,
(3) Narrative Development
(4) Referent Use.

(1) Basic Cognition.

The authors found that improvisers engage in cognitive processes – such as inference, schema generation, mental imagery, theory of mind, and decision-making – while performing a scene. One crucial point of improvisation is that actors have very little information about the scene, so the input has to be somehow enriched and completed using cognitive strategies. Improvisers would frequently report inferring information about the scene from another improviser's actions, specifically about the scene's location and the improviser's goals and knowledge in the scene. For example, the following inference occurred early in a game of Party Quirks. Improviser D1, watching the party host, D2, set up the scene, described the following:

> …and then I thought, 'Who is he trying to be?' Like 'is he having a house party or is he a college guy? Are we in the middle of the forest?' I mean I'm trying to picture where he is setting his party and who his character is. I thought eventually because there was a door [the host pantomimed opening a door], well, we have to be inside then. We have to be in someone's house. (Magerko et al., 2009, 121)

The authors highlight two strategies as of specific importance: (1) The improvisers reflected upon the 'reality of the scene', thus excluding many options. To do this, they not only had to memorize the established elements of the scene, but also find out *what kind of world* they were in and do this within a creative act of constructing a world-model out of elements that they were given explicitly. In a second step they would have to adjust this fictional world to the concepts of a partner. I will refer to this process again in the next section. (2) Improvisers were looking for the game of the scene' – in accordance with the Chicago-improvisation-style. The concept has evolved from Viola Spolin's work, starting in the 1930s in Chicago, but has grown into a more sophisticated concept. While in Spolin's work, games have a clear setting and well-defined rules (Spolin, 1977), in the concurrent Chicago school games emerge during the scene without a clear setting or rules; actors have no time to explicitly agree upon rules. Instead, they are taught to 'Find the game' (or 'Listen to the game') (Halpern, Close, & Johnson, 1994). Any interaction can be turned into a game. In a game the participants have interdependent objectives – usually in an antagonistic way:

The closer A gets to his or her objective, the more distanced B will be towards his or her objective and the other way around. For example when two characters are both trying to sit in the same chair they will inevitably have interdependent objectives and start a game. This can take the shape of a conflict, but improvisers are taught not to rely on conflict too much in order to be open to any type of game that might emerge. Usually a game appears within the first five stage actions. It should not be constructed or forced upon the fellow players, but should emerge 'by itself'. Once found, the game can be heightened until a maximum is reached (usually after three rounds, with every round taking things to a new extreme). The scene ends when the game ends.

How would a computer detect a game and participate in the game of a scene? First it would have to be able to detect numerous social schemes. This seems very ambitious today, but since computers are very good at pattern-detection in general, it might only be a matter of technical advancement. The second claim is presumably much harder: The computer would have to build up some kind of intention – and do this without being programmed for a specific intention (like 'Win the chess game'). So here computation hits a hard obstacle. While potentially being able to identify the emerging rules of a game, *engaging* in the emerging objective is not a computer's core competence since it is lacking intentionality. Here the difference between a *pattern* and a *game* becomes evident: A pattern can easily be detected and completed by a computer since it does not involve intention, but a game is more than a pattern. It involves intentions in order to build up expectations in order to participate in the interaction of the game.

Intentionality

This point seems crucial, since intentionality is one of the strongholds of humanism. We generally hesitate to ascribe intentions to computers – but in the case of games, doubts come up: Firstly the intentionality within a game might be very different from intentionality in general: A chess player does not really want to destroy an army, a card player does not really desire to possess as many cards as possible and a monopoly player does not really want to invest in hotels. Instead, humans seem to have the ability to engage in a kind of fictional intentionality that appears and ends with the game. Indeed, in many games the objective is simply predefined by the game builder and can be found in the instructions, so in certain sense one can consider the objective to be part of the rules. These 'adopted objectives' might be much easier to imitate for a computer than intentions in real life.

Secondly a computer can maybe circumvent this problem by processing anticipations and expectations, and *this might look very much like intention* – it might only be a matter of definition if we call this intention or not. After all, our assumption that all human beings possess something like freewill is just an assumption, more and more questioned by neuropsychology. Improvising computers work with something like a simulation of intentionality, one example being *voyager*, a pioneering software for improvised music. It uses specific algorithms for the representation of musical input and output, that try to *mirror the emotional state of the human partner* (Lewis, 1999). George Lewis calls this process 'transduction' and emphasizes its importance for the interaction between machine and human:

> The transduction of musical intentionality into or from sound, or 'emotional transduction', is important to the construction of interactive work. This notion constructs physicality and performance as an intentional act, that is, an act embodying meaning, and announcing emotional and mental intention. (Lewis, 1999, 9)

Mirroring the partner and transforming the response may *substitute* intentionality – even though Lewis seems to suggest, that it is the human partner who constructs intentionality by imputing it to the computer and reacting *as if* the computer were an intentional being.

(2) Shared Mental Models

Shared mental models consist of the common framework of knowledge (i.e. mental models) shared among the members of a group. In improvisation, shared mental models usually require effort to create. Sometimes the mental models of a group disagree, and *cognitive divergence* occurs (i.e. when improvisers have different internal models about what is going on in a scene). These disagreements are resolved through *cognitive convergence*, which is the process of building towards a goal state of *cognitive consensus* (i.e. the agreement of assumptions). When cognitive consensus is reached, mental models are shared among the group (at least partially).

Three steps of cognitive convergence usually occur before cognitive consensus is reached. First is observation, the point at which an improviser realizes that his mental model diverges from others'. Second is repair, which refers to all attempts to reconcile divergences. Repairs can either be attempted in order for an improviser to align himself with another improviser's mental model or in order for an improviser to align another improviser with his own mental model. The third, and final, step of cognitive convergence is acceptance, during which cognitive consensus may occur. It is also possible that consensus may be rejected or that an improviser will achieve perceived cognitive consensus, where they think that they have achieved consensus, but actually have not. Finally, when two improvisers reach consensus, there is usually an explicit external acknowledgment that they understand each other. The process of sharing mental models can thus be quite complicated and needs several feedback-loops to build up a stable basis. A computer would hate to detect cognitive divergence and use cognitive convergence to reach cognitive consensus with another computer or human partner. This requires enormous social knowledge, something psychologists refer to as *theory of mind* (TOM), the ability to take the other's perspective by reconstructing it.

For example in a game of chess both players will not only strive for the best move from their own perspective, but will try to predict, what the other player might have on his or her mind. In order to do so, he or she has to see the chess board from the other side and try to reconstruct the others way of thinking. This is more or less a cognitive task – while empathy is the emotional equivalent. For a computer this mastering TOM is ambitious, but not unthinkable, while empathy will probably be beyond the scope of machines. Within a given and rather formalized game like chess computers have already gained dominance over humans, but generally taking the perspective of a human might be just as difficult as our taking the perspective of a machine.

Still TOM might not be needed for improvisation. I even doubt that improvisation is about sharing mental models. Drawing from literature on improvisation I would rather suggest that the process of building a fiction or pattern in improvisation can better be described as a *process of constructing a social reality step by step*, thus not allowing fixed mental models to emerge in the first place (Sawyer, 2003) Lösel, 2013). Consensus, I would argue, is not reached through convergence but through co-creation. This process might be easier for computers than achieving a TOM.

(3) Narrative Developement

Magerko et al. (2009) suggest a differentiation between events and existents of a story, but they fail to connect this to the specific language of improvisers. This might be due to their reference of the Chicago school only. The British-Canadian school represented especially by Keith Johnstone has generated a multitude of rules for storytelling. For Johnstone storytelling is based on the performer's ability to reconstruct and predict the audience's expectations, prototypically trained in his famous game What Comes Next? (Johnstone, 1987). In this game an actor starts sitting on a chair in a neutral position and asking the audience 'What comes next?'. He or she will only follow the suggestions of the spectators without adding any material, step by step building up scenes that are completely controlled by the audience's expectations. Through this the improviser learns to stay within the 'circle of expectations' and to fulfill them. Only when he or she has learned to sense, predict and follow these expectations the improviser will train breaking expectations. One can easily think of a computer playing the game of 'What comes next?', making suggestions for a scene and slowly building up the capacity to predict, what kind of story will please the audience. From such a starting point the neuronal network could learn breaking expectations from time to time, for example by using some random generator. Improvisation as a dialogical form of art provides a rich environment for computers to learn about humans. For this kind of storytelling almost no dramaturgical rules are necessary to begin with. Instead, patterns of storytelling will emerge through experience and a feedback-loop with the audience. I propose that we distinguish between two phases of improvisation, which I call game-building and game breaking: In a phase of game-building expectations of the audience and fellow players are met, while in a phase of game-breaking expectations are broken (Lösel, 2013).

(4) Referent Use and Domain Knowledge

A key effect of improvisers' training and experience is the use of referents; specific terms or language referring to the improvisational techniques. As with many of the performing arts, improvisation has built up an extensive vocabulary, a set of widely held core principles (e.g., accept an offer and build on it), and a long list of techniques and games used within scenes.

Magerko et al. can show that improvisers use their specific language to describe the problems and choices in a scene. The language shapes what and how improvisers think about their improvising. It is both a tool for analyzing scenes and finding shared mental models. Examining the language of improvisation might be a direct way to understand the cognition of improvisation. This seems trivial, but has not been made evident: improvisers share similar concepts and rules; they have a common knowledge domain. While this study provides evidence for the use of domain knowledge in *post-performance* analysis, this might not apply to the cognitive processes of improvisers *while they are playing*. I will discuss the role of domain knowledge further down.

Modeling improvisation

Magerko et al. suggest that improvisation can be modeled using a well-established cognition model; the decision circle from Newell's Unified Theory of Cognition (Newell, 1990). This model is linked to computing because it is connected to a computational model called SOAR (State, Operator And Result), so it can be translated into coding rather directly. A visualization of SOAR can look like this:

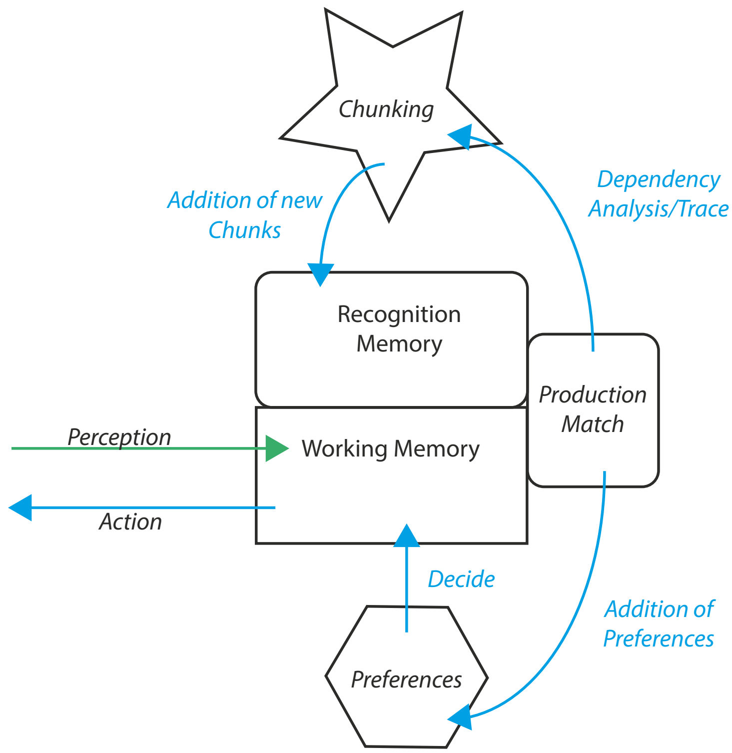

Fig. 3: Visualization of a SOAR

SOAR is an input/output concept with perception as input and action as output. Between input and output, working memory, recognition memory and production-match work closely together to generate two learning circles: one, displayed at the top, is designated to *proposing options*. If none of the stored options fit, the problem and strategies have to be broken down into smaller pieces – sub goals – until new options can be generated. The other circle, displayed at the bottom, serves as a *decision-maker*, using past experiences from an episodic memory that continually feeds preferences and is updated with every new experience.

SOAR uses production rules to generate states that gradually bring the system closer to the goal state. The main link to the outside world is the working memory, which controls the input (perception) and the output (action), completing and evaluating the input and selecting the most promising action. It operates on three structural levels that follow a sequence over time, but can be repeated as many times as necessary. Displayed in a temporal pattern, the

SOAR follows Newell's decision circle from, which states that, whenever reasonable, cognitive acts can be separated into five steps:

1. Receiving input
2. Elaboration of new knowledge based on other knowledge or inputs
3. Proposal of new operators / actions / goals to pursue
4. Selection of one of the proposed courses of action
5. Execution of the selected action

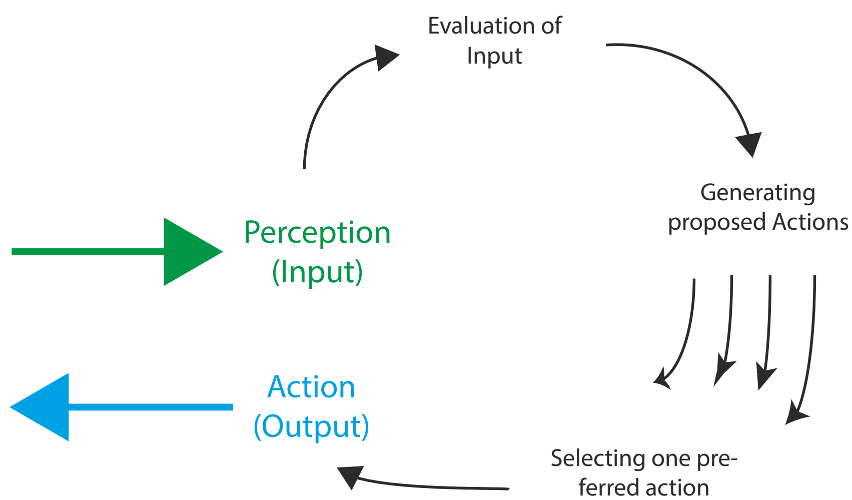

Fig. 4: Newell's Unified Theory of Cognition

Magerko et al. assume that this circle applies to improvisation in the same way as to other cognitive actions, but this assumption might be false. I propose that, when we apply the model to improvisation, major differences to the normal circle have to be considered and inserted. For improvisation, the normal decision circle seems to be the exception. Instead, the decision circle would look something like this:

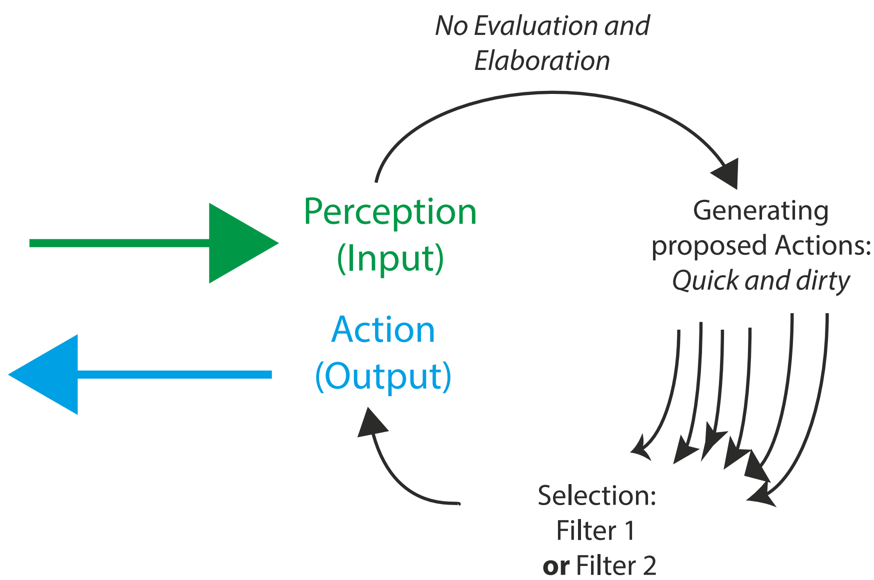

Fig. 5: Newell's model adapted to a mode of improvisation

Improvisers are trained *not* to evaluate the input, but to greet every impulse with joy and acceptance. Making offers and accepting them is a central rule in improvisational theater, shared by both the Chicago and the British-Canadian school (Salinsky & Frances-White, 2008, 57). An offer can be accepted in many ways – either in the obvious content or in the subtext – which makes offer/response much more complex than it first appears. The Chicago school coined the term the 'Yes and-principle', highlighting that accepting an offer and adding new information, and thus creating a new offer, are always connected and depend on each other. This principle guarantees that each player's contribution is integrated into a chain of communication that builds up a fictional reality: 'The whole point of the Yes And game is to build a chain of ideas, each linked to the previous one.' (Salinsky and Frances-White 2008, 59). It is not enough to soberly accept an offer, it should be accepted emotionally, enthusiastically, and meaningfully (Halpern, Close, & Johnson, 1994, 45-46) (Salinsky and Frances-White 2008, 94) (Johnstone 1999, 101-129). As any form of judgment will harm the collaborative relationship and responsiveness between the stage partners, improvising actors find ways to circumvent the process of input evaluation. While this is common ground for improvisation *in theater*, it might not hold true for music and dance, since they don't rely on a fictional reality as much as theater.

Improvisation certainly needs to go beyond affirmative action or the 'Yes-and-priciple', it has to stimulate an oscillation between affirmative and non-affirmative: While in a phase game-building the improviser is bound to stay inside the circle of expectations, in a phase of game-breaking the opposite is needed, strong, unpredictable, spontaneous action without any consideration of expectations. The selection of outputs thus differs in relation to the phase of improvisation, which means that two different strategies or codes for selection must exist:

Phase of Improvisation	Selection code
Game-Building	Filter 1 "Stay *inside* the circle of expectations" (Don't try to be original, don't make jokes, be obvious)
Game-Breaking	Filter 2 "Stay *outside* the circle of expectations" (Don't be afraid of spontaneous ideas, psychotic or obscene ideas)

Table 1: Selection codes for different phases of improvisation

In terms of cognitive psychology one might draw parallels to the dichotomy of convergent versus divergent cognition.

Could computers simulate the decision circle of improvisation? Actually this seems easier than simulating the standard decision circle, because no evaluation of input (which would involve some kind of episodic memory) is needed, nor is there a complicated code for the selection of output (which would call for domain knowledge). But there is still an obstacle: the computer or digital agent or robot would have to learn to predict human expectations. As sketched out above, this might be possible, using machine learning in a specific setting like 'What comes next?'. In other words, computers don not need to build up a complete representation of human cognition, but might be able to interact and learn as soon as they master predictive processing.

There is a general assumption behind this model: Improvisation is seen as a process of decision-making and/or problem solving. But a problem-solving process has to contain a mechanism for comparing the outcome of a process with the anticipated result, and in improvisation one might argue that there *is no anticipated result*. On the contrary, fulfilling a plan is exactly the opposite of improvisation. Also it seems that improvisation techniques are, to a certain extent, dedicated to *creating* problems instead of solving them. In the next section I will therefore explore the explanatory potential of a cognitive model that lately has been called predictive processing' (Clark, 2016), which focuses on the cognitive ability to anticipate.

2. Improvising Machines

Yoichiro Endo's (2008) dissertation at the Georgia Institute of Technology and strives for an even more formalized language to describe improvisation in order to construct robots. It attempts to build a computational framework and architecture that allows robots to act *proactively* in unpredictable situations. It draws heavily on cognitive studies, especially the Damasio's theory of embodied cognition, and neuroscience. Endo defines improvisation as a process to quickly generate solutions *without* having sufficient information. He then discusses the relationship between anticipation and improvisation:

> Anticipation here means that the robot can assess the current situation, predict the future consequence of the situation, and execute an action to have desired outcome based on the determined assessment and prediction. On the other hand, improvisation is performed when the consequence of the situation is not fully known. (2008, xxvi)

Anticipation is a way of predicting the future by simulating it, usually using large amounts of information. Improvisation, on the other hand, for Endo is something like an antagonistic concept that comes to action when anticipation fails. Improvisation is thus important for a robot because it enables the robot to solve a time-sensitive problem without complete knowledge of the situation. Current systems including the SOAR have a problem with fragmentary information. They will come to a screeching halt when there is not enough input to come up with a proposal for an action. Because of this, a digital agent, or robot, will behave exactly as a beginner improviser on stage; freeze and do nothing in order to minimize the risk of failing until there is enough information.

Anticipation and Predictive Processing

Endo sees improvisation as a cognitive mode to deal with situations where anticipation fails – either because there is not enough information, there is not enough processing time, or the situation seems to fall under the Markov assumption. He refers to Philip Agre's (1988) early contribution to Artificial Intelligence at the Massachusetts Institute for Technology. Agre's starting point is that improvisation is performed when the consequences of actions are not necessarily fully known and the time for data-processing is very limited. Under these circumstances, artificial intelligence has to avoid exhaustive computation. Agre relates this to *anytime algorithms*, in which a solution to a problem is considered in an incremental fashion. Anytime algorithms are constructed so that the quality of a solution improves monotonically with respect to the amount of time spent for computation. The computation can thus be interrupted anytime, yielding the solution attained within the given the time constraint. In this view improvisation would be linked to promptness, when time is too short for complete processing. This would certainly make sense from a biological perspective; a fast and imperfect reaction might in many cases be better than no reaction at all. From an improviser's perspective this supports the notion of embracing the imperfect and trying to avoid evaluation. Every idea is good, so the brain doesn't have to waste time finding the 'best' solution. This corresponds to the notion of following the first idea described above. Anytime algorithms correlate to what psychologists call heuristics. When there is insufficient information, the brain will draw on heuristics that allow for some probability of success. 'Chunking' is, then, a form of generating heuristics that allows for fast, imperfect responses. This is also in accord with Drinko's suggestion to conveive the improvisational mode of cognition as related to Kahnemann's system 1 (Drinko, 2013).

Endo's computational model uses an interesting technique for chunking. When episodic memory does not contribute to the solution for the new situation, the memory is converted into a *more abstract form*, which can then potentially serve as a base for a new answer. So, when anticipation fails and episodic memory cannot provide possible solutions, episodic memory is *transcoded* into more abstract data – resembling feelings or intuition, like a feeling of danger or a quality of movement – instead of a specific memory. This shifts the focus away from the *contents* of memory to the *coding* of memory. New options spring from transcoding experience into abstract forms. This solution certainly resembles to processes of improvisation. The proposal of new actions does not come from memory in a specific form, but in a very abstract form that is potentially beyond conscious control.

What Endo sketches fits well into neuropsychological concepts of seeing the brain as mainly designed for predictive processing continually, trying to predict the near future by actively projecting possible futures. Predictions are generated on every level of a multilevel system of processing. Only if prediction (or, in Endo's words anticipation) fails, will the higher levels of the system be activated, providing a new prediction involving higher – and more abstract – levels of coding and processing. This is also in accord with phenomenological approaches in the tradition of Edmund Husserl, who found that the experience of time is not only shaped by presence but also through a couple of seconds of past – which he calls *retentio* – and a short period of future – referred to as *protentio* (Husserl, 1964). According to the jazz-improviser and researcher David Borgo the automatic, unconscious prediction of the near future (=*protentio*), can be seen as the key for the analysis of free jazz with its mutual transitions and complex patterns (Borgo & Goguen, 2007).

Embodiment and Environment

In a wide improvisation is not an isolated cognitive process, it is embedded in the environments of the body and the surrounding situation, like the other improvisers, the stage, the audience and so on. It has to be conceptualized as a form of communication and dialogue. Prototypical for this is *Shimon*, probably the most advanced development in robotic musicianship (Hoffman & Weinberg, 2011). *Shimon* is a interactive robotic marimba player, that is working on a hybrid model. He uses domain knowledge of jazz musicians about standard chord progressions and beats, plus random inputs, generating patterns (Weinberg et al., 2007). While the development started with a classic cognitive approach to problem solving, the aspects of interactivity and embodiment became more and more important over the years Shimon was in development. Shimon has learned to listen to a human musician and continuously adapt its improvisation and choreography, while playing simultaneously with the human. He can identify gestures and build an improvisation on a gesture – a physical behavior – and play along with a human interaction partner in a call-and-response mode.

There are three modules involved.

1. The *Phrase-call and chord-response module* will detect the beat and the gestures of a human player and anticipate the next beats and phrases. It will also synchronize beat and chords and execute a sequence of simple and rhythmic chords. Thus, the robot tunes in with the chord sequence, just like a human musician.

2. The *Opportunistic overlay improvisation module*, draws on the bass-notes of the human player, tracking the beat and down beat. The robot will detect gestures, simulating the physical behavior a human would perform when playing the bass. This results in a dynamically changing confluence of two rhythms and one chord structure plus a choreografic element drawn from gesture detection.

3. The *Rhythmic phrase-matching improvisation module* will "listen" to the human players last beats and answer it with a phrase that oscillates between being very close to, or far away from, the human input. It clusters the input and random probabilities vary the degree of matching. This 'decay parameter' will determine the degree to which a robot will simply copy versus adding material that comes from somewhere else. The mechanism avoids copying completely. When the robot is adding material, it draws on previous human plays; the robot is using its episodic memory.

Improvising robots need to somehow (1) detect patterns like chord sequence, beats and baselines, or, in the case of theater, social schemas and/or games (2) create an output that can be understood by the human partner (3) decide, when to do the one or the other, defining phases of listening (pattern detection) and action (adding to the detected pattern). In this way chains of call-and-response cycles can evolve into complex, dynamical structures that facilitate the emergence of new material. The human player has to 'get the system's attention' (Lewis, 1999), Shimon has to detect the relevant patterns, and then decide to 'play along' (game-building) or go beyond expectations (game-breaking). This might not be creative in a strict sense, but does appear as responsive and spontaneous behavior and it is apt to inspire creative behaviour in the human partner. Responsivity and embodiment have already become reality in *Shimon* who, interestingly, is copying human behavior, including bodily restrictions in order to obtain results that humans can understand.

Creation, Chance and Emergence

In the discourse about creativity in machines, improvisation might add interesting points, because it is a form of creative processing that can do without complex planning, defining rules and environments that function on the base of simple rules but still generate complex, new and even artistic outputs. Johnson-Laird, being both a scientist and a jazz musician, was the first to apply this logic to improvisation (Johnson-Laird, 1989). According to him, only three types of algorithms can truly facilitate improvisation: neo-Darwinian, neo-Lamarckian, and a hybrid of the two. A neo-Darwinian algorithm generates a new piece by *randomly* blending different pieces together and thus generating a *large number of options*. Selection is then left to the environment, which, in the case of collective improvisation, are basically the other improvisers. If an idea is picked up by them, it will evolve, otherwise it will disappear. In a neo-Lamarckian algorithm, on the other hand, a new piece is derived from some relevant domain knowledge, which leads to more elaborated improvised outcomes that hardly require selection, because this is already in the decision circle within the individual improviser. The neo-Darwinian model applies a bottom-up concept, while the neo-Lamarckian model uses a top-down model. The third type, a hybrid, combines both top-down and bottom-up processes. Johnson-Laird noted that, even though the neo-Darwinian approach is fated to produce a substantial amount of unwanted pieces due to the randomness in the production process, it might be the only way one can improvise when no expert knowledge is available. On the other hand, the neo-Lamarckian approach produces a new piece by drawing on expert knowledge available to guide the production.

A related concept is the theory of social emergence, elaborated and applied to improvisation by Keith Sawyer (Sawyer, 2003). In this concept, the new material emerges through complex, dynamic interaction of elements on a microlevel of the improvising system. This material is unpredictable in detail, but has a certain probability to emerge within an 'interactional frame', that contains of material that has emerged before and has been marked as part of the shared world by being accepted by the fellow players. Emerging material thus will be confirmed and validated by the system – the improvising group. Only when somebody has said 'Yes', will the material become part of the frame for the following improvisation. The interactional frame provides a code for the selection of emerging options *and is emergent itself*. Sawyer's model implies both top-do and bottom-up relations.

I suggest summarizing these concepts under the term 'quasi-evolutionary models for improvisation'. Shimon incorporates both neo-Lamarckian and neo-Darwinian aspects and presumably improvisation needs both the uncensored production of random material and the elaboration of options with a high degree of domain knowledge. The neo-Lamarckian concept can apply for the phases of game-building: Domain knowledge, the specific language of improvisation and the rules of improvisation can be implemented in a robot (or taught to a human improviser). For the phases of game breaking the neo-Darwinian concept is more apt, since it produces surprising, unpredictable material. Quasi-evolutionary models focus on two processes: 1. How does new material appear (mutation)? 2. How is the selection organized (selection)?

1. How does new material emerge (mutation)?

As the works of Magerko et.al, Endo and Weinberg et al. suggest, there are several ways for a machine to generate new, surprising material:

Random generator: A computer can introduce unpredictable material by using a random generator or a pseudo random generator. In this aspect a computer is much more capable than a human, who depends on the associative structures of the brain, which are not quite unpredictable. Humans can generate divergent ideas, but this is not the same as random generation. A computer is free in selecting far-out ideas with no connection to previous input.

Reduce to abstract: Computers can transform the input/output by transcoding it into a more abstract form, like vectors or some kind of simulated somatic markers, which can resemble intuition, emotion, gesture. These structures imitate and strongly resemble neuronal structures of predictive processing.

Decay: Another form of coming up with new material is the concept of "decaying", which means that mistakes are introduced within the system by using fuzziness in pattern detection, option generation and/ or option selection. Through iterative circles of re-introducing mistakes, new material will emerge.

Emergence: In literature one can find another mechanism for generating new material, that draws from the theory of complex dynamical systems. In such systems, new material emerges from unpredictable events on a microlevel amplified through complex dynamics that lead to emergent phenomena on a macrolevel (Sawyer, 2003) (Borgo & Goguen, 2007) (Lösel, 2013).

The fine line between chance and emergence could make the difference between behaviour that simply *mimicks* improvisation and truly improvised behaviour. While this concept is very convincing, there is no actual application in digital agents or robots yet. In order to produce emergent phenomena computers would have to become *complex systems* and at the moment nobody seems to be able to tell, when this will be the case. Or if it is even possible at all.

2. How is the selection organized (selection)?

The selection of options is a crucial point. In a neo-Lamarckian model the selection is appointed to the individual, who, drawing on domain knowledge, will come up with some reasonably good output as outlined above. In a neo-Darwinian model, the selection is left to the environment of collaborating improvisers. For this, the environment has to have certain qualities to mark selected options and provide their survival, while eliminating options that were not selected. On the environmental level both the computer and the human partner must be able to confirm aspects of the partner's output, thus generating some common ground (or an interactional frame, or a circle of expectations) and highlighting the aspects selected in order to build up a response. So the improvising computer needs mechanisms to perceive, select and confirm options provided by the human partner. It is important to note that there is no universal code for the selection of options, instead it varies with the specific system or environment with its own emerging codes. In other words: What survives in one improvisation might not survive in the next. The improviser has to find out every single time. Unpredictability is a desired feature of the improvisational environment and, according to the theory of dynamical systems, it can be reached through high complexity. The ideal improvisational environment will never be completely predictable, but it will also never be completely unpredictable. It will oscillate between these poles.

Conclusions

In this article I followed previous attempts to conceptualize and build improvising machines. As sketched out in the beginning I will now follow the reverse path, asking the question of what improvising automata might tell us about human improvisation. First of all there is the simple observation that music is leading both in critical research on improvisation and in constructing mechanical or digital improvisers (Lewis, 2016). Dance and theater, though having comparable artistic traditions, either are harder to translate into algorithms – or simply have less closeness to academic research or technical thinking. Music certainly has some overlaps with mathematics, which is much less the case in other disciplines.

This raises the question if the cognitive processes of improvisation are more or less the same over the disciplines. Is there a universal model of improvisational cognition underlying all forms of improvisation? This paper strives to describe an 'improvisational mode of cognition', which differs from everyday cognition in specific ways:

1. The habitual evaluation of perceived input is turned off. Instead, improvisational cognition seems to apply a form of very subtle pattern detection, sometimes refered to as 'listening for patterns' or 'listening to games'. Episodic memory might not be needed for this.

2. Anticipation and prediction of the 'circle of expectations' might be crucial within the improvisational mode of cognition. Current research suggests that this is an automatic function of the human brain (Clark, 2016). It enables the improvisers to act without having comprehensive information.

3. There is a continuous effort to find common ground in communication. This does not necessarily mean to build up shared mental models of the world. Instead, marking the other's input as accepted will lead to the emergence of frames, games, codes and fictional worlds. Simply copying and slightly varying the partner's input will lead to surprising results, *if the environment has the right feedback qualities*.

4. Improvisational cognition will always introduce something new to the interaction. This does not necessarily imply creativity, since impulses can go through a quasi-evolutionary process, that will generate new, unpredictable material through emergence, or decay, or selection, or chance.

5. It seems useful to distinguish between cognition in the phase of game-building and cognition in the phase of game-breaking. This confirms an early cognitive model introduced by Jeffrey Pressing in 1984 (Pressing, 1984).

6. All in all a cognitive theory of improvisation has to integrate an environmental view, following concepts of embodied cognition and social interaction. The improvisational environment will encourage 'mistakes' and will lead to an emerging code for the selection of contributions. Only within this environment will improvisational cognition lead to meaningful results.

When I started this exploration I was secretly hoping to prove that improvisation is reserved to human beings. I tended to think of improvisation as being the peak of humanism, impossible to master for machines. But studying improvising automata that already exist, leads to a different conclusion, suggesting that the problem of machine creativity appears much smaller, when we demystify creativity in the light of improvisation: Improvisation is an artistic process, that does not need an artist. Improvisation can do without authorship, without intentionality and, in some sense, even without creativity. It does not rely on clever or original ideas, but on the contrary trusts in the emergence of new material from very small units of interaction. While maybe not being creative themselves, machines already seem to be quite capable in inspiring human partners to be creative, so they are already successfully part of the improvisational environment. Instead of thinking of improvisation as almost impossible for computers I now think that it might be one of the first areas where an artistic communication between humans and autonomous digital agents will be possible. And machines will learn a lot in this communication.

References

Agre, P. (1988). *The Dynamic Structure of Everyday Life*.

Benson, B. E. (2003). *The improvisation of musical dialogue : a phenomenology of music*. Cambridge : Cambridge University Press. http://doi.org/10.1017/CBO9780511615924

Borgo, D., & Goguen, J. (2007). Rivers of consciousness: The nonlinear dynamics of free jazz. *Journal of the International Association for Jazz Education*, 25, 1–14.

Clark, A. (2016). *Surfing uncertainty : prediction, action, and the embodied mind*. New York : Oxford University Press.

Craig, E. G. (1911). The Art of Theatre. London: Browners Bookstore Chicago.

Drinko, C. (2013). *Theatrical Improvisation, Consciousness, and Cognition*. New York: Palgrave Macmillan.

Endo, Y. (2008). Countering Murphy's Law: The use of anticipation and improvisation via an episodic memory in support of intelligent robot behavior. Dissertation, Georgia Tech University

Halpern, C., Close, D., & Johnson, K. (1994a). *Truth in Comedy – the Manual for Improvisation*. Colorado Springs: Meriwether Pub Ltd.

Hoffman, G., & Weinberg, G. (2011). Interactive improvisation with a robotic marimba player. *Autonomous Robots*, 31(2–3), 133–153. http://doi.org/10.1007/s10514-011-9237-0

Husserl, E. (1964). *Phenomenology of Internal Time-Consciousness*. Inciana: Indiana University Press.

Johnson-Laird, P. N. (1989). *The computer and the mind : an introduction to cognitive science*. London : Fontana Press.

Johnstone, K. (1987). *Impro - Improvisation and the Theatre*. New York: Routledge.

Lewis, G. (2016). *The Oxford handbook of critical improvisation studies. Critical improvisation studies*. 2 Vols. New York, NY, United States of America : Oxford University Press.

Lewis, G. E. (1999). Interacting with latter-day musical automata. *Contemporary Music Review*, *18*(3), 99–112. http://doi.org/10.1080/07494469900640381

Lösel, G. (2013). *Das Spiel mit dem Chaos - Zur Performativität des Improvisationstheaters*. Bielefeld: Transcript Verlag.

Magerko, B., Manzoul, W., Riedl, M., Baumer, A., Fuller, D., Luther, K., & Pearce, C. (2009). An Empirical Study of Cognition and Theatrical Improvisation. *Seventh ACM Conference on Creativity and Cognition*, (c), 117–126. http://doi.org/doi: 10.1145/1640233.1640253

Newell, A. (1990). *Unified theories of cognition* (Vol. 1987). Cambridge, Massachusetts [etc.] : Harvard University Press.

Pressing, J. (1984). Cognitive processes in improvisation. In *W.R. Crozier & A.J. Chapman (Eds.), Cognitive Processes in the Perception of Art*. Elsevier.

Salinsky, T., & Frances-White, D. (2008). *Improv Handbook: The ultimate Guide to Improvising in Theatre, Comedy and Beyond*. Continuum.
Sawyer, R. K. (2003). *Improvised Dialogues: Emergence and Creativity in Conversation*. Westport: Ablex Publishing.
Spolin, V. (1977). *Improvisation for the Theater*. London: Pitman Publishing.
Weinberg, G., Godfrey, M., Rae, A., & Rhodes, J. (2007). A Real-Time Genetic Algorithm in Human-Robot Music Improvisation. In *International Music Conference, Copenhagen* (pp. 192–195). Copenhagen.

List of Figures

List of Tables