Open access • Book Chapter • DOI:10.1007/978-3-642-33093-3_18

Towards Behavioral Consistency in Neuroevolution — Source link

Charles Ollion, Stéphane Doncieux

Institutions: International Society for Intelligence Research

Published on: 27 Aug 2012 - Simulation of Adaptive Behavior

Topics: Neuroevolution, Fitness function, Animat and Consistency (negotiation)

Towards behavioral consistency in neuroevolution

Charles Ollion, Stéphane Doncieux

Towards Behavioral Consistency in Neuroevolution

Charles Ollion, Stéphane Doncieux

ISIR, UPMC

Abstract. To survive in its environment, an animat must have a behavior that is not too disturbed by noise or any other distractor. Its behavior is supposed to be relatively unchanged when tested on similar situations. Evolving controllers that are robust and generalize well over similar contexts remains a challenge for several reasons. One of them comes from the evaluation: how to check a controller for such properties? The fitness may evaluate a distance towards a behavior known to be robust, but such an example is not always available. An alternative is to test the behavior in multiple conditions, actually as many as possible, to avoid overfitting, but this significantly slows down the search process. This issue is expected to become even more critical when evolving behaviors of increasing complexity. To tackle this issue, we propose to formulate it as a problem of behavioral consistency in different contexts. We then propose a fitness objective aimed at explicitly rewarding behavioral consistency. Its principle is to define different sets of contexts and compare the evolved system behavior on each of them. The fitness function thus defined aims at rewarding individuals that exhibit the expected consistency. We apply it to the evolution of two simple computational neuroscience models.

1 Introduction

While the use of evolutionary robotics has generated encouraging results [1], evolving neural networks for complex tasks still faces evolvability issues [2]. What gradient a fitness function should create? What should a fitness function reward? As of today, no straightforward methodology may help in designing a fitness function in a context of neuro-evolution. Several classifications have been made to take into account the features of a fitness function when comparing results. Floreano and Urzelai [3] thus proposed the *fitness space*, a framework for describing fitness functions in order to qualitatively compare them. Nelson et al [1] have made a review of the fitness functions used in the context of evolutionary robotics and classified them according to the degree of a priori knowledge that they include. If such work review the fitness functions used up to now, they are not aimed at proposing a method for fitness design. This question may be, at least for a part, problem specific, but we hypothesize here that there is a selection pressure that may be common to a lot of neuro-evolution problems and we propose a simple method to design it.

Neuro-evolution methods aim at generating neural networks that will exhibit a particular behavior in response to the inputs they receive. In the following, we focus on neural networks aimed at controlling the behavior of an animat (or subparts of such a controller). Our hypothesis is that a specific feature is explicitly sought when evolving neural networks in this case: the consistency of the neural network behavior in different contexts. A context will be refered as a particular setting, or environment, in which the animat is. Even if the exact behavior to be generated is not known, the experimenter may know in which contexts he expects a similar behavior and in which contexts he expects different behaviors. An animat moving in its environment should not be too sensitive to noise or other distractors and should behave the same in similar contexts. When designing a neural network with evolutionary methods, these properties are not guaranteed and it is common to observe neural networks that behave well only on the contexts used for evaluation, leading to a lack of generalization [4]. Ideally, such a consistency should emerge from the search process without being explicitly rewarded. However it would require to evaluate the behavior of the network in many different contexts so that this property has an impact on the fitness function —even in this case, the consistency is not guaranteed. We propose a method to design fitness functions that explicitly reward the consistency of neural network behavior. The approach consists in testing the network behavior in different contexts. The fitness function evaluates then how close or how different behaviors are, depending on what the experimenter expects.

Two models from computational neuroscience have been considered to test the approach: an attention selection model and an action selection model. Both represent functions that are expected to be useful for an animat to survive in its environment. The motivation behind this choice is the knowledge of one particular and efficient behavior from the litterature. Results generated with the consistency objective are thus compared to results generated with a more classic fitness function that rewards individuals close to this particular behavior. While requiring a less accurate knowledge, the consistency objective revealed to generate efficient solutions, some of them being original and different from the known behavior; all respecting the constraints.

2 Method

The consistency objective is based on an evaluation of behavioral consistency in different contexts. It relies on simulating one individual on a set S of different contexts, as shown on Figure 1. The consistency of the individual is then evaluated by comparing the behaviors (outputs). We denote $o_i(t)$ the simulated output of context i after t time-steps. Depending on the problem considered, the experimenter defines different contexts and different consistency constraints between contexts. Three possible constraints will be used in the following experiments:

- output of context i is exactly the same as output of context j: $o_i(t) = o_j(t)$
- output of context i is different from output of context j: $o_i(t) \neq o_j(t)$

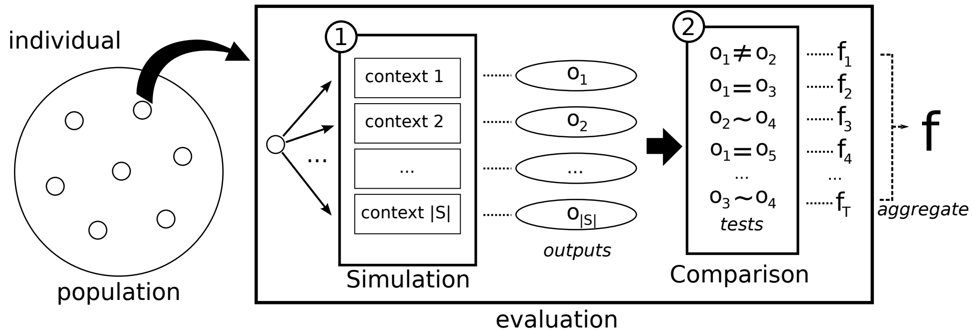


Fig. 1. One individual is evaluated in three steps: 1) simulation over a set of predefined contexts. 2) test of user-defined constraints satisfaction. The level of satisfaction of the constraint is f_i. 3) Aggregation of constraints into the fitness of the individual f.

- output of context i has similar properties as context j: $o_i(t) \sim o_j(t)$. The definition of similarity is specific to each experiment, but typically means equivalent to within a translation.

The constraints are computed in the following way:
- $o_i(t) \neq o_j(t)$: $f_t = d(o_i(t), o_j(t))$
- $o_i(t) = o_j(t)$: $f_t = 1 - d(o_i(t), o_j(t))$
- $o_i(t) \sim o_j(t)$: $f_t = 1 - d_s(o_i(t), o_j(t))$

The normalized distance d denotes the difference between outputs behaviors. d_s is a similarity distance that usually describes how close the shapes of outputs are. d and d_s are specific to the experimental setup.

The final assessment of the quality of an individual is then computed by aggregating the fitness terms. For the sake of simplicity, the simplest aggregation is used:

$$f(x) = \frac{1}{T} \sum_{t}^{T} f_t \qquad (1)$$

In short, building the consistency fitness objective involves three steps:

- Defining a collection of different contexts
- Defining constraints between outputs of contexts
- Defining how to compare outputs with distances d and d_s

It should be noticed here that although some knowledge is required, the exact behavior does not need to be known.

3 Attention Selection

3.1 Experimental Setup

The first experiment tackled here is inspired by the work of Quinton [5]. The goal is to use Continuous Neural Field Theory [6] to model attention selection.

The model is taken from [7] in which the neural field is known to output a robust neural activity able to track perceptual information in noisy contexts, even in the presence of distractors. In Quinton's work, the parameters of the neural field are optimized using an evolutionary algorithm.

The neural field is represented by a two dimensional map, and the potential at position vector $\mathbf{x}$ and time t is $u(\mathbf{x}, t)$, with $\mathbf{x}$ in $[-0.5, 0.5]^2$. The field is stimulated by perceptual input $s(\mathbf{x}, t)$, and lateral connections. The dynamics of the neural field follows the equation taken from [5]:

$$\tau \frac{\partial u(\mathbf{x}, t)}{\partial t} = -u(\mathbf{x}, t) + \int_{\mathbf{x}'} u(\mathbf{x}, t) w(\mathbf{x}, \mathbf{x}') d\mathbf{x} + s(\mathbf{x}, t) + h \tag{2}$$

h is the resting potential, set to 0 at first. The lateral connection weights follow the equation:

$$w(\mathbf{x}, \mathbf{x}') = Ae^{-\frac{|\mathbf{x}-\mathbf{x}'|^2}{a^2}} - Be^{-\frac{|\mathbf{x}-\mathbf{x}'|^2}{b^2}} \tag{3}$$

The lateral connection parameters A, B, a and b are evolved, as well as τ the inertial parameter of the dynamics. As in Quinton's work, the parameters are constrained in order to obtain a “mexican hat” like lateral connectivity: $A > B$ and $b > a$. In order to simulate the dynamics, the map is discretized onto a grid of 50×50 units. The dynamics of the neural fields are simulated on different contexts, in which the inputs of the neural field $s(\mathbf{x}, t)$ differ, as depicted in figure 2. The first and second contexts A and B correspond to constant inputs.

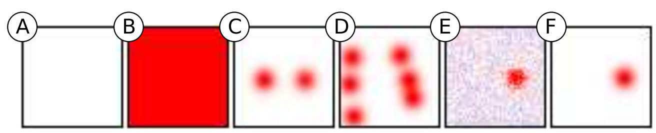
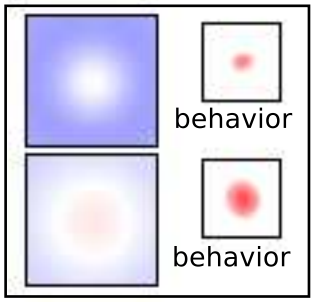


Fig. 2. Typical inputs (contexts) from left to right: A: empty B: full C: competition D: distractors E: noise F: simple

C, D, E are 3 contexts which correspond to Quinton's scenarios. In context C, a static input bubble is in competition with a second one of variable intensity. In contexts D and E, a rotating input bubble must be tracked, in presence of distractors (D) or noise (E). Finally, context F has a single static input bubble and no distractors. The details of those contexts can be found in [5].

Control experiment The fitness used in the original experiment is used as a control fitness. It is based on the three contexts C, D and E, and is described in [5]. It assumes the knowledge of the position and shape of the output at each time-step.

Context-based experiment In order to measure the bias added by the choice of contexts, various contexts are used to compute the fitness. The fitness includes



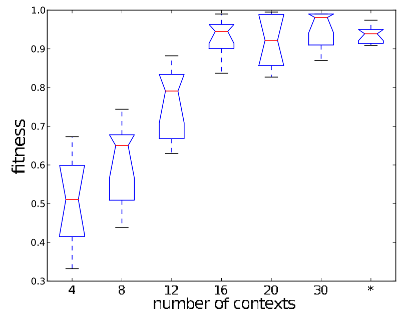


Fig. 3. (Left) Two resulting behaviors obtained with the context-based fitness. Left is the lateral weight profile and right the corresponding output activity. (Right) Influence of the number of contexts on the context-based fitness with 30 contexts. $*$ represents parameters obtained with the fitness of the original experiment.

up to $|S| = 30$ contexts. The first two correspond to contexts with a totally activated/deactivated neural field. The others are chosen randomly among C, D, E, or F —each run— to reduce any bias in the result statistics. In contexts C, D, E, or F, the initial position of the input bubble, the number of distractors or the noise level can vary. Four sets of contexts are defined in the following way: the first two contexts are in S_{const}, all others in S_{nconst}. Additional contexts derived from D and E have their input moving, therefore they should not converge towards a fixed output and are then in S_{nconv}. On the other hand, contexts derived from C and F are in S_{conv}. From those sets, the following constraints are defined:

- The output behavior of contexts from S_{nconst} should be different from constant behavior of S_{const} contexts: $\forall i \in S_{nconst}, j \in S_{const},\ o_i(t) \neq o_j(t)$
- The output of S_{nconst} should exhibit consistency in time: $\forall i \in S_{nconst}, \exists \epsilon_t\, \forall t, t' > \epsilon_t\; o_i(t) \sim o_i(t')$
- The output of S_{conv} should stabilize: $\forall i \in S_{conv}, \exists \epsilon_t\, \forall t, t' > \epsilon_t\; o_i(t) = o_i(t')$
- The output of S_{nconv} should be different over time: $\forall i \in S_{nconv}, \forall t > \epsilon_t, \exists t' > t, o_i(t) \neq o_i(t')$
- The outputs of different contexts of S_{nconst} should be similar: $\forall i, j \in S_{nconst},\ o_i(t) \sim o_j(t)$

Finally, the distance between behaviors is computed in the following way:

$$d(o_i(t), o_j(t)) = \int_{\mathbf{x}} ||o_i(\mathbf{x}, t) - o_j(\mathbf{x}, t)|| \tag{4}$$

The similarity distance computes the same distance, after aligning the outputs:

$$d_s(o_i(t), o_j(t)) = \min_{d} \int_{\mathbf{x},\mathbf{x}'} ||o_i(\mathbf{x}, t) - o_j(\mathbf{x} - \mathbf{d}, t)|| \tag{5}$$

d corresponds to the shift between the center of activities between the two outputs. Details of this computation can be found in [7]

Each individual is evaluated during 20 time-steps on each context. In order to avoid initial chaotic dynamics, the first 5 time-steps of the resulting behavior are discarded. Each experiment —with various numbers of contexts— is run 10 times, with a population size of 20 and the number of generations is 30 (same values as in Quinton's experiment). The evolutionary algorithm used is NSGA-II [8].

3.2 Results

The consistency objective with maximum number of contexts provides results with the expected qualitative behavior, in all 10 runs. This means that the output signal (a bubble of activity) is quickly able to follow the input with a signal consistent in time and shape, and is not damaged by noise or distractors. The evolved systems exhibit then the expected function. Furthermore, different behaviors have been discovered, as shown in Figure 3, left. This highlights an interesting property of the proposed method: it looks for behaviors respecting the given constraints, no matter how they manage to do it.

The number of contexts used greatly influences the effectiveness of the consistency objective. In order to study this influence, the individuals obtained with k contexts are tested on the fitness based on 30 contexts. Additionally, an individual obtained with Quinton's fitness was tested on the 30-context-based fitness. The graph (Figure 3 right) shows that:

- not surprisingly, individuals obtained with Quinton's fitness perform well on the context objective. It should be underlined here that the reverse is generally not true as Quinton's fitness looks for a particular shape that is not the unique solution respecting the constraints as mentioned above;
- the number of contexts needs to be high enough for the objective to be effective (around 16), but there is little difference in performance if the number of contexts is over 16 ($p > 0.1$). The difference between 4, 8, 12 and 16 contexts is significant ($p < 0.01$ for each).

4 Action Selection

This section describes the second experiment, based on a neuroevolution experiment on basal ganglia [9]. The goal of this experiment is to evolve a neural network able to perform action selection, a cognitive function supposed to be performed by the basal ganglia. The search space is much larger than in the previous experiment, as the structure and parameters of the neural network are evolved. Action selection in the brain is the problem of choosing an action, given external and internal sensory information. The focus is on the process of selection of a single action among conflicting ones, also known as a winner-takes-all (WTA) circuit. This section first describes the original experiment and fitness, and then the definition of contexts to build the new fitness function.

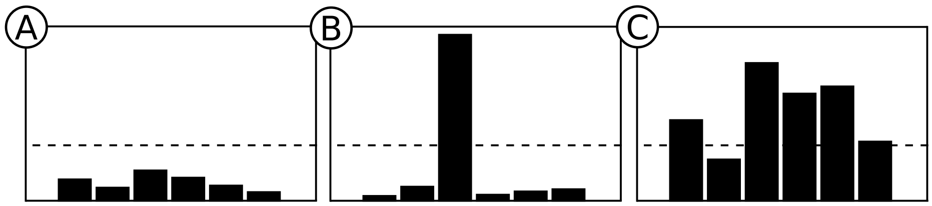


Fig. 4. Typical contexts from left to right (Input of the action selection model): A: empty or low noise B: easy selection C: competition

4.1 Experimental setup

Encoding In order to build an action selection model that generalizes to different number of action channels, Mouret et al. used a map-based encoding [9] in which parameters and structure of a neural network are evolved. The neural network is initially a feed-forward network, and the structure is modified through evolution by the addition of new neurons or maps of N neurons, and connections between them. Furthermore, connections between maps of neurons have additional evolved parameters that determine the nature of a connection: a 1-1 connection scheme connects each neuron of the input map to a single one in the output map, while a 1-all connection scheme connects one neuron to all neurons in the output map.

This experiment uses this map based encoding, with a modification: the connections between maps have an additional evolved parameter: an offset o. For instance, a 1-1 connection between two maps of size N with offset o connects neurons as follows: neuron i of the input map is connected to neuron $i + o \, modulo \, N$ of the output map.

Control experiment The authors of the original experiment expect to build a model that reproduces the functioning of basal ganglia in the brain. The fitness function assumes full knowledge of the output and is described in [10]. At rest the output of the basal ganglia is active, and represents inhibition to the target. The selected action will then be the channel where inhibition is removed. The neural networks have one input map and one output map of N neurons, N corresponding to the number of channels. Over a collection of $K = 500$ random inputs (inputs range from 0 to 1), the most activated input neuron has to be selected, which means that the corresponding output should be close to zero while others should be close to one. It rewards individuals that desinhibit the selected channel and inhibit all others.

As this setup is more challenging than the previous one due to the complex search space, a behavioral diversity helper objective is added, in a multi-objective scheme [9].

Consistency Objective The goal of the consistency objective is to evolve networks that perform action selection in any possible way, not only in a biologically plausible way.

Three types of contexts are displayed in Figure 4. Context type A corresponds to a very weak noise input. Context B corresponds to a very simple action selection: one channel is set to 1.0 while the others are set to 0 plus a uniform weak noise. C represents randomly generated contexts with random inputs.

Like in the previous experiment, we define 2 sets of contexts in the following way: S_{const} for context A, S_{nconst} for all others.

- All the outputs should stabilize: $\forall i \in S, \exists \epsilon_t \, \forall t, t' > \epsilon_t \, o_i(t) = o_i(t')$
- The output behavior of contexts from S_{nconst} should be different from constant behavior of S_{const} contexts: $\forall i \in S_{nconst}, j \in S_{const}, \; o_i(t) \neq o_j(t)$
- The outputs of two different contexts of S_{nconst} should have similar behaviors: $\forall i, j \in S_{nconst}, \; o_i(t) \sim o_j(t)$
- The outputs of two different contexts in S_{nconst} which select the same channel should be exactly the same: $\forall i, j \in S_{nconst}, max_i = max_j \Rightarrow \; o_i(t) = o_j(t)$
- The outputs of two different contexts in S_{nconst} which select different channels should be different: $\forall i, j \in S_{nconst}, max_i \neq max_j \Rightarrow \; o_i(t) \neq o_j(t)$

Finally the distances to compute difference and similarity are defined as follows:

$$\begin{aligned} d(o_i(t), o_j(t)) \quad &= \textstyle\sum_k ||o_i(k,t) - o_j(k,t)|| \\ d_s(o_i(t), o_j(t)) &= \textstyle\sum_k ||o_i(k,t) - o_j(k-\delta,t)|| \end{aligned}$$

where δ is the shift between centers of activity of the two outputs, computed in a similar fashion as in the first experiment.

The population size is 200 and the algorithm stops after 1000 generations. NSGA-II algorithm is used [8], and the source code is available online at `http://pages.isir.upmc.fr/evorob_db`

4.2 Results

Concerning action selection, the consistency objective was able to evolve successfully two main categories of solutions. Both of them realize action selection by outputting a coherent response for any output, and making a single output stand out from others. The first one (Figure 5, top), realizes the most intuitive action selection, and its corresponding neural network obtained has two internal maps of neurons and excitatory recurrent connections. The second category (Figure 5, bottom) of behaviors are similar to the Mouret et al. results, except for a possible shift in the output channel index. The corresponding neural network has two internal maps, but more connections, both excitatory and inhibitory.

The number of contexts have exactly the same influence as in the previous experiment: with a low number of contexts, the performance is not reliable, while with at least 15 contexts, the performance stabilizes. The difference between 15

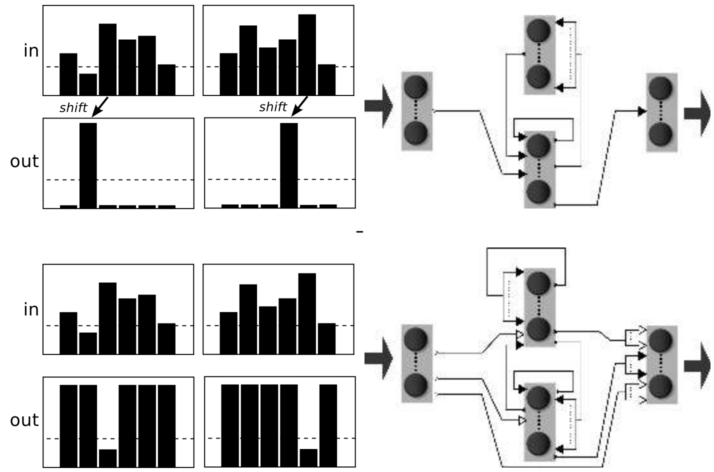


Fig. 5. (Left) 2 behaviors generated with the consistency objective. The graphs show a random input (in), and the output of the evolved model (out). Consistency objective fitness: 0.94 (top) and 0.96 (bottom). **(Right)** corresponding neural networks. Parameters and offsets are not displayed.

and 30 contexts is not statistically significant (p-value > 0.1 Mann-Withney U test). The solutions obtained with the original fitness perform very well when evaluated with consistency objective.

5 Conclusion

Properties such as cognitive abilities, robustness to noise, or generalization rarely emerge with fitness functions that only reward the completion of a task. Many evaluations over a lot of different contexts are required for those properties to emerge, and there is no guarantee that such properties actually emerge.

The consistency method can be used to explicitly drive the evolutionary search to the emergence of such properties. While it requires *a priori* knowledge on the expected property, a successful run ensures the emergence of this property, with a limited number of evaluations. The method is shown to successfully build two properties: attention selection and action selection.

Furthermore, the consistency method does not drive the evolutionary search to an explicit behavior. This means that the exact knowledge of a behavior presenting the desired property is not required. In addition, this does not restrain the evolutionary search to one particular solution, leading to the emergence of the property in many different ways.

It is important to note that even if the knowledge of a behavior is not required, the experimenter includes knowledge in building the different contexts and comparison between contexts. While the design of these contexts is straightforward in simple cases, one can expect more challenges for difficult properties to emerge.

The method is based on selection pressure rather than encoding, thus it could potentially be applied to any evolutionary algorithm. Future works include the application of the Consistency objective to the evolution of more complex computational neuroscience models. Furthermore, it could be used as a helper objective in a multi-objective scheme, alongside other objectives, such as a goal oriented objective, behavioral diversity or novelty objectives.

5.1 Acknowledgments

This work was funded by the ANR EvoNeuro project, ANR-09-EMER-005-01.

References

1. Nelson, A.L., Barlow, G.J., Doitsidis, L.: Fitness functions in evolutionary robotics: A survey and analysis. Robotics and Autonomous Systems **57**(4) (April 2009)
2. Blynel, J., Floreano, D.: Exploring the T-maze: Evolving learning-like robot behaviors using CTRNNs. Applications of Evolutionary Computing (2003) 593–604
3. Floreano, D., Urzelai, J.: Evolutionary robots with on-line self-organization and behavioral fitness. Neural Networks **13**(4) (2000) 431–443
4. Pinville, T., Koos, S., Mouret, J.B., Doncieux, S.: How to Promote Generalisation in Evolutionary Robotics : the ProGAb Approach Formalising the Generalisation Ability. In: GECCO '11: Proceedings of the 13 th annual conference on Genetic and Evolutionary Computation. (2011) 259–266
5. Quinton, J.c.: Exploring and Optimizing Dynamic Neural Fields Parameters Using Genetic Algorithms. Statistics (2010) 0–6
6. Amari, S.i.: Dynamics of pattern formation in lateral-inhibition type neural fields. Biological Cybernetics **87** (1977) 77–87
7. Rougier, N.P., Vitay, J.: Emergence of attention within a neural population. the official journal of the International Neural Network Society **19**(5) (June 2006)
8. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Parallel Problem Solving from Nature PPSN VI, Springer (2000) 849–858
9. Mouret, J.B., Doncieux, S., Girard, B.: Importing the computational neuroscience toolbox into neuro-evolution—application to basal ganglia. In: Proceedings of GECCO'10. (2010) 587–594
10. Liénard, J., Guillot, A., Girard, B.: Multi-objective evolutionary algorithms to investigate neurocomputational issues : The case study of basal ganglia models. In Doncieux, S., Girard, B., Guillot, A., Hallam, J., Meyer, J.A., Mouret, J.B., eds.: From animals to animats 11. Volume 6226 of LNAI., Springer (2010) 597–606