2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems

Designing comprehensible self-organising systems

Nicolas Höning
Centrum Wiskunde & Informatica (CWI)
Amsterdam, The Netherlands
Email: nicolas.honing@cwi.nl

Han La Poutré
Centrum Wiskunde & Informatica (CWI) and Utrecht University
Amsterdam/Utrecht, The Netherlands
Email: han.la.poutre@cwi.nl

***Abstract*—Self-organising systems are a popular engineering concept for designing decentralised autonomic computing systems. They are able to find solutions in complex and versatile problem domains, but as they capture more complexity in their own design, they are becoming less and less comprehensible to their users (be they humans or intelligent agents). We describe a design challenge that relates to usability theory in general and in particular resembles an observation made by Phoebe Senger, who noted that software agents tend to become incomprehensible in their behaviour as they grow more complex. In the manifestation of self-organising systems, the problem is more urgent (since we find ourselves using them more and more) and harder to solve at the same time (since these systems are not centrally controlled). We describe the problem domain and propose three system properties that could be used as quality indicators in this regard: Stability, Learnability and Engageability. We demonstrate their usage in a simple model of dynamic pricing markets (e.g. the electricity domain) and evaluate them in different ways.**

***Keywords*-Artificial intelligence, Self-organizing control, System analysis and design, User centered design, Optimization methods**

I. Introduction

When computer systems become more complex, their shortcomings in usability become clearer to their users. Phoebe Sengers [1] pointed this tendency out when she made clear how artificial agents resemble schizophrenic humans in their behaviour, being “fragmented“ and “incomprehensible“. This has never been stated for self-organising systems, but they exhibit the same problems and we find ourselves using them more and more.

Many contemporary examples of complex, automated computer systems who self-organise in a complex domain can be found in traffic control - for motorways (e.g. Smulders [2] describes the Dutch system of variable speed signs), public transport (e.g. Gershenson and Pineda [3] describe how buses can avoid the platooning effect if they follow system-internal schedules rather than to wait for all passengers to board) or urban traffic (e.g. Dusparic et al [4] model the interplay of inner city control mechanisms which optimise for private and public traffic in parallel). In Economics, complex trading systems have become reality and more are planned, for instance in deregulated electricity markets [5]. In addition, many models exist for applications in security/surveillance [6], e-health or e-government [7]. We note that many of these systems are hybrids in that they employ both artificial and human agents. We also make the same assumption about (future) users: automated systems should be comprehensible to external automated agents.

Many researchers seem to jump from the fact that self-organising systems support emergent behaviour to the conclusion that they are inherently easier to interact with, since they can adjust themselves to many circumstances. On the contrary, properties brought about by complex behaviour are hiding the systems inner states by abstraction, thus making the interaction problem even harder to solve than for single agents. To our best knowledge, this problem has not yet been described in the literature.

While similar shortcomings have been identified by other authors (e.g. [4], [8], [9], [10], [11]), our modelling of this problem reflects systems which do not lend themselves to semantic or game theoretic descriptions of all interactions. We take a signal processing approach, where we are interested in the interpretation of the systems output by its users.

Our approach considers similar settings as Decision Theory, being concerned with the question how better decision making can be enabled for actors in complex, uncertain and inter-temporal environments. We are not interested in usability concepts tailored specifically to humans (e.g. recent HCI approaches [12]), but rather use simple statistic analyses which could be performed by either humans or software. In a similar approach, Holzer et al [13] model self-organising systems as a multigraph and measure two system properties (autonomy and emergence) via the entropy over signals on the graph edges. We do not require a complete system model, as we are only interested in the edge between the system and a user.

In system design, current research focuses almost exclusively on internal performance objectives or the possible utilities achievable for users, each on its own. We propose comprehensibility as a third perspective, asking to what extent a system is dependable, learnable and engageable. A comprehensible system reduces the overhead costs that arise by using it and makes easier introduction and deployment possible.

The contributions of this work are to motivate why designing comprehensible self-organising systems is an especially hard task in the online, multi-user case, which we

978-0-7695-4232-4/10 $26.00 © 2010 IEEE
DOI 10.1109/SASO.2010.18

do in Section II, and to propose three system properties (as perceived by its users) that could be used as quality indicators for comprehensibility: Stability, Learnability and Engageability, which we review in Section III. In section IV, we model a simple example system to demonstrate how these indicators could be modelled and evaluated in practice. Section V concludes and Section VI discusses future work.

II. Online, multi-user, self-organising systems

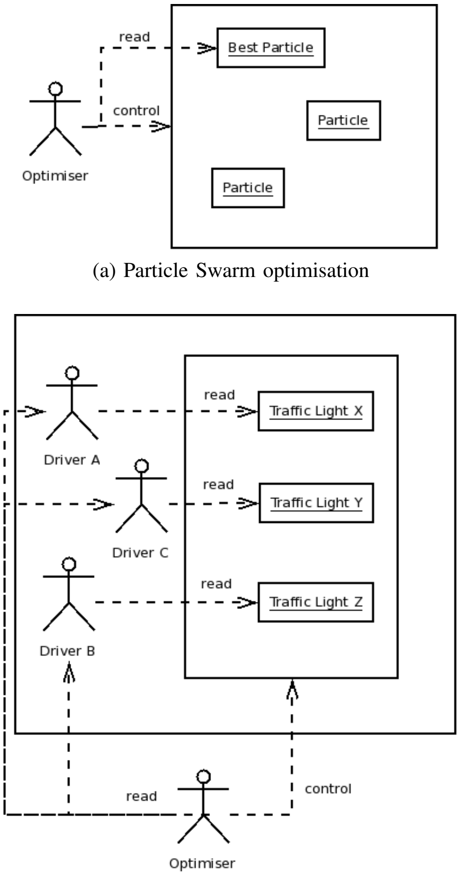
A. Definition

In this work, we treat self-organisation as an engineering concept, which fits well into the domain of autonomic computing [14]. Here, self-organisation can be seen as a process which enables a system to modify its organisation in case of changes in the environment, without explicit external commands. Instead, agents act inside of the system considering internal states, rules and constraints [15]. The reorganisation will put the system in a new state. This state can be vastly different from the previous state - the state transition is considered an appropriate response if it brings the system back to some internal performance objective.

In the sense of Von Foerster [16], the environment as an external, contingent actor is crucial to the operation of any self-organising system. Here, we are especially interested in an environment consisting of the systems users and in their ability to interact with the system. This relates to research in open systems. De Alfaro [8] defines an open system as "a system whose behavior is jointly determined by its internal structure, and by the input it receives from the environment."

The self-organising systems that are proposed in most of the literature behave to their environment much like distributed systems in that they abstract over their internal states, hiding them from their users. A distributed system, by design, provides its users a system view on itself which abstracts over several aspects of its distribution (e.g. access, location or failure)[17]. Thus, the user does not need to care which parts of the system he is interacting with - to him, the system is perceived as one single opponent.

We consider here self-organising systems that have *multiple users*, who are interested in states of the system at different time points (in its *online* behaviour) from distinct point of views. An example of an offline, single-user self-organising system is a Swarm within a Particle Swarm Optimisation [18]. There is only one user of such a system, who is only interested in one state at one time: the best position when the optimisation has stopped. Online multi-user systems, on the other hand, present several states to the outside world. Examples of multi-user systems are traffic regulation systems and markets. Figure 1 illustrates how design ('control') and interests in system states ('read') differ in both cases. In fact, the designer of a multi-user system faces a multi-objective optimisation problem as soon as the objectives of the users do not align with his (optimisation) objectives.



(a) Particle Swarm optimisation

(b) Traffic Control Scenario

Figure 1. Examples for an offline, single-user system and an online multi-user system

In this paper, we demonstrate our ideas with a simple market model. Markets fit our case as they are often described as self-organising systems (e.g. [19], [20]), and consist of multiple autonomous actors, who influence supply and demand via their actions over time [21]. In our model, consumers act as the users, from whose point of view we will evaluate the proposed indicators.

B. Problem Statement

The problem we outline here is rooted in the fact that a multi-user self-organising system is not a single opponent to its users. By definition, its agents act autonomously and not on behalf of a single, central opponent, who would ensure a stable behaviour that would be dependable or learnable. These systems abstract over their internal states, meaning that the causal connection between its outputs and inner states is hidden or too complex to infer. Users will find it hard to comprehend and thus depend on such systems. It has in fact been noted that human users tend to refer to complex self-organising systems in their language as single actors. In [22], Morris et al describe how using the agent metaphor (e.g. "the Nasdaq climbed") over the object metaphor ("the

Nasdaq bounced back") changes the users interpretation of the systems response. For lack of a better clue to the inner workings of the system, users are employing Occam's Razor by rationalising the system's response with a simple, but inaccurate interpretation. This may be a serious source for misunderstandings.

In addition to being dependable and learnable, several multi-user system designs hope for the user's active participation. This can be seen in partly liberalised markets (e.g. electricity markets), where hopes are high to create open frameworks, such that many players are able to access the market and drive it towards efficiency. The success of such an open system might partly depend on its engageability, the extent to which users are able to recognise the effect of their actions in the interaction with the system. To describe unengeaging user-system relations, Luhmann [23] suggests to use the term 'confidence' rather than the term 'trust', because trust "requires a previous engagement on your part". Users in a confidence relation "react to disappointment by external attribution" rather than to acknowledge their own contribution to the situation.

Senger [24] introduces the concept of comprehensible agents, who are not only supporting a function but are also usable in that they are comprehensible to its users. We argue that this concept needs to find its way into the design of self-organising systems, i.e. we are in need of comprehensible systems. This holds especially for systems whose users have little choice but to use them, as is for instance the case with traffic control systems and often for markets.

Any hidden complexity of current self-organising systems can possibly make its outputs and behaviours incomprehensible to its users. The hurdles which a design for comprehensibility would need to overcome lie at the heart of the concept of self-organisation: Those systems are designed to deal with very complex problems, thus their internal states can often not be reduced to simple concepts. Any constraints on state transitions might hinder the system in reacting flexible and its quest for optimising internal performance objectives. On the other hand, abstracting inner states and objectives away from users can often be a problem rather than a welcome feature for them. To feed the users desire to depend on, learn to use and engage with the system, some of these abstractions should be reconsidered, which is a challenging engineering problem.

Relations to Market Theory: There is a well-established discussion about hidden complexity in market theory: The efficient-market hypothesis claims that the prices in a market reflect the true value of products, i.e. all information about them. There is growing evidence against the hypothesis in its strong form (e.g. [25]), which claims that prices reflect even non-public insider information. Returning to our discussion of abstraction above, this would mean that markets abstract over their internal states. Hence, abstraction towards users must be seen as reducing this kind of informative efficiency: Pagano and Roell [26] investigate the effect of abstraction on liquidity and conclude "that if policy makers want to reduce trading costs for uninformed traders, they should publicly disseminate order flow information as promptly as possible". Their model predictions "may also help to understand why lately dealer markets are under increasing pressure from the more transparent automated auction systems."

We stated that designing for both complexity and comprehensibility constitutes a trade-off situation. This is illustrated by the concept of "Dynamic Efficiency" (e.g. [27]), denoting how well a market policy prepares its participants for possible future states, rather than (statically) optimising the present. In this context, Potts [28] argues that some sufficient amount of structural complexity is needed in a system to enable it to arrive at viable solutions which enable both adaptability to a current state and being prepared for change. In our view, complexity alone will hardly be sufficient. We think it is an important research question how (market) systems can contain more complexity while still enabling informed and cost-effective decision making for their users.

III. Quality Indicators

We propose three quality indicators and give some first approaches to measuring them. They all are an interpretation of the systems signals towards the user and use the concept of utility [29] to quantify this interpretation in different states. Thus, we use the term 'received utility' to denote how the systems output is interpreted by the user and the term 'action' to denote the users behaviour towards the system.

We believe that these indicators make a contribution towards more comprehensible self-organising systems, in that they constitute a universal set of interests inherent to most users who are interested in reducing their overhead costs, be they human or automated, which is commonly not captured by objective measures such as price. If they are used in system design, the user utility in these systems can become more comparable, at least the part which is influenced by the systems behaviour.

A. Stability

One aspect of comprehensibility is Stability. We claim here that a system with a low variance in transitions between received utilities can be more reliable to the user and thus the resources needed to use it can be reduced (or be put to use less frequently). Of course, Stability works against dynamicity, scalability and other desirable attributes of self-organising systems. It is hard to achieve Stability in those systems because they react to the dynamics of different influences - the received utilities have to be interpreted in the context of the environment. Often, the environment consists of other autonomous system-agents. If a self-organising system is decentralised, then another difficulty is that the system responds with utilities via several of its agents whose inner states might not be coherent. Cognitive Psychologists have developed relational complexity theory to describe this problem. According to relational complexity theory [30], the processing load of a cognitive task is determined by the complexity of the relations that must be processed in a given step.

A simple approach to measuring Stability could be done via a time series of measured data points, e.g. the price in a market system with dynamic pricing. The variance in this series is a simple indicator of Stability as in Equation 1:

$$s_X = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{1}$$

where N is the number of received utilities, x_i is an individual utility received by the agent and $\overline{x}$ is the population mean over all received utilities. Note that s_X is $\in [0, 1]$.

B. Learnability

The Learnability of a system is an established quality indicator in software quality evaluation [31]. To generalise this problem for all self-organising systems, we propose that the Learnability of a self-organising system indicates how well the likelihood of transitions between received utilities can be learned by a user. For this, additionally available local information might be helpful. A user can observe states in the systems environment (including his own) or get to know the intentions (i.e. inner optimisation objectives) of the system and to some extent learn why a new received utility supports these. Thus, we might also ask: How disentangled is a received utility from any users' local information?

Learnability does not come naturally to self-organising systems because of their inherent abstraction over complexity we discussed above. System agents do not expose all information available to them. In the decentralised case, they might even be restricted to giving only a partial account of the systems state from their viewpoint. For example, when a traffic-sign agent A on a Dutch motorway needs to signal a lower speed limit, this is due to congestions in another part of the system, but exact information about all problems in the system is not available at the agents location. And even if it were available, the response at A came about by complex interactions - it might even be part of an emergent high-level response and as such not easily reducible.

We present here the most simple approach to Learnability: To measure how hard it would be for the user to forecast utilities she receives from the system, given only other contextual information she has. For stable systems, (i.e. with low variance in received utilities) forecasting is not hard, but even with high variance, a user might be able to infer received utilities when she is given more information than just the received utilities themselves. If a user can observe a part of the environment (at the very least, each user knows about her own actions), she is able to see and learn possible correlations between changes in it and the received utilities. For example, the traffic-sign agent at location A could increase Learnability by telling slowed-down drivers about the reason he does so: maybe an average congestion index he derives from signals received from agents at location B, C and D. A running sample correlation coefficient would be computed like Equation 2:

$$r_{XY} = \frac{1}{N-1}\sum_{i=1}^{N}(\frac{x_i - \bar{x}}{s_X})(\frac{y_i - \bar{y}}{s_Y}) \tag{2}$$

where N is the number of pairs of system output x_i and local information y_i. Note that r_{XY} is $\in [-1, 1]$, where 0 means no correlation.

From the user's perspective, the Learnability problem resembles the General Game Playing Challenge ([11], [32]), where artificial agents have to play a game of which they are told the rules only when the game begins: They are not prepared to behave optimal (regarding their own utilities) in the realm of the system. An even harder case is when even the rules are unknown to the player - a situation which Levinson [33] calls the 'blind player'. Note that even when rules are accessible, blind players can be found in many domains, when they are not willing or interested enough to put in effort to learn the rules, for example in security or surveillance scenarios.

C. Engageability

Many claims exist that interaction patterns with users should matter. For self-organising systems, we can think of two reasons in favour of this:

1) Whenever the actions of users matter to the system context, the system design should give users incentives to make it individually rational [29] to work towards the systems objectives. In other words, the self-organising system should provide a conclusive relation between user actions and received utility, encouraging actions from the user which help the system and discouraging those that do not.
2) However, the system design should not only care about quality of user actions, but also quantity. When users feel they cannot do what they want or that their actions do not have an effect, they are likely to decrease their usage of the system or the efforts to engage.

If the systems user model is detailed enough and it engages the user with the fitting utilities, the user will notice an effect of a change in response to her actions. But this is hard to achieve, partly because of the abstraction of inner complexity, but also especially in systems with many users. For example, the presence of the driver on any dutch motorway at rush hour matters, but often only in infinitesimal amounts.

For the purposes of this work, we will again take the most simple view we can take to make a first approach to the this indicator. Engageability measures how much a change in user behaviour (her actions towards the system) a matters to the utility u received from the system in response. For example, consider a market system that engages a user by letting her know about a price peak in advance, suggesting to lower demand (the systems objective being here to lower overall demand). The amount of demand the user refrains from would be the action, an immediate price reduction (per unit) the received utility effect (in fact, this is how our example will proceed in Section IV). The indicator measures the ratio of changes in user action to effects in received utility:

$$\%\triangle a / \%\triangle u$$

In practice, it might make sense to consider specific ranges of time. Let T be a set of time points t at which the user had opportunity to perform a change in his action towards the system a_t (e.g. speed, usage amounts, price bids). We assume a utility effect $u_t \in [0,1]$ to every $a_t \in [0,1]$ (where in both cases 0 denotes no change or effect and 1 denotes the most complete change or effect possible), such that for every considered time point set T, we can calculate the rate of change in actions and the received utility effects for each time step t in T. If there exist timing difficulties in connecting utilities to actions, it might make sense to compare the sum of actions to net utility achieved over the course of a longer time period. We also assume that for both actions and utilities base values a_b and u_b are known, so we can compare the deviations (if no base value is known, one could simply take zero or the last value). Thus, a possible measure (in appropriate settings) could be:

$$\frac{\sum_{t=1}^{T} d^{T-t}((u_t - u_b) - (a_t - a_b))}{T} \tag{3}$$

where $d \in [0,1]$ denotes a discounting factor, which can discount older differences stronger than more recent ones.

Engageability gives us values $\in [-1,1]$, much as the correlation coefficient does, which we used for Learnability. If Engageability is zero, the received utility effect from the system matches the change in action by the user. If Engageability is lesser than zero, the utility effect does not match the change in action and if it is bigger than zero, the utility effect is even larger than the change in action. To rule out that randomly returned utilities average to zero, one should consider whether the variance in single Engagement indices is not too high.

D. Evaluation

In such complex environments as the domains in which self-organising systems are used, stochastic and parameterised simulations are a meaningful approach to study the different configurations a system might find itself in and then evaluate indicators based on the collected data. In the next section, we will put forward an example market system and simulate several scenarios in it. We will then use three methods to evaluate our indicators:

- It is important to see how the indicators respond to different system configurations. Which changes in parameter values of the system configuration increase or decrease them?
- For the comparison of system designs in multi-objective optimisation problems like these, pareto fronts between the three indicators can be a meaningful tool to study how much the indicators compete. Pareto fronts allow us to state if an indicator could be increased under a different configuration without decreasing another indicator (i.e. the new configuration pareto-dominates the current one).
- Not every user is the same - for a meaningful analysis, one should also look at distinct parts of the user population to see if the indicators behave differently to them.

Keep in mind that all indicators we propose denote a perceived value, from the users perspective. Their relevancy might differ, given the application or design situation. They can become less clear by noise and even influence each other (e.g. Stability decreases Engageability, as we will demonstrate later). In addition, the importance users assign to them are subjective and will vary between users.

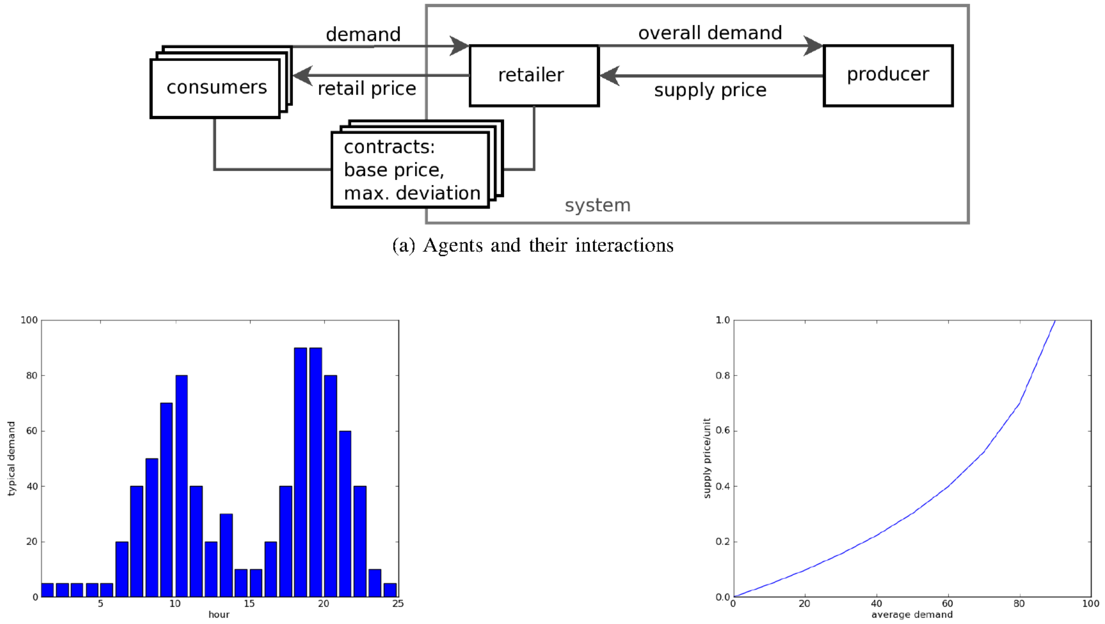
IV. Example

A. Model

To exemplify the indicators we proposed in the last section, we model a simple market example (see Figure 2a for illustration), where consumers consume a good continuously (e.g. electricity, but also water, communication services) and pay for it according to retail contracts. A supplier and a retailer self-organise dynamic pricing. Interestingly, the users of this system also constitute a significant part of the environment for other users via their behaviour. We will come back to this in the analysis. Note that in a full-fledged market model, we would find several retailer agents who would influence each other, rendering the situation far more complex. We could of course also add suppliers with different cost curves.

C consumers each consume an amount of the good $d_{c,h} \in [0,100]$ in 24 (hourly) intervals h. The typical demand profile over the course of one day is shown in Figure 2b, with two prominent spikes, one in the morning and one in the evening. We vary the demand of each consumer c in every hour h by a random amount $\in [0,15]$.

The system consists of a retailer agent and a production agent. The retailer agent has contracts with the local consumers. To match demand for the current hour, he buys the needed amount of the good from the producer - for a supply price p^s_h per unit $\in [0,1]$. The supply price reacts logarithmically to changes in average demand (see Figure 2c). Thus, it has high marginal prices when consumption is high and we write p^s_h short for $p^s_h(\frac{d_{C,h}}{C})$, where $d_{C,h} = \sum_{c=0}^{C} d_{c,h}$.



(a) Agents and their interactions

(b) Demand curve for typical consumption

(c) Supply Pricing

Figure 2. Overview over the example model and agent properties

The retailer agent has to pay the amount of money for fulfilling the local demand each hour h, shown in Formula 4, which is the production costs for the amount of goods that is consumed.

$$Costs_h = p^s_h * C \tag{4}$$

We now turn to the earnings of the retailer, with which he covers his costs. As is the case in retail markets, the retailer offers contracts to consumers. Via contracts, retailers burden risks (of price deviations). We model dynamic contracts in which the consumer might burden some of the risks. For every consumer $c \in C$, the contracts specify a baseline price per unit b_c and the maximal deviation in prices/unit dev_c that consumers might be subjected to in peak times. Every hour h, the retailer bills each consumer a retail price (per consumed unit) $p^r_{c,h} = b_c + x_{c,h}(d_{C,h})$, where $0 <= x_{c,h}(d_{C,h}) <= dev_c$.

Formulas 5, 6 and 7 show how each consumer is billed. First, the retailer calculates the amount of costs which is not covered by base prices in contracts, $Uncov_h$ - the amount he should try to bill via price deviations. If this amount is positive, for each consumer an extra charge $x_{c,h}(d_C)$ will be calculated and billed in addition to b_c. This extra charge is proportional to the consumption of all users, but limited by the maximal deviation of price per unit dev_c.

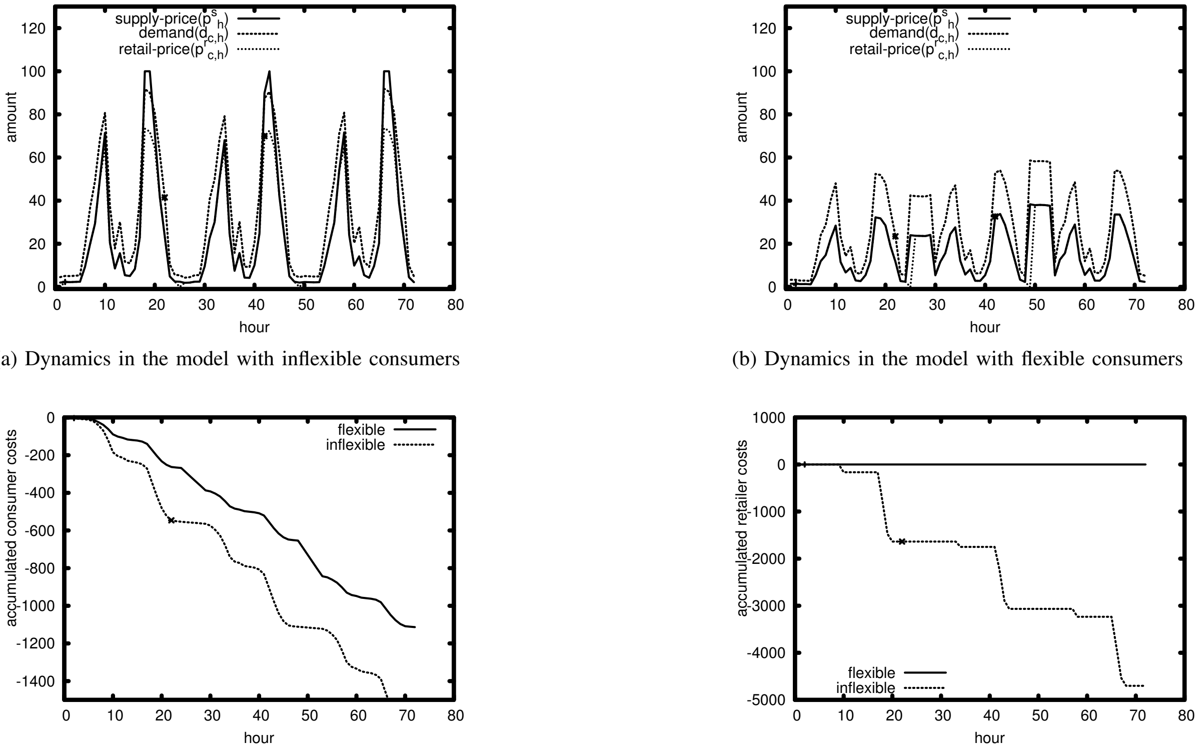
$$Uncov_h = max(0, Costs_h - \sum_{c=0}^{C}(d_{c,h} * b_c)) \tag{5}$$

$$\sum_{c=0}^{C} p^r_{c,h}(d_{C,h}, d_{c,h}) = \sum_{c=0}^{C} b_c + x_{c,h}(d_{C,h}, d_{c,h}) \tag{6}$$

$$x_{c,h}(d_C, d_{c,h}) = min(\frac{d_{c,h}}{d_{C,h}} * Uncov_h, d_{c,h} * dev_c) \tag{7}$$

Until here, we have modelled a relatively simple market, consisting of consumers and one retailer and one producer. Since demand is always met, we assume that retail prices can be treated as a suitable utility measure of consumers. Now we can model the interactions between consumers and retailers. Consumers can be flexible in the sense that they can adjust their demand to influence their retail price. Hourly announced retail prices act as an incentive signal, to which flexible consumers can react. However, we do not assume that demand can disappear completely. Each hour h, a flexible consumer c will delay 40% of his demand to the night (when demand is commonly low) if $p^r_{c,h} > b_c$, in order to achieve a lower price $p^r_{c,h}$. The amount that has been delayed over the course of one day is spreaded out evenly over the hours from 1 to 5 o'clock of the following day. Algorithm 1 explains this simple negotiation process. We call consumers inflexible who will never change their demand.

For one consumer at a time, who acts either flexible or inflexible, over the course of three days and with $dev_c = 0.4$ and $b_c = 0.3$, Figures 3a and 3b show the dynamics of demand $d_{c,h}$, retail prices $p^r_{c,h}$ and supply prices p^s_h. Notice how the flexible consumer spreads his demand and thus achieves fewer spikes. Supply prices are lower and thus are the retail prices, which are exactly the same. In the inflexible case, demand spikes are higher, which leads to much higher supply prices. Due to the base price settings in the contracts, the retail prices do not match the supply prices.



(a) Dynamics in the model with inflexible consumers

(b) Dynamics in the model with flexible consumers

(c) Accumulated costs of a flexible and an inflexible consumer

(d) Accumulated costs of the retailer in a flexible and an inflexible consumer market

Figure 3. Flexible vs Inflexible consumer scenario

Algorithm 1 Demand negotiations

```
while $h < runtime$ do
    All consumers $c$ announce their demand $d_{c,h}$
    Supplier calculates $p^s_h$
    Retailer calculates $p^r_{c,h}$ and announces results to all $c$
    for all $c \in C$ do
        if $c$ is flexible and $p^r_{c,h} > b_c$ then
            $c$ delays $40\%$ of $d_{c,h}$ to the night
        end if
    end for
    Supplier calculates $p^s_h$
    Retailer calculates $p^r_{c,h}$ and bills all $c$
    $h \leftarrow h + 1$
end while
```

Figures 3c and 3d show the costs for consumer and retailer, respectively. We first observe that flexibility in demand pays off for the consumer. His cost curve is much smoother and less steep. Note that this is even the case when he does not profit from the retail price being limited by var_c, as does the inflexible consumer in this case. His spiking demand behaviour is not only more costly to him, but also for the retailer.

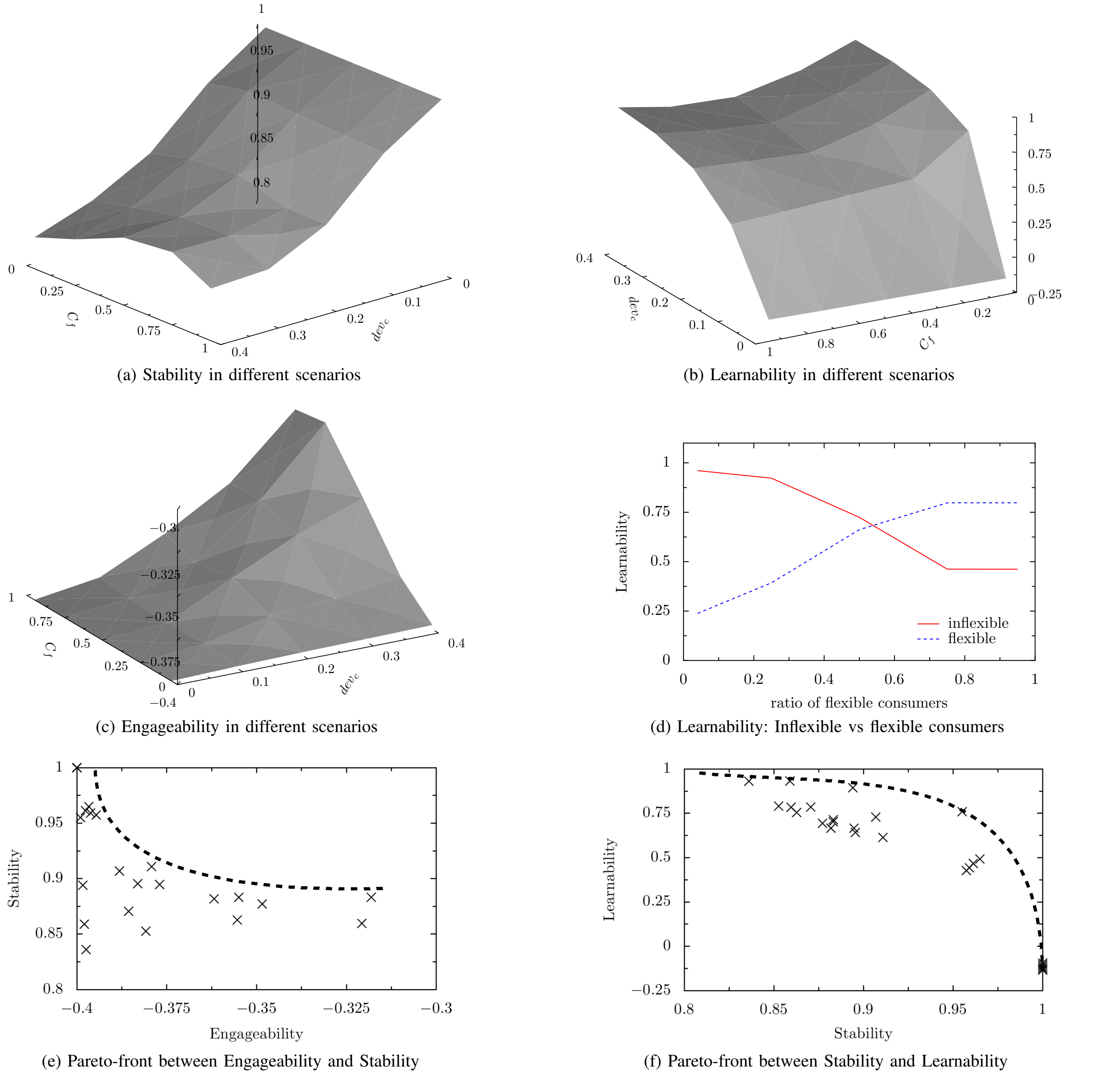
B. Measuring Indicators

We will now vary two important variables in our system and its environment and observe how this affects the indicators we proposed. We run several simulations with combinatorial experiment design in order to demonstrate the quality indicators we offered in Section III.

The retailer agent can hide some of the risk from the consumers, which is in our case the risk of supply price spikes. We modelled this by a maximal retail price deviation in contracts. We will now vary this maximal deviation $dev_c \in [0.0, 0.1, 0.2, 0.3, 0.4]$. An important (and unobservable) part of the environment are other consumers. Are they flexible and thus good for supply prices or not? To study effects of this part of the system environment, we vary the ratio of flexible consumers $C_f \in [0.04, 0.25, 0.5, 0.74, 0.958]$, such that there are $1, 6, 12, 18$ and 23 flexible consumers present, out of 24. We thus use $5 * 5 = 25$ settings. The model runs for 120 days and we conduct 10 runs per setting.

In our simple market example, a consumer has not much information at hand. He knows his personal retail price $p^r_{c,h}$ and he knows his own demand profile. In addition, he can observe how much the system lowers the retail price when he changes his demand profile. We always look at a time window over the last 24 hours.

To measure Stability, we consider the normalised re-



(a) Stability in different scenarios

(b) Learnability in different scenarios

(c) Engageability in different scenarios

(d) Learnability: Inflexible vs flexible consumers

(e) Pareto-front between Engageability and Stability

(f) Pareto-front between Stability and Learnability

Figure 4. a)-c): Results of varying the ratio of flexible consumers and maximal deviations in retail prices - note that the graphs have been rotated for best visibility of the plot; d): plotting Learnability for flexible versus inflexible users; e) & f): Pareto front analysis - data points and manually illustrated pareto fronts

tail prices consumers paid (per unit). This tells us how much fluctuation in prices is observed by a consumer. For Learnability, we correlate the normalised demand profile of consumers (before the flexible consumers delay parts of their demand) with the normalised announced retail prices that are based on them. This tells us how well the individual demand profile of a consumer c predicts retail price announcements he gets. For Engageability, the delays in demand which flexible consumers perform are compared to the reduction in retail price they achieve by this. This tells us about the utility effect a flexible consumer can achieve for himself via his actions. We compare deviations to known default states (original demand, original retail price), with $T = 24$ and $d = 1$ (we do not discount).

C. *Results*

1) System configuration analysis: Figures 4a, 4b and 4c show the outcome for all three indicators when the two variables are changing. In Figure 4a, we see that the significant influence on Stability is dev_c. It is highest when $dev_c = 0.0$. Retail prices are allowed to fluctuate by demand when dev_c is increased and thus the perceived Stability decreases. Learnability, in turn, is lowest when dev_C is low. When prices do not respond to changes in demand, they do not make sense to the consumers. Thus, Learnability increases with dev_c, and interestingly, this increase is considerably stronger when C_f is lower. This is due to flexible consumers shifting demand to the night. Less sharper spikes mean that the retail base price b_c is surpassed less often and less far, which leads to lesser correlations between demand and retail price. Engageability is highest when all consumers are flexible and the maximum retail price deviation is high and thus allows them to save money by delaying their demand: as the retailer will treat all consumers the same, the few flexible consumers will still suffer when the majority is inflexible. Note that -0.4 is the baseline for this indicator, as the delay amount is $0.4 * d_{c,h}$.

2) Pareto front analysis: In order to make a pareto front analysis, we plotted all indicators against each other. Figures 4e and 4f plot Engageability versus Stability and Stability versus Learnability, respectively. In both plots, it is desirable to move to the right (on the x-axis) and to the top (on the y-axis). We see that between these indicators a pareto front can easily be spotted. Experiments with much more variable configurations and careful exploration of the indicator space could produce more meaningful results, but this will suffice for this demonstration. A planner of our example market could make use of these trade-offs for design decisions.

3) Inner-usergroup analysis: We now look at the flexible and inflexible consumers distinctively. We are interested in how much the indicators differ among them and how much they change in relation to the prevalence of one group to the other. We fixed $dev_C = -0.4$. Figure 4d shows the only comparison with significant results: each user group perceives the highest Learnability when they are among consumers that behave just like them. This is again a direct consequence of the retailers method of distributing his cost among all consumers (if possible), without regard to their contribution in demand delay. Interestingly, both graphs level off after more than 80% of the consumers are flexible. Note that this significant distinction would not become visible when measuring only utility, as one lonely flexible consumer would still suffer from high average supply prices. Our proposed indicators make this clear.

V. Conclusion

In this paper, we have described a design challenge for self-organising systems, stating that the increasing complexity these systems are able to capture leads to decreasing comprehensibility of its behaviour to its users. We describe the kind of systems for which this problem arises and offer three quality indicators: Stability, Learnability and Engagement. We propose ways of computing them using basic statistics and conduct a simple study in a retail market model in order to demonstrate their evaluation. We analyse the results by comparing the outcome regarding various system configurations, plot pareto fronts and study how the indicators behave to different groups of users. We believe to have sufficiently motivated and shown in the results of our example (see Section IV-C) that these indicators are useful tools for planners of multi-user, online self-organising systems in order to find the best design trade-offs with respect to comprehensibility.

VI. Future work

The calculations of indicators we proposed are far from being optimal for every situation. For Learnability, agents could actually try to predict future system behaviour. They could use model selection over the results of a posteriori machine learning technique, for instance Bayesian Learning or Gaussian Processes. It would also be interesting for some domains to measure the minimal description length or entropy [13] of the systems responses to rate its Learnability. For Engageability, there are possible pitfalls worth studying. We already mentioned timing issues in comparing actions and received utilities. In addition, the interpretation of received utilities might be tricky (especially when dealing with human users). For instance, in the city of New York, pedestrian crossing buttons perform no actual function any more for decades, but many argue that they still raise Engageability for the pedestrians due to the placebo effect [34] (though they certainly do not facilitate Learnability).

Overall, the trade-off between system goals and the discussed indicators is worthy of more discussion (e.g. Stability most likely compromises adaptability). But this is not necessarily a one-way street. In more intricate market models, one could study if optimising towards these indicators actually makes a market more dynamically efficient [27] when faced with long-term change, as they empower users to adapt.

Acknowledgment

The authors thank the IOP EMVT programme (project IDeaNeD) for financial support.

References

[1] P. Sengers, “Schizophrenia and Narrative in Artifcial Agents,” *Science And Technology*, vol. 35, no. 4, pp. 427–431, 2002.

[2] S. Smulders, “Control of freeway traffic flow by variable speed signs,” *Transportation Research Part B: Methodological*, 1990.

[3] C. Gershenson and L. Pineda, “Why does public transport not arrive on time? The pervasiveness of equal headway instability.” *PloS one*, vol. 4, no. 10, p. e7292, 2009.

[4] I. Dusparic and V. Cahill, “Distributed W-Learning: Multi-Policy Optimization in Self-Organizing Systems,” in *Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, 2009.

[5] J. Li, G. James, and G. Poulton, “Set-Points Based Optimal Multi-Agent Coordination for Controlling Distributed Energy Loads,” *2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pp. 265–271, 2009.

[6] J. Kim and P. Bentley, “An Artificial Immune Model for Network Intrusion Detection,” in *7th European congress on intelligent techniques and soft computing (EUFIT’99)*, 1999.

[7] M. Ulieru, “Adaptive Information Infrastructures for the e-Society,” *Engineering Self-Organising Systems*, pp. 32–51, 2005.

[8] L. De Alfaro, “Game models for open systems,” *Lecture Notes in Computer Science, Verification: Theory and Practice*, pp. 269–289, 2004.

[9] P. Torroni, P. Yolum, M. Singh, M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, and P. Mello, “Modelling interactions via commitments and expectations,” *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, pp. 263–284, 2009.

[10] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden, “The Price of Stability for Network Design with Fair Cost Allocation,” *SIAM Journal on Computing*, vol. 38, no. 4, pp. 263–284, 2008.

[11] M. Genesereth and N. Love, “General Game Playing: Overview of the AAAI Competition,” *AI Magazine*, no. 26, 2005.

[12] H. Cramer, “People’s responses to autonomous & adaptive systems,” PhD thesis, Universiteit van Amsterdam, 2010.

[13] R. Holzer, H. D. Meer, and C. Bettstetter, “On autonomy and emergence in self-organizing systems,” *Self-Organizing Systems*, pp. 157–169, 2008.

[14] T. De Wolf and T. Holvoet, “Towards autonomic computing: agent-based modelling, dynamical systems analysis, and decentralised control,” in *Proceedings of the IEEE International Conference on Industrial Informatics*. IEEE, 2003, pp. 470–479.

[15] G. Serugendo, N. Foukia, S. Hassas, A. Karageorgos, S. Mostefaoui, O. Rana, M. Ulieru, P. Valckenaers, and C. Van Aart, “Self-organisation: Paradigms and applications,” *Engineering Self-Organising Systems*, vol. 2977, pp. 1–19, 2004.

[16] H. Von Foerster, “On self-organizing systems and their environments,” *Self-organizing systems*, vol. 50, pp. 31–50, 1960.

[17] M. Van Steen and A. Tanenbaum, *Distributed Systems: Principles and Paradigms*. Prentice Hall, 2002.

[18] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of IEEE international conference on neural networks*, vol. 4. Piscataway, NJ: IEEE, 1995, pp. 1942–1948.

[19] N. Johnson, S. Jarvis, R. Jonson, P. Cheung, Y. Kwong, and P. Hui, “Volatility and agent adaptability in a self-organizing market,” *Physica A: Statistical Mechanics and it Applications*, no. 258, pp. 230–236, 1998.

[20] A. Chakraborti, S. Pradhan, and B. Chakrabarti, “A self-organising model of market with single commodity,” *Physica A: Statistical Mechanics and its Applications*, vol. 297, no. 1-2, pp. 253–259, Aug. 2001.

[21] A. Mas-Colell, M. Whinston, and J. Green, *Microeconomic theory*, Oxford University Press, Ed. Oxford: Microeconomic theory, 1995.

[22] M. Morris, O. Sheldon, D. Ames, and M. Young, “Metaphors and the market: Consequences and preconditions of agent and object metaphors in stock market commentary,” *Organizational Behavior and Human Decision Processes*, vol. 102, no. 2, pp. 174–192, 2007.

[23] N. Luhmann, *Familiarity, Confidence, Trust: Problems and Alternatives*. New York: Basil Blackwell, 2000.

[24] P. Sengers, “Designing comprehensible agents,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 16. Lawrence Erlbaum, 1999, pp. 1227–1232.

[25] M. Jensen, “Some Anomalous Evidence Regarding Market Efficiency,” *Journal of Financial Economics*, vol. 6, no. 2-3, pp. 95–101, Sep. 1978.

[26] M. Pagano and A. Roell, “Transparency and Liquidity: A Comparison of Auction and Dealer Markets with Informed Trading,” *Journal Of Finance*, vol. 51, no. 2, pp. 579–611, 1996.

[27] A. Abel, N. Mankiw, L. Summers, and R. Zeckhauser, “Assessing Dynamic Efficiency: Theory and Evidence,” *The Review of Economic Studies*, vol. 56, no. 1, pp. 1–19, 1989.

[28] J. Potts, *The new evolutionary microeconomics: complexity, competence and adaptive behaviour*. Cheltenham: Edward Elgar Publishing, 2001.

[29] D. Fudenberg and J. Tirole, *Game theory*. Boston: MIT Press, 1991.

[30] B. Rumpler and W. Elmenreich, “Considerations on the Complexity of Embedded Real-Time System Design Tasks,” *2006 IEEE International Conference on Computational Cybernetics*, pp. 1–6, Aug. 2006.

[31] ISO/IEC, “International Standard 9126: Software engineering - Product quality,” Geneva, Switzerland, 2004.

[32] “General Game Playing.” [Online]. Available: http://games.stanford.edu

[33] R. Levinson, “General Game-Playing and Reinforcement Learning,” *Computational Intelligence*, vol. 12, no. 1, pp. 155–176, Feb. 1996.

[34] M. Luo, “For Exercise in New York Futility, Push Button,” 2004. [Online]. Available: http://www.nytimes.com/2004/02/27/nyregion/27BUTT.html