Evaluating an Emergent Behaviour Algorithm in JCSP for Energy Conservation in Lighting Systems

Anna KOSEK [a] and Aly SYED [b] and Jon KERRIDGE [c]
[a] *Risoe DTU National Laboratory of Sustainable Energy, Denmark;*
E-mail: amko@risoe.dtu.dk
[b] *NXP Semiconductors, The Netherlands; E-mail:* aly.syed@nxp.com
[c] *Edinburgh Napier University, United Kingdom; E-mail:* j.kerridge@napier.ac.uk

Abstract. Since the invention of the light bulb, artificial light is accompanying people all around the world every day and night. As the light bulb itself evolved a lot through years, light control systems are still switch-based and require users to make most of the decisions about its behaviour. This paper presents an algorithm for emergent behaviour in a lighting system to achieve stable, user defined light level, while saving energy. The algorithm employs a parallel design and was tested using JCSP.

Keywords. emergent behaviour, lighting system, energy saving

Introduction

The main energy sources on Earth are not renewable, according to EIA[1] statistics from 2008, renewable solar, geothermal, wind, hydro-power and biomass energy is only 7% of the world energy supply. Energy usage grows every year, according to EIA between 1997 and 2008 it grew by 604.5 Terawatt hours (TWh) that is 15.5% of total energy use [1]. There is a widely recognised need to reduce energy usage in any of the major sectors of the economy. Lighting takes on average 37.8% of the total energy use in buildings over $90m^2$ [2] and in houses it is 17% [3]. Therefore lighting usage has a big share of overall energy consumption and gives opportunity to conserve shrinking energy sources.

There are many ways to save energy on lighting, very obvious one is to replace existing high-power incandescent bulbs with energy-saving Compact Fluorescent Lamps (CFL) and Light Emitting Diode (LED) lights. This can bring 70% of savings when switching from incandescent to CFL [4] or LED [5]. Further reductions in energy use can be achieved by better controlling light, such as they consume energy sparingly. Statistics from the Dutch NEN[2] norm NEN2916 [6] shows an estimate of potential energy savings for different smart light controls.

Statistics from Table 1 indicate that daylight-dependent switching can save 60% of energy use in spaces with daylight, that is up to 22% of the total energy use for lighting in the building, if we replace on/off with dimming capability the reduction is 80%, that is overall on average 30% of enegry conservation in a building.

The number of devices controlled by home automation systems grows every day and many devices are already hidden for people's sight, becoming more and more pervasive. On-

[1]U.S. Energy Information Administration, http://www.eia.doe.gov
[2]Nederlandse Normalisatie-instituut (*eng.* Dutch Standards Institute), http://www.nni.nl

Table 1. Energy reduction possibilities for lighting.

Smart light control	Reduced number of light bulb burning hours (%) Spaces with daylight	Spaces with artificial light
Turning all lights off at a certain time	30	30
Manual switching	30	10
Daylight-dependent switching	60	10
Daylight-dependent switching and turning all lights off at a certain time	80	30
Daylight-dependent dimming	80	4

going device miniaturisation makes it possible to create a network of many dust-like devices that will eventually disappear from people's sight, therefore locating and managing them will become very difficult [7]. Building automation systems are usually centrally controlled, however the new trend in building automation systems is to have more distributed approach to control. Small systems are easy to manage with a central control, but for large systems problems with scalability appear and a central management unit becomes a system's bottleneck. The other reason to move toward distributed control system is that central control systems are difficult to change in response to changing occupancy and layout of the building.

When considering pervasive systems self-organisation, self-reconfiguration and adaptation to environmental conditions are desired control mechanisms. In distributed systems, it is possible to program devices to behave in a predefined manner, but it is also possible to give devices some simple behaviour and let the total distributed system's behaviour emerge out of actions of individual devices. This paper describes an emergent behaviour algorithm for energy conservation in a lighting system. The presented algorithm is inspired by the workings of human society. Section 1 presents an example of a lighting system and proposed scenario for light sources adaptation in a single space. Section 2 describes the Lazy and Enthusiastic Employee algorithm for emergent behaviour and Section 3 proposes a production regulation scheme used in this algorithm. In Section 4 of this paper we describe how the algorithm can be adjusted to achieve interesting behaviour. In Section 5 a proposed scenario implementation in JCSP [8,9] is discussed. Section 6 presents different experiments performed with implemented simulation and results are described in Section 7. Conclusions and Further work are presented in Section 8.

1. Scenario Description

The scenario chosen to describe the lighting system is presented in Figure 1, which shows a model of a room that is used as an office. The space is equipped with two different light source types: 16 ceiling lights bulbs that are capable of dimming, another source of light in this room is a mirror that is placed outside the window. This mirror can be actuated such that it reflects sun light to the ceiling in the room. A light sensor also placed outside the window detects and informs the mirror if there is enough sunlight to reflect into the room. It is possible to control the amount of sunlight reflected into the room by changing the angle of the mirror to the incident sunlight. The space is divided into 16 regions equipped with one light sensor each. Regions are associated with lamps and part of the ceiling that are illuminating it using reflected sunlight from the mirror.

The system is implemented such that all devices namely the 16 lamps, 16 light sensors, a mirror and the outside light detecting sensor are all autonomous devices that control their own actions without the influence of a central control device. This can be implemented for example using the architecture presented in [10].

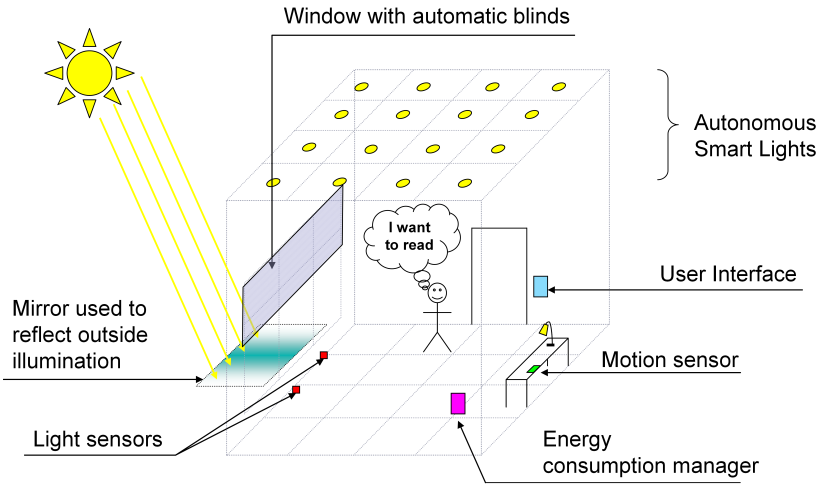

Figure 1. Proposed scenario.

When the sunlight conditions outside change and less light is reflected into the room, then the light sensors detect change in light intensity in the room. The light sensors then inform the lights, which adjust their dimming to compensate such that the amount of light in the room is maintained at the user defined level. As all devices in the room are autonomous and no central control is present, devices need to react to a signal from the sensor individually. This paper describes an algorithm that is responsible for emergent behaviour of the presented lighting system.

2. Lazy and Enthusiastic Employee Algorithm for Emergent Behaviour

Let's consider an example of a brick making company. The company can make bricks that have different cost price. One is made form local materials, therefore it has a lower production cost, but the number of bricks that can be made in certain time is limited, due to limited resources. The second type of brick is more expensive to produce, because it is made of imported materials, but the number of bricks that are being made, at a certain period of time, is not limited. The company has a pool of employees, they are divided into lazy and enthusiastic employees, a lazy worker is not very conducive to work, on the other hand an enthusiastic employee is always happy to answer a manager's request. However both lazy and enthusiastic employees are able make bricks at the same rate once they start working at full capacity. A manager monitors the factory's overall production and periodically informs the employees about the factory performance. The manager has no control over choosing what type of bricks are made, he only issues a feedback on the work that has been done by all employees. For optimal functioning of the factory the issues that need to be addressed are:

- Is it possible to make this brick making company reliable, as long as there are employees to work?
- How to minimise the production cost by using available employees and material?

Because the factory manager has no control over choosing the brick type that is being made, the algorithm to minimise production cost has to be deployed in employees.

Let's assign all lazy employees to production of expensive bricks and very enthusiastic employees to cheap bricks production. When the request to increase bricks production is triggered all available employees respond, but enthusiastic employees will start working immediately and try to increase brick production rapidly. Lazy employees, on the other hand, will try to avoid working, therefore they will wait for a while and then will increase bricks production by a small factor. If the production of cheap bricks is enough to fulfil a request, expensive bricks are not produced or only small number of those bricks is being produced. If the inexpensive production is not enough, then the lazy worker will eventually boost production of expensive bricks and take over the remaining percentage of the delivery.

When a request to decrease the production is triggered, the lazy workers quickly recognise opportunity to work less and are very happy to decrease the production. Enthusiastic employees, on the other hand, are very happy to work, so are not that inclined to decrease their workload. Enthusiastic employees will wait for a while for the situation to stabilise and, if it is necessary, they will decrease production only slightly.

The first reaction of both lazy and enthusiastic employees is significant for minimising cost of production. Because the brick company has to be reliable eventually lazy workers will have to work hard if the production of cheap bricks is not enough to fulfil the request. Similarly, the enthusiastic employees will have to work less if the production exceeds the required amount. As shown in Figure 2 the factory is designed to supply stable flow of bricks and react to request to increase and decrease production without any central control.

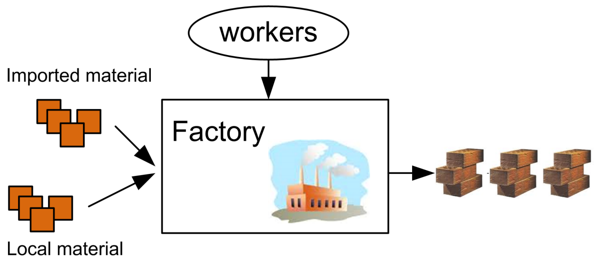

Figure 2. Factory scheme.

Another issue for emergent systems is a feedback loop problem. If, on a request, all employees are equally vigorous to change their behaviour, the behaviour if the system alters very quickly and it is proportional to the number of the employees answering a request. Therefore if employees are happy to answer a request to increase the production all at the same time, this behaviour will be escalated and trigger another request to decrease the production, that can lead to an infinite feedback loop. Introducing the Lazy and Enthusiastic Employee algorithm slows down the reaction of some parts of the factory, which reduces the rapid increase or decrease of production. The feedback loop can also appear, but it will truncate,as the reaction to decreasing and increasing production is not the same, the employees will eventually adapt to the situation and the loop will be broken.

3. The Used Production Regulation Scheme

For the purpose of this paper we present a production regulation scheme that is used with the emergent behaviour algorithm. In the factory all employees are capable of reacting to a value propagated by the manager and adapt their production rate. The ideal production rate in the factory is provided by the manager and is remembered by all employees. The ideal value is provided with a margin, that determines a range around this value where the production rate changes within an error range. Therefore the workers do not try to compensate any more while the production is within the margin. This is done to avoid production rate changing continuously when the value received from the manager is in a margin very close to the ideal value.

The production rate in an environment is divided into three groups: above agreed range, inside of the range and below it. If we let p_i be an ideal production rate value in the factory, $p_i > 0$, m be a margin and p_a be actual production of the factory, then the value of p_a is in the agreed range if $p_a \in (p_i - m, p_i + m)$ provided $m < p_i$ and $m > 0$.

The scheme that regulates production rate p_a in the factory takes a value from the manager at a particular time and if the value is above the specified range, therefore $p_a > p_i + m$, the production is decreased, if the value is below the range, therefore $p_a < p_i - m$, the production is increased. If the value p_a is inside of the range, there is no action taken.

Let d be a production rate of an employee of the factory, where $d > 0$ and let Δd be a factor of increase or decrease of d, where $\Delta d > 0$. Then if $p_a < p_i - m$, then workers production rate is increased $d = d + \Delta d$, if $p_a > p_i + m$, then workers production rate is decreased $d = d - \Delta d$, otherwise the request is dropped and the production rate d is kept constant until another request is issued.

The behaviour can be alternated by changing the value of delta Δd depending on some conditions. Let $c_{\{n\}}$ be a sequence of conditions $c_1, c_2, ..., c_n$, where $c_{\{n\}} = \{c_1, c_2, ..., c_n\}$, $n \in N_+$, $c_n \in R$ and let $g : c_{\{n\}} \to \Delta d$, $\Delta d > 0$ and $h : c_{\{n\}} \to \Delta d$, $\Delta d > 0$, be functions that generate the decrease/increase factor of the bricks production by each employee accordingly, then:

$$\Delta d = \begin{cases} g(c_{\{n\}}) \text{ if } p_a < p_i - m; \\ h(c_{\{n\}}) \text{ if } p_a > p_i + m. \end{cases}$$

Functions g and h that generate the decrease or increase factor of brick production by each employee can be adjusted to regulate factory behaviour. Functions g and h can be either directly dependent or independent of p_a, the actual value of production in the factory.

The first group of production decrease or increase factor functions (g' and h', where $g' \subset g$ and $h' \subset h$) produce values independent of the value of the actual production in the factory, therefore $c_{\{n\}}$ is not dependent on p_a. For example, let's assume the ideal production is $p_i = 100$ $bricks/min$ and the margin is $m = 5$ $bricks/min$ accepted production rate set is between 95 $bricks/min$ and 105 $bricks/min$. If the input from the manager informs that the actual production is $p_a = 50$ $bricks/min$, then the production is increased. An employee can decide to decrease the production by a factor of $\Delta d = 1$ $bricks/min$ or $\Delta d = 10$ $bricks/min$, that is independent of value of p_a. The production rate scheme with factor functions independent of the actual production only checks if the value belongs to any of three ranges and uses functions not based on the actual value of p_a. This function can be dependent of some other conditions used to calculate the increase or decrease factor. Therefore the employee that adapts to the manager's request only need to know if the value is outside of the set $(p_i - m, p_i + m)$ and react depending on the situation.

The second group of production decrease or increase factor functions (g'' and h'', where $g'' \subset g$ and $h'' \subset h$) produce values dependent of the value of the actual production in the factory, therefore p_a or function of p_a can be one of conditions in sequence $c_{\{n\}}$. For example one of the conditions of functions g'' and h'' can be the distance between actual and ideal production rate, therefore $c_1 = |p_i - p_a|$, where $c_1 \in c_{\{n\}}$.

4. Algorithm Adjustments

There are several factors that can be adjusted to achieve emergent behaviour when using Lazy and Enthusiastic Employee algorithm: in this paper we consider production decrease or increase factor functions and margin's m size. Production decrease or increase factor functions helps varying workers' behaviour. As mentioned in algorithm description any worker performs different behaviour when the overall production is too low and react differently in case of over-producing, therefore functions for production decrease or increase factor are different.

Let's consider a factory with 200 workers, where 100 of them are lazy employees and remaining 100 are enthusiastic employees. The ideal production rate is set to be 10000 *bricks/minute*, all workers start from individual production rate, that is 0 *bricks/minute*, the default increase/decrease rate (Δd) is 10 *bricks/minute*. Let's assign all enthusiastic employees to producing cheap bricks (1$ per unit) and lazy employees to production of expensive bricks (10$ per unit). By selecting appropriate production decrease or increase factor functions (g and h) we need to minimise the cost and maximise the factory reliability. At any time period the factory is reliable when actual production rate is within the accepted range, therefore $p_a \in (p_i - m, p_i + m)$ provided $m < p_i$ and $m > 0$, where p_i is an ideal overall production rate.

Within the proposed example let's consider decrease or increase factor functions (g' and h') that produce values independent of the value of the actual production in the factory. Considered functions for enthusiastic employees g'_a, h'_a, and lazy employees g'_b, h'_b, are as follows:

$$\begin{array}{ll} g'_{a_1}(\Delta d) = 3, & g'_{b_1}(\Delta d) = 1, \\ h'_{a_1}(\Delta d) = 1, & h'_{b_1}(\Delta d) = 3, \end{array} \tag{1}$$

Workers production rate factor in the first example of g' and h' functions is constant and reaction to a low production is faster than reaction to over-producing for enthusiastic employee ($g'_{a_1}(\Delta d) > h'_{a_1}(\Delta d)$), and opposite for lazy employee ($g'_{b_1}(\Delta d) < h'_{b_1}(\Delta d)$). This way the enthusiastic employee is increasing production faster than decreasing, independently of value of Δd. In this case the production increase/decrease functions u_1 and u_2 are constant and independent of vale of p_a. The second example functions are as follows:

$$\begin{array}{ll} g'_{a_2}(step, \Delta d) = \Delta d \cdot (step^4/100)/100, & \text{where } step \in [0, 10], \\ h'_{a_2}(step, \Delta d) = \Delta d \cdot ((step - 10)^4/100)/100, & \text{where } step \in [0, 10], \\ g'_{b_2}(step, \Delta d) = \Delta d \cdot ((step - 10)^3/10 + 100)/100, & \text{where } step \in [0, 10], \\ h'_{b_2}(step, \Delta d) = \Delta d \cdot (100 - (step)^3/10)/100, & \text{where } step \in [0, 10]. \end{array} \tag{2}$$

Workers production rate factor in the second example of g' and h' functions is modelled using curves presented in Figure 3. Curves in Figure 3 based on x^4 and x^3 are used to calculate fraction of Δd being added or subtracted. The shape of the curve influences worker's production rate and is one of the parameters of the algorithm. According to curves from Figure 3, enthusiastic workers increase their production rapidly at first and then stabilise, while decreasing production slowly and then picking up when no other employees are willing to decrease the production. The behaviour changes depending on a *step*, where $step \in [0, 10]$. At first an employee continues on the chosen curve and when the overall production reaches other side of the required range $(p_i - m, p_i + m)$, then the curve is changed and a worker behaves differently.

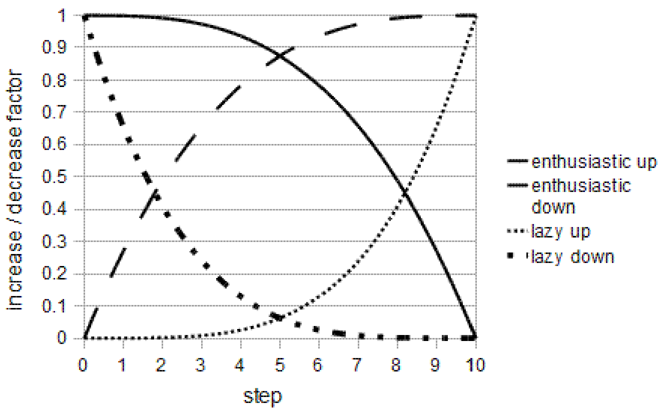

Figure 3. Functions used to decrease and increase bricks production in the factory.

The workers' behaviour for both sets of g' and h' functions is compared to the behaviour without the Lazy and Enthusiastic algorithm and presented in Figure 4. The g' and h' functions used for comparison are as follows:

$$\begin{aligned} g'_{a_3}(\Delta d) = 2, \quad g'_{b_3}(\Delta d) = 2, \\ h'_{a_3}(\Delta d) = 2, \quad h'_{b_3}(\Delta d) = 2. \end{aligned} \tag{3}$$

As presented in Figure 4, the Lazy and Enthusiastic Employee algorithm's performance depends on choice of g' and h' functions. Graphs 4.A1 and 4.A3 are very similar, therefore overall production of the factory is stable in both of those cases. When we look closer at individual behaviour of workers, graph 4.B3 shows that both work the same and try to sustain the production rate, in graph 4.B1 enthusiastic employees overtake the production with 9.9-10.2 $bricks/minute$, and force the lazy workers to decrease their production to around 0-2 $bricks/minute$. The third case (graph 4.B2), when using curves from Figure 3, enthusiastic workers take over the production completely, not letting lazy employees contribute to the overall production rate.

The second group of production decrease or increase factor functions (g'' and h'') produce values dependent of the value of the actual production in the factory. Therefore the value of increase/decrease of bricks production depends on the actual value of overall production p_a. An example of functions g'' and h'' depend of the distance between p_a and p_i, as introduced in previous section, and can be as follows:

$$\begin{aligned} g''_{a_4}(p_a, p_i) &= (1-(|p_i - p_a|/p_i)) \quad & p_i, p_a > 0, \\ h''_{a_4}(p_a, p_i) &= (|p_i - p_a|/p_i) \quad & p_i, p_a > 0, \\ g''_{b_4}(p_a, p_i) &= (|p_i - p_a|/p_i) \quad & p_i, p_a > 0, \\ h''_{b_4}(p_a, p_i) &= (1-(|p_i - p_a|/p_i)) \quad & p_i, p_a > 0. \end{aligned} \tag{4}$$

Functions g'' and h'' are designed to behave differently when distance between p_a and p_i is large and differently when both values are in close proximity. Special attention was paid to the region close to the value of p_i in order to enable enthusiastic employees to overtake work done by lazy workers and stabilise the production. The behaviour of the factory is presented in Figure 5.

As shown in Figures 4 and 5, the emergent behaviour of workers in a factory varies depending on chosen production decrease or increase factor functions. Table 2 presents combined results of the described behaviours, presenting cost of production and factory reliability. The cost of production is calculated with assumption that cheap bricks cost 1$ and expensive 10$ per unit. Factory reliability is measured from 100 samples of behaviour (100 minutes), the factory is reliable if the actual production is within the accepted range, therefore $p_a \in (p_i - m, p_i + m)$ provided $m < p_i$ and $m > 0$, where p_i is an ideal overall production rate. Note that the limitation in production of the cheap bricks is not included in results Table 2, as the aim is to compare speed of reaction of workers depending on increase and decrease factor functions. In experiments described in Section 6, the light production from the mirror is limited by the outdoor lighting conditions.

Table 2. Algorithm evaluation towards production cost and factory reliability.

Function used	cheap bricks production cost	expensive bricks production cost	overall production cost	factory reliability
(1)	798601	800520	**1599121**	72%
(2)	84390	0	**84390**	84%
(3)	435400	4354000	**4789400**	76%
(4)	891502	936440	**1827942**	94%

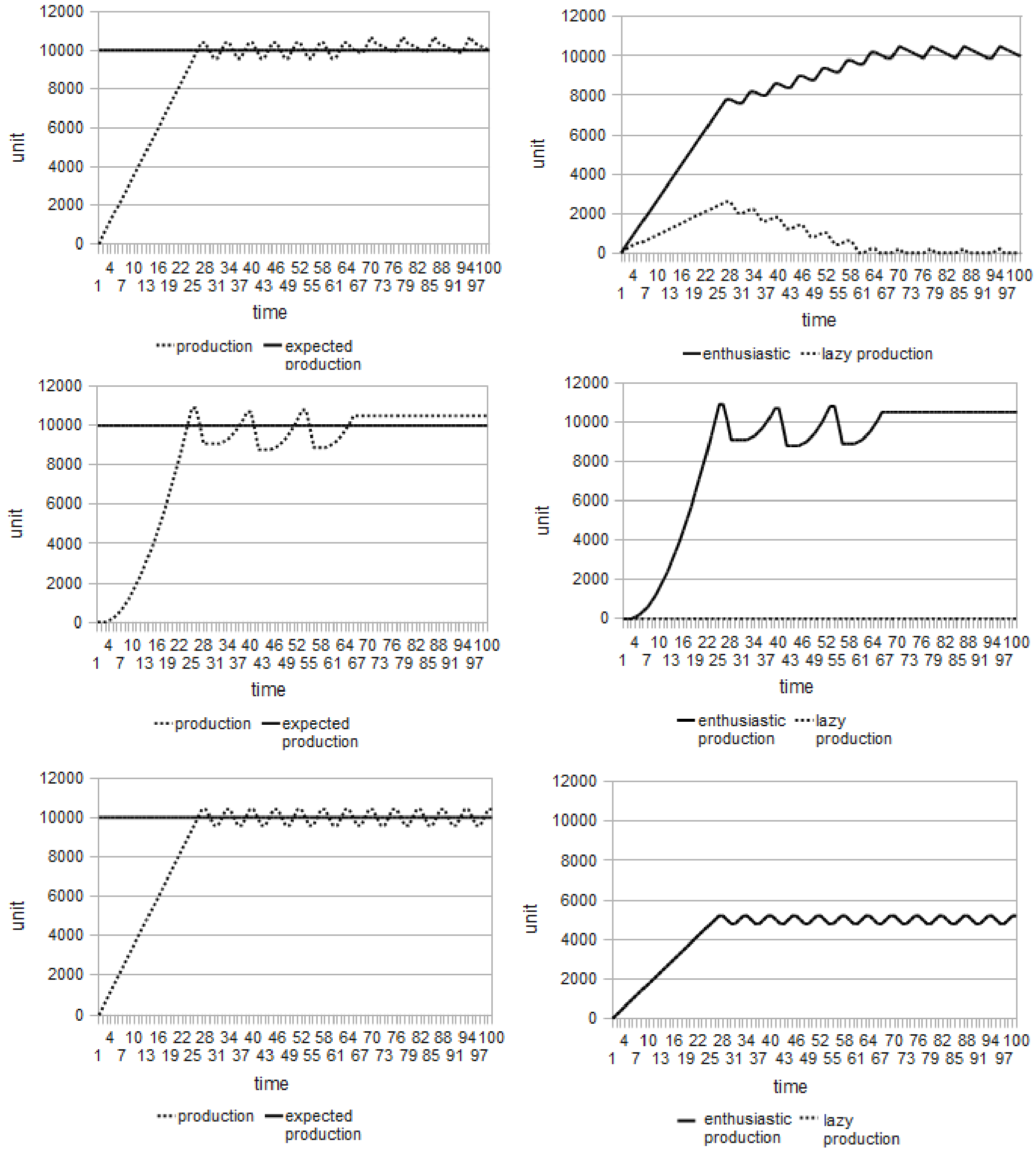

Figure 4. A1,A2,A3- Overall behaviour of workers with functions (1),(2) and (3) respectively; B1,B2,B3- Behaviour of groups of workers with functions (1),(2) and (3) respectively.

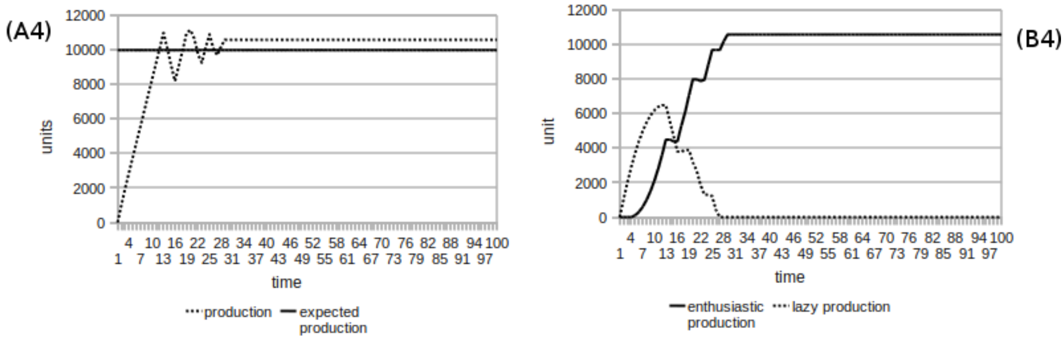

Figure 5. A4- Overall behaviour of workers with functions (4); B4- Behaviour of groups of workers with functions (4).

Based on data represented in Table 2, functions (2) minimised the cost considerably while maintaining factory reliability within 84%. The performance of factory reliability when using functions (4) is impressive, while the cost is around 20 times higher than the production with functions (2). The overall behaviour of the algorithm points out that functions regulating production decrease or increase factor need to be selected carefully. When maximising for production cost, functions (2) or their variations should be used.

5. Implementation

The Lazy and Enthusiastic Employee algorithm can be used to sustain a stable light level in a room with many autonomous light sources,as shown in Figure 1, while saving electricity. If we assume that light bulbs are lazy employees, as they need electricity to work, which is "expensive". Mirror using sunlight, which is "cheap", is assigned to be an enthusiastic employee and it is expected to give as much light as is needed and possible. This way it is possible to use the algorithm to save energy and ensure stable light level in the space as long as all light sources can function properly. This system can also exhibit emergent behaviour based on this simple model. All devices that are needed for the ligting system described in this paper are autonomous and collaborate by formatting ad-hoc networks. This means that there are many concurrent behaviours that represent a real word scenarios need to be simulated. We have chosen, therefore, the JCSP (*Communicating Sequential Processes (CSP)* [11] *for Java*) library for simulation of this system. Java was chosen as a programming language of the simulation because of its maturity and ease of programming CSP based networks of processes.

JCSP implements concurrent programming with processes running and communicating simultaneously. A parallel solution was chosen for a simulation to represent many devices, working autonomously, performing a behaviour depending on a value received from sensors. Devices are autonomous and do not rely on any global synchronisation; devices only synchronise on messages and, therefore, a CSP model is natural to represent the discussed pervasive system. A sensor reacts to change of the light intensity and periodically sends a broadcast signal to all available devices. Components of the presented system decide how to react to the received signal. As a broadcast mechanism is not available in CSP, a repeater was used to ensure that all devices receive a single signal. The broadcast mechanism is fixed to the number of available devices, repeating the message to all available devices. If any of devices is not available to receive the request, the broadcast process is blocked. All devices from the scenario in Figure 1 are CSP processes and use channels to communicate. For the purpose of this simulation, we assume perfect and reliable communication links. The behaviour of the presented system needs to be visualised. All devices send their state to a Graphical Interface (GI) in order to show results of the implemented behaviour. The GI accepts state information from devices on *any-to-one* channel, visualises the data and calculates values for indoor sensors. The GI of the presented simulation was built with *jcsp.awt* [8] active components to enable direct communication between the GI and other processes. The architecture implies broadcast communication between devices, as sensors do not know which devices are interested in the passed light intensity value. The connection between broadcast process and a device is one-to-one to make sure that the massage is delivered. The connection between device and GI is a many-to-one channel enabling all devices to write to GI.

As the control algorithm is implemented in individual lamps and the mirror, the behaviour of the system can only be observed when all devices run simultaneously. CSP has already been successfully used to simulate systems presenting emergent behaviour [12,13,14, 15], showing that a process-oriented approach is suitable for modelling a non-deterministic environment.

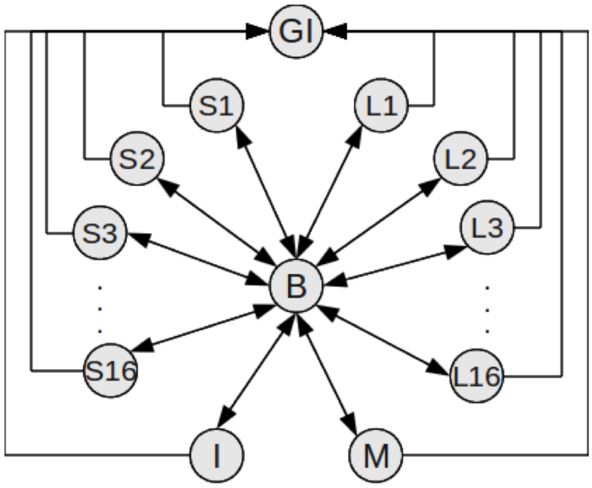

Figure 6. CSP process model diagram for the implemented lighting system.

All 16 lamps (in Figure 6: L1-L16), 17 sensors (in Figure 6: indoor sensors S1-S16, outdoor sensor I) and the mirror (in Figure 6: M) are CSP processes communicating over a broadcast process (in Figure 6: B) using one-to-one and one-to-many channels. Every message sent to the broadcast process is replicated and sent to all devices. Signal is next interpreted according to directions from the Lazy and Enthusiastic Employee algorithm in every device individually. All devices report their state to the graphical interface process (in Figure 6: GI). The first factor of the Lazy and Enthusiastic Employee algorithm is choice of the behaviour curve; according to results presented in Section 4, the chosen curves for lights and mirror are presented in Figure 3. The choice was based on the algorithm's high performance in overall production costs and user comfort, as presented in Table 2. The second factor that we consider in the simulation is the size and location of margins that define the range of reaction of the system. In this implementation, we use variable values of ideal light intensity in a space p_i, where $p_i > 0$, and fixed margin m=50. Therefore, a region $R_1 = [p_i, p_i + 50]$ for mirror and region $R_2 = [p_i - 50; p_i)$ for the lights is used. The regions are excluding ($R_1 \cap R_2 = \emptyset$), therefore lights and mirror stabilise on different values from light sensors. When the light is stable, the mirror is trying to go up and opposite; when the mirror is stable the lights want to dim down. Therefore, the light first reaction is always enthusiastic. This enables the behaviour of taking over the task to illuminate the space. Light can eventually become lazy when there is no need for a fast reaction or the space illumination should be stabilised.

For purpose of the simulation, we use arbitrary light intensity units to express values delivered by light intensity sensors. The ideal intensity, defined by user, is 500 units. We have also assumed that if the light is dimmed to x%, the light sensor senses $10 \cdot x$ units.

6. Experiments

The main goal of the system is to sustain user defined light intensity in a space while maintaining low energy use, using as much natural light as possible. We have performed two experiments using a different control algorithms for a space with 16 lamps and a mirror reflecting light into a ceiling. The aim of these two experiments is to compare energy use for a system with and without dimming control algorithm deployed. The space is divided into 16 regions, both a single lamp and a mirror has an influence on this space, also light from outside is simulated to influence the sensor value with 25%, 12.5%, 10% and 0% of environment's light intensity depending on the distance of regions from the window.

Experiment 1. Lights and mirror react to a sensor value and are designed to sustain user defined level of light. Both mirror and light are willing to accept every request and adapt to it, therefore the Δd is constant, therefore there is no algorithm used to regulate devices behaviour.

Experiment 2. Lights and mirror react to a sensor value and are designed to sustain user defined level of light and use the Lazy and Enthusiastic Employee Algorithm (L&EE Algorithm). The algorithm was implemented as described in Section 4. The simulation was

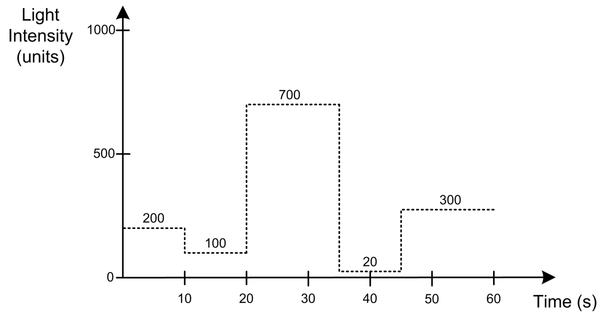

Figure 7. Experiments' input data for light intensity outside.

run for 60 seconds with identical input for all experiments. The light intensity of the environment was changed over time according to Figure 7. We have created a data reference set for proposed experiments. We have measured an energy consumption of 16 lamps that use no dimming, all light sustain constant light level that is 100%.

7. Results and Analysis

The simulation has a graphical interface to present results of experiments (Figure 8).The first part of the simulation GUI shows a 2D model of the considered space. There are 16 lamps on the ceiling and dimming percentage associated with a lamp (Figure 8,D).

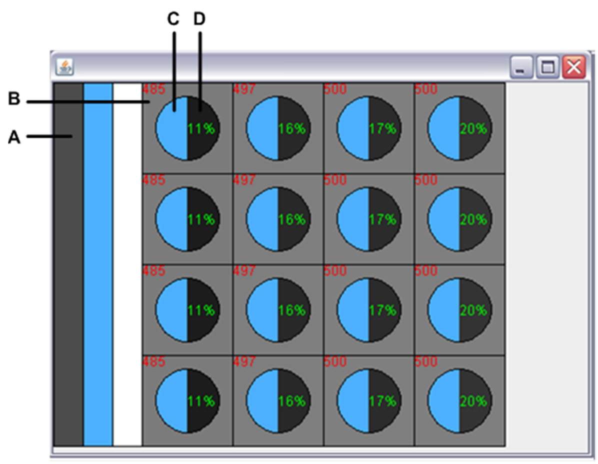

Figure 8. Simulation GUI.

The mirror's illumination is the same for whole room an is represented in the GUI by a half-circle (Figure 8,C). There are 16 regions and 16 sensors, each associated with each region. The value of the sensor in each region is the sum of light from a lamp, mirror and light spread from the window. The value of intensity in agreed units in a region is shown in Figure 8,B. For the purpose of this simulation we calculate the value of the sensor only using intensity from one lamp and mirror. The brightness of the environment due to sunlight is shown in a strip outside the room (Figure 8,A). Colours and brightness of all components in the simulation is adjusted depending on a data received from devices. The simulation real-time data is output to graphs as shown in Figure 9. To simplify the GIU only data from lamp 1 (top-left corner of the room), mirror and sensor 1 associated with region 1 are shown in graphs. The *x* axis in graphs is time measured in milliseconds.

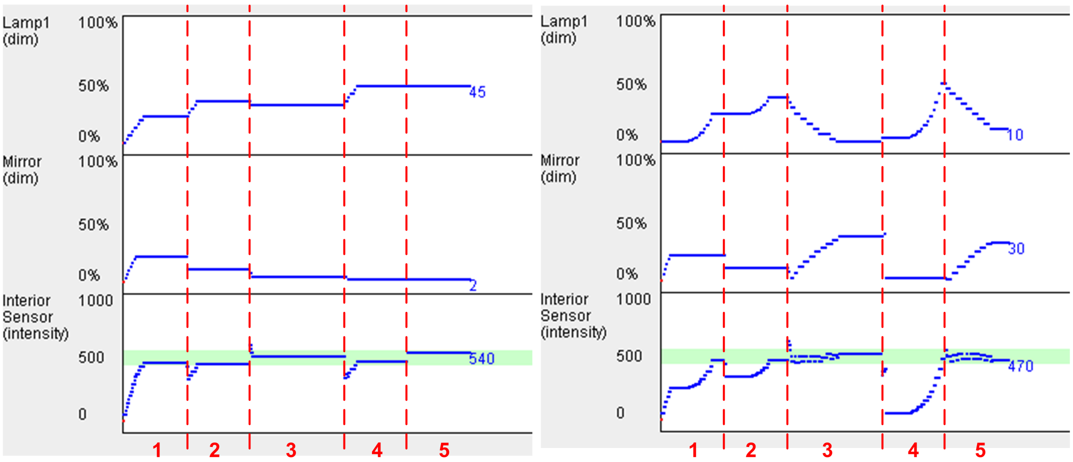

Figure 9. Light, mirror and indoor sensor behaviour for Experiment 1 and 2.

Experiment 1. In this experiment lights react to values from the sensor and try to fulfil the request to increase or decrease the light level in the space. All lights and mirror react the same to sensor values, increasing or decreasing light level with the same factor, therefore no algorithm regulating their behaviour is used. The graphs in Figure 9 are divided into time phases that were described in Figure 7.

In phase 1 light and mirror both are trying to dim up, both light sources stop as soon as they reach the defined range. In phase 2, as light outside decreases, mirror gives away less light and light has to compensate. Both light sources slightly decrease in phase 3. As environmental light decreases to 20 units in phase 4, light takes over lighting the space. In phase 5 both light and mirror are stable as both of them have reached the range. In this experiment we can observe that the mirror is usually not used, as it has a limit over its dimming up. The lamp can dim up easily, therefore it usually takes over lighting the space.

Experiment 2. In this experiment we use L&EE algorithm to control light intensity in the space. The graphs in Figure 9 are divided into the sample five time phases. At the start of the simulation, in phase 1, all the lamps are off, as sensors start sending values to lamps, lamps notice that the light intensity is smaller than defined by the user, therefore they start slowly dimming up. In phase 1 mirror is also willing to dim up, and as the light intensity outside is 200 units mirror fast dim up to 20%. Lamps waits for a while and then slowly starts dimming up to keep the desired light level. In phase 2 the environmental conditions change and intensity decreases to 100 units, the mirror gives less light, therefore lamps have to compensate. Soon lamp 1 becomes stable as the range of the ideal intensity was reached. In phase 3 the light from outside increases to 700 units and mirror takes this opportunity to dim up, meanwhile light notices a possibility to give away less light, therefore it dims down. After a while the mirror takes over lighting up the space and light 1 is off.

Table 3. Energy usage for all experiments.

Algorithm	**Energy usage (J)**
No control, lamps 100% dim	86700
Dimming control, no algorithm	31428
L&EE Algorithm	16019

In phase 4 light from the outside rapidly decreases to 20 units. The mirror stops giving light, so the light bulb is forced to dim up slowly, until the space will be illuminated within the agreed range. In phase 5 light outside increases to 300 units, therefore mirror goes up and lamp 1 dims down to 10%. As the light from outside spreads unevenly in the room. Regions closer to the window are brighter than regions further away from the window. At the end of this experiment lamp 1 is dimmed to only 10%, while lamp 4 (top-right corner of the room) needs to be dimmed to 20%.

Table 4. Energy savings when comparing energy usage from experiment 1 and 2 to the reference data.

Algorithm used	Energy savings (%)
No dimming algorithm	63.8
L&EE Algorithm	81.5

Energy usage results from two experiments are compared in Table 3. The energy usage data is calculated with assumption that all lamps are 100 Watt. From Table 3 we can further calculate the percentage of the energy savings while using L&EE algorithm and experiment with no dimming control algorithm used compared to the reference data. Results are shown in Table 4. Using data reference set and other experiments, we can see that Lazy and Enthusiastic Employee algorithm can reduce energy usage, when there is light outside that can be used to illuminate the space.

8. Further Work and Conclusions

In this paper we have shown an emergent behaviour arising from autonomous lighting system devices. This behaviour is based on a simple model inspired by human society. Process-oriented approach was chosen for representing this non-deterministic environment. We have used CSP to model and JCSP to implement this system of many concurrently executing devices and their message exchanges. The Graphical Interface benefits from use of *any-to-one* input channel for receiving information about devices' state in order to simulate the overall light intensity in the room. The algorithm helps saving energy in spaces with daylight and enables devices that use less energy to take over a task from devices that use more energy without a central control. The algorithm was tested with different parameters and a simulation of a lighting system in an office space was developed in order to show possible energy savings.

The algorithm presented in this paper is tested using simulation, we have chosen arbitrary units for light intensity as we did not use any lighting model. This algorithm can be tested in a real system with actual devices. In this simulation model we also assume that light sensor value is a sum of luminance from light that is above its location, the light distributed by the mirror and the light that gets into the room through the window. In real system, in general a sensor could be be affected by more than one light source but no light distribution model was used in this simulation. In a real system the value from the sensor is a sum of the actual lighting condition in the room, in the simulation, this value is calculated from lamps and light outside, but not according to any lighting model, therefore calculations can be inaccurate. In this paper we described algorithm with only two options for employees: lazy and enthusiastic, but its possible to make whole range of workers and assign them different behaviours using different behaviour curves, as presented in Figure 3.

Acknowledgements

The work presented in this paper and part of the experiments have been performed at NXP Semiconductors, The Netherlands.

References

[1] U.S. Energy Information Administration. Retail sales and direct use of electricity to ultimate consumers by sector, by provider, 1997 through 2008, 2008.

[2] U.S. Energy Information Administration. Electricity consumption (kwh) by end use for all buildings, 2003, 2008.

[3] Kurt W. Ruth and Kurtis McKenney. Energy consumption by consumer electronics in U.S. residences, Final report to the Consumer Electronics Association (CEA) January 2007, 2007.

[4] WorldwatchInstitute. Compact Fluorescent Lamps Could Nearly Halve Global Lighting Demand for Electricity, 2008.

[5] U.S. Department of Energy. Comparison of LEDs to Traditional Light Sources, 2009.

[6] NEN. NEN 2916:2004, Dutch Standards Institute Norm, Energy performance of residential buildings, 2004.

[7] Marco Mamei and Franco Zambonelli. *Field-Based Coordination for Pervasive Multiagent Systems*. Springer, 2006.

[8] P. H. Welch and P. D. Austin. The jcsp home page. http://www.cs.ukc.ac.uk/projects/ofa/jcsp/, 1999.

[9] Peter H. Welch, Neil C.C. Brown, J. Moores, K. Chalmers, and B. Sputh. Integrating and Extending JCSP. In Alistair A. McEwan, Steve Schneider, Wilson Ifill, and Peter Welch, editors, *Communicating Process Architectures 2007*, volume 65 of *Concurrent Systems Engineering Series*, pages 349–370, Amsterdam, The Netherlands, July 2007. IOS Press. ISBN: 978-1-58603-767-3.

[10] Aly A. Syed, Johan Lukkien, and Roxana Frunza. An ad hoc networking architecture for pervasive systems based on distributed knowledge. In *Proceedings of Date2010, Dresden*, 2010.

[11] C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978.

[12] C. G. Ritson and P. H. Welch. A process-oriented architecture for complex system modelling. *Concurr. Comput. : Pract. Exper.*, 22:965–980, June 2010.

[13] Christopher A. Rouff, Michael G. Hinchey, Walter F. Truszkowski, and James L. Rash. Experiences applying formal approaches in the development of swarm-based space exploration systems. *Int. J. Softw. Tools Technol. Transf.*, 8:587–603, October 2006.

[14] Peter H. Welch, Frederick R. M. Barnes, and Fiona A. C. Polack. Communicating complex systems. In *Proceedings of the 11th IEEE International Conference on Engineering of Complex Computer Systems*, pages 107–120, Washington, DC, USA, 2006. IEEE Computer Society.

[15] A.T. Sampson, P.H. Welch, and F.R.M. Barnes. Lazy Simulation of Cellular Automata with Communicating Processes. In J.F. Broenink, H.W. Roebbers, J.P.E. Sunter, P.H. Welch, and D.C. Wood, editors, *Communicating Process Architectures 2005*, volume 63 of *Concurrent Systems Engineering Series*, pages 165–175, Amsterdam, The Netherlands, September 2005. IOS Press. ISBN: 1-58603-561-4.