Journal of Applied Computer Science Methods
Published by University of Social Sciences

# ACSM

Volume 8 Number 1 2016
University of Social Sciences, IT Institute
Społeczna Akademia Nauk
Łódź
Instytut Technologii Informatycznych
ISSN 1689-9636

# International
# Journal of Applied Computer Science Methods

Journal of Applied Computer Science Methods

Published by University of Social Sciences

# ACSM

Volume 8 Number 1 2016

University of Social Sciences, IT Institute

Społeczna Akademia Nauk
Łódź

Instytut Technologii Informatycznych

**INTERNATIONAL JOURNAL OF APPLIED COMPUTER SCIENCE METHODS** (JACSM)
is a semi-annual periodical published by the University of Social Sciences (SAN)
in Lodz, Poland.

PUBLISHING AND EDITORIAL OFFICE:
University of Social Sciences (SAN)
Information Technology Institute (ITI)
Sienkiewicza 9
90-113 Lodz
Tel.: +48 42 6646654
Fax.: +48 42 6366251
E-mail: acsm@spoleczna.pl
URL: https://www.degruyter.com/view/j/jacsm

Print: Mazowieckie Centrum Poligrafii, ul. Słoneczna 3C, 05-270 Marki, www.c-p.com.pl, biuro@c-p.com.pl

---



---

## AIMS AND SCOPE:

The **International Journal of Applied Computer Science Methods is** a semi-annual, refereed periodical, publishes articles describing recent contributions in theory, practice and applications of computer science. The broad scope of the journal includes, but is not limited to, the following subject areas:
**Knowledge Engineering and Information Management:** *Knowledge Processing, Knowledge Representation, Data Mining, Machine Learning, Knowledge-based Systems, Knowledge Elicitation, Knowledge Acquisition, E-learning, Web-intelligence, Collective Intelligence, Language Processing, Approximate Reasoning, Information Archive and Processing, Distributed Information Systems.*
**Intelligent Systems:** *Intelligent Database Systems, Expert Systems, Decision Support Systems, Intelligent Agent Systems, Artificial Neural Networks, Fuzzy Sets and Systems, Evolutionary Methods and Systems, Hybrid Intelligent Systems, Cognitive Systems, Intelligent Systems and Internet, Complex Adaptive Systems.*
**Image Understanding and Processing:** *Computer Vision, Image Processing, Computer Graphics, Pattern Recognition, Virtual Reality, Multimedia Systems.*
**Computer Modeling, Simulation and Soft Computing:** *Applied Computer Modeling and Simulation, Intelligent Computing and Applications, Soft Computing Methods, Intelligent Data Analysis, Parallel Computing, Engineering Algorithms.*
**Applied Computer Methods and Computer Technology:** *Programming Technology, Database Systems, Computer Networks Technology, Human-computer Interface, Computer Hardware Engineering, Internet Technology, Biocybernetics.*

---

## DISTRIBUTION:

Apart from the standard way of distribution (in the conventional paper format), on-line dissemination of the JACSM is possible for interested readers.

# Contents


Yusong Liu, Zhixun Su, Bingjie Zhang,
Xiaoling Gong, Zhaoyang Sang
*Convergence Analysis of an Improved Extreme Learning Machine Based on Gradient Descent Method* 5

Grzegorz Sowa, Alina Marchlewska
*The Internet of Things: Technological and Social Aspects* 17

Zbigniew Filutowicz, Krzysztof Przybyszewski,
Józef Paszkowski
*Selected Examples of Applications of New Graphics and Animation Technologies* 29

Nagesha, Sunilkumar S. Manvi
*Resource Monitoring for Wireless Sensor Networks Using Anfis* 41

Konrad Grzanek
*Monadic Printing Revisited* 69

# Convergence Analysis of An Improved Extreme Learning Machine Based on Gradient Descent Method

Yusong Liu[1,2], Zhixun Su[1], Bingjie Zhang[2], Xiaoling Gong[3], Zhaoyang Sang[2]

[1] School of Mathematical Sciences,
Dalian University of Technology, Dalian 116024, China
*yangguolingfwz@163.com*
*ysliu1758@163.com*

[2] College of Science,
China University of Petroleum (Huadong), Qingdao 266580, China

[3] College of Information and Control Engineering,
China University of Petroleum (Huadong), Qingdao 266580, China

**Abstract**

Extreme learning machine (ELM) is an efficient algorithm, but it requires more hidden nodes than the BP algorithms to reach the matched performance. Recently, an efficient learning algorithm, the upper-layer-solution-unaware algorithm (USUA), is proposed for the single-hidden layer feed-forward neural network. It needs less number of hidden nodes and testing time than ELM. In this paper, we mainly give the theoretical analysis for USUA. Theoretical results show that the error function monotonously decreases in the training procedure, the gradient of the error function with respect to weights tends to zero (the weak convergence), and the weight sequence goes to a fixed point (the strong convergence) when the iterations approach positive infinity. An illustrated simulation has been implemented on the MNIST database of handwritten digits which effectively verifies the theoretical results..

**Key words:** Neural networks, Monotonicity, Weak convergence, Strong convergence, USUA, MNIST.

## 1 Introduction

Neural network has been a hot topic recently in many fields, such as cognitive science, prediction, classification, computational intelligence. The back-propagation (BP) algorithm is one of the most widely used techniques for training feed-forward neural networks (FNN), which was separately proposed by Werbos [1] and Rumelhart et al.[2]. The BP algorithm attempts to minimize the least squared error of objective function, which is defined by the differences between the actual outputs and the desired outputs [3]. In BP algorithm, all the weights of FNN need to be tuned along the negative gradient direction of the error function using the gradient descent method.

The BP algorithm for FNN has the ability of approximating nonlinear functions directly from the input samples. However, the training procedure of the BP algorithm is usually very time consuming. The reasons come from two aspects: (1) the gradient-based learning algorithms are used in training the neural networks, and (2) all weights of the neural networks are tuned in each iteration.

To overcome these shortcomings, Huang et al.[4] proposed a novel learning algorithm called extreme learning machine (ELM) for single-hidden layer feed-forward neural networks (SHLFN), which randomly chooses hidden weights and determines the output weights of SHLFN.

Specifically speaking, in ELM algorithm, the weights connecting the input and hidden layers are selected randomly, and the weights connecting the hidden and output layers are only calculated using the pseudo inverse once. There is no iteration step in the training procedure. In addition, the training speed of ELM is much faster than that of the BP learning algorithms when reaching the comparable performance.

Although ELM can be trained efficiently, it requires more hidden nodes than the BP algorithms for the trained neural networks. This apparently increases testing time which does not effectively work well in real applications.

Yu et al.[5] proposed a series of efficient learning algorithms for SHLFN. The main idea is that, giving the initial weights of FNN, the weights connecting the input and hidden layers are tuned in the negative gradient direction along which the square error is reduced the most, and then the weights connecting the hidden and output layers are calculated using the pseudo inverse. Numerical experiment shows that the proposed algorithms in [5] need less number of hidden nodes and testing time than ELM.

Unfortunately, there is little theoretical analysis to guarantee the convergent behavior during training. In this paper, we rigorously prove the theoretical results for the upper-layer-solution-unaware algorithm (USUA) proposed by Yu et al.[5]. The error function monotonously decreases during training. The weak convergence and the strong convergence show that the gradient of the error function goes to zero, and the weight sequence goes to a unique fixed point, respectively. Numerical experiment on the MNIST database of handwritten digits [6] verifies these theoretical results.

The rest of this paper is organized as follows. Section 2 gives a brief introduction to USUA. Section 3 presents the main theoretical results of USUA. Section 4 rigorously proves these theoretical results. A numerical experiment is simulated in Section 5.

## 2 USUA

The SHLFN is considered. The number of nodes of the input, hidden and output layers are set to be $D$, $L$ and $C$, respectively.

The matrix $\mathbf{W}=(\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_L)\in R^{D\times L}$ represents the weight connections between the input and hidden layers, where $\mathbf{w}_i=(w_{1i},w_{2i},\cdots,w_{Di})^T\in R^D$ is the weight vector connecting the input nodes and the i-th hidden node. $\mathbf{U}=(\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_C)\in R^{L\times C}$ denotes the weight matrix connecting the hidden and output layers, where $\mathbf{u}_i=(u_{1i},u_{2i},\cdots,u_{Li})^T\in R^L$ is the weight vector connecting the hidden nodes and the i-th output node. For simplicity, $\mathbf{W}$ is rewritten as $\mathbf{V}=(\mathbf{w}_1{}^T,\mathbf{w}_2{}^T,\cdots,\mathbf{w}_L{}^T)^T\in R^{DL}$.

Let $g,f:R\to R$ be the given activation functions of the hidden and output layers, respectively. For any given vector $\mathbf{z}=(z_1,z_2,\cdots,z_L)^T\in R^L$, the vector valued function is introduced, denoting as

$$\mathbf{G}(\mathbf{z})=(g(z_1),g(z_2),\cdots,g(z_L))^T\in R^L. \tag{1}$$

For any given output vector $\mathbf{x}\in R^D$, the actual output vector of the neural network is $\mathbf{y}\in R^C$, i.e.

$$\mathbf{y}=f(\mathbf{U}^T\mathbf{G}(\mathbf{W}^T\mathbf{x})).$$

Yu et al.[5] present an efficient and effective algorithm for training SHLFN named USUA. The basic idea of USUA is as follows: when the initial value of **V** and **U** are given, the weight matrix **U** is then fixed, and the weight matrix **V** is updated by using the gradient descend method until it reaches the stop criteria. Then, **U** is calculated using the pseudo inverse. The detailed description is as follows.

Given a training sample set with ***N*** samples, $\mathbf{X}=\{\mathbf{x}_i\}_1^N$ is the set of the input vectors, $\mathbf{T}=\{\mathbf{t}_i\}_1^N$ is the set of the corresponding ideal outputs, and the actual output of the output layer are $\mathbf{Y}=\{\mathbf{y}_i\}_1^N$, where $\mathbf{x}_i\in R^D$, $\mathbf{t}_i\in R^C$, $\mathbf{y}_i\in R^C$. The objective function of the neural networks is defined as follows,

$$\begin{aligned}E(\mathbf{V})&=\frac{1}{2}\left\|\mathbf{Y}-\mathbf{T}\right\|_{\mathrm{F}}^2=\frac{1}{2}\sum_{i=1}^N\left\|\mathbf{y}_i-\mathbf{t}_i\right\|^2\\&=\frac{1}{2}\sum_{i=1}^N\left\|f(\mathbf{U}^T\mathbf{G}(\mathbf{W}^T\mathbf{x}_i))-\mathbf{t}_i\right\|^2\end{aligned}$$

$$=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{C}(f(\mathbf{u}_j^T\mathbf{G}(\mathbf{W}^T\mathbf{x}_i))-t_{ji})^2$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{C}f_{ji}(\mathbf{u}_j^T\mathbf{G}(\mathbf{W}^T\mathbf{x}_i)), \tag{2}$$

where $\|\cdot\|_F$ and $\|\cdot\|$ stand for the Frobenius norm of matrix and the Euclidean norm of vector, respectively, and $f_{ji}(s)=\frac{1}{2}(f(s)-t_{ji})^2, s\in R$.

The gradients of the error function $E(\mathbf{V})$ with respect to $\mathbf{w}_k$ $(k=1,2,\cdots,L)$ are

$$E_{\mathbf{w}_k}(\mathbf{V})=\sum_{i=1}^{N}\sum_{j=1}^{C}f_{ji}^{'}(\mathbf{u}_j^T\mathbf{G}(\mathbf{W}^T\mathbf{x}_i))u_{kj}g^{'}(\mathbf{w}_k^T\mathbf{x}_i)\mathbf{x}_i\ . \tag{3}$$

Denote

$$E_{\mathbf{V}}(\mathbf{V})=((E_{\mathbf{w}_1}(\mathbf{V}))^T,(E_{\mathbf{w}_2}(\mathbf{V}))^T,\cdots,(E_{\mathbf{w}_L}(\mathbf{V}))^T)^T\ . \tag{4}$$

For any given initial weight vector $\mathbf{V}^0$ and $\mathbf{U}$, $\mathbf{V}$ can be iterated by the following formula

$$\mathbf{V}^{n+1}=\mathbf{V}^n+\Delta\mathbf{V}^n,\quad n=0,1,2,\cdots, \tag{5}$$

where $\Delta\mathbf{V}^n=((\Delta\mathbf{w}_1^n)^T,(\Delta\mathbf{w}_2^n)^T,\cdots,(\Delta\mathbf{w}_L^n)^T)^T$, and

$$\Delta\mathbf{w}_k^n=-\eta\sum_{i=1}^{N}\sum_{j=1}^{C}f_{ji}^{'}(\mathbf{u}_j^T\mathbf{G}(\mathbf{W}^T\mathbf{x}_i))u_{kj}g^{'}(\mathbf{w}_k^T\mathbf{x}_i)\mathbf{x}_i\ , \tag{6}$$

where $\eta>0$ is the learning rate.

At last, **U** is calculated using the pseudo inverse.

## 3 The main convergence results

To analyze the convergence of USUA, the following assumptions are needed.

(A1)The activation functions $g$ and $f$ satisfy that, $|g(s)|$, $|f(s)|$, $|g^{'}(s)|$, $|f^{'}(s)|$, $|g^{"}(s)|$ and $|f^{"}(s)|$ are all uniformly bounded for any $s\in R$.

(A2) There are finitely many points in the set $\Omega_0=\{\mathbf{V}\in\Omega: E_{\mathbf{V}}(\mathbf{V})=0\}$, where $\Omega$ is a bounded closed region.

**Theorem 1.** Assume that assumption (A1) is valid, and the learning rate $\eta$ satisfies the formula (21) behind. Then, for any arbitrary initial weight vector $\mathbf{V}^0$, the sequence $\{E(\mathbf{V}^n)\}$ monotonously decreases, i.e.

$$E(\mathbf{V}^{n+1}) \le E(\mathbf{V}^{n}); \tag{7}$$

there exists $E^* \ge 0$, such that

$$\lim_{n\to\infty} E(\mathbf{V}^{n}) = E^*; \tag{8}$$

and the weak convergence result holds,

$$\lim_{n\to\infty} \left\| E_{\mathbf{V}}(\mathbf{V}^{n}) \right\| = 0 . \tag{9}$$

In addition, if assumption (A2) is also valid, then the strong convergence result holds, i.e. there exists $\mathbf{V}^* \in \Omega_0$, such that

$$\lim_{n\to\infty} \mathbf{V}^{n} = \mathbf{V}^* . \tag{10}$$

## 4 The Proofs

The proofs of the convergence results (Theorem 1) are presented as follows. Firstly, two useful lemmas are given. For sake of consistency, denote

$$\Delta \mathbf{w}_k^n = \mathbf{w}_k^{n+1} - \mathbf{w}_k^n , \tag{11}$$

$$\mathbf{G}^{n,i} = \mathbf{G}((\mathbf{W}^n)^T \mathbf{x}_i), \ \boldsymbol{\varphi}^{n,i} = \mathbf{G}^{n+1,i} - \mathbf{G}^{n,i} . \tag{12}$$

**Lemma 1.** If assumption (A1) is valid, then there exist $c_1 > 0$ and $c_2 > 0$, satisfying

$$\left\|\boldsymbol{\varphi}^{n,i}\right\|^2 \le c_1 \sum_{k=1}^{L} \left\|\Delta \mathbf{w}_k^n\right\|^2, \quad i = 1,2,\cdots,N, \ n = 1,2,\cdots, \tag{13}$$

$$\left|f_{ji}^{'}(s)\right| \le c_2, \left|f_{ji}^{''}(s)\right| \le c_2, \ s \in R, i = 1,2,\cdots,N, \ j = 1,2,\cdots,C . \tag{14}$$

**Proof.** According to assumption (A1) and the Taylor expansion, we get

$$\left\|\boldsymbol{\varphi}^{n,i}\right\|^2 = \left\|\mathbf{G}^{n+1,i} - \mathbf{G}^{n,i}\right\|^2 = \left\| \begin{pmatrix} g((\mathbf{w}_1^{n+1})^T \mathbf{x}_i) - g((\mathbf{w}_1^{n})^T \mathbf{x}_i) \\ g((\mathbf{w}_2^{n+1})^T \mathbf{x}_i) - g((\mathbf{w}_2^{n})^T \mathbf{x}_i) \\ \vdots \\ g((\mathbf{w}_L^{n+1})^T \mathbf{x}_i) - g((\mathbf{w}_L^{n})^T \mathbf{x}_i) \end{pmatrix} \right\|^2$$

$$= \left\| \begin{pmatrix} g^{'}(s_{1,i,n}) \Delta(\mathbf{w}_1^n)^T \mathbf{x}_i \\ g^{'}(s_{2,i,n}) \Delta(\mathbf{w}_2^n)^T \mathbf{x}_i \\ \vdots \\ g^{'}(s_{L,i,n}) \Delta(\mathbf{w}_L^n)^T \mathbf{x}_i \end{pmatrix} \right\|^2$$

$$\leq (\sup_{s\in R}\left|g^{'}(s)\right|\max_{1\leq i\leq N}\|\mathbf{x}_i\|)^2\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2$$

$$= c_1\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2,$$

where $c_1 = (\sup_{s\in R}\left|g^{'}(s)\right|\max_{1\leq i\leq N}\|\mathbf{x}_i\|)^2$, and $s_{k,i,n}(k=1,2,\cdots,L)$ lies between $(\mathbf{w}_k^{n+1})^T\mathbf{x}_i$ and $(\mathbf{w}_k^n)^T\mathbf{x}_i$.

By the expression of $f_{ji}(s)$ and assumption (A1), it is easily known that

$$\left|f_{ji}^{'}(s)\right|\leq c_2,\ \left|f_{ji}^{"}(s)\right|\leq c_2,\ \ i=1,2,\cdots,N,\ \ j=1,2,\cdots,C,\ \ s\in R,$$

where $c_2 = \max\{\sup_{s\in R}\left|(f(s)-t_{ji})f^{'}(s)\right|,\ \sup_{s\in R}\left|(f^{'}(s))^2+(f(s)-t_{ji})f^{"}(s)\right|\}$.

The following lemma is the same as Theorem 14.1.5 [7], therefore we only list it below without proof.

**Lemma 2.** [7] Let $F:\Omega\subset R^n\rightarrow R^m(n,m\geq 1)$ be continuous on a bounded closed region $\Omega\subset R^n$, and $\Omega_0=\{\mathbf{z}\in\Omega:F(\mathbf{z})=0\}$ be a finite set. Let $\{\mathbf{z}^k\}\subset\Omega$ be a sequence satisfying

(1) $\lim_{k\to\infty}F(\mathbf{z}^k)=0$,

(2) $\lim_{k\to\infty}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|=0$,

then, there exists a $\mathbf{z}^*\in\Omega_0$ such that $\lim_{k\to\infty}\mathbf{z}^k=\mathbf{z}^*$.

Next, the proofs for (7)-(10) are successively presented as follows.

**Proof for (7).**

By (2) and the Taylor expansion, we have

$$\begin{aligned}
&E(\mathbf{V}^{n+1})-E(\mathbf{V}^n)\\
&=\sum_{i=1}^{N}\sum_{j=1}^{C}[f_{ji}(\mathbf{u}_j^T\mathbf{G}^{n+1,i})-f_{ji}(\mathbf{u}_j^T\mathbf{G}^{n,i})]\\
&=\sum_{i=1}^{N}\sum_{j=1}^{C}[f_{ji}^{'}(\mathbf{u}_j^T\mathbf{G}^{n,i})\mathbf{u}_j^T(\mathbf{G}^{n+1,i}-\mathbf{G}^{n,i})+\frac{1}{2}f_{ji}^{"}(\tilde{s}_{n,i})(\mathbf{u}_j^T(\mathbf{G}^{n+1,i}-\mathbf{G}^{n,i}))^2]\\
&=\delta_0+\delta_1,
\end{aligned}\tag{15}$$

where $\delta_0 = \sum_{i=1}^{N}\sum_{j=1}^{C} f_{ji}^{'}(\mathbf{u}_j^T\mathbf{G}^{n,i})\mathbf{u}_j^T(\mathbf{G}^{n+1,i} - \mathbf{G}^{n,i})$,

$\delta_1 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{C} f_{ji}^{''}(\tilde{s}_{n,i})(\mathbf{u}_j^T(\mathbf{G}^{n+1,i} - \mathbf{G}^{n,i}))^2$, and $\tilde{s}_{n,i}$ lies between $\mathbf{u}_j^T\mathbf{G}^{n+1,i}$ and $\mathbf{u}_j^T\mathbf{G}^{n,i}$.

By the Taylor expansion and (6),

$$\delta_0 = \sum_{i=1}^{N}\sum_{j=1}^{C} f_{ji}^{'}(\mathbf{u}_j^T\mathbf{G}^{n,i})[\sum_{k=1}^{L} u_{kj}(g((\mathbf{w}_k^{n+1})^T\mathbf{x}_i) - g((\mathbf{w}_k^{n})^T\mathbf{x}_i))]$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{C} f_{ji}^{'}(\mathbf{u}_j^T\mathbf{G}^{n,i})[\sum_{k=1}^{L} u_{kj}g^{'}((\mathbf{w}_k^{n})^T\mathbf{x}_i)(\Delta\mathbf{w}_k^n)^T\mathbf{x}_i + \frac{1}{2}\sum_{k=1}^{L} u_{kj}g^{''}(\bar{s}_{k,i,n})((\Delta\mathbf{w}_k^n)^T\mathbf{x}_i)^2]$$

$$= -\frac{1}{\eta}\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2 + \delta_2, \qquad (16)$$

where $\delta_2 = \frac{1}{2}\sum_{k=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{C} f_{ji}^{'}(\mathbf{u}_j^T\mathbf{G}^{n,i})u_{kj}g^{''}(\bar{s}_{k,i,n})((\Delta\mathbf{w}_k^n)^T\mathbf{x}_i)^2$, and $\bar{s}_{k,i,n}$ lies between $(\mathbf{w}_k^{n+1})^T\mathbf{x}_i$ and $(\mathbf{w}_k^{n})^T\mathbf{x}_i$.

By (15) and (16),

$$E(\mathbf{V}^{n+1}) - E(\mathbf{V}^n) = -\frac{1}{\eta}\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2 + \delta_2 + \delta_1. \qquad (17)$$

As (15), $\mathbf{U}$ is fixed and the triangle inequality,

$$\delta_1 \le \frac{1}{2}c_2\sum_{i=1}^{N}\sum_{j=1}^{C}\left\|\mathbf{u}_j^T\boldsymbol{\varphi}^{n,i}\right\|^2$$

$$\le \frac{1}{2}c_2\sum_{i=1}^{N}\sum_{j=1}^{C}(\max_{1\le j\le C}\left\|\mathbf{u}_j\right\|)^2 c_1\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2$$

$$= \frac{1}{2}c_2NCc_1(\max_{1\le j\le C}\left\|\mathbf{u}_j\right\|)^2\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2$$

$$= c_3\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2, \qquad (18)$$

where $c_3 = \frac{1}{2}c_2NCc_1(\max_{1\le j\le C}\left\|\mathbf{u}_j\right\|)^2$.

By assumption (A1) and (14),

$$\delta_2 \le \frac{1}{2}\sum_{k=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{C} c_2\max_{1\le j\le C}\left\|\mathbf{u}_j\right\|\sup_{s\in R}\left|g^{''}(s)\right|\max_{1\le i\le N}\left\|\mathbf{x}_i\right\|^2\left\|\Delta\mathbf{w}_k^n\right\|^2$$

$$=\frac{1}{2}NCc_2\max_{1\le j\le C}\left\|\mathbf{u}_j\right\|\sup_{s\in R}\left|g''(s)\right|\max_{1\le i\le N}\left\|\mathbf{x}_i\right\|^2\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2$$

$$=c_4\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2, \tag{19}$$

where $c_4=\frac{1}{2}NCc_2\max_{1\le j\le C}\left\|\mathbf{u}_j\right\|\sup_{s\in R}\left|g''(s)\right|\max_{1\le i\le N}\left\|\mathbf{x}_i\right\|^2$.

Therefore, by (17), (18) and (19)

$$\begin{aligned}&E(\mathbf{V}^{n+1})-E(\mathbf{V}^n)\\ &\le-\frac{1}{\eta}\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2+c_4\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2+c_3\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2\\ &=-(\frac{1}{\eta}-c_5)\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2\\ &=-\alpha\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2,\end{aligned} \tag{20}$$

where $c_5=c_3+c_4$, $\alpha=\frac{1}{\eta}-c_5$. Set

$$0<\eta<\frac{1}{c_5}, \tag{21}$$

then, $E(\mathbf{V}^{n+1})\le E(\mathbf{V}^n)$. The monotonicity is proved.

**Proof for (8).**

For any $n=0,1,2,\cdots$, $E(\mathbf{V}^n)\ge 0$. Then, by (7), the sequence $\{E(\mathbf{V}^n)\}$ monotonously decreases. Therefore, there exists $E^*\ge 0$ such that $\lim_{n\to\infty}E(\mathbf{V}^n)=E^*$.

**Proof for (9).**

By (20), we obtain

$$\begin{aligned}E(\mathbf{V}^{n+1})&\le E(\mathbf{V}^n)-\alpha\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2\\ &\le E(\mathbf{V}^{n-1})-\alpha\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^{n-1}\right\|^2-\alpha\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^n\right\|^2\\ &\le\cdots\le E(\mathbf{V}^0)-\alpha\sum_{i=0}^{n}\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^i\right\|^2.\end{aligned}$$

For any $n\ge 0$, we have $E(\mathbf{V}^n)\ge 0$. Therefore, $\alpha\sum_{i=0}^{n}\sum_{k=1}^{L}\left\|\Delta\mathbf{w}_k^i\right\|^2\le E(\mathbf{V}^0)$.

By (3), (4) and (6), taking $n \to \infty$, and changing indexes,

$$\alpha \sum_{n=0}^{\infty} \sum_{k=1}^{L} \left\| \Delta \mathbf{w}_k^n \right\|^2 = \alpha \eta^2 \sum_{n=0}^{\infty} \left\| E_{\mathbf{V}}(\mathbf{V}^n) \right\|^2 \le E(\mathbf{V}^0) < \infty .$$

Then, we have $\lim_{n\to\infty} \left\| E_{\mathbf{V}}(\mathbf{V}^n) \right\| = 0$ . The weak convergence is proved.

**Proof for (10).**
By (3)-(6),

$$\left\| \mathbf{V}^{n+1} - \mathbf{V}^n \right\| = \left\| \Delta \mathbf{V}^n \right\| = \eta \left\| E_{\mathbf{V}}(\mathbf{V}^n) \right\|.$$

Thus, using (9), we get

$$\lim_{n\to\infty} \left\| \mathbf{V}^{n+1} - \mathbf{V}^n \right\| = 0 .$$

By (A2), the conditions of lemma 2 are valid. Therefore, there exists $\mathbf{V}^* \in \Omega_0$ satisfying $\lim_{n\to\infty} \mathbf{V}^n = \mathbf{V}^*$ .The strong convergence is proved.

## 5 Numerical experiment

The MNIST database of handwritten digits contains 60,000 training samples and 10,000 testing samples. Each digital image has been normalized to an image 28×28 pixels, and expanded as a 784×1 vector. The elements of these digital vectors are the integer numbers between 0-255.

According to the property of MNIST, we construct a network model whose structure is set to be 784-128-10. The learning rate is selected as a constant 0.0007. The activation functions of hidden and output layers are with the common sigmoid function $g(x) = \frac{1}{1+e^{-x}}$ and the linear function, respectively. The initial weights are randomly assigned in the interval $[-1,1]$ . The stop criteria are set to be: 1,000 training epochs or the error below 0.01.

Figure 1 and Figure 2 display the classification ability of the USUA on training and testing samples. To show the details clearly, the accuracies are recorded for each training epoch. We observe that the USUA has the similar performance on both training and testing samples. In addition, the two curves drastically increase in the early training stage and then maintain with a stable status.

In Figure 3, it shows the error values of each training epoch. Corresponding to the training performance in Figure 2, the errors sharply decrease in the early training epochs, and the approach the minimum. This effectively verifies the monotonicity of error function which is proved in Theorem 1.

For the last Figure 4, the norms of the gradient of error function with respect to weight vectors have been graphed along with epochs. Although the curve shows the oscillation behavior in the training process, it still demonstrates that the norms tend to small values near zero along with the increasing epochs. This then illustrates the proved weak convergence of USUA in Theorem 1.

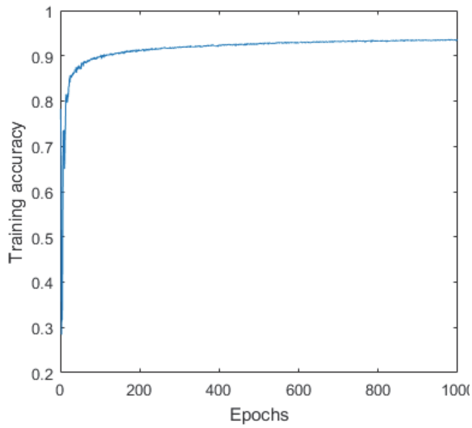



**Figure 1.** The curve of training accuracy

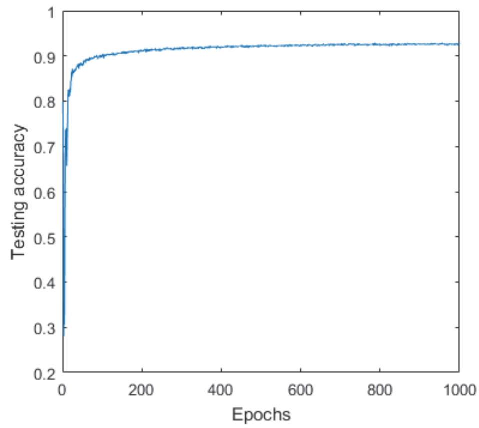



**Figure 2.** The curve of testing accuracy

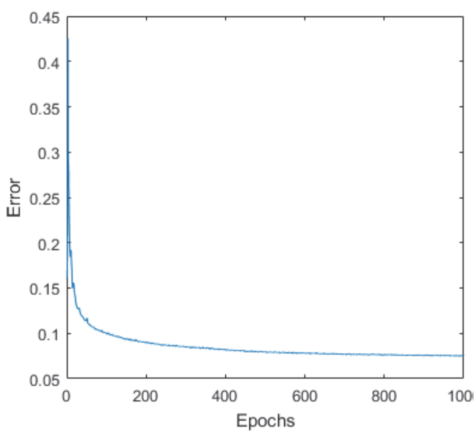



**Figure 3.** The curve of error function

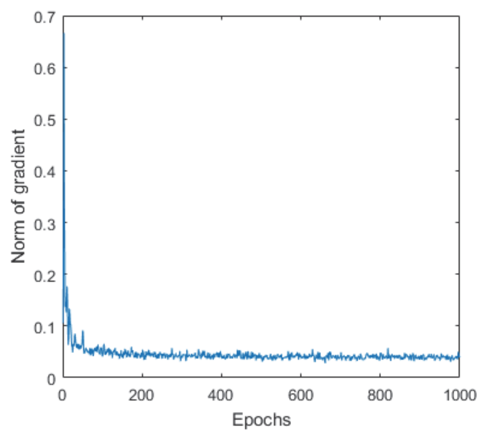



**Figure 4.** The norms of the gradient of error function with respect to weight vectors

## 6 Conclusion

In this paper, we mainly rigorously prove the theoretical results of USUA proposed by Yu et al.[5], including the monotonicity of error function, the weak and strong convergence. The error function monotonously decreases in the training procedure. The weak and strong convergence indicate that the gradient of the error function with respect to weights tends to zero and the weight sequence goes to a fixed point when the iterations approach positive infinity, respectively. Numerical experiment on the MNIST database of handwritten digits support these theoretical results.

## Acknowledgments

The authors wish to thank the anonymous reviewers for careful error proofing of the manuscript and many insightful comments and suggestions which greatly improved this work.

This project was supported in part by the National Natural Science Foundation of China (No. 61305075, 61173103, 61572099, 61320106008, 91230103), the China Postdoctoral Science Foundation (No. 2012M520624), Natural Science Foundation of Shandong Province (No. ZR2013FQ004, ZR2013DM015), the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20130133120014) and the Fundamental Research Funds for the Central Universities (No. 15CX05053A, 15CX08011A).

## References

1. Werbos, P. J., 1974, *Beyond regression: new tools for prediction and analysis in the behavioral sciences*, Ph.D. thesis, Harvard University, Cambridge, MA
2. Rumelhart D. E., Hinton G. E., Williams R. J., 1986, *Learning representations by back-propagating errors*, Nature, Vol. 323, pp. 533-536
3. J.H. Goodband, O.C.L. Haas, J.A. Mills, 2008, *A comparison of neural network approaches for on-line prediction in IGRT*, Medical Physics, Vol. 35, No. 3, pp. 1113–1122
4. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., 2006, *Extreme learning machine: theory and applications*, Neurocomputing, Vol. 70, No. 1-3, pp. 489-501
5. D. Yu and L. Deng, 2012, *Efficient and effective algorithms for training single-hidden-layer neural networks*, Pattern Recognition Letters, Vol. 33, No. 5, pp. 554–558
6. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, 1998, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278-2324
7. Y. Yuan, W. Sun, 2001, *Optimization Theory and Methods*, Science Press, Beijing

# THE INTERNET OF THINGS: TECHNOLOGICAL AND SOCIAL ASPECTS

Grzegorz Sowa, Alina Marchlewska

IT Institute, University of Social Sciences, Łódź, Poland
*g22sowa@gmail.com, amarchlewska@spoleczna.pl*

**Abstract**

The basic idea of is the Internet of Things is presence around us of a variety of things – such as RFID tags, sensors, actuators, mobile phones, etc. – which are able to interact with each other and cooperate with their neighbors to reach common goals. Semantic oriented IoT visions have also been proposed. The number of items involved in the future Internet is destined to become extremely high. Therefore, issues pertaining how to represent, store, interconnect, search, and organize information generated by the IoT will become very challenging. In this context, semantic technologies will play a key role. And creative approaches to visualizing data – humans are far better than computers at seeing patterns –frequently prove integral to the process of creating knowledge. From a privacy perspective, IoT is challenging because it operates in private settings, like homes, and presents an attack target that is harder to secure.

**Key words:** Internet of Things, RFID systems, Big Data, privacy, user interface

## 1 Introduction

The development of information technology has brought about universal information access and measurements. This can be explained by way of a simple example: the measurement of time. We often continue to wear watches; clocks hang on the wall largely for decorative purposes. Yet time is displayed on mobile phones as a basic non-core function. And it is also displayed on electric ovens, microwaves, computer screens, television screens, television decoders, weather stations, in the car – wherever we look, we will see the current time. The everyday items that surround us have functions that cross over, and they are driven by electronic devices of a level of complexity similar to that of computers. The items are often connected to external networks – even a simple weather station may be synchronising with a central clock somewhere in Europe. And of course computer-smartphone-tablets make automatic updates not only of the date and time but of their operating systems or anti-virus protection.

Another noteworthy characteristic is the connection of these devices to the internet, which enables them to communicate with each other. This creates completely new technological potential; we are not far away from a refrigerator which can let us know by mobile phone when its stock of a product has fallen below a prescribed minimum, allowing you to make that same order in a shop. This is what the Internet of Things looks like, and it is fast becoming a reality. There are several features of this phenomenon that are worth a closer look. Our starting point is with the technical solutions, which are already quite advanced, although some standards are still to be developed. One interesting aspect here is the problem of analysing the large amounts of data which are a by-product of these systems - the problem of ‘big data’ takes on a new dimension. Another issue is that of the interface. All the devices have an interface – even if it is only a standard time display. These need to be carefully designed – and this must be done from the starting-point of the needs and capabilities of users and not of the limitations of hardware or software which are ever-decreasing. Today’s interface designers must be psychologists, sociologists and artists and not just specialists. And lastly let us not omit to consider security issues and the protection of personal data.

## 2 The concept of the Internet of Things

The Internet of Things (IoT) is formed as a result of the presence around us of a variety of devices such as RFID tags, sensors, actuators and smartphones, which when covered in a uniform addressing scheme can work together pursuing common tasks. Linking the sensors and control systems of the various devices creates an opportunity to exchange information between different platforms with a uniform standard. This brings potential for the development of innovative systems. Such an objective is achieved via comprehensive measurement, data analysis and information presentation, usually with the use of cloud computing [Atzori, 2010, p.1].

## 3 The technology and architecture of the Internet of Things

As is the case in conventional computer networks, the Internet of Things is built in layers. At the lowest level, that of hardware, we have firstly the measuring and implementing components: RFID tags and implanting components (actuators) associated with wireless WSN (Wireless Sensor Network) networks and NFC (Near Field Communication) networks. The key elements are the ‘spims’ – self-sufficient objects whose positions can be tracked over time. This category includes ‘smart items’ which provide wireless connectivity and memory and that are autonomous in action [Fritzche, 2015, p. 20].

Such equipment fulfils basic functions:
- Identification and storage of information (RFID)
- Communication (WNS)
- Information processing
- Physical communication, control and operations

A general overview of the layers of the Internet of Things can be presented as follows:
- Application layer
- Intermediary layer (‘middleware’)
- Internet layer
- Access goals layer (‘access gateway’)
- Technology layer (‘hardware’)

The intermediary layer - ‘middleware’ - provides an interface between the hardware and the applications. In recent years, there has been a proliferation of service-orientated architecture (SOA), which allows the creation of applications with well-defined components. This layer is responsible for the management of devices and information, the filtering and aggregation of data, semantic analysis, and access control.

The application layer includes various relevant solutions e.g. in the fields of logistics, healthcare, environmental protection [Bandyopadhyay, 2011, p. 5].

The internet of the future will be very different to the one that we know today. Currently, communication is generally host-to-host, and we mainly use the network for acquisition or publication of information. However, the networks of the future will be built around the data itself: this assumes that both the data and queries to it will be self-addressing and self-routable (have built-in protocols). Such features already exist of course in the object-orientated programming era. Research carried out by Koponen’s team suggested the need for change in addressing practises in the Internet. They proposed replacing the hierarchical DNS system into a ‘flat’ one, in which names and addresses would include everything above the IP layer; each object (‘thing’) would have its own address and be available on the Internet. This proposal is called DONA: Data-Orientated Network Architecture [Koponen, 2010, p. 2]. Internet Ø is spoken of differently here, in terms of a transformation of today’s ‘internet devices’ in the ‘Internet of Things’. The IPSO (Internet Protocol for Smart Objects) was developed by this team.

A recent UN report [Botterman, 2009] states that in the coming era humans will be a minority of recipients and senders of data on the Internet; most communication will be between smart objects. The basic concepts in the field of the Internet of Things are the oriented network and object, but the overriding approach (semantic-orientated) focuses on meaning [Botterman, 2009, p.

8]. While network and web approaches are focussed on protocols and programming languages, the semantic approach focuses on the meaning of the data and how the information can be represented.

The semantic layer includes [Atzori, 2010, p. 3]:

- Semantic technologies
- Inferences based on data
- Semantic environmental regulations
- Semantic middleware

Two components made possible the early applications of the Internet of Things: the RFID - Radio Frequency Identification and the WSN - Wireless Sensor Network. RFID is still the key technology due to technological maturity, low costs, and strong resulting support enterprises [Pang, 2013, p. 43]. Passive and active RFID chips are widely used in logistics and are gradually replacing bar codes in commerce. WSN networks combine sensors and implanting components (actuators) in the network and integrate them with higher level systems throughout the net. However it is already apparent that a wider spectrum of technologies supporting the Internet of Things has been created: Near Field Communications (NFC) and Wireless Sensor and Actuator Networks (WSAN) are now elementary components that combine the world of things with the digital world. The development of supporting platforms such as Wireless Identification and Sensing Platforms (WISP) is significant.

Sensor nodes transmit their data (location, temperature, movement) to the so-called ‘sink’ – a special node that collects information. Such measurement nodes are lightweight, inexpensive and easy to install and maintain but their capabilities are limited. Their constituent elements are: power, processing unit, communication and sensing element. It is possible to create a network of thousands of sensors that gathers, processes and analyses information. The principal relevant issues are: energy efficiency, scalability, reliability and resistance to interference.

Sensors gather information about objects and processes and events. Technically they are transducers which convert physical signals into electronic impulses. Sensors link the physical world to digital measurements of real processes and events, and convert them into a form that can be stored and processed. Placing a large number of sensors in many places brings great benefits – from improvements in logistics to enhanced levels of safety. Sensor networks sometimes contain implementing components that manipulate the real world, such as boiler switches.

There is a distinction between active sensors that are equipped with an external power supply, and passive ones that obtain their energy from the measured object e.g. by infrared waves. Another classification is for sensors that are resistive, capacitive, inductive or piezoelectric.

A comparison of technologies is presented in this table:

**Table 1.** Comparison between RFID systems, wireless sensor networks, and RFID sensor networks. Source: [Atzori 2010, p. 5].

| technolo-gy | pro-cessing | collecting of information | communica-tion | range (metres) | power | durabil-ity | Size | standard |
|---|---|---|---|---|---|---|---|---|
| RFID | No | No | assymetric | 10 | harvested | unlim-ited | very small | ISO 1 8000 |
| WSN | Yes | Yes | peer-to-peer | 100 | battery | up to 3 yrs | small | IEEE 802.15.4 |
| RSN | Yes | Yes | assymetric | 3 | harvested | unlim-ited | small | none |

Smart objects are those objects in our surroundings which we can identify – address and communicate with. Examples of such objects are [Nguyen, 2015, p. 58]:

- Smart key: contains an RFID tag, so that it is possible, for example, among other functions, to find its location
- Smart lighting: linked to a wireless controller that allows different settings, such as on/off
- Smart plate: able to identify the types of meals that are placed on it and to transmit this information to a database about the dietary behaviour of the user
- Smart air conditioning: scans data on the number of people in a room thus facilitating automatic temperature control
- Smart fridge: keeps track of stored products thus enabling automatic re-ordering [The Only Fridge as Smart as You, 2015]

## 4 Typical applications

In transport and logistics, communications and sensors (video, sonar, radar, induction loops and magnetometers) that enable the exchange of information between cars, and between cars and the store, are already in common use. They enable traffic jams and journey times to be reduced. Sensors placed in cars and trains, and on their routes, can provide information to a collision avoidance system, monitor the transport of hazardous materials and help in directing vehicles (assisted driving). Each step of the logistics chain (design, purchasing, production, transportation, storage, sales, after-sales service) can be monitored and reinforced with the help of RFID tags. Real-time visibility of a company's operations and customers can bring about more efficient supply chain management. It is possible to determine the location of goods being transported, their status, time of delivery, and any delays or errors [Xiao, 2011, p. 4].

An example within the field of environmental monitoring would be the tracking of perishable food goods at every stage (collection, processing, transportation, storage and consumption) to ensure supply chain efficiency. Temperature and humidity can be monitored/controlled at every stage. Another important application is tracking of the status of major engineering structures such as bridges or pipelines, which facilitates their maintenance. The incorporation of sensors that enable fast collection of measurement data in hard to reach places is possible during the construction phase.

In the field of healthcare, the ability to identify moving objects and people (e.g. children) brings about significant safety improvements. Identifying patients prevents confusion and provides an up-to-date treatment record. Specialised indicators of temperature, pressure and breathing can be used to monitor patients and, in combination with wireless networks, provide information about their health status in real time. This information can be displayed where it is needed, reducing the number of medical errors. Most of the sensors used in medicine are placed on the bodies of patients and it is possible to use them in patients' own homes, allowing them a safe life under medical supervision.

Recently, attention has been focussed on a rather startling issue. The aim of protecting individuals' privacy is straightforward; rather murkier is the use of so-called persuasion technologies which shape correct behaviour (from the point of view of, for example, patient care or the environment [Fogg, 2009]. A simple example of such a technology is already prevalent; beeping that reminds us of the need to fasten seat-belts in cars or dashboard display of information about current fuel consumption. The idea here is to influence the behaviour of users ('educating them') without compulsion or forcing them to do anything. Another example, and an amusing one, is a doll which behaves like a baby and demands constant attention in order to discourage teens from early parenthood [Fogg, 2009, str. 4].

## 5 The Internet of Things and Big Data

A problem specific to the Internet of Things is the huge amount of data collected by different measuring devices. Such data collection and, particularly, sound interpretation of the data is no longer possible using traditional tools – we are dealing with the issue of 'Big Data'. The growth in data has been unimaginable (in 2012 data from sensors was ten times greater in volume than total Internet traffic in 2000 [Richards, 2013, p. 1]).

However, this increase in the amount of collected and processed data is only an external symptom. More significant is the change in approaches to the calculation of results. Traditional experimental science is based around controlled experiments; current approaches are about recognizing patterns within chaotic incoming data and seeking to understand them. This approach is referred to as 'data dredging'. In other words, we make the assumption that there is information lying within the data, even if we cannot yet understand it. Counter-intuitively, an active role for human operators here is often critical. 'A creative approach to visualising data – people are much better at pattern recognition than computers – often proves to be crucial in the creation of knowledge' [Shaw, 2014 p. 30].

Examples of meaningful discoveries of completely non-obvious patterns of behaviour have been given e.g. the deduction that a woman is pregnant based on the type of deodorant she purchases.

There is an interesting analogy here to another field of 'big data' – that of the analysis of social media. 'The huge amount of information created by Internet users offers new potential for social analysis. The tools for analysing this data are still just beginning to be developed. Tools for monitoring social networking sites are already a major source of marketing information. However, we are dealing with a form of digital divide; only the reviews of customers who are online matter. From the standpoint of social marketing that's simply an issue for commerce, but unrepresentative social research (e.g. electoral polling) is a political problem. The issue is of course well-known and it has been noticed for a long time that, for example, telephone users are not a good representative sample. Yet these days the temptation to limit oneself to analysing more easily available digital information is even greater.' [Sowa, 2012, p. 13]

## 6 Protecting privacy in the era of the Internet of Things

RFID tags attached/worn by users create a completely new social world. When meeting someone we can automatically exchange information straightaway, we can provide live updates to social networking data about our location and activities. Such data can be used to communicate with other users or be saved as a digital diary. Of course, access to such information should be restricted. There are already sophisticated algorithms in practise which allow, for example, the identification of a person based on her behaviour patterns: how she moves, how she uses a mouse, etc. [Ziegler, 2015, p. 102].

Another example of an application is in security. We can find lost objects; a user can also be kept informed when an object (e.g. a valuable painting) changes its usual position.

Unfortunately, the Internet of Things is extremely sensitive to hacker attack [Atzori, 2010, p. 2]. Firstly, most of its components are inactive most of the time; they are also readily physically accessible. Wireless communication is easy to wiretap and most of the hardware components have little computing capability and so cannot effectively use sophisticated programming methods as a defence. Authentication is difficult – that requires infrastructure and servers that exchange information – it is hard to imagine it in place for RFID tags at present. Similarly, it is difficult to control for accurate data integrity. Several solutions have been proposed, based generally on the principle of a hierarchy of nodes, with those at higher levels treated as normal Internet nodes and made subject to normal authentication procedures.

Privacy and data protection are currently at the forefront of attention, and policy is 'privacy at the design stage' (Privacy by design – PbD). Internet of Things system designers are required to incorporate personal data protection considerations as early as possible, especially in the field of smart homes, and before the system is completely constructed [Urquart, 2016, p. 7]. What is essential here is the time and manner in which legal regulations will be made. This pyramid of regulatory strategies is interesting [Urquart, 2016, p. 13]:

- Self-regulation
- Forced self-regulation
- Organised self-regulation with discreet penalty
- Organised self-regulation with public penalty

And a pyramid of sanctions:

- Persuasion
- Letter of warning
- Civil trial
- Criminal trial
- Suspension of licence
- Removal of license

While learning from previous experience we are currently trying to keep up with current regulations for the development of the technology. In October 2014 a conference in Mauritius adopted a resolution on Big Data and the Internet of Things. The main recommendations were:

- Implement privacy at the design stage
- Transparency in data collection, processing and transmission
- Define the purpose of data collection
- Obtaining consent, limiting access
- Collecting only the data that is needed
- Data made available to those concerned, possible to correct and make changes to profiles
- Data anonymity if possible

In this day and age, the choice between safety and privacy has become critical.

## 7 The interface of the Internet of Things

The prevalence in the 1980s of last-century personal computers with popular computer applications (word processors, spreadsheets etc.) meant that literally everyone was a potential computer user. This was a drastic contrast with an earlier era where a very few computers were in use by professionals. This universality of usage led to an increase in the importance of interfaces, the development of which was as rapid as the development of the equipment itself; they were in fact the major component of software. Traditional interfaces based on a keyboard and drop-down menu (still in use) were replaced largely by GUI interfaces, where icons supplement or replace text. Still under development are voice interfaces, and more recently interfaces based on the interpretation of gestures (Gestalt User Interfaces).

In the era of the Internet of Things we make contact with new devices that continuously measure and observe our surroundings. In the past they operated independently (as thermostats, boilers and refrigerators), now they will be part of the Internet, and will exchange information. Interfaces different to the graphical user interfaces that we are used to will become necessary. Different types of 'responsive objects' are being researched. One of them is the touch-based interface TUI - Tangible User Interface [Sharlin, 2004]. Basically it is a digital world adaption of physical contact with the driven object. But here automation brings with it new opportunities: an object can change visually e.g. shapes can give us information about their state, can vibrate etc.

The concept has its own history. Kurt Koffka wrote in 1935: 'To a primitive man each thing tells us what it is and what is to be done with it: fruit says: 'Eat me', water says ' Drink me', thunder says 'Fear me' and a woman says 'Love me''. [Koffka, 1935, p. 3]

Freedom of manoeuvre (DOF – degree of freedom) is essential to interface design and so is linking it to the controlled object. From this point of view mouse cursor control has promising properties – its movements correspond exactly to the movements of the cursor. But it is not suitable for steering a model airplane, for example, moving in three-dimensional space [Sharlin, 2004, p. 2].

Intricate relationships between different systems increase the complexity of the design. One can imagine, for example, a system which supports the care of patients living at home detecting depression and deciding to increase light levels, whilst another application that manages energy gauges that nothing is moving in the house – it is empty and lights should be put out. When overly complex devices communicate directly with each other using wireless networks, mistakes in print execution by a neighbouring printer are common [Alur, 2015, p. 9].

This brings us to the idea of Cyber-Physical-Human Systems (CPHS) systems. In addition to a set of computing devices that communicate with each other and affect the physical environment, a place for the user must be found – for a person who must be meaningly integrated into the control loops. A human must be able to intervene, interact with and use these systems.

The visualisation of information is an essential part of most applications because of the need to communicate with their users. Measurements which are not incorporated are useless. The purpose of visualisation is to provide information drawn from raw data. They must be presented in such a way that the user can utilise them. A simple example is the representation of real-world objects as virtual ones that can be depicted, accessible and show their states. Their representation can be physical or – more commonly – virtual. The basic devices in use here are smartphones, tablets and notebooks.

## 8 Conclusions

The Internet of things brings with it a qualitative change for the development of computer science. Computers are not only ubiquitous but are beginning to communicate directly with each other. This creates enormous opportunities to improve our quality of life but, as usual, also poses new risks. A car that reminds us constantly to fasten our seatbelts can be troublesome enough; a plate that vibrates unpleasantly when we place our pork knuckle onto it is perhaps a step too far. Maybe not everything that is technically possible is worth pursuing.

## References

1. Alur, R., Berger E., Drobnis A.W., Fix L., Fu K., Hager G. D., Lopresti D., Nahrstedt K., Mynattm E., Patel S., Rexford J., Stankovic J., A., Zorn B., 2015, *Systems Computing Challenges in the Internet of Things*, Computer Community Consortium, September 22
2. Atzori L., Iera A., Morabito G., 2010, *The Internet of Things: A survey*, Computer Networks 54 (2010) 2787–2805
3. Bandyopadhyay D., 2011, Sen J., *Internet of Things - Applications and Challenges in Technology and Standardization*, Wireless Personal Communications, May 2011, Volume 58, Issue 1, pp 49-69
4. Botterman M., May 2009, *Internet of Things: An Early Reality of the Future Internet*, Report of the Internet of Things Workshop, Prague, Czech Republic
5. Fogg, B. J., Cuellar, G. & Danielson, D., 2009, *Motivating, influencing, and persuading users: an introduction to captology*. In A. Sears & J. A. Jacko (Eds.),

Human-computer interaction. Fundamentals (pp. 109-122). London: CRC Press. Taylor & Francis Group

6. Fritzsche B., 2015, *Revealing the Invisible - Information Visualization in the Internet of Things Era*, Human Computer Interaction in the Internet of Things Era, Hauptseminar Medieninformatik SoSe
7. Technical Report LMU-MI-2015-2, September, 2015 ISSN 1862-5207, 2015
8. Gershenfeld N., Krikorian R., Cohen D., 2004, *The internet of things*, Scientific American 291 (4) (2004) 76–81.
9. Koffka K., 1935, *Principles of Gestalt psychology*, New York
10. Koponen T., Chawla M., Byung-Gon Chun, Ermolinskiy A., Kye Hyun Kim, Shenker S., Stoica I., August 2007, *A Data-Oriented (and Beyond) Network Architecture*, in: Proceedings of ACM SIGCOMM'07, Kyoto, Japan
11. *Mauritius Declaration on the Internet of Things*, 14 October 2014, 36th International Conference of Data Protection and Privacy Commissioners, Balaclava
12. Nguyen M., 2015, *Designing Smart Interactions for Smart Objects*, Human Computer Interaction in the Internet of Things Era, Hauptseminar Medieninformatik SoSe 2015, Technical Report LMU-MI-2015-2, September, 2015 ISSN 1862-5207
13. Pang Z.,2013, *Technologies and Architectures of the Internet-of-Things (IoT) for Health and Well-being*, Doctoral Thesis in Electronic and Computer Systems KTH – Royal Institute of Technology Stockholm, Sweden
14. Richards N. M., King J. H., 2013, *Three Paradoxes of Big Data*, September 3, 66 STAN. L. REV. ONLINE 41
15. Rieder A., 2015, *Health and Everyday Life: The Potential of Self-Monitoring in Managing the Own Health*, Human Computer Interaction in the Internet of Things Era, Hauptseminar Medieninformatik SoSe 2015, Technical Report LMU-MI-2015-2, September, 2015 ISSN 1862-5207
16. Sharlin E., Watson B., Kitamura Y., Kishino F., Itoh Y., 2004, *On tangible user interfaces, humans and spatiality*, Personal and Ubiquitous Computing, 8(5): 338-346
17. J. Shaw., 2014, *Why big data is a big deal*, Harvard Magazine
18. Sowa G., Filutowicz Z., Paszkowski J., 2012, Social Media Monitoring Tools, Computer Methods in Practice; W-wa, ISBN 978-83-60434-94-9
19. *The Only Fridge as Smart as You*, 2015, to Launch at CES, GE Appliances Pressroom LAS VEGAS—January 5

# SELECTED EXAMPLES OF APPLICATIONS OF NEW GRAPHICS AND ANIMATION TECHNOLOGIES

Zbigniew Filutowicz, Krzysztof Przybyszewski, Józef Paszkowski

IT Institute, University of Social Sciences, Łódź, Poland
*(zfilutowicz, kprzybyszewski)@spoleczna.pl*

**Abstract**

In recent years there has been a marked increase in the competitiveness of some very interesting (user) applications software within the field of computer graphics and animation. This paper presents an analysis of selected examples of the use of graphic applications software designed for professional use within various areas of human activity, and also focusses on the potential for further development of this software. Graphic applications software that makes use of motion capture, performance capture, time-lapse, morphing, Augmented Reality and the use of avatars in human-computer communication has become increasingly popular, cheap and simple.

**Key words:** computer graphics, computer animation, performance capture, human-computer communication

## 1 Introduction

In the past, the market for (user) applications software was monopolized by a few leading and well-known companies. The market valued cult products, and the data recording formats associated with them. The last decade has seen the spread of the use of cloud computing, which allows end users to run computer programs online in a web (internet) browser. Previously, all programs had to be installed on one's own computer and were run from the desktop. Since the prices of user programs of that time were a result of the laws of supply and demand, they differed significantly from the cost of production. It turned out that for entry-level companies it paid to provide them on the cloud for free to a certain extent or even completely. The phenomenon of free software, and the movement within it known as Open Source, has been around for a long time of course. From the year 2010 sites with online applications in the cloud started to appear en masse, and so it was in the field of computer graphics. For example, competitors to the cult program Microsoft Vision (acquired in 2000 from the Shapeware Corporation) appeared in the form of online programs such as Gliffy, Draw.io, Lucid chart and dozens of others [1].

In the last five years such has been the story of the new technologies of human-computer communication: Kinect and Leap Motion. They allow the control of a program in real-time by using body movements, for example by pointing a finger rather than moving a computer mouse. Many computer technologies are descendants of much earlier technologies, for ex-ample, in films, photography and animation. Similiarly older computer technologies influence the development of ones that are currently popular. An example is motion capture technology, which used to be expensive and require recording studios and the installation of markers on moving actors. Today it is very cheap and accessible in games consoles and computers in the form of the technologies Kinect or Leap Motion.

This paper presents an overview of some selected applications of new computer graphics and computer animation technologies. Also discussed are examples of software invaluable for fast and cost-effective creation of interesting applications.

## 2 Rotoskop stop-motion animation technology

Stop-motion animation is one technology for drawing, imaging and computer animation. It involves displaying images at specific intervals, e.g. 24 frames per second. The history of animation dates back to the end of the nineteenth century and its progenitor was Léon Gaumont (1864-1946), who was awarded a French patent for stop-motion animation in 1900. As with animations, so with film and digital video. An example of Time-lapse stop-motion animation is a time-lapse movie composed of many images made with, for example, a camera. The resulting film is very impressive in that it allows one to view phenomena imperceptible to the human eye because of their very slow pace. In this way one can present an animation of the growth of a plant taking place at a faster rate than in reality [8] [21].

An interesting method of stop-motion animation is rotoskop, in which movie frames with live actors are substituted for ones that are manually drawn by a cartoonist. A new use of rotoskop animation technology is a first full-length animated film which is dedicated to the Dutch painter Vincent van Gogh.

**Figure 1.** Movie frame with the image of an actor hand-painted with oil paints on canvas [13]

The director and designer – Dorota Kobiela – is Polish. The lead animation producer is Hugh Welchman, who won an Oscar for best short animated film with „Peter and the Wolf" in 2008. The screenplay was a collaboration by Kobiela and Jack Dehnel, a Polish poet, translator and painter. „Twój Vincent" („Loving Vincent") will be an 80-minute film about the life of van Gogh. It will consist of more than 56,000 frames – photographs of oil-on-canvas paintings done in the Expressionist style characteristic of the artist Figure 1. About 100 Polish painters, working in studios in Gdansk, Wroclaw and Great Britain, were invited to participate by the producers. Together they created more than 43,200 images, making 12 images for each second of film. About 120 original paintings by van Gogh feature in the film [13].

## 3 Computer applications that simulate the knowledge and skills of a painter

Research is being carried out into automating the process of painting pictures. An inter-national team consisting of world-class scientists, engineers, programmers and art historians decided to take a challenge and see if a program could be developed to enable a computer to simulate the skills of one of the greatest and most innovative artists of all time - Rembrandt – and control a 3D printer [2]. All of Rembrandt's paintings were scanned using a 3D scanner in order to analyse his style and to assess the thicknesses of the layers of paint he applied. After the scanning of 346 of the artist's works and their analysis by a team of researchers, software with the knowledge and skills of the brilliant painter was developed. A significant par of Rembrandt's work is portraits, so the researchers focused on those in particular Figure 2.


**Figure 2.** 3D scan and analysis of the painter's works [2]

Once the software was developed, a made-up portrait of the Dutch painter was produced with a 3D printer based on a database containing information about his original portraiture. The results of such research can be used to create accurate reproductions, and contribute to making the achievements of the great painter more widely known. The software created by the researchers could also be used for commercial purposes, and then we would all be able to order our own portrait created by a virtual seventeenth century Rembrandt for our homes.

## 4 Animation techniques based on motion capture

The first type of stop-motion animation was motion capture technology, the capture of movements of selected points of three dimensional real objects e.g. the human body. An actor puts on a vest with markers (sensors) attached, the movements of which are captured with a camera onto film frames Figure 3. Markers are also attached to the face, in order to record facial expressions. The positions of these markers on individual animation frames helps graphic designers to create very lifelike motion-animated characters. Studio SoInteractive is an example of a company involved in animation design based on the technique of motion capture [9] [28].

**Figure 3.** The use of markers to record the movements of points on 3D objects [9]

In 2010 there was a revolutionary change in the animation technique of motion capture because two new technologies appeared on the market. Microsoft brought out the technology Kinect [19] and a similiar solution appeared from the Leap Motion company [15]. A snap-in to a computer with two cameras registers the movement of 3D objects. The application of these new technologies no longer requires the use of markers because the computer programs can detect characteristic points of the observed objects, for example moving fingers or hands Figure 4.

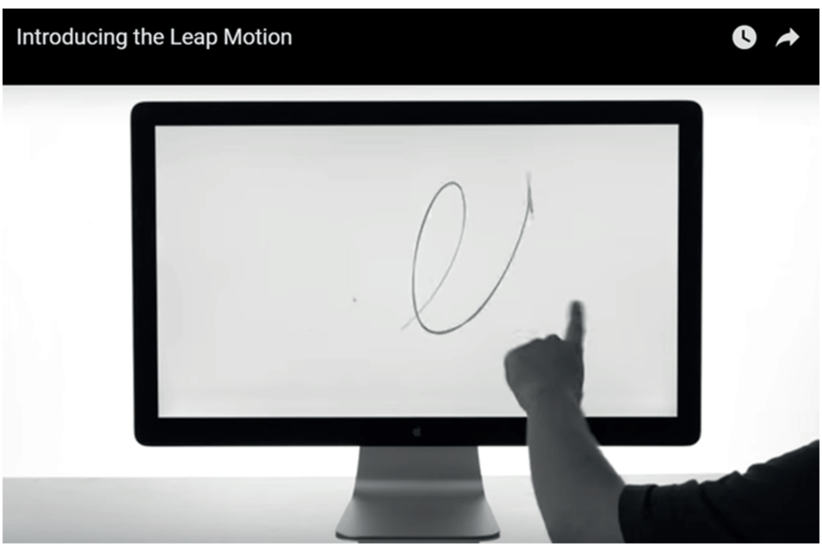



**Figure 4.** Capturing the movement of real objects [14]

## 5 Inbetweening animation technology

Morphing technology is the simplest example of automatic animation also known as inbetweening. It relies on software to generate graphics for intermediate frames. Thus we enter images, for example of a face, for two frames which are called key frames. For both raster images the program identifies characteristic points, such as the corners of the eyes, and on this basis determines the position of these points in the intermediate frames using a calculation algorithm. Then the filling in between the points is rendered with fragments of raster graphics derived from both key frames [16].

An example of the use of morphing would be an animation that shows a person's face changing over the course of a lifetime. Photographs of the face taken at various life stages are entered into the morphing program. The applications software generates intermediate frames creating an animation of the person's face [20].

Another example of programs with software that generates animation frames is provided by applications for creating avatars [22] [3]. An avatar can be animated from a drawing of any object, or from a 2D photograph of a human. Many companies providing software and services that produce various types of avatars have appeared on the market [31]. Avatars can consist of only part of an object, for example a head [5], or the whole object [17]. One of the most popular programs for animation of the head, along with facial expressions and lip movements, is the program CrazyTalk, currently in version 8, from the company Reallusion [25].


**Figure 5.** The CrazyTalk application site, selecting an avatar's teeth [25]

This program enables the animation of a 3D avatar head from photographs or 2D drawings. During the automatic animation facial expression and lip movements are added, which can articulate words that are requested with a text or audio file (Figure 5). Such an avatar is con-trolled on the one hand with a dynamic behaviour program, and on the other with a script for its speech.

There are also free sites, deserving of wider use, on which it is possible to design video animations for your specific requirements, without any need for programming knowledge [32] [7] [27] [6].

## 6 Controlling applications in real-time using body motion capture - performance capture

Motion capture technology enables the use of natural movements of selected points of the body in animation, which indicate where the same points will be located in animated shapes at a given moment in time. The current development of this technology is referred to as performance (facial motion) capture and involves not only mapping body movements, but also scanning human facial expressions [10]. Performance computing along with the very cheap technologies Kinect or Leap Motion allow the capture and use of whole-body movements to control computer programs in real time Figure 6.

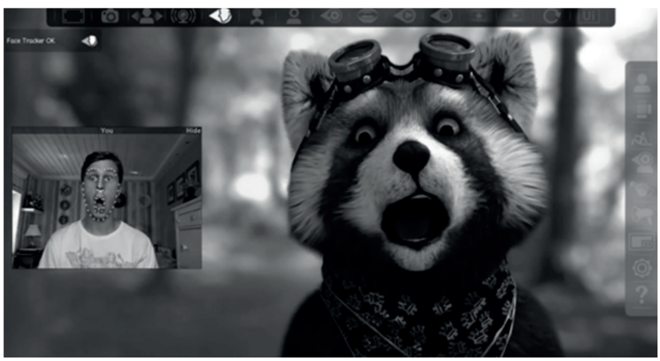



**Figure 6.** Scanning facial expressions and transposing them on the face of an avatar [10]

A company called Reallusion, which has been known for a long time for its program CrazyTalk, offers among other things a program called iClone, which allows 3D animation of a whole figure along with the head in Kinect and Kinect Mocap plug-in technology [26].

## 7 Linking computer graphics with real world images – augmented reality

Virtual Reality is an example of the use of 3D computer graphics and has become increasingly popular. Its popularity, even on 2D monitors, stems from the potential of creating increasingly realistic 2D images with it, which are deceptively reminiscent of real world images viewed through human eyes. Of course stereoscopic viewing of 3D objects or watching 2D animations in 3D technology makes an even greater impression. An interesting example is the reconstruction of Jan Matejko's painting „Battle of Grunwald" with stereoscopic technology Figure 7. This project was commissioned by the National Museum in Warsaw and created by the studio Platige Image [23] [12].

An interesting idea is to combine two views – from the real world and the virtual one – and view them on the screen at the same time. Such a solution is called Augmented Reality (AR). A Virtual Reality (VR) view can also be combined with another one to create an Augmented Virtuality (AV). The synchronisation of these two worlds can be performed using point markers or by reference to a geographic position. An example of the use of augmented reality are Microsoft Hololens glasses [18], which are specially tailored to this kind of visualisation. Computer graphics can become complementary to the real world with this technology. For example, with Hololens glasses a lineman sees the repaired part of a sewerage system and at the same time carries out a conversation with an expert on Skype (seeing it on a virtual screen Figure 7, and also sees a graphic which shows which components can be loosened and with which movements [18].

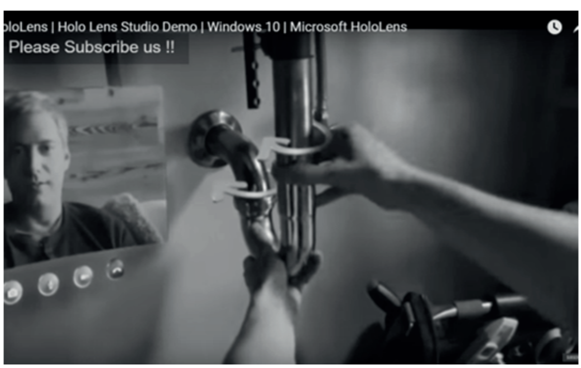



**Figure 7.** With Hololens glasses a lineman sees a sewage system, virtual computer screen and an infographic [18]

In museums virtual avatars of historical figures may appear as guides. Strong interest in using augmented reality is shown in web and traditional marketing, for example to create a virtual fitting room [4].

## 8 Facial recognition and identification of facial expressions

The development of these technologies in recent times is linked not only with animation, but also with facial recognition. The programmers of Facebook have built software which recognises faces with 97.25% accuracy, which approaches the capabilities of a human. The program works regardless of prevailing light conditions or the angle of inclination of the head [30].

In 2016, Amazon filed an application for a patent that describes a shopping method which will make use of facial recognition and facial expressions to authorise transactions [24]. After accepting a transaction in an online store, the site launches a camera application and performs a two-step verification process. The first, of course, is facial recognition. At the second stage, the site launches a camera that records facial expressions. For example, it could request that the user of the site winks with the right eye or smiles.

In 2013, at the Techfest conference, Microsoft presented the results of research carried out in the area of recognition of facial expressions and voice tone [29]. The aim of the project „turning monolingual speaker into multilingual" was, in the first instance, the simulation of a human face speaking a native language. Further research led to the possibility of real-time simulation of the face of the same human speaking 26 languages.

A team of researchers from the University of Washington designed software which analysed thousands of photos posted on the Internet from the point of view of the incremental changes in the appearance of a human face in the course of a lifetime, and on this basis is able to estimate the appearance of a human face in any period of life. The program is so powerful that it can picture a face at any given time in a human life even in different poses, with different expressions and in unfavourable light. It is particularly difficult to predict how a child will look like in later years of life, because the shape of a face changes dramatically with age. This time however, the researchers say, the program is so accurate that it can calculate each dimension of the head and shape of the face, taking into account different age groups; it generates the target portrait on that basis. The study was financed by Google and Intel Corporation [11].

## 9 Conclusions

In the examples given of computer graphics and animation the appeal of the applications software and its increasing capabilities are apparent. There is also potential for further development. There is nothing to prevent film frames with actors from the van Gogh film being transformed by computer into frames of animated film using the program that was created for the works of Rembrandt. The automation of film production with computer applications is a major technological achievement. Museums will become more attractive and valuable exhibits more popular as, with low costs and seamless technologies, it will be possible to display faithful copies of those exhibits anywhere. Graphics programs that permit the analysis of photographs of the face, eyes or hands for the purpose of medical diagnosis will emerge.

Currently, graphics applications software is designed for an even wider range of uses. The tools are better adapted to the needs of artists and more convenient to use. Continuous progress in the field means that it is necessary to follow the innovations and potential of soft-ware from different producers, and not be tied to just one firm as in the past. The examples of the technologies Kinect and Leap Motion are evidence of the scale of change of the last five years. Performance computing and the very cheap technologies Kinect and Leap Motion al-low the capture and use of whole body movements for real-time control of computer pro-grams. The use of graphic applications software with motion capture technology, performance capture, time-lapse, morphing, Augmented Reality or the use of avatars in human-computer communication is becoming increasingly popular, cheap and simple.

## References


1. AN Srikanth, 2015, *Visio Alternatives: 10 Best Diagramming Software*, http://beebom.com/visio-alternatives/
2. Brown M., 2016, *'New Rembrandt' to be unveiled in Amsterdam*, https://www.theguardian.com/artanddesign/2016/apr/05/new-rembrandt-to-be-unveiled-in-amsterdam
3. Endicott M. L., 2014, 100 *Best Talking Avatar Videos*, http://meta-guide.com/videography/100-best-talking-avatar-videos
4. Filutowicz Z., Przybyszewski K., Sowa G., Paszkowski J., 2012, *Techniques of digital photography and computer graphics visualization e-business*, Studies & Proceedings of Polish Association for Knowledge Management" Tom 62, Nr 1, pp17-31
5. Fleming B, Dobbs D., 2002 *Animating Facial Features and Expressions Paperback*, Helion, 2002
6. FreeMake, 2016*, How to Make a Cartoon Yourself: Top 7 Animated Video Makers Compared*, http://www.freemake.com/blog/5-best-sites-to-make-animated-video-trouble-free/
7. GoAnimate, 2016, *Make Professional Animated Videos GoAnimate*, https://goanimate.com/

8. Google, 2016, *Google Earth Engine Timelapse*, https://earthengine.google.com/timelapse/
9. Hildebrand K., 2012, *Polygamia.pl: How does the motion capture studio?* https://www.youtube.com/watch?v=R-N-BpeHm28
10. Holotech Studios, 2016, *FaceRig*, https://facerig.com/
11. Kemelmacher-Shlizerman I., Suwajanakorn S., Seitz S. M., 2014, *Illumination-Aware Age Progression*, http://grail.cs.washington.edu/aging/
12. Kidziński R., Bagiński T., 2010, *Jan Matejko's "Battle of Grunwald" in 3D*, http://www.playpoland.org.uk//index.php?name=movies_more&oId=80
13. Kobiela D., Welchman H., 2016, *Loving Vincent*, http://lovingvincent.com/
14. Leap Motion, 2012, *Introducing the Leap Motion*, https://www.youtube.com/watch?v=_d6KuiuteIA
15. Leap Motion, 2016, *Reach into new worlds*, https://www.leapmotion.com/
16. Lindquist T., 2011, *Cat-to-dog morph*, https://www.youtube.com/watch?v=VV807_NqGto
17. Maestri G., 2000, *Character Animation*, Helion
18. Microsoft HoloLens, 2015, *Mixed reality: Your world is the canvas*, https://www.microsoft.com/microsoft-hololens/en-us
19. Microsoft Kinect, 2016, *Meet Kinect for Windows*, https://www.developer.microsoft.com/en-us/windows/kinect
20. MovieTrailersNET, 2012, *Morphing 33 years in 60 seconds*, https://www.youtube.com/watch?v=lAE-gUuu814
21. National Geographic, 2014, *Time-Lapse: Watch Flowers Bloom Before Your Eyes,* https://www.youtube.com/watch?v=LjCzPp-MK48
22. Plantec P., Kurzwell R. M., *2003Virtual Humans: A Build-It-Yourself Kit, Complete with Software and Step-by-Step Instructions*, New York: AMACOM
23. Platige Image, 2016, *Platige to Artistic Conglomerate*, http://www.platige.com/pl/
24. Rao L., 2016, *Amazon Is Trying to Patent Paying With a Selfie*, http://fortune.com/2016/03/14/amazon-patent-selfie/
25. Reallusion, 2016, *CrazyTalk*, http://www.reallusion.com/crazytalk/default.html
26. Reallusion, 2016, *iClone*, http://www.reallusion.com/iclone/default.html
27. Skavish D., *That's how the idea for Animatron was born,* https://editor.animatron.com/
28. SoInteractive, 2012, *The film presents the possibilities of studying SoInteractive confirms the good condition of the Polish sector of motion capture*, http://www.gry-online.pl/S013.asp?ID=68466
29. Soong F., 2012, *Turning a Monolingual Speaker Into Multi-Lingual Speaker*, http://research.microsoft.com/en-us/projects/mixedlangtts/
30. TaigmanY., Yang M., Ranzato M. A., Wolf L., 2014, *DeepFace: Closing the Gap to Human-Level Performance in Face Verification*,

https://research.fb.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/

31. van Lun E., 2016, *Virtual Agents / Chatbots Directory*, https://www.chatbots.org/
32. Wideo.co, 2016, *Create videos easily for your marketing campaigns*, http://get.wideo.co/en/

# Resource Monitoring for Wireless Sensor Networks using ANFIS

Nagesha[1], Sunilkumar S. Manvi[2]

[1] Department of Electronics and Instrumentation Engineering
JSS Academy of Technical Education, Bengaluru, India
*nageshashiva@gmail.com*

[2] School of Computing and Information Technology
Reva University, Bengaluru, India
*sunil.manvi@revainstituion.org*

## Abstract

Wireless sensor networks (WSNs) are usually a resource constrained networks which have limited energy, bandwidth, processing power, memory etc. These networks are now part of Internet by the name Internet of Things (IoT). To get many services from WSNs, we may need to run many applications in the sensor nodes which consumes resources. Ideally, the resources availability of all sensor nodes should be known to the sink before it requests for any further service(s) from the sensor node(s). Hence, continuous monitoring of the resources of the sensor nodes by the sink is essential. The proposed work is a framework for monitoring certain important resources of sensor network using Adaptive-Neuro Fuzzy Inference System (ANFIS) and Constrained Application Protocol (CoAP). The ANFIS is trained with these resources consumption patterns. The input to ANFIS is the resources consumption levels and the output is the resources consumed levels that needs to be sent to the sink which may be individual or combinations of resources. The trained ANFIS generates the output periodically which determines resources consumption levels that needs to be sent to the sink. Also, ANFIS continuously learns using hybrid learning algorithm (which is basically a combination of back propagation and least squares method) and updates its parameters for better results. The CoAP protocol with its observe option is used to transport the resource monitoring data from the sensor nodes to the cluster head, then from the cluster head to the sink. The sensor nodes runs coap server, the cluster head runs both coap client and server and the sink runs coap client. The performance of the proposed work is compared with LoWPAN network management protocol (LNMP) and EmNets Network Management Protocol (EMP) in terms of bandwidth and energy overheads. It is observed that proposed work performs better when compared to the existing works.

**Key words:** Wireless sensor networks (WSNs); Resource management; Resource Monitoring; Constrained application protocol (CoAP); ANFIS; Fuzzy inference system.

# 1 Introduction

The Wireless Sensor Networks as described by many researchers [1], is mainly used for monitoring the physical world. It is now connected to the Internet using new communication technologies like IEEE 802.15.4 standard, 6LoWPAN, RPL, CoAP etc. Internet of Things [2] is the larger technological term used to refer the connectivity of building automation, industrial automation, transportation, logistics, wireless sensor networks etc. with the Internet. Sensor nodes or any other real world things (embedded devices) can be connected to other sensor nodes or things in the other part of the world through global Internet infrastructure for timely collection and sharing of data as part of various physical world monitoring applications.

The design and development of WSNs is influenced by managing resources which are usually limited in WSNs, wireless radio characteristics, middleware, application specific QoS requirements, etc. Limited resources of sensor nodes which mainly include energy, bandwidth, memory space, processing power, etc. These resources of sensor nodes have to be managed effectively to provide better QoS services, reliability, greater performance and long life span of the WSNs. Resource allocation, resource mapping, resource adaptation, resource monitoring, resource discovery and selection, resource estimation, resource scheduling and resource modeling are some issues pertaining to resource management in WSNs. This paper addresses the monitoring of resources of sensor nodes.

Resource Monitoring is a systematic process of observing, tracking and recording data about resources of sensor nodes for the purposes of utilizing the services of sensor nodes and network to the maximum possible extent. Energy, bandwidth, processing capability, memory space, etc. are some of the main resources of sensor nodes. These resources are utilized by sensor node to run applications, to transmit/receive data, temporary storage of data, etc. Sink is a device which is responsible for monitoring sensor nodes resources, routinely gathers information about resources of the sensor nodes.

Resource monitoring is required in WSNs for the following reasons. (1) Gathering of information about sensor nodes of the network which can be used to make decisions about the demand for better services or more services from the sensor nodes. (2) To learn from experiences to improve in demanding services from the sensor nodes. (3) To have an accountability of resources used for different purposes and the obtained results. (4) Information gathered through monitoring which could be used to analyze, evaluate all components of the network in order to measure its effectiveness and adjust inputs where ever necessary. (5) Monitoring allows results, processes and experiences to be documented and used as a basis to steer decision making and learning processes. (6) The data acquired through monitoring is used for evaluation.

The existing IP based solutions for the resource monitoring in WSNs are based on periodic reporting about resources or use query-response techniques. The periodic reporting about the resources by the sensor nodes to the sink is irrespective of the resources consumed by different processes or applications. Information about resources are reported even when they are not consumed. This reporting is unnecessary and results in the consumption of resources (bandwidth and energy for the transport of resource monitoring data from sensor nodes to sink) which are precious in WSNs. Query processing is a two way communication which results in delay in obtaining the information about the resources and consumes extra bandwidth and energy. These are some of the problems that are addressed in this paper with solutions provided by designing Adaptive-Neuro Fuzzy Inference System (ANFIS) and using Constrained Application Protocol (CoAP) along with its observe option.

Resources to be monitored are fed as inputs to the multi-layered ANFIS [3] whose output depends on training with some input-output data pairs and subsequent learning. The consumption level of the resources and which resources need to be transported to the sink from sensor nodes makes input-output data pair. The ANFIS is trained to indicate resources consumption level to be transported when consumption exceeds 40%. If there are three resources to be monitored, then ANFIS may indicate to either transport consumption level of any one resource or combination of any two resources or all three resources depending on the resources consumed by the sensor node. The ANFIS also learns to serve better using back propagation and least squares method.

The Constrained Application Protocol [4] along with its observe option [5] and client-server architecture is used to collect the information about resources of sensor nodes. The WSN is organized into clusters where the cluster head transports the information to the sink. The cluster head is the client which collects the resources information of sensor nodes. The sensor nodes are the servers which supply their resources consumption information to the cluster head. The cluster head acts as the client while collecting resource monitored data from sensor nodes and acts as a server when it supplies the same to the sink. The client endpoint of the CoAP device registers as a client to the server(s) with its observe option. After the registration, the CoAP server with the observe option sends the resource information whenever there is a change in the resource consumption.

Our contributions are as listed below.

1. IP based resource monitoring technique is developed to efficiently collect resources consumption information from sensor nodes of the WSNs.
2. The developed resource monitoring technique uses ANFIS which is designed to enhance the efficiency of resources information collection. The ANFIS is trained to intelligently decide which resources information has to be sent to the sink.

3. The ANFIS training data is developed based on intuition and is used to optimize the membership functions of the ANFIS.
4. Finding the efficiency of the developed technique in terms of bandwidth and energy at node level, cluster level and network level.
5. Using the CoAP with observe option enhances the efficiency of developed resource monitoring technique. The combination of the coap with its observe option and ANFIS gives a superior performance (in terms of bandwidth utilization and energy consumption to transport the monitoring data) compared to LNMP (query based) and EMP (periodic reporting) based resource monitoring techniques.

The rest of the paper is organized as follows. The related work is given in section 2. The protocol stack of the sensor node is discussed in section 3. The complete work description is provided in section 4. Simulation of the work at node level, cluster level, network level and comparison with other works is described in the section 5. The section 6 concludes the paper

## 2 Related Work

Energy of sensor nodes is continuously monitored to avoid sensor node failure which in turn may result in the sensor network failure. Some of energy monitoring techniques are as follows. eScan [6] is a monitoring technique developed to monitor the energy levels (energy map) of sensor nodes of the network using aggregation based approach. Later predication based energy maps are developed [7] to monitor the energy levels of the sensor network. Further, monitoring the energy with low overhead is proposed [8] using the techniques such as hierarchical monitoring structure, in-network aggregation and cluster heads rotation.

Similarly monitoring techniques are developed to monitor the link quality, congestion level, bandwidth, buffer length etc. Snooping based link quality monitoring is designed [9] which listens to the channel and infers the success and loss rates. CODA [10] and ECODA [11] are receiver based congestion detection and monitoring mechanisms. All these resource monitoring mechanism deals with only one resource and that is communicated to the necessary ends (e.g. sink). Sending monitoring data using piggybacking is proposed [12]. Our work is not to improve any of these works of resource monitoring, instead, designing a framework to send consumption level of all necessary resources (like residual energy, bandwidth, buffer length, link quality) collectively and efficiently (based on some criteria) to the sink which uses these resources consumption level to monitor the sensor network services to provide services to its users.

Other related works are as follows. Some of the applications which provide sensor network management schemes in terms of controlling and monitoring [13] are BOSS, MANNA, WinMS, TinyDB, Mote-View, TP. All these are non-IP based network management systems. The LowPAN Network Management protocol (LNMP) [14] and EmNets Network Management Protocol (EMP) [15] are IP based sensor network management protocols. The LNMP uses query processing and EMP uses periodic reporting to fetch information from the sensor network. Our work uses ANFIS and CoAP with its observe option to push resource monitored data from sensor nodes to the sink.

Now sensor networks are viewed as part of Internet of things. Web services are provided with two different styles: Big Web services (SOAP) and RESTful web services. REST style web service is more suitable for sensor devices because of its simplicity. Recent work on management of sensor devices uses REST style [16] web service. This work is on complete sensor device management but not addressed networking issues. Our proposed work is not on complete device management but considers collecting resources consumption level of all sensor nodes of the network.

We have not found any literature which uses the ANFIS model in WSNs for Resource Monitoring. Recent work on Resource Mapping [17] uses ANFIS for video communication. ANFIS is used to decide which resources level has to be notified to the registered client(s) using coap with observe option.

## 3 Sensor node Protocol Stack

A low power, highly reliable and Internet enabled standardized protocol stack for IoT [18] is discussed by Maria Rita Palattella et al. The same protocol stack is adopted in this proposed work. Figure 1 illustrates IoT protocol stack along with protocols from physical layer to application layer.

### 3.1 Low Power PHY and MAC Layer – IEEE 802.15.4(2011)

The IEEE 802.15.4 - 2011 [19] defines the specifications for physical layer (PHY) and medium access layer (MAC) for low cost, low-power and low data rate wireless connectivity. The physical layer provides two services (1) The PHY data service: This service enables the reception and transmission of PHY protocol data units across the physical radio channel. (2) The PHY management service: This service provides activation and deactivation of the radio transceiver, energy detection within the channel, link quality indicator, clear channel assessment etc. Some of the PHYs defined in this standard are O-QPSK PHY, BPSK PHY, ASK PHY, CSS PHY, UWB PHY etc.

The MAC sublayer provides two services (1) The MAC data service: This service enables the reception and the transmission of MAC protocol data units across the physical data service. (2) The MAC management service: This service provides beacon management, guaranteed time slot (GTS) management, channel access, frame validation etc.

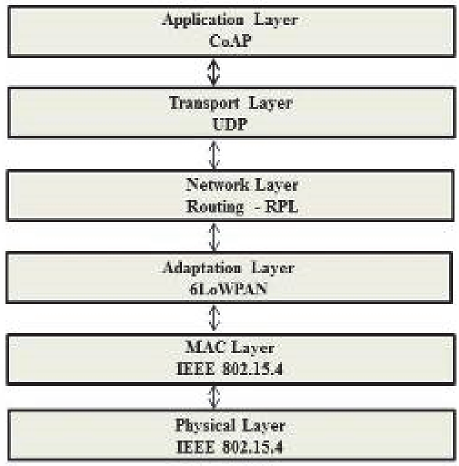



**Figure 1.** IoT Protocol Stack

### 3.2 Adaptation layer - IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)

The 6LoWPAN, an IETF standard [20] is necessary between IPv6 layer and IEEE 802.15.4 MAC layer of the protocol stack to fit the IEEE 802.15.4 into the stack. The IEEE 802.15.4 standard has a frame size of 127-byte, with payload size of layer 2 as low as 72 bytes. The IPv6 standard has a minimum packet size of 1280 bytes, thus requiring fragmentation and reassembly. This is done by the adaptation layer 6LoWPAN.

The IPv6 and UDP headers consumes significant portion of the payload space in a single IEEE802.15.4 packet. To reduce the overhead, the header compression is provided by the 6LoWPAN

### 3.3 Routing – RPL

The IPv6 routing protocol for low power and lossy networks (RPL) [21] is designed considering the constrained nodes (with limited energy, memory and processing power) which supports only low data rates. It supports three basic traffic flows (1) point-to-point (2) point-to-multipoint (3) multipoint-to-point. The RPL does not rely on any particular link layer technology and designed to operate over different link layers.

The RPL is basically a distance vector routing protocol which does not require predefined topology but is able to build the topology of the network. The routes of RPL are optimized for traffic from (or to) one or more sinks (roots) of the topology. It organizes a topology of the network as a Directed Acyclic Graph (DAG). The DAG is partitioned into many Destination Oriented DAGs (DODAGs), one DODAG per sink. In multiple root DAG, the roots are joined by common backbone (transit link).

### 3.4 Transport Layer – UDP

The user datagram protocol (UDP) provides minimum protocol mechanism for application programs to send packets to other application programs. This protocol is transaction based and does not provide any guaranteed delivery of packet and duplicate protection. The application layer protocol is responsible for reliability (i.e. retransmission of lost packets).

### 3.5 Application Layer – CoAP

The Constrained Application Protocol (CoAP) is a REST style and specially designed web transfer protocol (like HTTP) for use with constrained networks and constrained nodes. It includes key concepts of the web such as built in discovery of services and resources, multicast support, Internet media types and URIs. It has two main features (1) Messaging: deals with UDP to exchange messages asynchronously between CoAP endpoints. CoAP defines four types of messages: confirmable (for reliability), Non-confirmable, acknowledgment, reset. (2) Requests and Responses: deals with the application using methods (GET, PUT, POST, DELETE) and response codes (2.xx, 4.xx, 5.xx).

The CoAP's interaction model is similar to the client-server model of HTTP. The observe extension to the CoAP offers a mechanism for a CoAP client to observe a resource on a CoAP server. With this extension of CoAP, client can retrieve a representation of the resource and request this representation to be updated by the server over a period of time as long as it is interested in the resource. The sensor node which has resources to be monitored acts as a server and sink acts as a client. The registration by the client and subsequent notifications by the server is as shown in the Figure 2. The client uses the GET request with observe option (observe = 0) of the CoAP (extended GET) for registration. After registration, the server immediately responds with current resource level with observe option. The observe option of the
CoAP when included in a response (server side), identifies the message as a notification. The server sets the value of the observe option of each notification in the increasing sequence number order. This is helpful in reordering the

notifications if a notification arrives at the client later than a newer notification from the server. The request by the client (registration) carries a self-generated token that is echoed by the server in the resulting subsequent notifications. The notification uses the 2.05 (content) as a response code which indicates that the payload in the response is a representation of the requested resource.

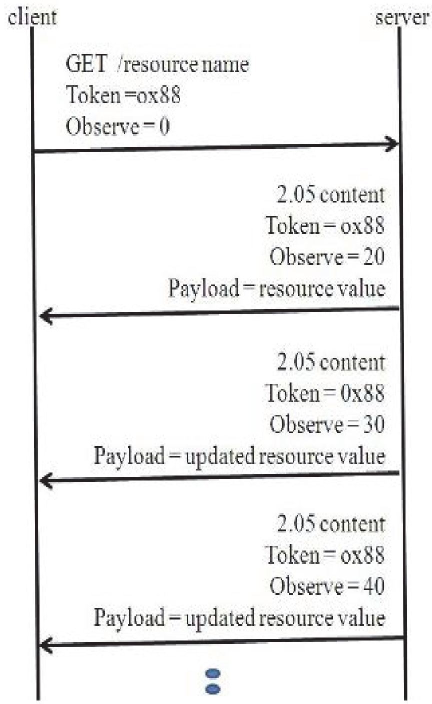



**Figure 2.** CoAP with observe option

## 4 ANFIS based Resource Monitoring using CoAP in Wireless Sensor Networks

### 4.1 Resources of the Sensor Networks

There are mainly two types of resources in sensor networks: physical and logical. The processor, memory, sensors etc. are important physical resources of the sensor nodes whereas energy, bandwidth, operating system, network throughput etc. are important logical resources. The extent of involvement of the processor, amount of memory occupied by the current processes, residual energy, bandwidth are the important resources of sensor nodes to be monitored in WSNs. In general, consider the resources to be monitored as R1, R2, R3, ..., Rn. The patterns of consumption of these resources are observed over the period of time. The consumption patterns of these resources are finalized and used for monitoring of the resources.

### 4.2 Resource Monitoring Policy

The proposed work is the framework for the monitoring of three resources R1, R2 and R3. We assume that the consumption patterns of these resources are as shown in the Figure 3. These consumption patterns can be used for the three most important resources of sensor node: energy, memory and processing speed. These are the inputs to the ANFIS which gives the output as which resources level should be notified to the registered client based on the policy shown in the Table 1. The policy is based on the assumed resource consumption pattern and is not to notify the resources level whose consumption level is below 40% to the registered client.

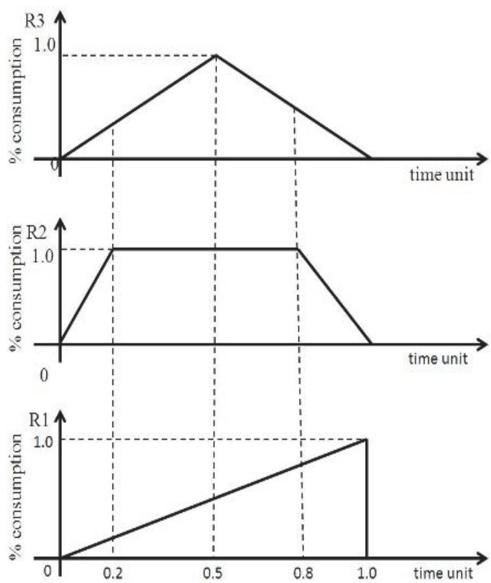



**Figure 3.** Resource consumption patterns

### 4.3 Adaptive-Neuro Fuzzy Inference System (ANFIS)

The ANFIS is a class of adaptive neural networks that are functionally same as fuzzy inference systems. The fuzzy inference system is embedded into the framework of adaptive neural networks to obtain the ANFIS. Fuzzy if-then rules and membership functions based on the zero-order Sugeno type fuzzy inference system [22] are used to construct the ANFIS which is used to generate the required input-output data pairs of the proposed work as shown in the Table 2. The membership functions parameters are tuned using the input-output training data set and the hybrid learning algorithm (i.e. combination of back propagation and least squares method).

**Table 1.** Resource Monitoring Policy

| Resources consumed | Resources to be notified | code |
|---|---|---|
| R1 < 0.4, R2 < 0.4, R3 < 0.4 | NULL | 0 |
| R1 > 0.4, R2 < 0.4, R3 < 0.4 | R1 | 1 |
| R1 > 0.4, R2 > 0.4, R3 < 0.4 | R2 R1 | 2 |
| R2 > 0.4, R1 < 0.4, R3 < 0.4 | R2 | 3 |
| R2 > 0.4, R3 > 0.4, R1 < 0.4 | R3 R2 | 4 |
| R3 > 0.4, R1 < 0.4, R2 < 0.4 | R3 | 3 |
| R3 > 0.4, R1 > 0.4, R2 < 0.4 | R3 R1 | 6 |
| R1 > 0.4, R2 > 0.4, R3 > 0.4 | R3 R2 R1 | 7 |

**Table 2.** Required input-output combination of ANFIS

| Time unit | R3 | R2 | R1 | ANFIS output | Resources to be notified |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0 |
| 0.1 | 0.2 | 0.5 | 0.1 | 3 | R2 |
| 0.2 | 0.4 | 1.0 | 0.2 | 4 | R3 R2 |
| 0.3 | 0.6 | 1.0 | 0.3 | 4 | R3 R2 |
| 0.4 | 0.8 | 1.0 | 0.4 | 7 | R3 R2 R1 |
| 0.5 | 1.0 | 1.0 | 0.5 | 7 | R3 R2 R1 |
| 0.6 | 0.8 | 1.0 | 0.6 | 7 | R3 R2 R1 |
| 0.7 | 0.6 | 1.0 | 0.7 | 7 | R3 R2 R1 |
| 0.8 | 0.4 | 1.0 | 0.8 | 7 | R3 R2 R1 |
| 0.9 | 0.2 | 0.5 | 0.9 | 2 | R2 R1 |
| 1.0 | 0.0 | 0.0 | 1.0 | 1 | R1 |

The ANFIS structure of the proposed work consists of three inputs R3, R2 and R1, six rules, one output and zero-order Sugeno fuzzy model. The computational efficiency of Sugeno model is high and is best suited for the development of fuzzy inference system from given input-output training data set. The three inputs to the ANFIS are resource consumption patterns of R3, R2 and R1 and the single output is the code which represents which resource(s) needs to be notified to the registered client by the server. The cluster heads are the registered clients and leaf nodes are servers in a clustered tree WSNs.

**List of Parameters and Notations:** List of parameters and notations used in ANFIS are listed in Table 3.

**Rules:** The fuzzy inference system considered in the proposed work is zero order Sugeno fuzzy model which has three inputs R3, R2 and R1 and one output. Fuzzy if-then rules with R3, R2 and R1 as linguistic variables and less, more as linguistic labels is as given below.

Rule 1: if R3 is less AND R2 is less AND R1 is less then output1 = 0;
Rule 2: if R3 is less AND R2 is less AND R1 is more then output2 = 1;
Rule 3: if R3 is less AND R2 is more AND R1 is less then output3 = 3;
Rule 4: if R3 is less AND R2 is more AND R1 is more then output4 = 2;
Rule 5: if R3 is more AND R2 is more AND R1 is less then output5 = 4;
Rule 6: if R3 is more AND R2 is more AND R1 is more then output6 = 7;

where the output is represented by constant output (singular membership function).

**Table 3.** Parameters and notations used in ANFIS

| Notation | Description |
|---|---|
| (a,b,c,d) | Premise parameter set of membership functions |
| less, more | Linguistic variables |
| $\mu_{less}(R3)$, $\mu_{more}(R3)$ | Membership functions of resource R3 |
| $\mu_{less}(R2)$, $\mu_{more}(R2)$ | Membership functions of resource R2 |
| $\mu_{less}(R1)$, $\mu_{more}(R1)$ | Membership functions of resource R1 |
| $w_i$ | firing strength of rule i |
| $\overline{w_i}$ | Normalized firing strength of rule i |
| $O_i^1$ | i th node output of layer 1 |
| $O_i^2$ | i th node output of layer 2 |
| $O_i^3$ | i th node output of layer 3 |
| $O_i^4$ | i th node output of layer 4 |
| $O_i^5$ | i th node output of layer 5 |

**Membership functions**: The reasoning mechanism for the above Sugeno model is illustrated in the Figure 4. The membership function shown for the inputs R3, R2 and R1 with the linguistic labels less and more. The corresponding functionally equivalent ANFIS architecture which is a multi-layer network is shown in the Figure 5.

**ANFIS structure**: Each layer of the ANFIS performs specific task and nodes of the particular layer performs similar functions. The layers of the ANFIS are fuzzification layer, rules layer, normalization layer, defuzzification layer and output layer. The square represents a adaptive node whereas the circle indicates a fixed node. The output signals from nodes of a layer are fed as inputs to the next layer as shown in the Figure 5.

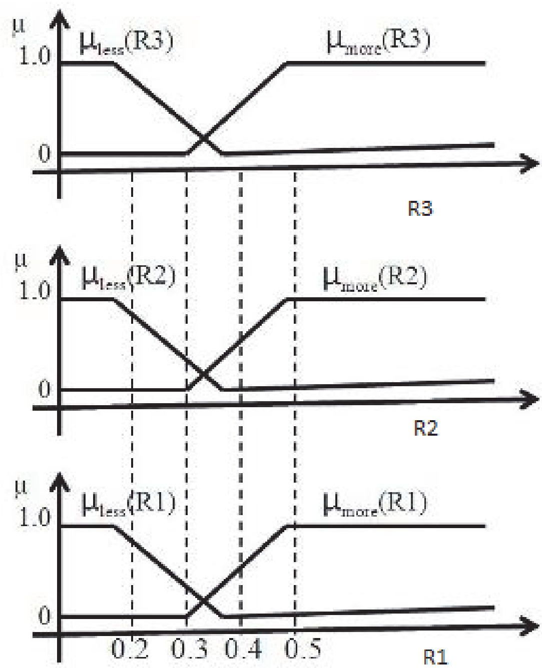



**Figure 4.** Membership functions

Layer 1: This layer is called the fuzzification layer which fuzzifies the inputs R3, R2 and R1. Nodes of this layer are adaptive nodes with node function

$O_1^1 = \mu_{less}$ (R3)

$O_2^1 = \mu_{more}$ (R3)

$O_3^1 = \mu_{less}$ (R2)

$O_4^1 = \mu_{more}$ (R2)

$O_5^1 = \mu_{less}$ (R1)

$O_6^1 = \mu_{more}$ (R1)

where R3, R2 and R1 are inputs to the nodes at this layer and less/more is the linguistic label associated with the node function. In general, $O_i^1$ (the function of the $i^{th}$ node) is the membership function of less/more and it specifies the degree to which the given input (R3/R2/R1) satisfies the quantifier less/more. We choose the membership function to be trapezoidal in shape

$$\text{trap}(x{:}a,b,c,d) = \max(\min((x-a) / ( b-a),1,(d-x) / (d-c)),0) \qquad (1)$$

where {a,b,c,d} is the premise parameter set (with $a < b <= c < d$) which determines the x coordinates of the four corners of the trapezoidal membership functions. The change in the values of these parameters varies the trapezoidal function accordingly and produce various forms of the membership functions.

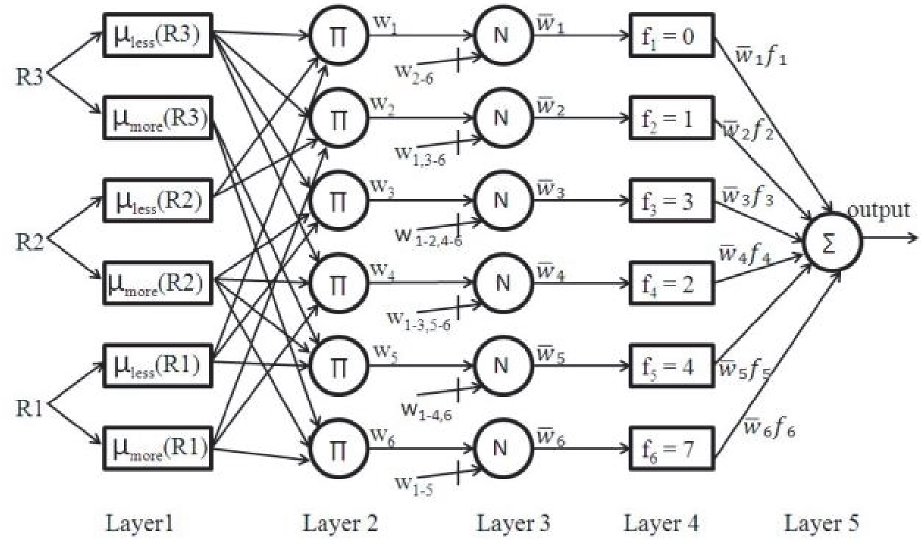



**Figure 5.** ANFIS structure

Layer 2: This layer is called the rules layer. All nodes of this layer are fixed nodes (represented by π) and performs similar functions. The inputs of each node are connected by the intersection operator (fuzzy AND) as shown in the if-then rules. The output of each node is the product of all its incoming signals. Six nodes are used to implement six rules whose outputs are as follows.

$O_1^2 = \mu_{less}(R3) \times \mu_{less}(R2) \times \mu_{less}(R1)$

$O_2^2 = \mu_{less}(R3) \times \mu_{less}(R2) \times \mu_{more}(R1)$

$O_3^2 = \mu_{less}(R3) \times \mu_{more}(R2) \times \mu_{less}(R1)$

$O_4^2 = \mu_{less}(R3) \times \mu_{more}(R2) \times \mu_{more}(R1)$

$O_5^2 = \mu_{more}(R3) \times \mu_{more}(R2) \times \mu_{less}(R1)$

$O_6^2 = \mu_{more}(R3) \times \mu_{more}(R2) \times \mu_{more}(R1)$

$\mu_{less}(R3)$, $\mu_{more}(R3)$, $\mu_{less}(R2)$, $\mu_{more}(R2)$, $\mu_{less}(R1)$, and $\mu_{more}(R1)$ are fuzzified inputs to the nodes of the rules layer as shown in the Figure 5. The output of each node represents the firing strength of the rule.
Layer 3: Nodes in this layer are fixed nodes and they are represented by N. The nodes output of this layer are called the normalized firing strengths. The $i^{th}$ node function is to calculate the ratio of firing strength of the rule i to the sum of firing strengths of all rules.

$$\overline{w_i} = \frac{w_i}{\sum_{k=1}^{6} w_k} \tag{2}$$

where i = 1,2,3,4,5,6

Layer 4: Nodes of this layer are adaptive nodes with a node function

$$O_i^4 = \overline{w_i} * f_i = \overline{w_i} * \text{constant} \tag{3}$$

where $\overline{w}_i$ is the normalized firing strength.

Layer 5: The only node in this layer is a fixed node (represented by ∑) which computes the overall output as the sum of all the incoming signals.

$$O_i^5 = \sum_{i=1}^{6} \overline{w}_i f_i = \frac{\sum_{i=1}^{6} w_i f_i}{\sum_{i=1}^{6} w_i} \tag{4}$$

The learning mechanism has to tune only the premise parameters to fine tune the membership functions. In the proposed work, the ANFIS is modelled as zero-order Sugeno fuzzy inference system and there is no consequent parameter set.

**ANFIS input-output training data:** The range of three inputs R3, R2 and R1 is from 0.0 to 1.0 which corresponds to 0% to 100% of the resource consumed and the output code is from 0 to 7 which corresponds to the resources to be notified to registered client like cluster head.The ANFIS input-output training data set is prepared as follows. The resources R3 and R2 are fixed at 0.0 and R1 is varied from 0.0 to 1.0 in steps of 0.1. Then R2 is changed (keeping R3 same) to 0.1 and R1 is varied again from 0.0 to 1.0 in steps of 0.1. This is repeated by changing R2 (keeping R3 same) in steps of 0.1 upto 1.0. Now R3 is changed to 0.1 (with R2 = R1 = 0) and the above procedure is repeated. The R3 is changed upto 1.0 in steps of 0.1 and the procedure is repeated for every incremental value of R3. The output code for the combination of R3, R2 and R1 consumption levels is in accordance with the Table 2. Part of the training data with R3 = 0.3 is shown in the Table 4.

### 4.4 Pushing monitoring data from sensor nodes to sink using CoAP's observe option

This Resource Monitoring application uses CoAP's observe option to push the monitoring data from sensor nodes to sink. This is done at three levels. (1) Sensor node level (2) Cluster level (3) Network level. The clustered tree network structure is considered in the proposed work and is as shown in the Figure 6. The tree like structure is considered for simplicity, although the IPv6 routing protocol allows for each sensor node to have multiple parents when the node's connectivity supports it.

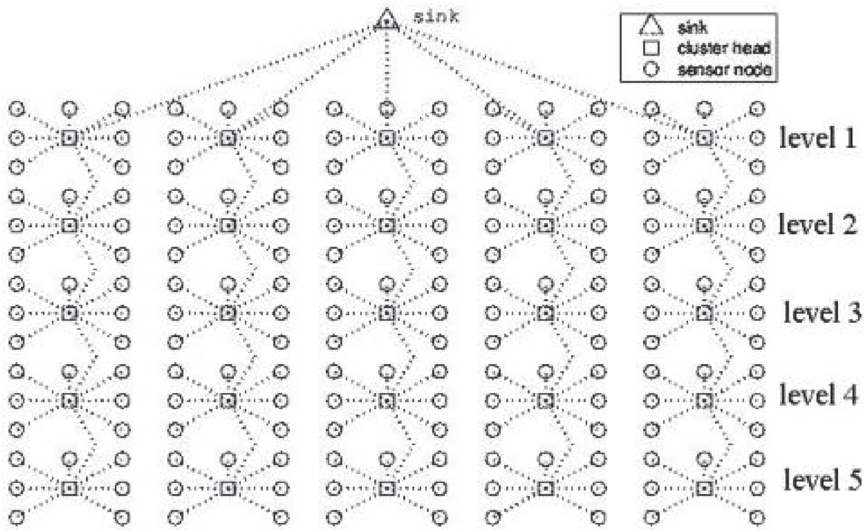



**Figure 6.** Clustered Tree Network

**Sensor node level:** At this level, the single sensor node which serves as a server (has resource monitored data) connected with the sink which is a client. The client registers to the server its interest in resources R3, R2 and R1 using an extended GET request (i.e. with observe option). The server sends notification to the client as indicated in Table 2 and resource consumption patterns in Figure 3.

**Cluster level:** The cluster head is a special node which is rich in resources and capabilities (e.g. routing) compared to sensor nodes. Sensor nodes are connected to cluster head to form star network inside the cluster. The cluster head (client) registers its interest in resources R1, R2 and R3 by initiating an extended GET request (i.e. with observe option) to all sensor nodes (servers)

**Table 4.** ANFIS Training Data

| R3 | R2 | R1 | o/p | R3 | R2 | R1 | o/p | R3 | R2 | R1 | o/p |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.3 | 0.0 | 0.0 | 0 | 0.3 | 0.3 | 0.7 | 1 | 0.3 | 0.7 | 0.3 | 3 |
| 0.3 | 0.0 | 0.1 | 0 | 0.3 | 0.3 | 0.8 | 1 | 0.3 | 0.7 | 0.4 | 2 |
| 0.3 | 0.0 | 0.2 | 0 | 0.3 | 0.3 | 0.9 | 1 | 0.3 | 0.7 | 0.5 | 2 |
| 0.3 | 0.0 | 0.3 | 0 | 0.3 | 0.3 | 1.0 | 1 | 0.3 | 0.7 | 0.6 | 2 |
| 0.3 | 0.0 | 0.4 | 1 | 0.3 | 0.4 | 0.0 | 3 | 0.3 | 0.7 | 0.7 | 2 |
| 0.3 | 0.0 | 0.5 | 1 | 0.3 | 0.4 | 0.1 | 3 | 0.3 | 0.7 | 0.8 | 2 |
| 0.3 | 0.0 | 0.6 | 1 | 0.3 | 0.4 | 0.2 | 3 | 0.3 | 0.7 | 0.9 | 2 |
| 0.3 | 0.0 | 0.7 | 1 | 0.3 | 0.4 | 0.3 | 3 | 0.3 | 0.7 | 1.0 | 2 |
| 0.3 | 0.0 | 0.8 | 1 | 0.3 | 0.4 | 0.4 | 2 | 0.3 | 0.8 | 0.0 | 3 |
| 0.3 | 0.0 | 0.9 | 1 | 0.3 | 0.4 | 0.5 | 2 | 0.3 | 0.8 | 0.1 | 3 |
| 0.3 | 0.0 | 1.0 | 1 | 0.3 | 0.4 | 0.6 | 2 | 0.3 | 0.8 | 0.2 | 3 |
| 0.3 | 0.1 | 0.0 | 0 | 0.3 | 0.4 | 0.7 | 2 | 0.3 | 0.8 | 0.3 | 3 |
| 0.3 | 0.1 | 0.1 | 0 | 0.3 | 0.4 | 0.8 | 2 | 0.3 | 0.8 | 0.4 | 2 |
| 0.3 | 0.1 | 0.2 | 0 | 0.3 | 0.4 | 0.9 | 2 | 0.3 | 0.8 | 0.5 | 2 |
| 0.3 | 0.1 | 0.3 | 0 | 0.3 | 0.4 | 1.0 | 2 | 0.3 | 0.8 | 0.6 | 2 |
| 0.3 | 0.1 | 0.4 | 1 | 0.3 | 0.5 | 0.0 | 3 | 0.3 | 0.8 | 0.7 | 2 |
| 0.3 | 0.1 | 0.5 | 1 | 0.3 | 0.5 | 0.1 | 3 | 0.3 | 0.8 | 0.8 | 2 |
| 0.3 | 0.1 | 0.6 | 1 | 0.3 | 0.5 | 0.2 | 3 | 0.3 | 0.8 | 0.9 | 2 |
| 0.3 | 0.1 | 0.7 | 1 | 0.3 | 0.5 | 0.3 | 3 | 0.3 | 0.8 | 1.0 | 2 |
| 0.3 | 0.1 | 0.8 | 1 | 0.3 | 0.5 | 0.4 | 2 | 0.3 | 0.9 | 0.0 | 3 |
| 0.3 | 0.1 | 0.9 | 1 | 0.3 | 0.5 | 0.5 | 2 | 0.3 | 0.9 | 0.1 | 3 |
| 0.3 | 0.1 | 1.0 | 1 | 0.3 | 0.5 | 0.6 | 2 | 0.3 | 0.9 | 0.2 | 3 |
| 0.3 | 0.2 | 0.0 | 0 | 0.3 | 0.5 | 0.7 | 2 | 0.3 | 0.9 | 0.3 | 3 |
| 0.3 | 0.2 | 0.1 | 0 | 0.3 | 0.5 | 0.8 | 2 | 0.3 | 0.9 | 0.4 | 2 |
| 0.3 | 0.2 | 0.2 | 0 | 0.3 | 0.5 | 0.9 | 2 | 0.3 | 0.9 | 0.5 | 2 |
| 0.3 | 0.2 | 0.3 | 0 | 0.3 | 0.5 | 1.0 | 2 | 0.3 | 0.9 | 0.6 | 2 |
| 0.3 | 0.2 | 0.4 | 1 | 0.3 | 0.6 | 0.0 | 3 | 0.3 | 0.9 | 0.7 | 2 |
| 0.3 | 0.2 | 0.5 | 1 | 0.3 | 0.6 | 0.1 | 3 | 0.3 | 0.9 | 0.8 | 2 |
| 0.3 | 0.2 | 0.6 | 1 | 0.3 | 0.6 | 0.2 | 3 | 0.3 | 0.9 | 0.9 | 2 |
| 0.3 | 0.2 | 0.7 | 1 | 0.3 | 0.6 | 0.3 | 3 | 0.3 | 0.9 | 1.0 | 2 |
| 0.3 | 0.2 | 0.8 | 1 | 0.3 | 0.6 | 0.4 | 2 | 0.3 | 1.0 | 0.0 | 3 |
| 0.3 | 0.2 | 0.9 | 1 | 0.3 | 0.6 | 0.5 | 2 | 0.3 | 1.0 | 0.1 | 3 |
| 0.3 | 0.2 | 1.0 | 1 | 0.3 | 0.6 | 0.6 | 2 | 0.3 | 1.0 | 0.2 | 3 |
| 0.3 | 0.3 | 0.0 | 0 | 0.3 | 0.6 | 0.7 | 2 | 0.3 | 1.0 | 0.3 | 3 |
| 0.3 | 0.3 | 0.1 | 0 | 0.3 | 0.6 | 0.8 | 2 | 0.3 | 1.0 | 0.4 | 2 |
| 0.3 | 0.3 | 0.2 | 0 | 0.3 | 0.6 | 0.9 | 2 | 0.3 | 1.0 | 0.5 | 2 |
| 0.3 | 0.3 | 0.3 | 0 | 0.3 | 0.6 | 1.0 | 2 | 0.3 | 1.0 | 0.6 | 2 |
| 0.3 | 0.3 | 0.4 | 1 | 0.3 | 0.7 | 0.0 | 3 | 0.3 | 1.0 | 0.7 | 2 |
| 0.3 | 0.3 | 0.5 | 1 | 0.3 | 0.7 | 0.1 | 3 | 0.3 | 1.0 | 0.8 | 2 |
| 0.3 | 0.3 | 0.6 | 1 | 0.3 | 0.7 | 0.2 | 3 | 0.3 | 1.0 | 0.9 | 2 |
| **Contd.** | | | | **Contd.** | | | | 0.3 | 1.0 | 1.0 | 2 |

in the cluster. Sink (client) registers its interest in resources R1, R2 and R3 of all sensor nodes of cluster by initiating a extended GET request to the cluster head (acts as server to the sink).

**Network level:** In the hierarchical tree structure shown in the Figure 6, the sink is at depth 0, and clusters are arranged at different depths 1, 2, 3, ..., n. The cluster head acts as client for sensor nodes inside its cluster and server for cluster head of upper cluster head and acts as client of lower cluster head.

## 5 Simulation, Results and Analysis

### 5.1 Simulation Environment

Matlab 7 (R2010a) and its fuzzy logic tool box is used to simulate the ANFIS to generate the membership functions and rules from the training data. We have reduced the ANFIS generated rules from eight to six. The GUI tool anfisedit is used in the simulation process. ANFIS is trained using hybrid learning algorithm.

The sensor node, cluster head, and network environment is simulated using the Java programming language and CoAP with observe option used for transporting resources information is implemented using the Californium (CoAP framework).

### 5.2 Node level

**Simulation:** The sensor node is equipped with ANFIS model. The ANFIS model consists of 25 nodes, three inputs (R3, R2 and R1), six membership functions, six rules and one output. It is trained with 1,331 pairs of input and output training data (Table 4). The parameters (a,b,c,d) of trapezoidal membership functions (two membership functions for each input) $\mu_{less}$(R3), $\mu_{more}$(R3), $\mu_{less}$(R2), $\mu_{more}$(R2), $\mu_{less}$(R1) and $\mu_{more}$(R1) are (-0.7, -0.3, 0.14, 0.39), (0.3, 0.54, 1.3, 1.7), (-0.7, -0.3, 0.14, 0.39), (0.3, 0.54, 1.3, 1.7), (-0.7, -0.3, 0.14, 0.39) and (0.3, 0.54, 1.3, 1.7) respectively. The only output of the ANFIS (Table 2) is the coded output which indicates which resources needs to be pushed (notified) to the client.

The sensor node which has resources to be monitored acts as a server and the sink acts as the client. The registration by the client and subsequent notifications by the server is as shown in the Figure 7. The client uses the GET request with observe option of the CoAP (extended GET) for registration (observe = 0). After registration how communication happens between client and server using token, observe option, response code and payload is already explained under protocol stack. The Figure 7 is simplified version of Figure 2.

All the assumed resources consumption patterns are as shown in the Figure 3. After registration from the client, the server sends the notifications (which are based on Table 2) for every 0.1 time unit up to 1.0 time unit as shown in the Table 2. The detailed notifications for one complete cycle of resources consumption patterns are shown in the Figure 7. The payload of the notification is the resources consumption level which may be just any one resource level (R3 or R2 or R1) or combination of any two or all three resources level as shown in the notifications.

The frequency of resource monitoring data movement from the sensor nodes to the sink depends on the magnitude of the resources of the sensor node and resources consumption patterns. The time unit may vary from network to network.

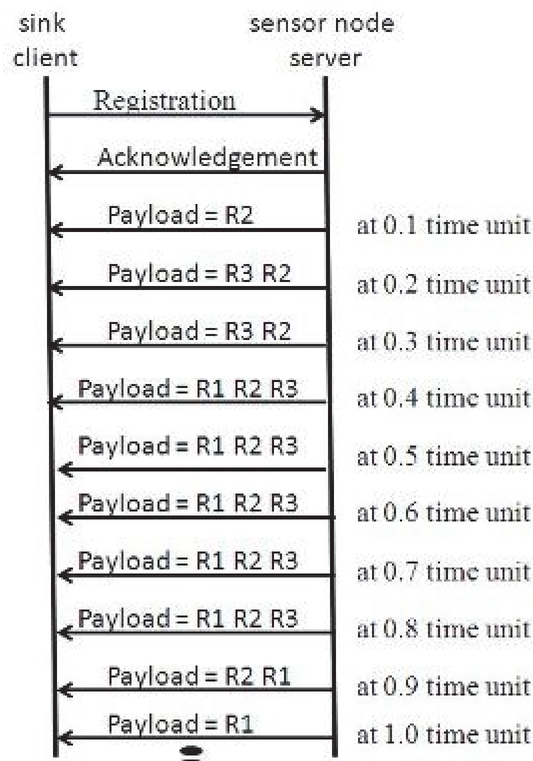



**Figure 7.** Interaction between node and sink

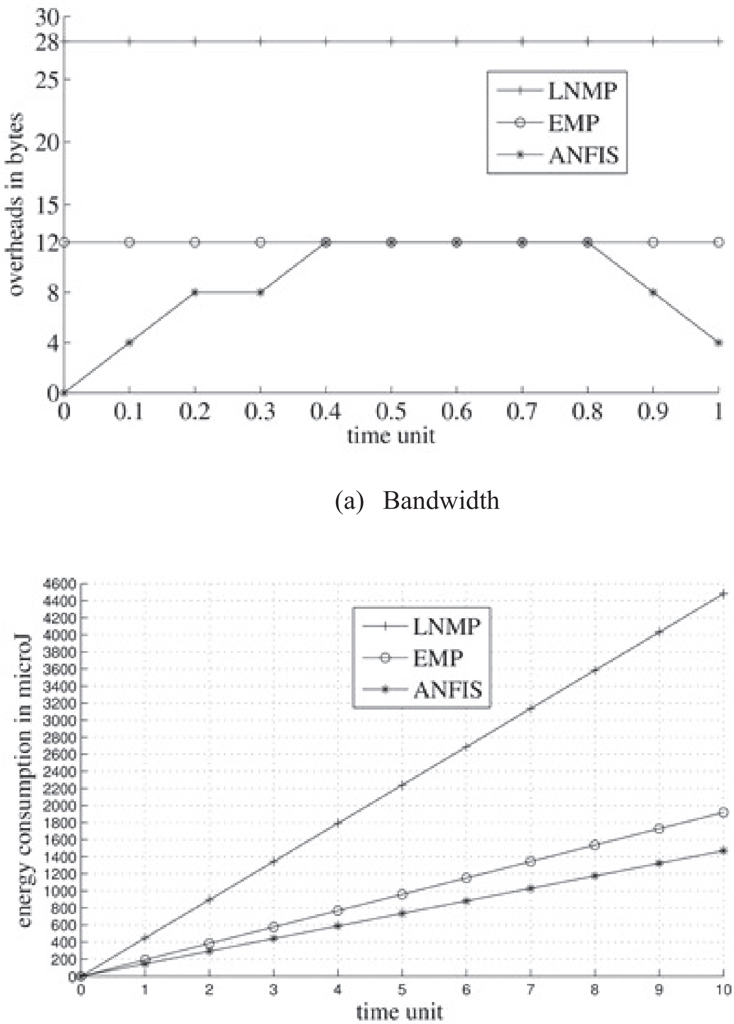
**Results and Analysis:** Resources consumption level cannot be sent in % as the network may have heterogeneous sensor nodes which have different size of memory, processor with different speed and capabilities etc. Let us assume that each resource consumption level indication needs 4 bytes. Then for three resources, 12 bytes required. These resources consumption level is notified to the client for every 0.1 time unit. The complete cycle of one time unit then needs 12 x 10 = 120 bytes of transmission to client. Instead of notifying all three resources to the client (EMP, periodic reporting), the ANFIS decides which resources level needs to be notified to the client based data provided in the Table 2. For the assumed resources consumption pattern (Figure 3), the resources need to be notified is already shown in the Figure 7. The number of bytes to be transmitted for one cycle is graphically shown in the Figure 8(a). Total number of bytes needs to be transmitted is 92 bytes which is 23% less compared to notifying all resources consumption level. This clearly indicates a 23% less consumption of energy by the sensor node and 23% less bandwidth required for the transmission of resource monitoring data. Here the saving of sensor node energy and required bandwidth is only indicative and vary based on resource consumption pattern.

The query processing used in LNMP demands a query request from the client for every fetch of resources information. The CoAP protocol consumes 16 bytes for (ignoring the headers from other layers) a query request. We need to query the server 10 times for one complete cycle. A one complete cycle demand the transmission of 12x10 + 16x10 = 280 bytes (includes both request and response) which is 67% higher compared to our work. Hence, the saving of 67% energy and 67% less bandwidth requirement for the transport of resource monitoring data at the node level when ANFIS and CoAP is used.

The radio model of IEEE 802.15.4 standard approximately consumes 200nJ/bit (based on products survey) and 1600nJ/byte for transmission or reception. Figure 8(b) shows the comparison of energy consumption in LNMP, EMP and ANFIS when used for resource monitoring for 10 time units. The graph clearly shows the energy saving (in sensor node with ANFIS and CoAP) of 23% when compared to EMP and 67% when compared to LNMP.

### 5.3 Cluster level

**Simulation:** Let us consider the cluster of sensor nodes with one special node which functions as cluster head. All sensor nodes are similar and equipped with the ANFIS as described in the node level section. The resources consumption in all sensor nodes is assumed to be same. In the Figure 6 shown, there are seven sensor nodes which are wirelessly connected to the cluster head which in turn wirelessly connected to the sink. The sink which needs resource monitoring data of all the sensor nodes registers itself as a client to the cluster head which acts as server capable of feeding the monitoring data of all sensor nodes to the sink. The cluster head registers as client to all sensor nodes separately for requesting monitoring data. The process of registration and resource monitoring data movement from sensor nodes to cluster head and then from cluster head to sink is shown in the Figure 9. The details of process of registration by the client and notifications from the server are exactly same as details presented in the node level section.




(a) Bandwidth

(b) Energy

**Figure 8.** Performance at node level

The notifications from the sensor node (servers) at 0.1 unit time is collected by the cluster head (acts as client) and all these notifications are sent by cluster head (acts as server) to sink (client) in a single packet. Subsequent notifications from the sensor nodes at 0.2, 0.3, 0.4, ..., 0.9 and 1.0 time unit is collected by the cluster head (client) and sent to the sink in sequence as shown in the Figure 9.

**Results and Analysis:** The total number of bytes required to send all resources monitoring data (R3, R2, R1) of all sensor nodes in a cluster from the cluster head to the sink is 7 sensors x 3 resources x 4 bytes/resource x 10 times/cycle = 840 bytes. The sensor nodes equipped with ANFIS are capable of deciding which resources consumption level needs to be sent to the cluster head. The Figure 10(a) indicates how many bytes needs to be transmitted for one complete cycle of consumption pattern of resources R3, R2, R1. At 0.1 time unit, only R2 of all sensor nodes is sent to cluster head which is 7 sensor nodes x 1 resource x 4 bytes = 28 bytes. For one complete cycle of resources consumption pattern, the total number of bytes received by the cluster head from all sensor nodes together is 644 bytes which is sent to sink.

This is approximately 23% less compared to 840 bytes (EMP, periodic reporting) which indicates 23% saving in energy consumption and bandwidth requirement for transporting monitoring data from cluster head to the sink.

The resources information of sensor nodes reaches the sink in two hops. There is a 23% saving of energy and bandwidth for each hop.

The query processing used in LNMP demands the transmission of (84 bytes x 10 times + 16 bytes x 10 times x 7 sensors) = 1960 bytes which is 67% more compare to our work. Hence, the saving of 67% energy and bandwidth for each hop.

Figure 10(b) shows the comparison of energy consumption in LNMP, EMP and ANFIS when used for resource monitoring for 10 time units and one hop (cluster head to sink). The graph clearly shows the energy saving of 23\% when compared to EMP and 67\% when compared to LNMP.

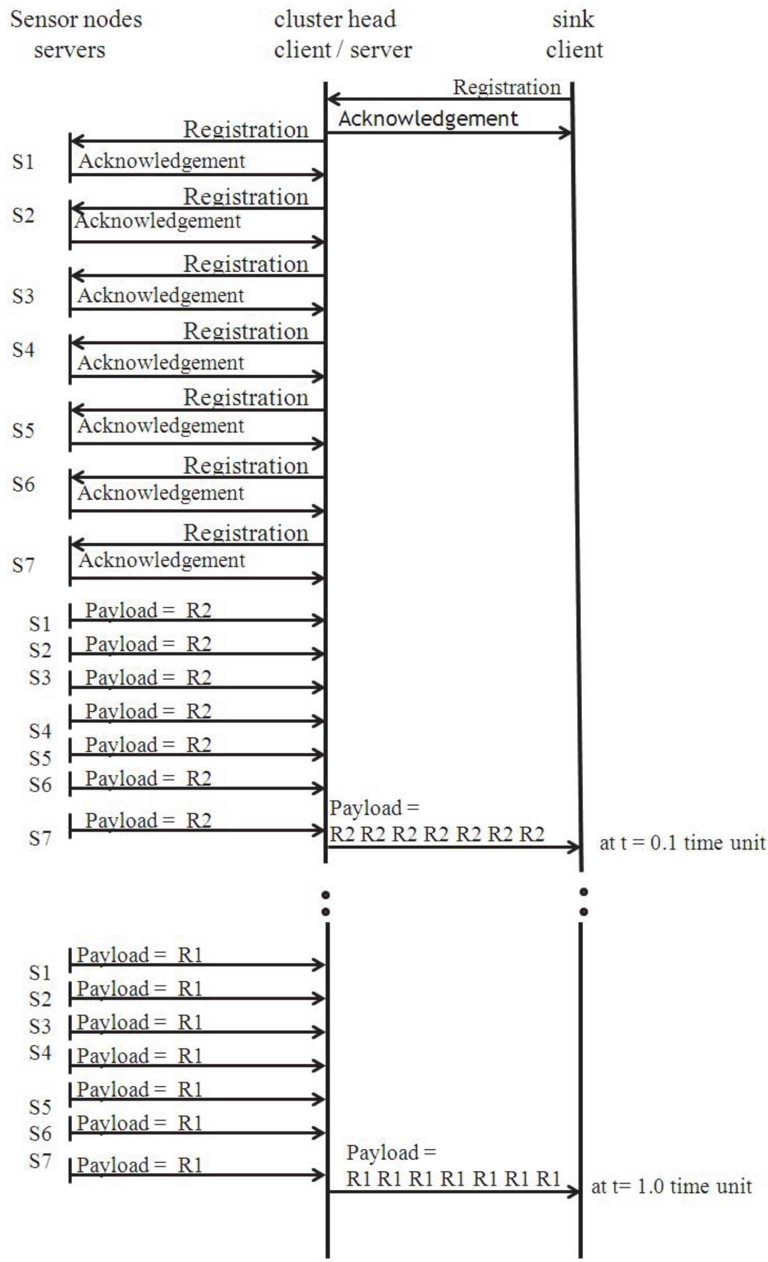
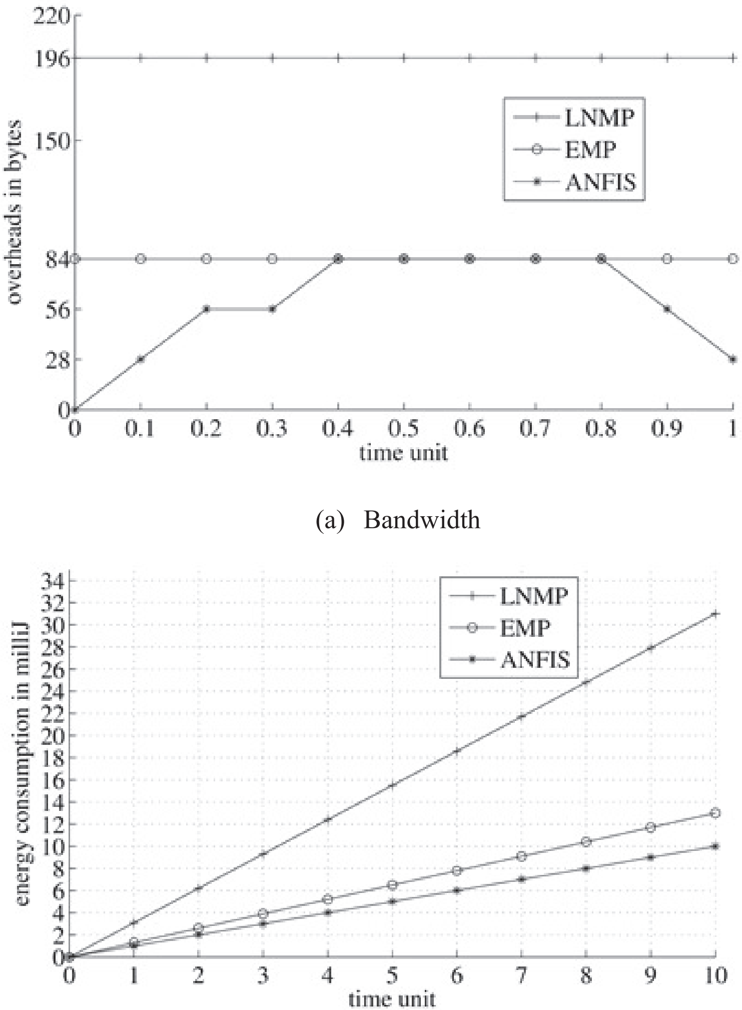



**Figure 9.** Interaction between sensor nodes, cluster head and sink




(a) Bandwidth



(b) Energy

**Figure 10.** Performance at cluster level

## 5.4 Network level

**Analysis:** By using the ANFIS, the transmission of resource monitoring data from sensor node to cluster head is reduced from 120 bytes/time unit (EMP, periodic reporting) into 92 bytes/time unit. The network of n sensor nodes with ANFIS in each node reduces the need of transmitting 120 x n bytes into 92 x n bytes which is 23% saving of energy and bandwidth requirement for transporting monitoring data from sensor nodes to the clustered head.

Let us consider the clustered tree network with equal number of sensor nodes in each cluster for analysis. The number of bytes saved is 28 bytes x number of sensor nodes in the cluster x number of branches from the root (sink) x depth of tree (sink at depth 0). For 175 sensor nodes clustered tree network with 7 sensor nodes in each cluster, with five branches and five levels as shown in the Figure 6, the number of bytes saved is 28 x 7 x 5 x 5 = 4900 bytes/time unit which is 23% saving compared to 21000 bytes/time unit (EMP, periodic reporting) as shown in the Figure 11(a). Again there is a saving of 23% of energy and bandwidth while transporting each cluster monitoring data from cluster head to sink. The monitoring data movement from cluster head to sink takes more than one hop if the cluster is at depth greater than one. There is a saving of 23% of energy and bandwidth for each hop of data movement from cluster head.

The query processing for fetching resources information using LNMP demands 280 bytes/time unit as compared to 92 bytes/time unit required by ANFIS. This is about 67% less data movement from sensor node to cluster head and saves 67% energy and bandwidth. For the above mentioned clustered tree network, it is the saving of 188 x 7 x 5 x 5 = 32,900 bytes/time unit which is 67% saving compared to 49,000 bytes/time unit (LNMP, query processing) as shown in the Figure 11(a). Hence, there is saving of 67% of energy and bandwidth for each hop of the data movement from cluster head.

Figure 11(b) shows the comparison of energy consumption in LNMP, EMP and ANFIS when used for resource monitoring for 10 time units. The graph clearly shows the energy saving in sensor node with ANFIS and CoAP (257 mJ for 10 time units) of 23% when compared to EMP (336 mJ for 10 time units) and 67% when compared to LNMP (784 mJ for 10 time units).

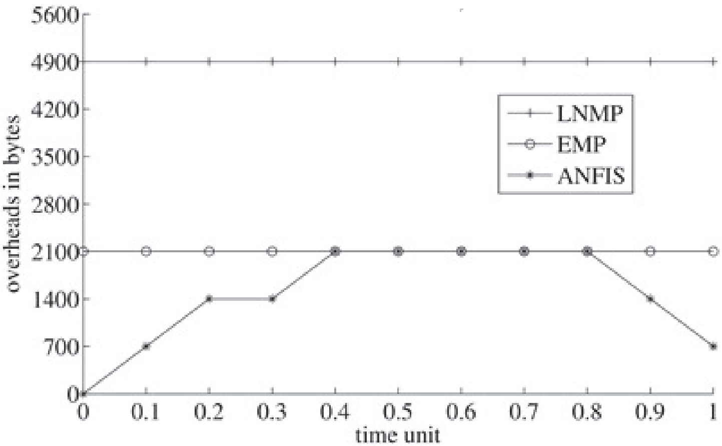
If there are two different resources consumption patterns (R3, R2, R1 for 'a' nodes and R6, R5, R4 for 'b' nodes where a + b = 7) in each cluster, then number bytes saved from transmission is different for different resource consumption pattern (say p \& q). Then bytes saved from transmission is (p x a x 5 x 5) + (q x b x 5 x 5).

In general, for a sensor network of n nodes with different resources consumption patterns for different sensor nodes

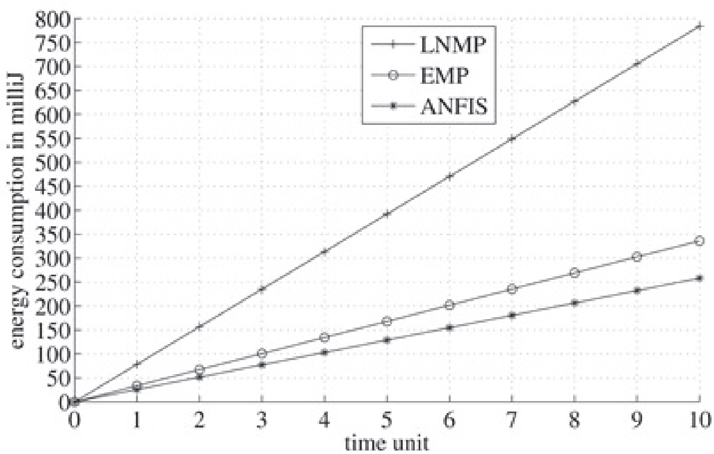
Total number of bytes saved from transmission (nodes to cluster head)

$$= \sum_{i=1}^{n} a_i$$

where $a_i$ - bytes saved from transmission for $i^{th}$ sensor node per hop, i = 1, 2, 3, ..., n-1,n;




(a) Bandwidth




(b) Energy

**Figure 11.** Performance at network level

The saving of energy in all the above cases is at the transmitting end as well as at the receiving end (receiving also consumes energy).

## 6 Conclusions

WSNs are becoming part of Internet of Things and getting merged into mainstream Internet. In the proposed work for resource monitoring of WSNs, sensor nodes and network are considered as Internet of Things. Each and every sensor node is simulated as web server, cluster head as both web server and web client, and sink as web client. Data related to monitoring of resources of sensor node (processing speed, memory, energy, bandwidth) are pushed to the client (cluster head) using the observe option of the CoAP protocol. The sensor nodes are equipped with the ANFIS which decides which resources are to be pushed to the client. Use of ANFIS in sensor nodes reduces resource monitoring data size by 23\% per hop when compared to EMP (periodic reporting) and 67\% per hop when compared to LNMP (query processing). This results in the saving of energy and bandwidth requirement (in sensor node with ANFIS) by 23\% when compared to EMP and 67\% when compared to LNMP. The simulation is conducted at two levels: (1) node level: A sensor node which is equipped with ANFIS notifies resource monitoring data to sink which is a client. (2) Cluster level: All sensor nodes (web servers) of the cluster are equipped with the ANFIS and notify resource monitoring data to the cluster head (web client) which in turn notifies (now acts as web server) the same to the sink (web client). Effect of using ANFIS in senor node for resource monitoring in clustered tree network is analyzed.

## References


1. Ian. F. Akyildiz, Weillan Su, Yogesh Sankarasubramaniam and Erdal Cayirci, 2002, *A Survey on Sensor Networks*, IEEE Communication Magazine Vol.40, No.8, pp.102-114.
2. Zack Shelby, 2010. *Embedded Web Services*, IEEE Wireless Communications, Vol.17, No.6, pp.52-57.
3. Jyh-shing Roger Jang, 1993, *ANFIS: Adaptive network base Fuzzy Inference System*, IEEE Transactions on Systems, Man and Cybernetics Vol.23, No.03, pp.665-685
4. Shelby Z., Hartke K. and Bormann C., 2014, *The IETF's, The Constrained Application Protocol*, https://datatracker.ietf.org/doc/rfc7252/
5. Hartke K., 2015, *Observing Resources in the Constrained Application Protocol (CoAP)*, http://www.rfc-editor.org/info/rfc7641/
6. Yonggang Jerry Zhao, Ramesh Govindhan and Deborah Estrin, 2002, *Residual Energy scan for Monitoring Sensor Networks*, https://escholarship.org/uc/item/2st0t8cf
7. Raquel A. F. Mini, Max do Val Machado, Antonio A. F. Loureiro and Badri Nath, 2005, *Prediction - based energy map for wireless sensor networks*, Ad Hoc Networks, Vol 3, No. 2, pp. 235-253
8. Edward Chan and Song Han, 2009*, Energy Efficient Residual Energy Monitoring in Wireless Sensor Networks*, International Journal of Distributed Sensor Networks, Vol 5, pp. 748-770
9. Alec Woo, Terence Tong and David Culler, 2003, *Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks,* SenSys' 03, November 5-7, 2003, Los Angeles, California, USA.

10. Chieh-Yih, Shane B. Eisenman and Andrew T. Campbell, 2003*, CODA:Congestion Detection and Avoidance in Sensor Networks*, SenSys' 03, November 5-7, 2003, Los Angeles, California, USA.
11. Li Qiang Tao and Feng Qi Yu, 2010, *ECODA:enhanced congestion detection and avoidance for multiple class of traffic in sensor networks*, IEEE Transaction on Consumer Electronics, Vol 56, No. 3, pp.1387-1394
12. Falko Dressler and Dominik Neuner, 2013, *Energy-Efficient Monitoring of Distributed System Resources for Self-Organizing Sensor Networks*, IEEE Topical conference on wireless sensors networks (WiSNET), Jan 20-23, 2013 Austin, TX, PP.145-147
13. Winnie Louis Lee, Amitava Datta and Rachel Cardell-Oliver, 2007, *Network Management in Wireless Sensor Networks,* https://cpn.unl.edu/
14. Hamid Mukhtar, Kim Kang-Myo, Shafique Ahmad Chaudhry, Ali Hammad Akbar, Kim Ki-Hyung, Seung-Wha Yoo, 2008, *LNMP-Management architecture of IPv6 based low-power wireless Personal Area Networks (6LoWPAN)*, IEEE Network operations and Management Symposium, April 7-11 2008, Salvador Bahia, pp.417-424
15. Shafique Ahmad Chaudhry, Weiping Song, Muhammad Habeeb Vulla, Cormac Sreenan, 2011, *EMP:A Protocol for IP Based Wireless Sensor Networks Management*, Journal of Ubiquitous Systems and Pervasive Networks, Vol. 2, No. 1, pp. 15-22.
16. Zhengguo Sheng, Hao Wang, Changchuan Yin, Xiping Hu, Shusen Yang, Victor C. M. Leung, 2015, *Lightweight Management of Resource Constrained Sensor Devices in Internet of Things*, IEEE Internet of Things Journal, Vol. 2, No.5, pp. 402-411.
17. Nagesha and Sunilkumar S. Manvi, 2016, *ANFIS based Resource Mapping for Query Processing in Wireless Multimedia Sensor Networks,* Journal of Intelligent Systems, accepted, http://www.degruyter.com/printahead/j/jisys
18. Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Gennaro Boggia, Mischa Dohler, 2013, *Standardized protocol stack for the Internet of (Important) Things*, IEEE Communication Surveys and Tutorials, Vol. 15, No. 3, pp.1389-1406.
19. Wireless personal area network (WPAN) working group, 2011, *Low-Rate Wireless Personal Area Networks (LR-WPANs),* http://standards.ieee.org.
20. Kushalnagar N., Montenegro G. and Schumacher C., 2007, *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, problem Statement, and Goals*, https://datatracker.ietf.org/doc/rfc4919/
21. Winter,T., Thubert,P., Brandt,A., Hui,J., Kelsey,R., Levis,P., Pister,K., Struik,R., JP.Vasseur and Alexander,R., 2012, *RPL:IPv6 Routing Protocol for Low-Power and Lossy Networks,* https://datatracker.ietf.org/doc/rfc6550/
22. Tomohiro Takagi and Michio Sugeno, 1985, *Fuzzy Identification of Systems and its Application to Modelling and Contro*l, IEEE Transactions on Systems, Man and Cybernetics, Vol. 15, pp.116-132

# MONADIC PRINTING REVISITED

Konrad Grzanek

IT Institute, University of Social Sciences
9 Sienkiewicza St., 90-113 Łódź, Poland
*kgrzanek@spoleczna.pl*

**Abstract**

Expressive and clear implementation of monadic printing requires an amount of work to define and design proper abstractions to rely upon when performing the actual programming works. Our previous realization of tree printing library left us with a sense of lack with respect to these considerations. This is why we decided to re-design and re-implement the library with core algorithms based upon new, effective and expressive text printing and concatenation routines. This paper presents the results of our work.

**Keywords:** Functional programming, monads, Haskell, polymorphism

## 1 Introduction

Textual presentation of data structures is invariably one of the most effective ways to visualize them, especially when it comes to presentation of large data structures. The ability to display textual content and working on the presentation results with automated text-processing tools sometimes makes this way of visualizing much more appealing to the end-user than displaying using GUI views. The data structure that is especially susceptible to this approach is tree, or – even more generally – DAG (Directed Acyclic Graph).

Our previous work on this subject aimed towards creating a library for visualizing trees and DAGs. Our few years old paper [5] presented a library for Haskell [1, 2], the purely functional and statically typed programming language. The library described there possessed the following properties:

- The ability to generate representations of arbitrary DAGs.
- Writing to any monad including IO. This also means it was capable of writing to normal Haskel Strings (lists of Char) via Identity monad.
- Extensive use of Haskell type-system to verify correctness of the usage scenarios.

Unfortunately, the design and implementation of this library was not perfect. It missed expressiveness and the clarity of algorithm formulation. These issues led to extensive re-design of the library. The updated architecture of the library consists of:

– Printing abstraction,
– String/Text concatenation routines,
– Re-designed tree printing implemented on top of the two previous ones.

This paper is an attempt to present all the details of the refreshed library.

## 2 Printing Abstraction and Its Implementations

Generic printing mechanisms are defined in *Kask.Print* module [6]. All its contents are defined in the presence of the following import clauses:

```
import qualified Control.Monad.State.Strict as S
import Data.Monoid ((<>))
import qualified Data.Text as T
import qualified Data.Text.IO as TIO
import qualified Data.Text.Lazy as TL
import qualified Data.Text.Lazy.Builder as TLB
import qualified Data.Text.Lazy.IO as TLIO
import Prelude hiding (print)
```

The most essential abstraction is a *type-class* called *Printable*. It is parameterized by two *type-arguments* out of which the first one, *m*, is a *monad* [4].

We have two procedures defined here, namely *print* and *printLn*. They both return a *unit-type* in the monad *m*. The *printLn* works exactly like *print*, but it adds a newline character to the end of the printed entity of type *p*:

```
class Monad m ⇒ Printable m p where
  print :: p → m ()
  printLn :: p → m ()
```

### 2.1 IO Monad

The *Printable* type-class is implemented within the *IO monad* for a collection of textual data-types, like *String*, *ShowS*, and *Text*, either lazily and eagerly evaluated. See the listing below:

```
instance Printable IO String where
  print = putStr
  printLn = putStrLn
   {-# INLINE print #-}
   {-# INLINE printLn #-}
instance Printable IO ShowS where
```

$$
\begin{array}{l}
\quad \mathit{print} = \mathit{print} \circ \mathit{evalShowS} \\
\quad \mathit{printLn} = \mathit{putStrLn} \circ \mathit{evalShowS} \\
\qquad \{\text{-\# INLINE print \#-}\} \\
\qquad \{\text{-\# INLINE printLn \#-}\} \\
\mathbf{instance}\ \mathit{Printable\ IO\ T.Text}\ \mathbf{where} \\
\quad \mathit{print} = \mathit{TIO.putStr} \\
\quad \mathit{printLn} = \mathit{TIO.putStrLn} \\
\qquad \{\text{-\# INLINE print \#-}\} \\
\qquad \{\text{-\# INLINE printLn \#-}\} \\
\mathbf{instance}\ \mathit{Printable\ IO\ TL.Text}\ \mathbf{where} \\
\quad \mathit{print} = \mathit{TLIO.putStr} \\
\quad \mathit{printLn} = \mathit{TLIO.putStrLn} \\
\qquad \{\text{-\# INLINE print \#-}\} \\
\qquad \{\text{-\# INLINE printLn \#-}\}
\end{array}
$$

We also provide an *IO-monadic* implementation for an effective textual builder defined in *Data.Text.Lazy.Builder*, like:

$$
\begin{array}{l}
\mathbf{instance}\ \mathit{Printable\ IO\ TLB.Builder}\ \mathbf{where} \\
\quad \mathit{print} = \mathit{print} \circ \mathit{TLB.toLazyText} \\
\quad \mathit{printLn} = \mathit{printLn} \circ \mathit{TLB.toLazyText} \\
\qquad \{\text{-\# INLINE print \#-}\} \\
\qquad \{\text{-\# INLINE printLn \#-}\}
\end{array}
$$

## 2.2 Text in the State Monad

Another interesting monad to mention here is the *state monad*, as defined in *Control.Monad.State.Stric*. We define a special type *TextBuilder* to wrap the textual state management within an useful text-coercible abstraction:

$$
\begin{array}{l}
\mathbf{type}\ \mathit{TextBuilder} = \mathit{S.State\ T.Text} \\
\mathit{toText} :: \mathit{TextBuilder}\ () \to \mathit{T.Text} \\
\mathit{toText\ tb} = \mathit{snd}\ (\mathit{S.runState\ tb}\ \texttt{""}) \\
\quad \{\text{-\# INLINE toText \#-}\}
\end{array}
$$

The *TextBuilder* monad has the following *Printable* implementations for *String* and *ShowS*:

$$
\begin{array}{l}
\mathbf{instance}\ \mathit{Printable\ TextBuilder\ String}\ \mathbf{where} \\
\quad \mathit{print} = \mathit{print} \circ \mathit{T.pack} \\
\quad \mathit{printLn} = \mathit{printLn} \circ \mathit{T.pack} \\
\qquad \{\text{-\# INLINE print \#-}\} \\
\qquad \{\text{-\# INLINE printLn \#-}\} \\
\mathbf{instance}\ \mathit{Printable\ TextBuilder\ ShowS}\ \mathbf{where}
\end{array}
$$

```
    print = print ∘ evalShowS
    printLn = printLn ∘ evalShowS
      {-# INLINE print #-}
      {-# INLINE printLn #-}
```

as well as for eagerly, and lazily evaluated *Text*:

```
instance Printable TextBuilder T.Text where
    print txt = do
        buf ← S.get
        S.put (T.append buf txt)
      {-# INLINE print #-}
    printLn txt = do
        buf ← S.get
        S.put (T.append (T.append buf txt) "\n")
      {-# INLINE printLn #-}
instance Printable TextBuilder TL.Text where
    print = print ∘ TL.toStrict
    printLn = printLn ∘ TL.toStrict
      {-# INLINE print #-}
      {-# INLINE printLn #-}
```

We also provide implementation for *Data.Text.Lazy.Builder* like in the case of *IO* monad:

```
instance Printable TextBuilder TLB.Builder where
    print = print ∘ TLB.toLazyText
    printLn = printLn ∘ TLB.toLazyText
      {-# INLINE print #-}
      {-# INLINE printLn #-}
```

## 2.3 Lazy Text Builder in the State Monad

Eagerly evaluated state monad may be also used as a basis for a lazily evaluated string builder, as defined below, together with two state evaluators:

```
type LazyTextBuilder = S.State TLB.Builder
toLazyTextBuilder :: LazyTextBuilder () → TLB.Builder
toLazyTextBuilder tb = snd $ S.runState tb $ TLB.fromString ""
  {-# INLINE toLazyTextBuilder #-}
toLazyText :: LazyTextBuilder () → TL.Text
toLazyText = TLB.toLazyText ∘ toLazyTextBuilder
  {-# INLINE toLazyText #-}
```

Like in the case of the previous monadic implementations, firstly we define the implementations for *String* and *ShowS*:

```
instance Printable LazyTextBuilder String where
  print = print ∘ T.pack
  printLn = printLn ∘ T.pack
   {-# INLINE print #-}
   {-# INLINE printLn #-}
instance Printable LazyTextBuilder ShowS where
  print = print ∘ evalShowS
  printLn = printLn ∘ evalShowS
   {-# INLINE print #-}
   {-# INLINE printLn #-}
```

as well as for strictly and lazily evaluated *Text* instances:

```
instance Printable LazyTextBuilder T.Text where
  print = print ∘ TLB.fromText
  printLn = printLn ∘ TLB.fromText
   {-# INLINE print #-}
   {-# INLINE printLn #-}
instance Printable LazyTextBuilder TL.Text where
  print = print ∘ TLB.fromLazyText
  printLn = printLn ∘ TLB.fromLazyText
   {-# INLINE print #-}
   {-# INLINE printLn #-}
```

To make this realization conceptually coherent with the previous ones, we also provide an implementation for *TLB.Builder* (as it was presented in the previous sub-sections):

```
instance Printable LazyTextBuilder TLB.Builder where
  print b = do
    builder ← S.get
    S.put (builder <> b)
   {-# INLINE print #-}
  printLn b = do
    builder ← S.get
    S.put (builder <> b <> TLB.fromLazyText "\n")
   {-# INLINE printLn #-}
```

## 2.4 ShowS in the State Monad

For *ShowS* type we define a separate State Monad instance, together with the following evaluators:

```
type StringBuilder = S.State ShowS
evalShowS :: ShowS → String
```

```
evalShowS s = s ""
  {-# INLINE evalShowS #-}
toShowS :: StringBuilder () → ShowS
toShowS tb = snd (S.runState tb (showString ""))
  {-# INLINE toShowS #-}
toString :: StringBuilder () → String
toString = evalShowS ∘ toShowS
  {-# INLINE toString #-}
```

The *String* and *ShowS* instances of the *Printable* type-class raise up in a natural way:

```
instance Printable StringBuilder String where
  print = print ∘ showString
  printLn = printLn ∘ showString
    {-# INLINE print #-}
    {-# INLINE printLn #-}
instance Printable StringBuilder ShowS where
  print s = do
    buf ← S.get
    S.put (buf ∘ s)
   {-# INLINE print #-}
  printLn s = do
    buf ← S.get
    S.put (buf ∘ s ∘ showString "\n")
   {-# INLINE printLn #-}
```

along with *Text* instances, like in the following listing:

```
instance Printable StringBuilder T.Text where
  print = print ∘ T.unpack
  printLn = printLn ∘ T.unpack
    {-# INLINE print #-}
    {-# INLINE printLn #-}
instance Printable StringBuilder TL.Text where
  print = print ∘ TL.toStrict
  printLn = printLn ∘ TL.toStrict
    {-# INLINE print #-}
    {-# INLINE printLn #-}
instance Printable StringBuilder TLB.Builder where
  print = print ∘ TLB.toLazyText
  printLn = printLn ∘ TLB.toLazyText
    {-# INLINE print #-}
    {-# INLINE printLn #-}
```

## 3 Compatible Abstraction for Concatenation

Early in the design phase it became apparent that we might use the *Printable* for string concatenation. After all the concatenation may be viewed here as printing into the concatenating (string/text builder) object. To make things clear we provide the following *StrCat* type-class, that is another useful abstraction in our library:

**class** *StrCat c* **where**
  $strCat :: (Foldable\ t) \Rightarrow t\ c \rightarrow c$

Concatenation is being treated as a *fold* (e.g. see [3]) operation, that's why we define the *strCat* mechanism as taking place inside a *Foldable*.

Functional merging of *StrCat* and *Printable* takes place via the following *strCatWith* procedure:

$strCatWith :: (Printable\ m\ c, Foldable\ t) \Rightarrow (m\ () \rightarrow c) \rightarrow t\ c \rightarrow c$
$strCatWith\ f = f \circ mapM\_\ print$
 {-# INLINE strCatWith #-}

This immediately allows us to provide *StrCat* implementations for *String* and *ShowS*:

**instance** *StrCat String* **where**
  *strCat = strCatWith toString*
   {-# INLINE strCat #-}
**instance** *StrCat ShowS* **where**
  *strCat = strCatWith toShowS*
   {-# INLINE strCat #-}

The same approach applies to *Text* and *TLB.Builder*:

**instance** *StrCat T.Text* **where**
  *strCat = strCatWith toText*
   {-# INLINE strCat #-}
**instance** *StrCat TL.Text* **where**
  *strCat = strCatWith toLazyText*
   {-# INLINE strCat #-}
**instance** *StrCat TLB.Builder* **where**
  *strCat = strCatWith toLazyTextBuilder*
   {-# INLINE strCat #-}

## 4 Re-designed Tree Printing

All abstractions and their implementations described so far allow us to provide an updated realization of tree printing, previously defined and presented in [5]. The new realization can be viewed as a whole in *Kask.Data.Tree.Print* module [7]. In the presence of the following import clauses:

**import** $Control.Monad\ (unless, forM\_)$
**import** $Data.Foldable\ (toList)$
**import** $qualified\ Data.Text\ as\ T$
**import** $qualified\ Data.Text.Lazy\ as\ TL$
**import** $qualified\ Data.Text.Lazy.Builder\ as\ TLB$
**import** $qualified\ Kask.Constr\ as\ C$
**import** $Kask.Data.List\ (markLast)$
**import** $qualified\ Kask.Print\ as\ P$
**import** $Prelude\ hiding\ (Show, show)$

we have the basic type definitions like below:

**type** $Adjs\ a\ t = Foldable\ t \Rightarrow a \to t\ a$
**type** $Show\ a\ s = Symbolic\ s \Rightarrow a \to s$
**type** $Depth = C.Constr\ (C.BoundsConstr\ C.Positive)\ Int$

One additional visible change with respect to mechanisms defined in [5] relates to *Depth* - a new data type that describes the maximum depth of tree-printing. Currently it is a positive integer, with the contract enforced by using *Constr* and *BoundsConstr*, an effective compile-time contract definition routines, also provided by the *kask* repository.

## 4.1 Tree Printing API

Essentially it consists of a single procedure *printTree* with the following signature and implementation:

$printTree :: (P.Printable\ m\ s, Symbolic\ s, Foldable\ t) \Rightarrow$
  $a \to Adjs\ a\ t \to Show\ a\ s \to Maybe\ Depth \to m\ ()$
$printTree\ node\ adjacent\ show\ maxDepth =$
  $doPrintTree\ node\ adjacent\ show$ (**case** $maxDepth$ **of**
    $Just\ d \to C.unconstr\ d - 1$
    $Nothing \to maxBound)$
    $0$ -- initial level is 0-th
    $[True]$ -- node has no siblings
    $True$ -- .. and it is the first one

## 4.2 Simplified and More Expressive Tree Printing Implementation

The procedure takes the following form:

$doPrintTree :: (P.Printable\ m\ s, Symbolic\ s, Foldable\ t) \Rightarrow$
  $a \to Adjs\ a\ t \to Show\ a\ s \to Int \to Int \to [Bool] \to Bool \to m\ ()$

```
doPrintTree node adjacent show maxDepth level
  lastChildMarks isFirst = do
  let s     = show node
      pfx = if isFirst then empty else eol
      repr = if level ≡ 0
        then P.strCat [pfx, s]
        else P.strCat [pfx, genIndent lastChildMarks, s]
  P.print repr
  unless (level ≡ maxDepth) $ do
    let children = toList $ adjacent node
    forM_ (zip children (markLast children)) $ λ(child, isLast) →
      doPrintTree child adjacent show maxDepth (level + 1)
        (isLast : lastChildMarks) False
```

All the printing, concatenation and string-building abstractions allowed us to achieve two goals:

1. Make the implementation clear and obvious.
2. Make the API expressive.

The clarification seems apparent here, and the expressiveness enhancement takes place thanks to powerful compile time abstractions provided in the signature: *Printable*, *Foldable*, *Adjs*, *Show*.

## 4.3 Further Implementation Details

String concatenation abstraction is also used to implement properly the indentation used to layout the printed tree:

```
genIndent :: Symbolic s ⇒ [Bool] → s
genIndent [ ] = empty   -- should not happen anyway
genIndent (isLast : lastChildMarks) = P.strCat [prefix, suffix]
  where
    indentSymbol True = emptyIndent
    indentSymbol False = indent
    suffix = if isLast then forLastChild else forChild
    prefix = P.strCat $ fmap indentSymbol $ reverse $ init lastChildMarks
```

Additionally we use a *Symbolic* type class that holds the information about all textual elements forming the tree printing layout. The abstraction is defined as:

```
class P.StrCat s ⇒ Symbolic s where
  indent       :: s
  emptyIndent  :: s
  forChild     :: s
  forLastChild :: s
  eol          :: s
  empty        :: s
```

with the following realization for *String* and *ShowS*:

```
instance Symbolic String where
  indent       = "    "
  emptyIndent  = "     "
  forChild     = "  "
  forLastChild = "  "
  eol          = "\n"
  empty        = ""
instance Symbolic ShowS where
  indent       = showString (indent       :: String)
  emptyIndent  = showString (emptyIndent  :: String)
  forChild     = showString (forChild     :: String)
  forLastChild = showString (forLastChild :: String)
  eol          = showString (eol          :: String)
  empty        = showString (empty        :: String)
```

Finally, we also provide an implementation for *Text*:

```
instance Symbolic T.Text where
  indent       = T.pack (indent       :: String)
  emptyIndent  = T.pack (emptyIndent  :: String)
  forChild     = T.pack (forChild     :: String)
  forLastChild = T.pack (forLastChild :: String)
  eol          = T.pack (eol          :: String)
  empty        = T.pack (empty        :: String)
instance Symbolic TL.Text where
  indent       = TL.pack (indent       :: String)
  emptyIndent  = TL.pack (emptyIndent  :: String)
  forChild     = TL.pack (forChild     :: String)
  forLastChild = TL.pack (forLastChild :: String)
  eol          = TL.pack (eol          :: String)
  empty        = TL.pack (empty        :: String)
```

and for *TLB.Builder*:

```
instance Symbolic TLB.Builder where
  indent       = TLB.fromText (indent       :: T.Text)
  emptyIndent  = TLB.fromText (emptyIndent  :: T.Text)
  forChild     = TLB.fromText (forChild     :: T.Text)
  forLastChild = TLB.fromText (forLastChild :: T.Text)
  eol          = TLB.fromText (eol          :: T.Text)
  empty        = TLB.fromText (empty        :: T.Text)
```

## References


1. Peyton Jones S., 1987, The Implementation of Functional Programming Languages, Prentice-Hall International Series in Computer Science. Prentice Hall International (UK) Ltd

2. Lipovaca M., 2011, Learn You a Haskell for Great Good!: A Beginners Guide, No Starch Press; 1st edition (April 21, 2011)

3. Bird R., Wadler R., 1988, Introduction to Functional Programming. Series in Computer Science (Editor: C.A.R. Hoare), Prentice Hall International (UK) Ltd

4. Awodey S., 2010, Category Theory, Second Edition, Oxford University Press

5. Grzanek K., 2014, Monadic Tree Print, JACSM 2014, Vol. 6, No. 2, pp. 147-157

6. GitHub, 2016, Kask.Print module: Kask repository, https://github.com/kongra/kask/blob/master/src/Kask/Print.hs

7. GitHub, 2016, Kask.Data.Tree.Print module: https://github.com/kongra/kask/blob/master/src/Kask/Data/Tree/Print.hs

# Information for Authors

Authors are invited to submit original papers, within the scope of the JACSM, with the understanding that their contents are unpublished and are not being actively under consideration for publication elsewhere.

For any previously published and copyrighted material, a special permission from the copyright owner is required. This concerns, for instance, figures or tables for which copyright exists. In such a case, it is necessary to mention by the author(s), in the paper, that this material is reprinted with the permission.

Manuscripts, in English, with an abstract and the key words, are to be submitted to the Editorial Office in electronic version (both: source and pdf formats) via e-mail. A decision to accept/revise/reject the manuscript will be sent to the Author along with the recommendations made by at least two referees. Suitability for publications will be assessed on the basis of the relevance of the paper contents, its originality, technical quality, accuracy and language correctness.

Upon acceptance, Authors will transfer copyright of the paper to the publisher, i.e. the Academy of Management, in Lodz, Poland, by sending the paper to the JACSM Editorial Office.

The papers accepted for publication in the JACSM must be typeset by their Authors, according to the requirements concerning final version of the manuscripts. However, minor corrections may have been carried out by the publisher, if necessary.

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm.

The first page of each paper should contain its title, the name of the author(s) with the affiliation(s) and e-mail address(es), and then the abstract and keywords, as show above. Capital letters must be applied in the title of a paper, and 14 pt. boldface font should be used, as in the example at the first page. Use: 11-point type for the name(s) of the author(s), 10-point type for the address(es) and the title of the abstract, 9-points type for the abstract and the key words.

For the main text, please use 11-point type and single-line spacing. We recommend using Times New Roman (TNR) fonts in its normal format. Italic type may be used to emphasize words in running text. Bold type and underlining should be avoided. With these sizes, the interline distance should be set so that some 43 lines occur on a full-text page.

The papers should show no printed page numbers; these are allocated by the Editorial Office.

Authors have the right to post their Academy of Management-copyrighted material from JACSM on their own servers without permission, provided that the server displays a prominent notice alerting readers to their obligations with respect to copyrighted material. Posted work has to be the final version as printed, include the copyright notice and all necessary citation details (vol, pp, no, …).

An example of an acceptable notice is:
"This material is presented to ensure timely dissemination of scholarly work, and its personal or educational use is permitted. Copyright and all rights therein are retained by the Copyright holder. Reprinting or re-using it in any form requires permission of the Copyright holder.

The submission format for the final version of the manuscript
and the MS Word template are available on our web site at:
**http://acsm.spoleczna.pl**

# Journal of Applied Computer Science Methods

## Contents

**Yusong Liu, Zhixun Su, Bingjie Zhang,
Xiaoling Gong, Zhaoyang Sang**
Convergence Analysis of an Improved Extreme
Learning Machine Based on Gradient Descent Method 5

**Grzegorz Sowa, Alina Marchlewska**
The Internet of Things:
Technological and Social Aspects 17

**Zbigniew Filutowicz, Krzysztof Przybyszewski,
Józef Paszkowski**
Selected Examples of Applications of New
Graphics and Animation Technologies 29

**Nagesha, Sunilkumar S. Manvi**
Resource Monitoring for Wireless
Sensor Networks Using Anfis 41

**Konrad Grzanek**
Monadic Printing Revisited 69