# Modeling the value chain with object-valued Petri Nets

JAROSLAV ZACEK
Faculty of Science, Department of Computer Science, University of Ostrava
Ostrava, CZECH REPUBLIC

FRANTISEK HUNKA
Faculty of Science, Department of Computer Science, University of Ostrava,
Ostrava, CZECH REPUBLIC

ZDENEK MELIS
Faculty of Science, Department of Computer Science, University of Ostrava,
Ostrava, CZECH REPUBLIC

Abstract: A substantial part of the economic theories is based on conversion and exchange process. This process can be described as a value chain, which can be considered as a cyclic model with complex attributes. There is a serious problem how to express resources and their conversions in a complex cyclic model during the simulation and how to identify these converted resources in every step of the simulation. This paper introduces the Objectvalued Petri (OV-PN) modification as a new formalism to create a cyclic model of the value chain. According to the modification we had to define a new path and pass of the OV-PN. We also had to determine new properties. Properties are based on the OV-PN and reflect needs of model requirements. A new formalism is verified on a common enterprise value chain.

Keywords: Value chain; Object-valued Petri nets; cycle Petri nets; simulation; model validation
Received: February 25, 2021. Revised: November 28, 2021. Accepted: December 19, 2021. Published: December 31, 2021.

## 1. Introduction

The value chain is a modelling technique to formalise and monitor the competitiveness of the business. It focuses on the flow of resources between internal business processes that are interconnected to each other. A product increases its value when it passes through a stream of production chain. That is the fundamental notion in value chain analysis [3]. The REA value chain is a network of business processes. The purpose of the network is to directly or indirectly contribute to the creation of the desired features of the final product or service, and to exchange it with other economic agents for a resource that has a greater value for the enterprise [10]. The value chain definition implies that it is important to find a suitable formalism for the simulation of the model run for practical realization. Existing theories such as state machines, Petri nets [5], or neural networks were considered while searching for a correct formalism. Value chains have specific requirements for descriptive formalism allowing their validation and simulation (according to [17]). Specific type of the value chain is supply chain [12]. State machines are not expressive enough to solve this problem. Despite the fact that neural networks are expressive enough for describing processes in the value chain there is significant complexity in simulation. Two independent neural networks must be created for the simulation of the value chain. The first network is able to validate the model and the second implements simulation steps. In both cases, the neural network must learn these properties. Therefore the process becomes time and implementation consuming. On the other hand the neural network approach is very flexible and can be used to solve multilevel problems (for example multilevel SPAM control [11]).

The Petri Net theory matches the description of the model states more closely, but its expressivity, especially for P/T Petri nets, is very limited [7]. General token is not able to capture such a complex structure, for example an object representing the resource. Therefore this article suggests using the object-valued Petri nets (OV-PN) to ensure the simulation and the validation of value chains. It also discusses some specific properties of the OV-PN and defines new properties for the value chain domain. Main advantage of using the Petri net theory is possibility to create an automatic deterministic process of the code generation [13].

### THE VALUE CHAIN AND ITS SIMULATION PROCESS

The value chain consists of two main parts: processes and links that form a chain with other processes (similar to supply chain described in [15]). These parts create the interconnected network of processes increasing the value of the resource. The value chain creates a cyclic bond that means all processes have their inputs and outputs connected together and form a full closed chain. Each process can have more than one input and more than one output. Multiple types of resources can form the input and the output. In figure 1 there are two significant examples of resource distribution. Resources *Plan* and *Money*, needed for purchasing the *Material*, enter to the *Purchase process*. Sales process produces output *Money* that enters into two another processes - *Purchase process* and *Acquisition process*.

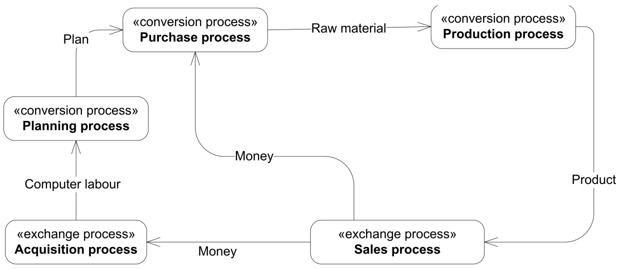


Fig. 1. Enterprise Value Chain

The simulation of the value chain process is used to monitor the competitiveness of the business. Model elements show processes that increasing the value of corporate resources. Each step of the process increases the value of the company output, and therefore it can be understood as a value chain [1]. The value chain is the set of mutually interdependent activities that are interrelated together by their inputs and outputs.

Enterprise value system shows the flow of resources between participants within the enterprise and it can be obtained by analysis of the company value chains. The value chain shows the flow of resources across business processes [14].

Figure 1 shows an example of the value chain depicting the flow of resources between the enterprise and business partners. The model example consists of 5 processes:

- *Purchase process* expresses the purchase of the raw material from vendors.
- *Production process* is internal conversion process creating the product.
- *Sales process* illustrates the sales of the created product to the customer.
- *Acquisition process* arranges labour for *Planning process*.
- *Planning process* prepares purchase plane.

For purposes of the planning and detailed analysis of business value chains it is necessary to record the flow of resources and their changes in time. For these cases, it is possible to use the simulation of the flow of resources across business processes. In every step of the simulation an exchange process performs exchanging of resources and the conversion process creates a new product or modifies characteristics of an existing product. In case of complex value chain the simulation can determine which links are inefficient and where is the place for the subsequent optimization. The simulation can be also used for analysis of the economic situation of the enterprise, such as stores status, financial estimates, an efficiency of the production, or logistics. You can also simulate the way of a product from an initial purchase of raw materials, through its production and finishing by sale to determine the total financial and time costs for one product.

# 2. Object-valued Petri Net

Object-valued Petri net is an extension of P/T Petri net. This extension has been introduced in [8]. Object-valued Petri nets are used as formalism for validation and synchronization of complex object models.

***Definition 3.1: Object-valued Petri net***

*Petri net is extended to a 6-tuple (P, T, F, V, R, C), where:*

- $P$ *is a finite nonempty set of places,*
- $N$ *is a finite nonempty set of transitions,*
- $T \cap P = \emptyset$ *(P and T are disjoint),*
- $F \subseteq (P \times T) \cup (T \times P)$ *is a finite set of arcs (flow relation),*
- *V is a finite set of object data types,*
- *R is a finite set of transforming functions* $R: P \cup T \rightarrow \psi(V)$*, where* $\psi(V)$ *is the power of the set of object data types.*
- *C is a set of capacity function.* $C: P \rightarrow N \cup \omega, N \subseteq \mathbb{N}$ *and* $\omega$ *denotes infinite.*
- $M_0: P \rightarrow V_{MS}$ *is the initial marking of the token.* $\forall p \in P: M_0(p) \in R(p)_{MS}$*, where* $R(p)_{MS}$ *is the multiple set of the object data type tokens in p.*

The main idea of the Object-valued Petri net is an object-valued token that provides adequate expressivity to describe resources represented by complex object structures. The token carries basic information to identify the specific object instance. Initial marking consists of the multiple set of object data types deployed across the net. Firing of each token means change in marking of the net and also change of the token type. However token identification remains and therefore we can identify the token in every step of the simulation process. If the model is partly linked with the Object-valued Petri net theory we have to define the path of tokens. Formalism itself defines necessary basis to create the model, unfortunatelly that does not ensures the sequence of movements into desirable result. Object-valued Petri net realizes transition as soon as the transition is feasible. Nevertheless the real model can require other conditions to realize the transition (for instance lazy constructions). Therefore we have to state the new definition of the path and pass of the model.

***Definition 3.2: Path of the OV-PN***

*Let OV-PN = (P, T, F, V, R, C) be an Object-valued Petri net with initial marking* $M_0$*. The path from the place* $u_1 \in P \cup T$ *to following place* $u_n \in P \cup T$ *is the sequence* $(u_1, u_2, \dots, u_n)$*, where* $(u_i, u_{i+1})$ *for 1 ≤fi ≤ n.*

***Definition 3.3: Pass of the OV-PN***

*Object-valued Petri net OV-PN = (P, T, F, V, R, C) with initial marking $M_0$ is feasible when:*

1. *Must exist an initial place* $i \in P$ *where* $\bullet i = \varnothing$.
2. *Must exist exactly one final place* $o \in P$ *where* $o\bullet = \varnothing$.
3. *Every place* $u \in P \cup T$ *lies on the pass between initial place **i** and final place **o**.*

In this context the first condition is understood as a marking of the input of the model that can be represented by more than one input parameter. If this condition is set to be strict to the value of input marks the model cannot realize calling of the method with more than one input parameter. The output of the model is usually one because of the standard method construction in object-oriented paradigm [2]. Third condition expresses the fact that every place and every transition exists on the path between the initial place and the final place. Therefore the Object-valued Petri net should not have blind paths and every call in the model should be reachable form initial place by passing finite number of transitions representing a flow relation *F (similarly to [16])*. Similarly to the initial place every place in the model exists in the flow relation *F* is able to reach the final place of the model.

**Boundedness and safeness**

The ordinary Petri net (P/T Petri net) defines the boudedness mechanism to limit the tokens in all reachable markings. A place in the Petri net is called *k*-bounded if it does not contain more then *k* tokens in every making in the net, including the initial marking. Moreover the Petri net is bounded if and only if its reachability graph is finite. The special case of the boundedness is safeness attribute. If the net is 1-bounded it is called *safe*.

Object model synchronized by Petri net mechanism can be bounded at the places level as in ordinary Petri net. Every method can produce more than one output during the simulation. Places may store these outputs as Object-valued tokens (similar to colour evaluation in [6]). By applying safeness rule the places in model stores only one object-valued token and the model becomes less complex.

**Conservation**

Created model cannot be strictly conservative. In the first step in figure 1 the *Purchase process* consumes two inputs and produces one output. The *Sales process* consumes only one input and produces two outputs - two object-valued tokens parameterized as *Money*. Moreover the *Purchase process* requires a synchronization mechanism. The model cannot have a constant token count for every marking from set of the reachability set

$$\Re(M_0): \sum_{p_i \in S} M(p_i) \neq \sum_{p_i \in S} M_0(p_i)$$

**Liveness and deadlock**

All methods in object-oriented paradigm can be executed more than once [4]. However by executing some method an internal state of the object can be altered. That means if we need to apply liveness property to whole model, every method must be considered as an atomistic operation.

*Generally the transition $t \in T$ is alive if:*

$$\forall p \in \bullet t: M(p) \neq \emptyset \text{ and } p \in t\bullet: M(p) = \emptyset.$$

It means that transition becomes active, if there are tokens on all transition's entrances and the place that follows the transition is empty. The net is alive if there is at least one live transition in every step of simulation process otherwise a deadlock occurs. Deadlock is solved on a higher abstraction level and requires user intervention.

# 3. Object-valued Petri Net Extensions

The main condition of the value chain is cyclicality. However the Object-valued Petri net has two definitions that limit the path of the net (definition 3.2) and pass of the net (definition 3.3). First definition says that O-V Petri net with a specific marking $M_0$ has a specific sequence from one place to another. The value chain has also specific sequence that defines the path of the chain. Moreover the cyclic chain consists of many single paths connected to each other. To express a general value chain principle with the Petri net theory we have to define a cyclic Petri net:

**Definition 4.1: Cyclic Object-valued Petri net**

*A marked Petri net (OV-PN, $M_0$) is cyclic Petri net if from every reachable marking M it is possible to return into $M_0$ (i.e.* $M \in \Re(OV-PN, M_0 \Rightarrow M_0 \in \Re(OV-PN, M)$*).*

According to [9] we must also define the inverse of an ordinary Petri net:

**Definition 4.2: The inverse of an Object-valued Petri net**

*For a Petri net OV-PN, its inverse* $\overline{OV-PN} = (P, \bar{T}, \bar{F})$ *is given by:*

- $\bar{T} = \{\bar{t} | t \in T\}$ *and*
- $\bar{F}(\bar{t}, p) = F(p, t)$ *and* $\bar{F}(p, \bar{t}) = F(t, p)$ *for every* $p \in P$ *and* $t \in T$

The definition of the inversion of the Object-valued Petri net is presented for completeness only. In the real model of the value chain there is usually no backward path. For instance the company cannot convert the product to the raw material. On the other hand this definition gives the robust tool to verify cyclicality of the net. Algorithms to verify cyclicality could be simplified to perform the token verification. Every token in the Object-valued Petri net have the unique instantiation number. The inner value of the Object-valued token is changed during the pass of the net, however instantiation number stays unchanged despite the value transformation. The modelling tool can set up the initial marking and make finite steps of the firing. If the net is cyclic the specific instantiation returns to the initial marking.

According to the facts above we can redefine the original Object-valued Petri net tuple for modelling the value chain:

**Definition 4.3: Petri net for a value chain simulation**

*Petri net for value chain simulation is a 5-tuple (P, T, F, S, R), where:*

- $P$ *is a finite nonempty set of places,*
- $T$ *is a finite nonempty set of transitions,*
- $T \cap P = \emptyset$ *(P and T are disjoint),*
- $F \subseteq (P \times T) \cup (T \times P)$ *is a finite set of arcs (flow relation),*
- *S is a finite set of resources,*
- *R is a finite set of transforming functions* $R: P \cup T \to \psi(S)$*, where* $\psi(S)$ *is the power of the set of resources.*

Moreover we must claim boundedness of elements:

- $\forall p \in P \exists t \in T\colon\ F(p,t) \in F$,
- $\forall p \in P \exists t \in T\colon\ F(t,p) \in F$,
- $\forall t \in T \exists p \in P\colon\ F(p,t) \in F$,
- $\forall t \in T \exists p \in P\colon\ F(t,p) \in F$

and their connection to cyclic model:

- $\forall p \in P\colon M(p) \in \Re(M_0), M_0 \in \Re\big(M(p)\big)$,
- *every* $t \in T$ *is reachable from any* $p \in P$ *in limited count of steps.*

Naturally we also must specify the properties of the new definition:

**Boundedness and safeness**

In the value chain the one resource can be transferred into the more than one process (i.e. money to buy a new material and money to fund innovations).

The safeness of the net is the matter of discussion. In fact there are two possibilities. The net can be safe and that means one place stores only one object-valued token. That property can be convenient to verify the whole conversion process and user can focus to one resource and transformation process. This simulation is similar to redefining business processes in the company. By applying the safeness property the whole model became simple to understand and verification of the process flow is much easier.

The second view on the value chain simulation is to get statistic data and optimize workflow parameters. The model must simulate the conversion process with more than one Object-valued token. A typical example is a production process creating the specific product. At the beginning of the simulation the company needs to know how many products must be created to cover money for a product development. In the short term the first view can set the margin of the seller and express the production process. The simulation of the second view takes longer and works with multiple tokens. The price of the product decreases with time and by the long term simulation the company can reveal if the price model has been set correctly. Therefore the second view cannot be safe form a Petri net point of view.

**Liveness and deadlock**

Object-valued Petri net must fulfill liveness property because of the object-oriented paradigm construction. Cyclic Petri nets are based on general OV-PN, but it differs on boudedness and safeness property. Therefore liveness property must be changed.

*Transition* $t \in T$ *is alive if:*

- $\forall p \in \bullet t\colon M(p) \neq \emptyset$
- $p \in t\bullet\colon |M(p)| + U(t) \leq K(p)$*, where* $U(t)$ *is a number of resources produced by transition t*

*The net is live if there is at least one live transition in every step of the simulation process.*

The value chain consists of processes and links. Process itself consists of atomic operations that can be repeated infinitely with the same result. For example: a production process is defined by precise methodology how to produce a product on the serial assembly. The parameters of the process are set at the beginning of the serial assembly (i.e. speed of the line) and usually remains unchanged during production. Therefore we naturally apply the liveness property to the Object-Valued Petri net model of the value chain. All processes remain the same despite the fact that the Object-valued token flow through the process.

The process itself can have more than one input link. In figure 1 the *Purchasing process* requires the *Money* and the *Plan*. *Plan process* takes more time to create a specific *purchasing plan* and *sales process* delivers the *Money* immediately after the product has been sold. In this specific step of the model simulation the deadlock occurs. That means the execution of the *Purchasing process* is delayed until both inputs provided with links are available. These cases can be problematic and generally can be solved on a higher abstraction level, i.e. modelling tool. If the model is validated a deadlock cannot occur because of the cyclicality property of the value chain. Moreover that implies that every process must be reachable.

**Conservation**

The conservation property means that one object-valued token cannot be duplicated when it passes the transition and the transition has the same number of inputs and outputs. In other words the count of the Object-valued tokens is same in every step of the simulation. The basic models of the value chain can be conservative. However most models in the real world are more complex and there is big challenge to apply the conservation rule to express a chain of resources. For example money in the real world is an input to more than one process - production process, planning, development, etc.. From a Petri net point of view we must duplicate tokens with specific inner attributes and sends them to other transitions. Therefore the model does not have the constant token count for every marking from the reachability set.

# 4. Value Chain Simulation Example

During the transformation the value chain elements are mapped into the modified cyclic Object-Valued Petri net elements. Similar transformation process can be found in [18]. Processes that exist in the value chain will be represented as transitions and all properties mentioned above will be applied. Links that exist in the value chain will be composed of two arcs and one place. The Arc indicates the direction of an Object-valued token and the Place carries the Object-valued token(s) that represents information.

For example we used the value chain from figure 1. The transformed result is shown in Figure 2. The Petri Net consists of five transitions and six places. The simulation starts with the Token set on the place *Product* and it takes five cycles before repeating:

1. The company has product to sell. The *Product* enters into the *Sales process*. That generates *Money* for the *Purchase process* and the *Acquisition process* (token is divided into two places).
2. *Money* enters into the *Acquisition process* and creates the *Computer work*. The *Purchase process* is not executed because of insufficient *Plan* input.
3. The *Computer work* enters into the *Planning process* and generates the *Plan*.
4. The *Purchase process* transition has all needed inputs to perform firing. *Money* and *Plan* tokens are exchanged for *Material*.
5. In the last step the *Material* enters into the *Production process* and creates a new *Product*.

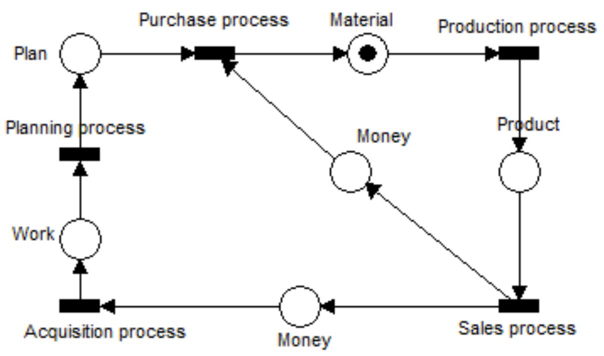


Fig. 2. Transformed value chain

The model is cyclic. That means the cycle 1 does not have to be the first and all steps are realized infinitely. The order is only that matters. The key in the value chain simulation process, except the synchronization primitives, is the Object-valued token. In the beginning of the simulation the Object-valued token is parameterized as a *Product*. In the first step the token is transformed into the *Money* and split into the two tokens with specific ratio used to determine the inner value. The association of the tokens to the value chain can be identified through the instantiation ID, and inner values can be changed as needed. Moreover in the second cycle, one of tokens enters to *Acquisition process* and transforms (parameterizes) into the *Computer work*. That means the token has different inner values and even a data structure. Analogical transformation changes the *Computer work* into the *Plan* structure in the third cycle. In the fourth cycle two tokens are merged together by *Purchase process* and the result is the Object-valued token parameterized as a *Material*. The merge process can be performed because of the same token instantiation ID. In the last step the token is transformed to the *Product* by the *Production process* and the chain is closed.

There is only one token in figure 2 and all links are limited to 1. We can simulate the whole conversion process with more than one token and we can establish a capacity function on every place in the net. All splits and merges of tokens are identified by instantiation ID and therefore they are distinguishable. That means we can recognize the specific token as a part of the cyclic chain and the base for optimization of processes in the model.

# 5. Conclusion

The paper introduced a new formalism based on the Object-Valued Petri net to create, synchronize and manage cyclic models of the value chain. The paper described a basic theory of economic models based on conversion and exchange processes and introduced a value chain term in the first part. The paper also described why current formalisms such as neural network and state machines are not suitable to build a value chain model. Paragraph 3 shows an Object-value Petri net theory focused on the path and pass of the net. This theory is suitable to build a value chain model because the Object-valued token can be used to express resources of the value chain and their transformations. However Object-valued Petri net are not cyclic and have strictly defined pass and path of the net. The definition of an extended Object-valued Petri net formalism - definition 4.3 - solves this problem and adds the cyclicality. All basic properties are discussed and redefined for the new cyclic object-oriented model. An extended Object-Valued Petri net formalism solves all problems mentioned in the second paragraph and can be applied to any cyclic model. The proposed formalism has been verified on the ordinary value chain and basic steps of the simulation are described in paragraph 4.

# Acknowledgment

The paper was supported by the grant reference no. SGS08/PRF/2013 provided by Ministry of Education, Youth and Sports.

## *References*

[1] Fiala, J., Ministr, J.: *Pruvodce analyzou a modelovanim procesu*, VB-TU Ostrava, 2003, ISBN 20-248- 0500-6.

[2] Shilling, J.: *Three Steps to Views: Extending the Object-Oriented Paradigm*, OOPSLA 89 Proceedings, 1989.

[3] Hunka, F., Zacek, J., Melis, Z., Sevcik, J.: REA *Value Chain versus Supply Chain*, Scientific Papers of the University of Pardubice, 2011, s. 68-77, ISSN 1211-555X.

[4] Zacek J., Hunka F.: *CEM: Class executing modeling*, World Conference on Information Technology, 2010, page 1597-1601, ISSN 1877-0509.

[5] Sklenar, J., Caruana, E.: *Using Timed Petri Nets in Discrete Simulation*, Proceedings of the Industrial Simulation Conference, 2004 (ISC-2004), Malaga, 2004, page 7-11.

[6] Zhao, X., Wei, C., Lin, M., Feng, X., Lan, W: *Petri Nets Hierarchical Modelling Framework of Active Products Community,* Advances in Petri

Net Theory and Applications, 2010, page 153-174, ISBN 978-953- 307-108-4.

[7] Martinik, I.: *Methodology of object-oriented programmatic system development using theory of Object Petri nets*, dissertation thesis, VB-TU Ostrava, 1999.

[8] Zacek, J., Hunka, F.: *Object model synchronization based on Petri net*, 17th International Conference on Soft Computing MENDEL 2011, Brno University of Technology, Faculty of Mechanical Engineering, 2011, s. 523-527, ISBN 978-80-214-4302-0.

[9] Bouziane, Z., Finkel, A.: *Cyclic Petri Net Reachability Sets are Semi-linear Effectively Constructible*, CONCUR'11 Proceedings of the 22nd international conference on Concurrency theory, Springer-Verlag, Berlin, ISBN: 978-3-642-23216-9.

[10] Geerts, G. L., McCarthy, W. E.: *Using Object Oriented Templates from the REA Accounting Model to Engineer Business Process and Tasks*, Paper presented at European Accounting Congress, Gratz, Austria, 1997.

[11] Sochor T., Davidova A.: Potential of Multilevel SPAM Protection in the Light of Current SPAM Trends, 10th International Conference on Networking, Sensing and Control, Paris-Evry University, France, 2013.

[12] Kersten W., Blecker T., Ringle M. Ch.: Managing the Future of Supply Chain, Eul Verlag, ISBN 978-3-8441-0180-5, 2012.

[13] Ding Z., Liu J., Wang J.: A Petri Net Based Automatic Executable Code Generation Method for Web Service Composition, Proceedings of the 2012 International Conference on Information Technology and Software Engineering, pp 39-48, ISBN 978-3-642-34530-2, 2013.

[14] van Hee M. K., Sidorova N., van der Werf M. J.: Business Process Modeling Using Petri Nets, Transactions on Petri Nets and Other Models of Concurrency VII, pp 116-161, ISBN 978-3-642-38142-3, 2013

[15] Wang W. J., Ip H. W., Muddada R. R., Huang L. J., Zhang J. W.: On Petri net implementation of proactive resilient holistic supply chain networks, The International Journal of Advanced Manufacturing Technology, ISSN 0268-3768, Springer-Verlag, 2013.

[16] Darondeau P., Demri S., Meyer R., Morvan Ch.: Petri Net Reachability Graphs: Decidability Status of First Order Properties, Logical Methods in Computer Science, Vol. 8(4:9), pp. 1–28, 2012.

[17] Bocewicz G., Wójcik R., Banaszak Z.: Cyclic Scheduling for Supply Chain Network, Trends in Practical Applications of Agents and Multiagent Systems, Springer Berlin Heidelberg, pp 39-47, 2012.

[18] Li J., Zhou M., Xianzhong D.: Reduction and Refinement by Algebraic Operations for Petri Net Transformation, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on (Volume:42 , Issue: 5), pp 1244 - 1255, ISSN 1083-4427, 2012.