academicJournals

Vol. 4(1), pp. 1-10, July, 2016
DOI : 10.5897/JECI2015.0183
Article Number: 7C47A8159582
ISSN 2141-6508

http://www.academicjournals.org/JECI

Journal of Engineering and Computer Innovations

Full Length Research Paper

Creating E-pedigree Kanban to secure customers' chain using multi-agent image technique

Ahmed M. Abed

Industrial Engineering Department, Zagazig University, Zagazig, 44519, Egypt.

Received 3 April, 2015; Accepted 8 March, 2016

Egypt is one of the countries mostly faced with the problem of China's fake spare parts used in replacing original components. The spare parts components trade is close to 2.7 billion LE out of 9.3 billion LE, that is, the volume of imported spare parts in Egypt, as announced by the official speaker of the Ministry of Trade and Industry in Egypt. Hence, it can be seen that China's components adulteration problem is threatening nationally and internationally; and it is easier due to the great expansion of supply chains. Therefore, this paper aims to deliver the original parts to customers via three-bin-system (Kanban) by using e-kanban in creating e-pedigree kanban. Many aspects are discussed to control the transportation system with minimum costs. These include e-Pedigree, Databases, RFID, 2D barcodes, multi-agent systems, and the integration of these tools to develop a system to secure the importing spare parts via VSM (Value Stream Map). This paper proposed E-PKS (Electronic Pedigree Kanban System) for track and traces any products' logistic via using tracking procedures supported by image code technique. The E-PKS are joints of VSM and consist of two phases: the first control the VSM in different agents and the second phase focuses on the development of the 2D BC of the product to guarantee right channel to customers by using the proposed image code technique.

Key words: China items adulteration, supply chain, multi-agent systems, E-Kanban, e-pedigree.

INTRODUCTION

The spread of counterfeit goods (commonly called "knockoffs") has become global in recent years and the range of goods subjected to infringement has increased significantly. According to the study of Counterfeiting Intelligence Bureau (CIB) of the International Chamber of Commerce (ICC), counterfeit goods make up 5 to 7% of World Trade; however these figures cannot be substantiated due to the secretive nature of the industry (ICC, 2007). A report by the Organization for Economic Co-operation and Development indicates that up to US$200 billion of international trade involved counterfeit and illegally copied goods in 2005 (OECD 2007). In November 2009, the OECD (Organization for Economic Co-operation and Development) updated these estimates, concluding that the share of counterfeit and illegitimate goods in world trade had increased from 1.85% in 2000 to 1.95% in 2013. That represents an increase of US$250 billion worldwide (CNET 2013).

E-mail: gembaconsultingoffice@gmail.com. Tel: +20-1022956165. Fax: +20-055-2304987.

Most counterfeit goods are produced and manufactured in China, making it the counterfeit capital of the world. In fact, the counterfeiting industry accounts for 8% of China's GDP (Gross domestic product) (Hopkins et al., 2010). This paper combats counterfeiting by creating e-pedigree Kanban system (E-PKS) for the spare parts via joining both customers and suppliers in one secure system.

A famous Kanban system implementation is a “three-bin system or three-agent system” for the supplied parts, where there is no in-house manufacturing. While one bin is on the factory floor, another is in the factory store, and yet another is at the supplier store. The bins usually have a removable card containing the product details and other relevant information (classic Kanban card). David Anderson was among the first to formulate the Kanban Method for application of IT and software development in his 2010 book *Kanban*. A notable earliest contribution came from Corey Ladas, whose 2009 book *Scrumban* (Ladas, 2009) was the first to suggest that Kanban was a superior alternative for software development. Most factories that use kanban today use Electronic kanban (E-kanban) systems (Bruce, 2008). These help to eliminate common problems such as manual entry errors and lost cards (David, 2008). E-kanban systems integrated into enterprise resource planning (ERP) systems to real- time tracking across the VSM and improved visibility. Data pulled from e-kanban system can be used to secure transportation levels by creating related serial number making a link between the part and its supplier (Cutler, 2013). The product details recorded on the E-kanban are numerical values for specific part (e.g., X_1) in specific position (e.g., X_2) for any spare part. We try to create coded serial number having some information picked from the e-kanban in different stations in VSM. These details may be surface roughness or tolerance for specific dimension in this part. These values are randomly picked from the E-kanban during manufacturing process and vary from part to part. E-kanban often uses the internet as a method of routing messages to external customers.

Organizations such as the Ford Motor Company (ISO, 2014) and Bombardier Aerospace have used E-kanban systems in tracking transportation process. Systems are now widespread from single solutions or bolt on modules to ERP systems. According to Tam (2010), root causes of adulterating China components include the opportunity to make money, with the growing number of expensive China components, which are indispensable such as all aspects of the spare parts and cosmetics products, and the advancement of computer technology. With computers, it is becoming easier to forge labels and create a packaged product that closely resembles the original; also the growing complexity and size of the VSM enables forgers to introduce their adulteration China components into the VSM (Govindu, 2007; NNE, 2014). Adulteration found its opportunity due to the absence of link between the product, serial number, suppliers, and customer.

Critical elements to combat adulteration components (Kovalchuk et al., 2008; Savic, 2013) are: New technologies, stricter licensing requirements as well as establishing close VSM cycle. The beginning and the end of the chain is the customer. Till date, no technique has been effective in eliminating the adulteration problem. Most detection procedures currently in place rely on manual product inspection by quality technical or sales representatives to check for evidence of adulteration. In the absence of automated inspection technology, these methods are often too costly to do adulteration inspection on a broad, periodic basis (Castro et al., 2007; Kwok, 2008). If positive detection of adulteration does occur, it is not clear what action to take because current methods provide incomplete information about the scope of adulteration for a particular China’s component. Anti-adulteration product technologies fall into two broad categories: covert or overt (Koh et al., 2003; Lehtonen et al., 2007; Savic, 2013). The covert concept uses security markers or tags on the components coating and packaging material. Tags are made of China components grade materials incorporated into the packaging as adhesive, coating on a label, or in the cap. Tag code is read using a micro-imaging reader and software (Jung et al., 2006), and allows authentication at the pill level. The overt concept tries to secure supply chains using visible markers like hologram and colour-shifting ink (Mitkas et al., 2008). Data are the basic building blocks of today’s information economy and knowledge-based businesses.

This paper explores how to track and trace expressive data, which originates from reading unique and creative serial number for each product at different locations in a supply chain to detect suspicious movements of products. The tools required to build a secure VSM system to combat adulterated China’s components include database, appropriate method of pedigree tracking (RFID, *2D*, other), hardware and equipment to serve the system (Castro et al., 2007). The appropriate software entities to serve the system and the appropriate contact method between several levels of the VSM (Huang et al., 2009) are developed in this paper.

The paper focuses on creating a link between supplier and customer through a secure and closed system, along a VSM, where the customer is satisfied by receiving the original product through the scratched warranty card, which illustrates the anonymous value ($x_1, x_2, x_3, x_4\ldots$). These values must be the same as in the serial number barcode of the product and have real equality in the product.

STRUCTURE OF THE VSM USING MULTI-AGENT SYSTEMS

A VSM illustrated in Figure 1 is a network of autonomous entities, engaged in procurement of raw materials, manufacturing

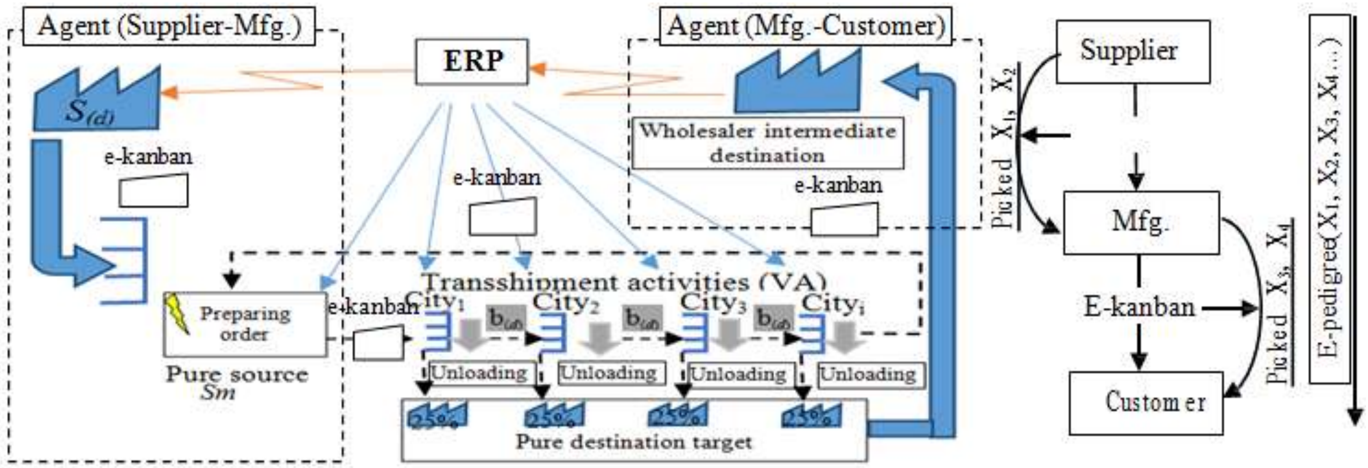

Figure 1. The VSM cycle for e-pedigree kanban.

and converting raw materials into finished products and distribution of finished products (Lu et al., 2008; Zanjirani Farahani et al., 2013). Distribution, manufacturing and purchasing organizations along the VSM often operate independently and have their own objectives, which may be in conflict (Kwok, 2008; Blackstone, 2014; ISO, 2014). The VSM failed due to counterfeiting. The traceability system delivers the right product, at the right time, competitive cost, and with customers' satisfaction (Meng and Ming, 2012). These are possible only by using e-kanban card, either between suppliers and manufacturers (Mfg.), or during manufacturing (Mfg.), or between manufacturers (Mfg.) and customers. The three e-kanban formulates product e-pedigree.

The evolution of information technology for VSM

Information is the key to the success of a VSM because it enables management to make decisions over a broad scope that crosses both functions and companies (Ritesh, 2011). The use of (E-KPS) paradigm has increased sharply as an important field of research within the Artificial Intelligence area. An agent is defined by its flexibility, which implies that an agent is:

Reactive: Agents perceive the context in which they operate and react to it appropriately.
MRP man: An agent has to try to fulfill his own objectives.
Coordinator: An agent has to be able to communicate with other agents.
Stakeholder: An agent has capabilities of task selection, prioritization, goal-directed behaviour, decision-making without human intervention.
Processor: Agents perform information processing and reasoning, based on their internal knowledge base and rules.
Communication: An agent participates in communication acts, interacting and sharing knowledge with other agents of the E-PKS.

Technologies agent seems to be the most appropriate choice to solve most supply chains problems. This is because agents have the ability to provide solutions in a characterized form by the distributed nature of data (Li et al., 2014), the complexity of the software solution, lack of centralized control. Multi-agent technology could be a favourite alternative to model and simulate the collaboration mechanisms and processes (Wang, 2014). Therefore, the combination of supply chain process definitions with an advanced infrastructure in terms of multi-agent systems has the potential to make possible a real strategic competitive advantage for the entire VSM and will enable new forms of business and work.

The goal is to create and evaluate sets of intelligent agent that can cooperatively support production and logistics network decisions. Every agent in the SCM support model owns its knowledge, interests, status information, message handlers, process element executors and policies.

E-PKS AS JOINTS OF VSM

The proposed system in this research, E-PKS (Electronic Pedigree Kanban System) includes four main agents: Supplier agent, Customer agent, Distributor agent and Super-Market agent. A local database is used in each phase, and is related to the corresponding agent; a central database usually accessed by all the agents. Figure 2 illustrates the E-PKS structure while Table 1 illustrates the role of each part in the system.

Phase 1: E-PKS agents

Here, the first part of the outlet products, that is, Super-Market agent process as a representative example is taken to see how agents work. Incoming messages are dealt with by the agent according to the event selection mechanism that is subject to some pre-determined policies such as first in first out (FIFO). All the messages are based on Knowledge Query and Manipulation Language (KQML), which is a language and protocol for exchanging information and knowledge. Then the selected messages passed to the message handlers to determine how to process the message and which process element executors are to be executed. Meanwhile, the policies, knowledge and status information are applied and retrieved. At the same time, if the outgoing messages are generated, they are sent to the distributor agents. Of course, the local performance of each agent may need to be measured and the policies may be revised according to the new performance information. It should be mentioned that the process element executors, which are derived from the process element framework, are the kernels as the agents are designed to be process- focused. This framework is illustrated in Figure 3.

A. Supplier-Mfg. agent

Verify EBC for product x, for (x=1: x=X)
Insert product x into local database
Delete product x from local database
for package y, for (y=1, y=Y)

If Valid; then
otherwise
Send message (invalid product) to Shop Floor agent Verify EBC
If Valid; then; Insert package y into local database

B. Mfg. -Customer agent

Receive message (product information, package information) from Packaging agent
Check arrival time **t**
Send message (Alarm_ Delay) to Packaging agent
Send message (Query_for_Authentication) to central database
If (true); then
otherwise
Delete products and packages from local database
Send message (ePedigree) to central database
Send message (product_information, package_information, lot_information, shipping_information) to Distributor agent
Send message (Alarm_Delay) to Finished Goods agent; Locate truck; Contact truck
then
Send message (Alarm_Adulteration) to Finished Goods agent

If ($t \geq 0.8T$); then
Verify EBC for products X and packages Y
Receive message (Authentication) from central database
Insert products and packages into local database
Send message (Alarm_Adulteration) to Packaging agent
Receive message (Order) from Distributor agent
If Receive message (Alarm_Delay) from Distributor agent; then
If Receive message (Alarm_Adulteration) from Distributor agent
Send message (Query_for_Shipping_Information) to local database

C. Distributor agent

Receive message (product_information, package_information, lot_information, shipping_information) from Finished Goods agent

Verify EBC for products X, packages Y, lots Z
Receive message (Authentication) from central database
Insert products and packages and lots into local database
Delete products and packages and lots from local database
Send message (ePedigree) to central database
Send message (product_information, package_information, shipping_information) to Super-Market agent
Send message (Alarm_Delay) to Distributor agent
If Receive message (Alarm_Adulteration) from Super-Market agent; then
Send message (Query_for_Shipping_Information) to local database
Send message (Order) to Distributor agent

Check arrival time t; If ($t \geq 0.8T$); then
Send message (Alarm_ Delay) to FinishedGoods agent

Send message (Query_for_Authentication) to central database
If (true); then otherwise
Send message (Alarm_Adulteration) to Finished Goods agent

Receive message (Order) from Super-Market agent
If Receive message (Alarm_Delay) from Super- Market agent; then
Locate truck; Contact truck
Send message (Alarm_Adulteration) to Distributor agent Paint-Shop Super-Market agent
Receive message (product_information, package_information, shipping_information) from Distributor agent

Check arrival time t ; If ($t \geq 0.8T$); then
Verify EBC for products X, packages Y
Receive message (Authentication) from central database
Insert products and packages into local database; otherwise
Send message (Alarm_Adulteration) to Distributor agent
Send message (selling_information) to local database.

Send message (Alarm_ Delay) to Distributor agent
Send message (Query_for_Authentication) to central database
If (true); then
Delete products and packages from local database
Send message (ePedigree) to central database

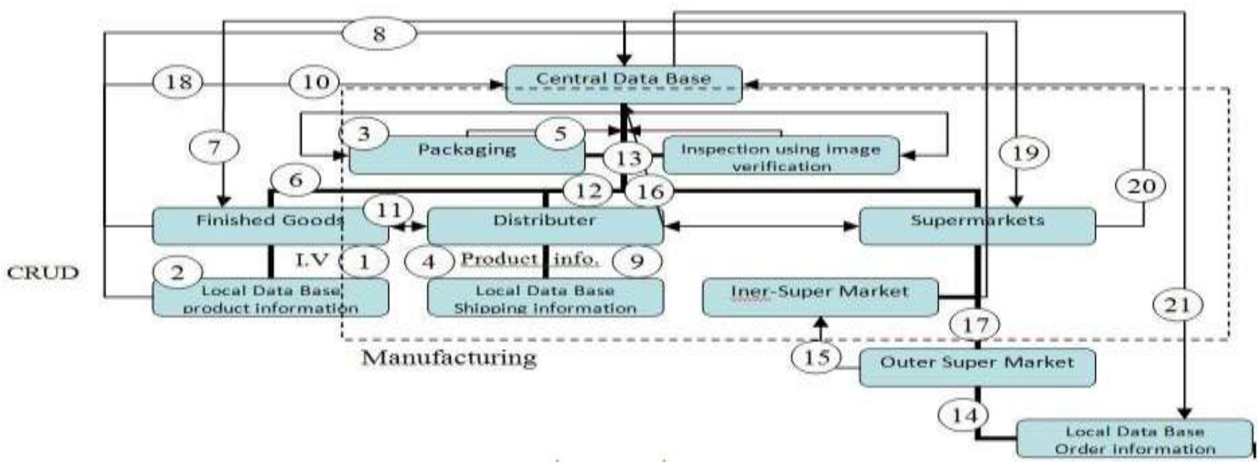

Figure 2. The E-PKS structure via VSM.

Table 1. E-PKS functions.

S/N	Action
1	Check for products and extract x_1, x_2…
2	Insert valid products and delete invalid products from local database
3	Inform the agent
4	Check for packages and send data to local database
5	**Update** the e-Pedigree at the central database (x_1)
6	Send an e-mail containing product and package information to Customer agent
7	Send Query for Authentication to central database
8	Receive Authentication approval from central database
9	Send data of valid products to local database
10	**Update** the e-Pedigree at the central database (x_2)
11	Send an e-mail containing product, package and lot of information to Distributor agent
12	Send Query for Authentication to central database
13	Receive Authentication approval from central database
14	Send data of valid products to local database
15	**Update** the e-Pedigree at the central database (x_3)
16	Receive Order from Super-Market agent
17	Send an e-mail containing product information to Super-Market agent
18	Send Query for Authentication to central database
19	Receive Authentication approval from central database
20	**Update** the e-Pedigree at the central database (x_4)
21	Send data of selling information to local database

Phase 2: Securing the customers' components packaging with image code (Encrypted Bar Code: EBC)

Agents take care to ensure the packaging of customers' components, by putting code that we recognize on the packaging. Codes related to the product are divided into parts. Their places are changed so that the carrier or user is not recognized on the image, every part in the code related to specific value, (e.g., the code consists of four variables, x_1: specific dimension, x_2: its position, x_3: supplier identification and x_4: relation between the previous variables and transforming this values to serial number). Upon reception, the goods are stored in the same image database; selected parts of the picture are ordered randomly to add information package on it. Images are matched with the database and part of the image is read. Product information is reprocessed. The production engineer and receiver of retail shop goods should not define the part of the image.

This code is constructed by C#, to scramble the shape and split the original shape into small picture randomly. The quality manager chooses a part and put the information of the package on it. The comparing image occurs with every part of the original image. The image may be barcode or any other image selected by the factory. An example is in Figure 4.

THE C# STATEMENTS

```
namespace picture_Code
{
   public partial class formPictureCode : Form
   {
      enum directions { left, right, up, down };            directions dirRememberLastMove;          int intVerticalOffset = 75;      // vertical gap
      int intSquareSize = 100;          // tile size               int intWidth, intHeight;                      int intScrambleCounter = 0;
      int intScrambleViewCounter = 0;                           bool bolScrambling = false;
   udtCartesian udrMousePos;                              udtCartesian udrHoleLoc;                                    udtCartesian[,] udrImageIndices;
   PictureBox pic = new PictureBox( );                               Bitmap[,] bmp;
   Label lblFeedback = new Label( );                      PictureBox picSolution = new PictureBox( );           bool bolStopScramble = false;
         public struct udtCartesian
         {
            public int x;
            public int y;
         }
         public formPictureCode( )
         {
      InitializeComponent();                           pic.BackColor = Color.Black;      pic.SizeMode = PictureBoxSizeMode.Normal;
      Controls.Add(picSolution);                       picSolution.Visible = false;          Controls.Add(lblFeedback);
      lblFeedback.BringToFront( );                     lblFeedback.Font = new Font("ms sans-serif", 14);
      lblFeedback.AutoSize = true;
      lblFeedback.ForeColor = Color.White;      SizeChanged += new EventHandler(formPictureCode_SizeChanged);
   string strCurrentDirectory = System.IO.Directory.GetCurrentDirectory( );
   newCode(picture_Code.Properties.Resources.nessa18, false);
         }
   void formPictureCode_SizeChanged(object sender, EventArgs e)
         {
            placeLabelFeedback( );                    pic.Top = intVerticalOffset;                       pic.Left = 5;
         }
   void placeSolution()
         {
        picSolution.Left= (Width - picSolution.Width) / 2;                    picSolution.Top=(Height- picSolution.Height) / 2;
         }
   void placeLabelFeedback( )
         {
    lblFeedback.Left=(Width- lblFeedback.Width) / 2;                        lblFeedback.Top=(Height- lblFeedback.Height) / 2;
         }
   void showFeedback(string strText)
         {
            lblFeedback.Text = strText;               placeLabelFeedback( );                lblFeedback.Visible = true;
            lblFeedback.Refresh( );                   lblFeedback.BringToFront( );
         }
   intWidth = (int)Math.Floor((double)bmpPic.Width / (double)intSquareSize);
   intHeight = (int)Math.Floor((double)bmpPic.Height / (double)intSquareSize);
   picSolution.Image =new Bitmap(bmpPic, (int)((intWidth * intSquareSize) * .24), (int)((intHeight * intSquareSize) * .24));
   picSolution.Width = picSolution.Image.Width;                          picSolution.Height = picSolution.Image.Height;
   bmp = new Bitmap[intWidth, intHeight];  // create 2d array of bitmap pointers
   pic.Width = intWidth * (1 + intSquareSize); // plus one to show a gap bet.tile
   pic.Height = intHeight * (1 + intSquareSize);
   // cycle through every tile and copy the image from the source to the appropriate tile
      for (int intX = 0; intX < intWidth; intX++)
           for (int intY = 0; intY < intHeight; intY++)
                    {
                       // calculate the area which needs to be copied
   Rectangle recSaveArea = new Rectangle(intX * intSquareSize, intY * intSquareSize, intSquareSize, intSquareSize);
                       if (!(intX == intWidth - 1 && intY == intHeight - 1)) // leave bottom right blank
                       {
   Rectangle recDestination = new Rectangle(0, 0, intSquareSize, intSquareSize);
   // create a new bitmap of the appropriate size
   bmp[intX, intY] = new Bitmap(recDestination.Width, recDestination.Height);
   // copy the cropped source image onto the new bitmap
   using (Graphics g = Graphics.FromImage(bmp[intX, intY]))
   { g.DrawImage(bmpPic, recDestination, recSource, GraphicsUnit.Pixel); }
                       }
                       else
   bmp[intX, intY] = new Bitmap(picture_Code.Properties.Resources.blank_tile, intSquareSize, intSquareSize);  // create a new bitmap
   udrImageIndices[intX, intY] = getCartesian(intX, intY);
                    }
   bool CodeSolved( )
         {
            for (int intX = 0; intX < intWidth; intX++)
               for (int intY = 0; intY < intHeight; intY++)
                  if (udrImageIndices[intX, intY].x != intX || udrImageIndices[intX, intY].y != intY)
                     return false;
            return true;
         }
```

```
void pic_Click(object sender, EventArgs e)
      {
udtCartesian udrOldHole = udrHoleLoc;
if (udrMousePos.x > 0)          { if (udrHoleLoc.x == udrMousePos.x-1 && udrHoleLoc.y == udrMousePos.y)
               movePiece(directions.left, true);
         }
         if (udrMousePos.x < intWidth - 1)
         {
if (udrHoleLoc.x == udrMousePos.x +1 && udrHoleLoc.y == udrMousePos.y)          movePiece(directions.right, true);
         }
         if (udrMousePos.y > 0)
         {
if (udrHoleLoc.x == udrMousePos.x && udrHoleLoc.y ==udrMousePos.y-1)          movePiece(directions.up, true);
         }
if (udrMousePos.y < intHeight - 1)
         {
if (udrHoleLoc.x == udrMousePos.x && udrHoleLoc.y == udrMousePos.y + 1)          movePiece(directions.down, true);
         }
      }
private void mnuLoadPicture_Click(object sender, EventArgs e)
      {
         OpenFileDialog ofd = new OpenFileDialog( );          ofd.Filter = "pics | *.bmp;*.jpg;*.gif;*.png";
         ofd.InitialDirectory = System.IO.Directory.GetCurrentDirectory( );     if (ofd.ShowDialog() == DialogResult.OK)
         {
            PictureBox pic = new PictureBox( );          bolStopScramble = true;          pic.Load(ofd.FileName);
            newCode(pic.Image, true);
         }
      }
private void mnuShowSolution_Click(object sender, EventArgs e)
      {
         picSolution.Visible = !picSolution.Visible;          picSolution.BringToFront();
      }
private void mnuFile_Scramble_Click(object sender, EventArgs e)
      {
         bolStopScramble = false;          Scramble();
      }
private void mnuFile_Exit_Click(object sender, EventArgs e)
      {
         Dispose( );
      }          }          }
public class classMessageBox : Form
{
string reply;          int intButtonLeft;     public System.Windows.Forms.Button[] btns = new System.Windows.Forms.Button[0];
public System.Windows.Forms.Label lbl = new System.Windows.Forms.Label();
public classMessageBox(string strTitle, string strText,params string[] strButtons)
   {
      Text = strTitle;
      Controls.Add(lbl);          lbl.Text = strText;          lbl.Top = 5; lbl.Left = 5;
      lbl.AutoSize = true;          foreach (string strButton in strButtons)     addButton(strButton);          resizeForm( );
   }
   void addButton(string strThisButton)
   {
      System.Windows.Forms.Button btnNew = new System.Windows.Forms.Button();
      Controls.Add(btnNew);          btnNew.Visible = true;
      btnNew.AutoSize = true; btnNew.AutoSizeMode = AutoSizeMode.GrowAndShrink;
      btnNew.Text = strThisButton.Trim( );          btnNew.Click += new EventHandler(btn_Click);
      btnNew.FontChanged += new EventHandler(btnNew_FontChanged);
      Array.Resize<System.Windows.Forms.Button>(ref btns, btns.Length + 1);          btns[btns.Length - 1] = btnNew;
   }
void btn_Click(object sender, EventArgs e)
   {
System.Windows.Forms.Button btnThis = (System.Windows.Forms.Button)sender;
      reply = btnThis.Text;          Close( );
   }

void btnNew_FontChanged(object sender, EventArgs e)          { initResize( ); }          }          }
```

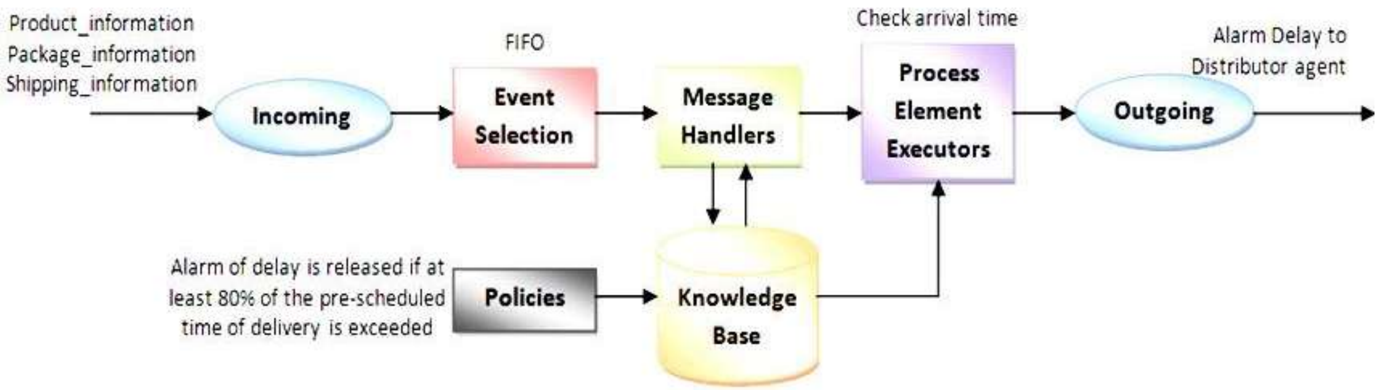

Figure 3. Agent operations.

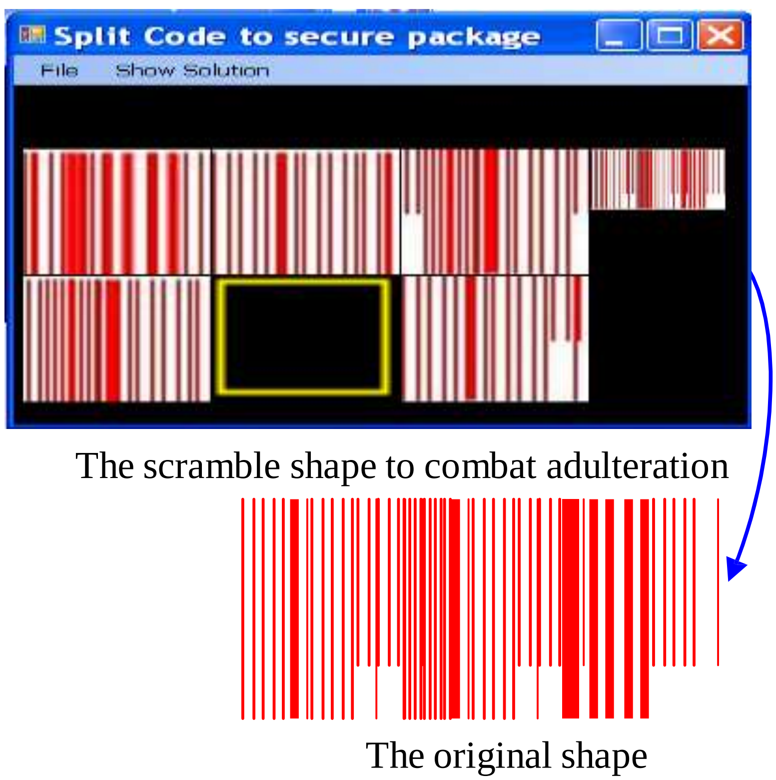

Figure 4. Secure the package with image code.

ANALYSIS OF IMPLEMENT E-KPS

The Multi-Agent System (E-PKS) approach developed in this paper combines Multi-agent systems (Fu et al., 2000; Zhang, 2014), EBC|EPC, Databases and Internet capabilities (D. Fery et al 2003) to provide an effective and efficient system to combat product adulteration and provide a reliable e-pedigree for products (Davidson and Wernstedt, 2004; Zhang, 2014). The four agents linked together, both the packaging and finished goods agents, are the manufacturers; they start the e-Pedigree of products and are responsible for the product until it reaches the distributors on time. The distributor agent ensures that products are received on time and checks their authentication information by decryption EBC. The super-market agent sends orders to the distributor agent, ensures that products are received on time, and checks its authentication. The four agents connect through the central database, which provides the full e-pedigree kanban of the products.

Thus, both track and trace, and product verification are feasible through the implementation of the E-PKS approach. E-PKS assumes that all product manufacturers, carriers, wholesalers and outlet products have the necessary hardware and computing ability to read and process EPC™ information. The E-PKS approach would have to be fine-tuned in terms of information synchronization among many different VSM partners to ensure a high level of reliability for pedigree and product verification information. If a single VSM partner does not properly handle information, pedigrees might show gaps that would raise adulteration questions. The E-PKS approach assumes different entities within the VSM to achieve a common level of cooperation in supporting the information infrastructure. Besides proposed applications in improving track and trace, and product verification, the E-PKS approach also serves as the foundation for future applications of importance to the health care industry. The E-PKS approach is a natural fit to certain classes of dynamic VSM problems because the paradigm focuses on coordinating the activities of loosely coupled distributed entities, e.g., manufacturers, distribution agents, and Super-Markets (where an agent represents each of these). One goal of the paradigm is to enable agents to meet deadlines and resource constraints; it is also flexible, robust, responsive, and adaptive.

Conclusions

The VSM formation is an important problem in the commercial world and can be improved using multi-agent systems. Products adulteration is a great problem and a great challenge for China's components industry. In this work, the use of reliable track and trace technology provides the required security and authenticity. This would help secure the integrity of the VSM, "e-pedigree" and a record documenting via E-kanban for the product manufactured and distributed under secure conditions. This paper advocated for the implementation of electronic track and traces mechanisms and noted that EBC is the most promising technology revamping image verification to meet this need. Implementation of EBC will allow customer chain stakeholders to track the chain of pedigree of every package of product. By tying each discrete product unit to a unique electronic serial number, a product is tracked electronically through every step of the supply chain. Over the last year, stakeholders have made tremendous progress in the development and implementation of EPC/EBC. This endeavor requires close collaboration among all distribution systems. The proposed technique introduces novel approaches and technical methods.

The E-PKS system follows new conceptual models that resonate with the latest advances in Informational Society, such as web services and multi-agent technologies for implementation of complex distributed systems (e.g. functions of communication management). EBC technologies are employed to identify products. The proposed E- PKS ensures confidentiality, integrity and security of data. Hence it is suggested to introduce a EPC to be stored in the e-kanban cards. Most of these techniques are relatively new, and can be used to cope with the supply chains strategies in developed countries. E-PKS is systemic, instance, proactive, dynamic, global, flexible, technology- based, comprehensive, and responsive.

RECOMMENDATION AND FUTURE WORKS

The tracking and tracing network along with generic technology behind the tracking and tracing network needs to be discussed. In this context, available tracking and tracing software was presented and their operational activities are explained briefly. All these software tools support real-time tracking technologies required for online tracking information across logistics chains. In this paper, we have also presented the methodology to support the collaborative logistics tracking and tracing via multi-agents, where the networks partners could track their shipments easily and efficiency on real-time basis. The researcher recommends transmitting the messages through automatically EDI or XML message interface to the partners' internal receivers. Considering the implementation structure of the tracking technology, future research will continue to describe practical business cases in detail and assess the sophistication of the technology applied.

Conflict of Interests

The authors have not declared any conflict of interests.

REFERENCES

Anderson D (2010). Kanban - Successful Evolutionary Change for your Technology Business. Blue Hole Press. ISBN 0-9845214- 0-2.

Blackstone EA, Fuhr JP, Pociask S (2014). The Health and Economic Effects of Counterfeit Drugs. Retrieved from Am. Health Drug 7(4).

Castro L, Wamba SF (2007). An inside look at "RFID technology". J. Technol. Manag. Innov. 2(1):128-141.

Cutler TR (2013). Examining Lean Manufacturing Promise. SoftwareMag.com. Retrieved January 29, 2013.

Davidsson P, Wernstedt F (2004). A framework for evaluation of multi-agent system approaches to logistics network management. Multi-Agent Systems. An Application Science, Kluwer, 2004.

Fery D, Stockheim T, Woelk PO, Zimmermann R (2003). Integrated Multi-agent-based Supply Chain Management. In Proceedings of 1st International Workshop on Agent-based Computing for Enterprise Collaboration.

Hopkins D, Lewis K, Mark T (2010). Counterfeiting Exposed: How to Protect Your Brand and Market Share (Wiley and Sons 2003) "Intellectual Property Rights" (PDF). Retrieved 15 May 2010.

Huang D, Verma M, Ramachandran A, Zhou Z (2009). EPCA Distributed e-Pedigree Architecture. Accepted by the 11th International Workshop on Future Trends of Distributed Computing Systems (FTDCS).

ICC (Counterfeiting Intelligence Bureau) (1997). Countering Counterfeiting: A Guide to Protecting and Enforcing Intellectual Property Rights, United Kingdom.

ISO (2014). Crackdown on counterfeiting". International Organization for Standardization (ISO), Jan. 8, 2014. Cited in: https://en.wikipedia.org/wiki/Counterfeit_consumer_goods

Jung EH, Cho SY (2006). A Robust Digital Watermarking System Adopting 2D Barcode against Digital Piracy on P2P Network. Int. J. Comp. Sci. Netw. Secur. 6(10):263-268.

Koh HL, Hou P, Zou P, Low MY, Chan E (2006). Structural identification of a new acetildenafil analogue from pre–mixed bulk powder intended as a dietary supplement. Food Addit. Contam. 23(9):870-875.

Ladas C (2009). Scrumban: Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press. ISBN 978-0578002149.

Lehtonen M, Al-Kassab J, Michahelles F, Kasten O (2007). Anti-counterfeiting Business Case Report. Available at www.bridge-project.eu

Li ZK, Duan ZS, Lewis FL (2014). Distributed robust consensus control of multi-agent systems with heterogeneous matching uncertainties. Automatica 50:883-889.

Lu L, Wang G (2008). A study on multi-agent supply chain framework based on network economy. Comput. Ind. Eng. 54:288-300.

Meng B, Ming YD (2012). Research on the Lean Process Reengineering Based on Value Stream Mapping. Manag. Sci. Eng. 6(2):103-106.

NNE (2014). Pharmaplan. Serialization: “Get on the Right Track” Retrieved from http://www.nnepharmaplan.com/Global/Downloads/Newspapers/ebook/Serialization_Newspaper_Ebook/index.html#/4/

OECD (2007). The Economic Effect of Counterfeiting and Piracy, Executive Summary" (PDF). OECD, Paris. 2007. Retrieved 2007.

OECD (2009). Magnitude of counterfeiting and piracy of tangible products – November 2009 update" (PDF). OECD, Paris. 2009. Retrieved 2010.

Savic J (2013). Drug Serialization. Retrieved from Pharma Compliance Monitor.

CNET (2013). Shanzhai JI -All you need to know about fake phones - Mobile Phones". CNET. Retrieved 5 August 2013.

WHO H (2005). Inside a counterfeiting Investigation and World Health Organization (WHO 2005). 16-July-2012. www.ebnonline.com/author.asp?section_id=1084&doc_id=247423

Tam YK (2010). Measuring the true costs of counterfeiting and piracy, Assistant Commissioner (intelligence and investigation) Hong Kong Customs 2010. Cited in http://slideplayer.com/slide/698490/

Wang J, Chen K, Ma Q (2014) Adaptive Leader-Following Consensus of Multi-Agent Systems with Unknown Nonlinear Dynamics. Entropy 16:5020-5031.