Making Product Customization More Feasible for Flexible Manufacturing Systems

Hoejin Kim* and David E. Culler

Department of Manufacturing and Mechanical Engineering Technology
Oregon Institute of Technology
Klamath Falls, OR, 97601, USA

ABSTRACT

A big challenge for companies is to make the steps between product design and process configuration more agile and adaptable. Some potential areas of performance improvement include; connecting product design to process design and automating the set-up of manufacturing equipment for material handling, machine tools, bar-code readers and vision stations. Progress in these areas will allow for customer requirements to be utilized efficiently in developing instructions for a Flexible Manufacturing System (FMS). The result would be a faster turnaround for new products and changes to customer roles in developing the products. One option is to streamline the creation of routing sheets, work instructions and programs that contain all product information and the strategies for making and inspecting the product. A key to this effort are the Application Programmable Interface (API) tools available in software throughout the product development and manufacturing cycle. If a library of programs can be created and made available for each station in the cell for different product configurations, the API could select and automatically load these programs including data (e.g. tooling, jigs, and fixtures) and make the cell very easy to set-up. A company could increase customer involvement in product development without a huge cost associated with re-configuring the automated system.

1. INTRODUCTION

Industrial manufacturing is globally adjusting to rapid changes in customer requirements and product customization. A company's degree of competitiveness can be seen as a function of its ability to react quickly to these changes and reconfigure or reprogram processes to handle these varying product requirements. One place that this concept is being experienced is when customers participate in designing their own product by using visual or web based interfaces to specify options or draw their own design for example. Using this method, companies are promptly able to establish and attempt to meet customer preferences. At the same time, engineers are developing and improving software tools such as Computer Aided Design (CAD)/Computer Aided Manufacturing (CAM) for designing, engineering and manufacturing quality customized products economically.

However, the problems arise in tasks such as manufacturing process design, production planning, and programming the machines and devices on the production line. These jobs have not been streamlined or automated to allow rapid changeover from one product to another. Manual and hands on skills such as converting data or reprogramming still are required by engineers working in these fields so that the manufacturing process is not responding quickly to customers' varying needs. Devices, such as robots, Programmable Logic Controller (PLC), Computer Numerical Control (CNC) machines, cameras and bar code readers take a lot of time to set-up, configure and program. If product data could be converted to manufacturing process instructions in a more automated way, then processes could be streamlined and customers, designers, and marketing people could interact more directly in the process from design to production. This way, the manufacturing process could be modified quickly according to changes in the customer's specification.

First and foremost, more efficient management of product data will be necessary to address these issues. Currently, CAD/CAM software and work instructions determined by manufacturing engineers are used to determine the correct routing of the product and what must happen at each station that the product passes through on its way to completion. However, if the initial product data could be efficiently converted into programs and setup values for each machine in the production line and transmitted to the communications hub or flexible cell controller via Application Program Interface (API) utilities, then automated or flexible equipment can adapt to new customer designs orders.

---

* Corresponding author: Tel.: (541) 581-5761; E-mail: hoejin.kim@oit.edu

To demonstrate the concept explained above, a set of API utilities are developed so that a centralized cell controller (PC) can configure and prepare an inspection station for a new product using images produced from the solid model in a CAD system. The cell controller would receive data from the CAD program and then transfer it to the machine in a production line as converted data which software in the machine (using its own API) is able to read and load into the program. This would help so that reprogramming or re-setup by engineers is unnecessary. Similar utilities could be used so that the correct robot program, CNC file, Bar-Code library or PLC program could be loaded and run automatically. This is a developmental system to research the possibilities made possible by improved API functionality in software across the board from design to planning to production to inspection. Although Flexible Manufacturing Systems and centralized databases exist in industry, the use of didactical cells can be used for researching ideas and developing the kinds of tools presented in this paper.

2. LITERATURE REVIEW

Flexible manufacturing is rapidly changing to catch up with industry needs and the preferences of more demanding customers. One of the areas which engineers can improve in Flexible Manufacturing Systems (FMS) is to shorten and streamline the work occurring during process planning, production planning, and programming of the production line. This field currently has many limitations, but improvements are being made on a continuous basis [1].

One of the approaches proposed in the literature is that machine components, machines, cells, or material handling units can be added, removed, modified, or interchanged as needed to respond quickly to changing requirements [2][3]. Such a fully automated manufacturing system does not yet exist today. However, it is the subject of major research efforts around the world, with special emphasis on the hardware and software aspects [1]. Also, several flexibilities of manufacturing systems are required for this challenge; machine flexibility for various operations performed without set-up change, process flexibility for set of part types that can be produced without major set-up change, routing flexibility for number of feasible routes of all part types, control program flexibility for the ability of a system to run virtually uninterrupted due to the availability of intelligent machines and system control software [4].

For process flexibility, set-up changes in the production process need to be streamlined to speed up manufacturing systems. One of the innovative approaches includes the utilization of Information Technology (IT) that can help reprogramming and re-setting machines reconfigure tooling, coolant, fixtures/fixturing to be used to machine any particular component, or can recognize geometrical shapes in downloaded customer CAD files and automatically generate CNC part programs that will include appropriate speed, feed rates, tools/fixtures selections, and process/operation sequencing [5].

Vogel-Heuser mentioned that new concepts from the IT domain need introduction to the automation industry because the IT is becoming more and more important for fully automated manufacturing [6]. A standard software architecture featuring a modular construction and component layout known from other domains would enable a fast and inexpensive reconfiguration of production lines. The suggested approach is based on the concept of models @ run-time. In this concept, models are not only used during design time but also during run-time. The system can change the models to reflect changes at modeling level. This is similar to our approach, but limited to product modeling and having one station in the production line. Our approach is not restricted to a specific station but proposes a cell controller managing product information for use in cell reconfiguration, and converting and distributing re-setup program into machines at each station [7].

The approach described by Naumann introduces the concept of capability descriptions to automatically integrate new devices. Unlike our approach, they only use the approach for robot cells and do not extend it to include the rest of the manufacturing systems [8]. Besides, they focus more on reducing the programming effort, whereas our approach aims at reducing configuration time and increasing integration of production systems.

The literature discussed above cites examples of similar works which have been completed or are currently in progress and explains different approaches that help to understand and reinforce why improvements of current manufacturing systems are needed.

3. DEVELOPING DIDACTICAL MANUFACTURING CELL

To explore possible solutions to the product customization problem, a small automated cell was developed using components from different manufacturers. The didactical manufacturing cell is a scaled down version of real systems that are widely used in industry. This cell consists of a vision system, robotic arm, milling machine, c-more control panel, Variable Frequency Drive (VFD), PLC, and a barcode reader. The conveyor belt, operated by a VFD through the PLC program, carries products to different independent pieces of equipment such as a camera, a milling machine, and a robotic arm the PLC can deliver a part through the entire process. The camera and the RoboRealm® software for the vision system are used for the image acquisition, recognition, and analysis. For instance, when the product passes in front of the camera, switches on the conveyor belt would be on and the vision system would capture, recognize shapes, and analyze it by matching it to a reference image as soon as the product has stopped. If it has defects or appears different than the reference image, then the vision system sends information to the PLC which can then send signals to the robotic arm and milling machine. The robotic arm can then pick up the product and put it back in the milling machine for rework or place in a reject pile (Figure 1). The barcode scanner reads the bar code on top of the product and sends information to the computer to provide part information to the vision system and allow for faster image analysis. The C-more panel connects to the PLC through an Ethernet cable and provides a Human Machine Interface (HMI) that provides control of the PLC through a touch screen. The cell controller computer is in charge of data management and the communication interfaces.

Table 1. Machines & softwares used for FMS.

Name	Type/Company	Use for	Programming language	API
Milling Machine	eXpertMill/Intelitek	Milling materials	CNCBASE	✓
Robot Arm	SCORBOT-ER 4u/Intelitek	Distributing materials	SCORBASE	✓
Barcode Reader	Voyager 9540/Metrologic Inc.	Scanning barcode	ABM Commander	✓
C-more panel	EA7-T12C/AutomationDirect	Touch panel as HMI	EA-PGNSW	✓
Camera	DFK 21BUCO3/ImagingSource	Image aquisition		
PLC	DL205/Automation direct	Operating conveyer belt	DirectSoft5 program	✓
Feature CAM	Feature CAM2010/Delcam	Computer-aided mfg		✓
AutoCAD	AutoCAD2012/Autodesk	2D/3D computer design		✓
RoboRealm	Academic version/RoboRealm	Image processing		✓

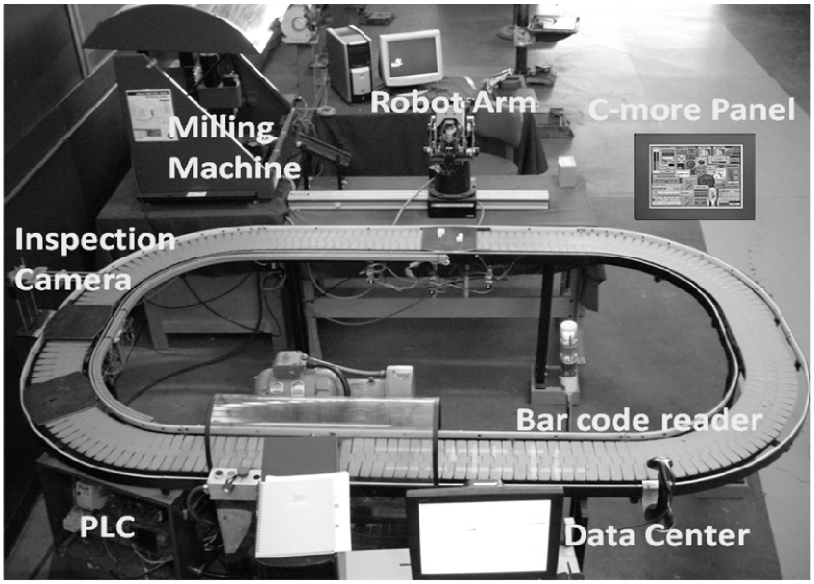


Figure 1. Layout of the didactical manufacturing cell.

4. STREAMLINING INFORMATION FLOW USING API UTILITIES

One problem in manufacturing industry is that a product specification might change over time due to feature enhancement or customer's preference. This slows and complicates automated manufacturing systems because it is time consuming to set up machines and reconfigure cells to make process flow seamlessly. One solution might be to connect CAD/CAM/Planning/Programming to a cell controller and distribute information to each station/machine that will be utilized. The cell controller would receive product data information from CAD/CAM process and convert it into specific languages which are recognizable to each station's software program. The data can be converted, transmitted and loaded into the program, so that the machines and cells in the production line are able to change working procedures quickly by reprogramming for the next product. This way the cell can be ready for the next job quickly as soon as the new product information is updated (Figure 2).

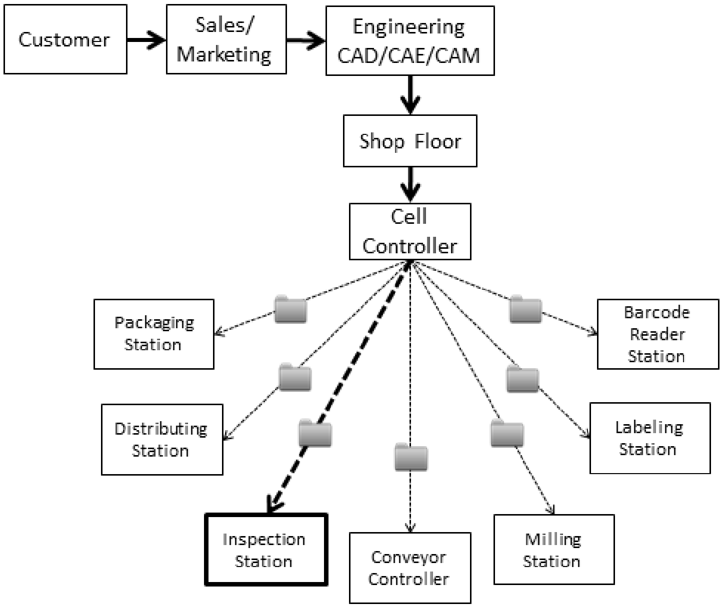


Figure 2. Layout of product data distribution and role of Cell Controller.

A GUI developed in this API has four sections representing the four steps of processing new orders. The buttons at the top of each section would be pressed to deliver the contents from each stage to the next stage (Figure 3). This data could be converted into different types of languages as information files, and then sent to each station by clicking the last button called 'Manufacturing Factory'. Another example would create routing sheets and send them to the conveyor controller (i.e. PLC) for reconfiguring the steps around the conveyor (Figure 4).

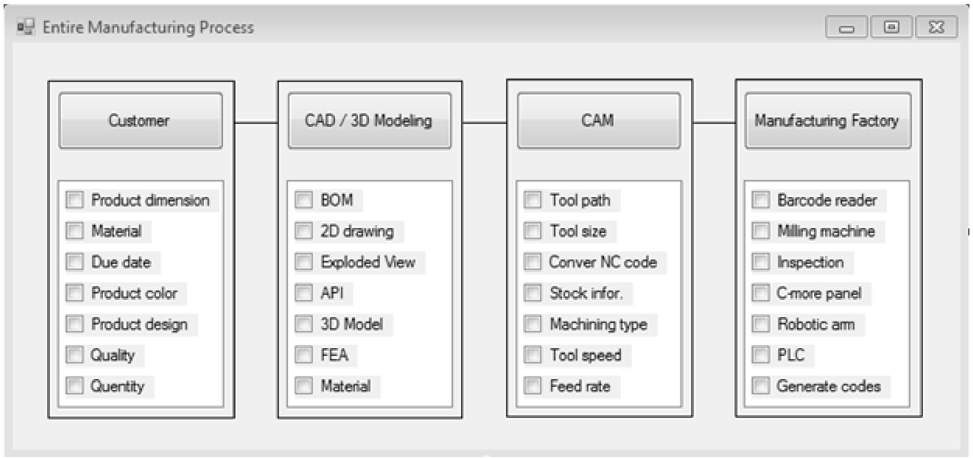


Figure 3. GUI for Data Management using API Utilities.

Table 2. Flow of Product Data Information from Origin to Destination Using API.

Data Origin	Content of data package	Interface	Data Destination
CAD	Inspection Specification - CAD Images, Known Defects	RoboRealm	Vision Camera
CAM	Routing Sheet - I/O Configuration, Sensors, Feedback	PLC	Conveyor Systems
CAD	Position & Orientation, Coordinates, Program, Simulation-offline	Scorebot	Robot Arm
CAM	Part ID, Quantity, Material, Customer info.	Zebra API	Barcode Reader
PLANNING	Cell Configuration, Station Configuration, Monitoring, Performance, Display/Control	C-more	C-more HMI Panel

In this way, the cell controllers' ability to receive and send data can shorten the time of production planning and preparation of the entire cell to handle different products. In addition, it allows the entire product development cycle to be streamlined and data to be shared efficiently. The idea is to utilize API languages along the way and make the time to production quicker, keeping up with customers' varying demands and preferences.

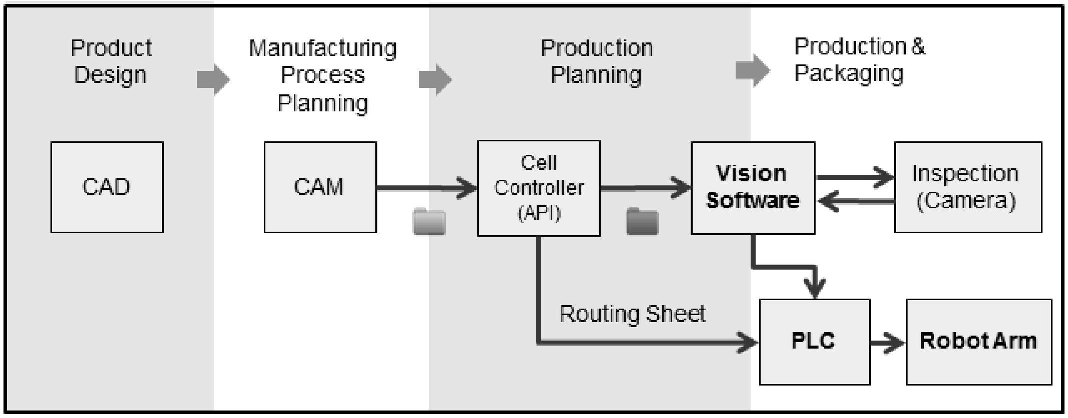


Figure 4. Product information flow on inspection.

5. APPLICATIONS BETWEEN CELL CONTROLLER AND INSPECTION STATION

To demonstrate one area of the concept described in the previous sections related to automatic configuration and programming of the cell equipment using API's, the part inspection station operation will be described. Vision systems are widespread and have been utilized in manufacturing industry to increase productivity. RoboRealm® vision software is a widely used and basic software package for use in computer vision, image analysis, and robotic vision systems. Using an easy 'point and click' interface, RoboRealm® drastically simplifies vision programming. Image or video processing can be technically difficult. RoboRealm® provides the software needed to get such a system up and running. It has compiled many image processing functions into an easy to use windows based application that operators can use with any digital image.

To explain the steps, RoboRealm® acquires and processes an image, analyzes it, and sends the results to your computer, robot, HMI, etc. It also has API utilities for automating much of the functionality from C, C++, C#, and Visual Basic, also making it possible to communicate with other devices. There are three types of shapes representing different products as examples (Figure 5). There are a circle, a square, and a triangle. Each shape has some small circles inside representing part features. If there are an incorrect number of circles or the circles are an incorrect size, the product is regarded as defective (Figure 6). If images of a new product were exported from the CAD system, RoboRealm®'s API would allow us to pre-load those images from CAD and use them as the library of images to compare the manufactured part to the designed part as well as images of the part with known defects or inaccuracies (i.e. shape, orientation, # holes or size).

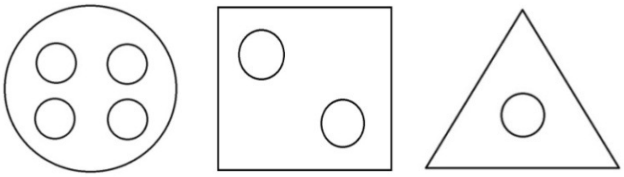
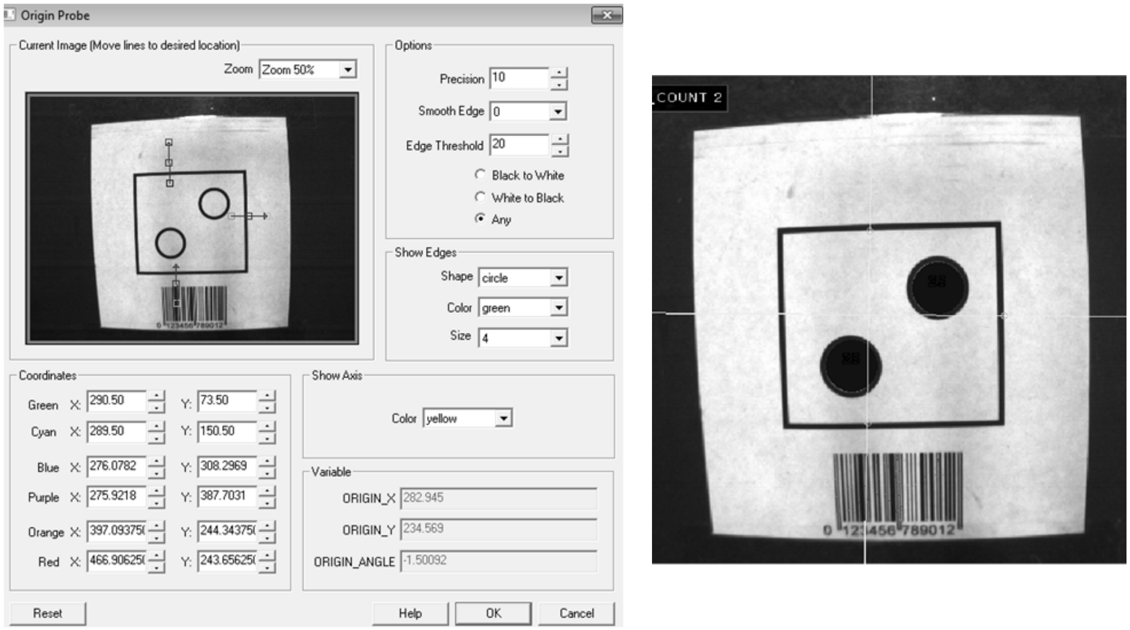


Figure 5. Examples of Shapes for Use in RoboRealm$^{®}$ program.



Zoom: Zoom 50%

Precision: 10
Smooth Edge: 0
Edge Threshold: 20
○ Black to White
○ White to Black
● Any

Shape: circle
Color: green
Size: 4

Coordinates

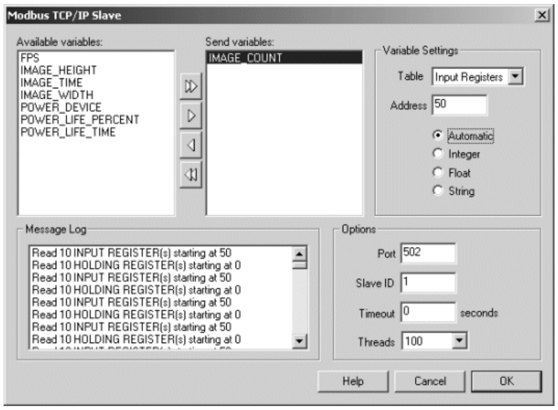
	X:	Y:
Green	290.50	73.50
Cyan	289.50	150.50
Blue	276.0782	308.2969
Purple	275.9218	387.7031
Orange	397.09375(	244.34375(
Red	466.90625(	243.65625(

Color: yellow

ORIGIN_X: 282.945
ORIGIN_Y: 234.569
ORIGIN_ANGLE: -1.50092

Figure 6. Screen showing results of image analysis.

The API in RoboRealm$^{®}$ has a built-in-function, called 'Modbus TCP/IP Slave'. It acts as a slave and requires PLC information such as address and port in order to send a data pack to the PLC acting as a client (Figure 7). For instance, RoboRealm$^{®}$ counts the number of small circles inside and sends it to PLC which could be used to re-route the part automatically, generating a new ladder logic program reflecting the re-work that must take place. However, another API in PLC is necessary to be able to create a new ladder logic program in order to reconfigure a new product on the conveyor. PLC doesn’t have API function to generate its own program co-working with other software, but it requires a sub-API program to generate ladder logic codes automatically according to new product data, receiving from the routing sheet. This problem could require further research to develop this paper.



Available variables:
FPS
IMAGE_HEIGHT
IMAGE_TIME
IMAGE_WIDTH
POWER_DEVICE
POWER_LIFE_PERCENT
POWER_LIFE_TIME

Send variables:
IMAGE_COUNT

Table: Input Registers
Address: 50
● Automatic
○ Integer
○ Float
○ String

```
Read 10 INPUT REGISTER(s) starting at 50
Read 10 HOLDING REGISTER(s) starting at 0
Read 10 INPUT REGISTER(s) starting at 50
Read 10 HOLDING REGISTER(s) starting at 0
Read 10 INPUT REGISTER(s) starting at 50
Read 10 HOLDING REGISTER(s) starting at 0
Read 10 INPUT REGISTER(s) starting at 50
Read 10 HOLDING REGISTER(s) starting at 0
```

Port: 502
Slave ID: 1
Timeout: 0 seconds
Threads: 100

Figure 7. GUI of Modbus TCP/IP Slave.

To solve this problem without the sub-API program, three ladder logic codes are already generated in one program. It will select one of three routes where the product will follow according to the number of small circles received from RoboRealm® (Figures 8).

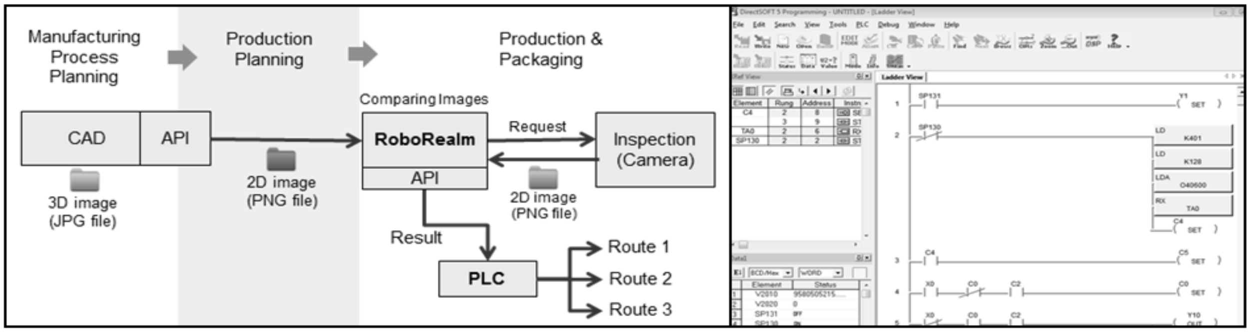


Figure 8. Product data flow for application and resulting PLC ladder logic program.

6. DISCUSSION AND CONCLUSIONS

In this paper, we proposed the cell controller connected to all machine to enable the product data information to flow seamlessly from customers to the production line by utilizing the API utilities. This approach helps minimizing obstacles such as converting product data and reprogramming occurring between the manufacturing process planning and the production. We presented how the product information data flow without these obstacles by developing the cell controller (API utilities). Using this method, customers as well as designers and marketers are able to participate in and control the manufacturing process indirectly. Also, they are able to easily change their specification and preferences because the manual efforts and all steps required for reconfiguring the system and the product data can be reduced while the manufacturing process.

More works and ideas are needed to continue developing this research and put it into practice. In order to have the cell controller and the API of each machine communicate with one another without errors, the good cell controller needs to be developed by the computer programmer to cover all different kinds of machines in production line. Also, physical problems can arise. For example, if the robotic arm doesn't have all types of grippers within reach, then physical restraints can hinder API workloads. This could be solved in a way to invent innovative grippers with adopting vision technology or develop the API in the robotic arm software.

This research utilizes materials, machines and software programs that are accessible to other university students. Although much more sophisticated and expensive systems exist in industry, many of the ideas for automating the configuration and operation of the stations that make up a CIM cell can be studied and improved by using this basic equipment. This project is representative of a larger body of work that will require the combined efforts of electrical engineers, mechanical engineers, manufacturing engineers and software developers to move forward and find innovative and effective solutions. It is important to recognize that as technology continues to grow and more students are exposed to the challenges facing automated manufacturing, innovative ideas and creative solutions will help to make the idea of product customization and its impact on engineering and manufacturing more of a topic of discussion among a diverse group of experts from different fields.

ACKNOWLEDGEMENTS

The author gratefully acknowledges support and the innovative ideas of Dr. David E. Culler and MMET department in Oregon Institute of Technology, and prayer of One-way church and the DPM family.

REFERENCES

[1] Hoda A. ElMaraghy: "Flexible and reconfigurable manufacturing systems paradigms", Springer Science + Business Media, pp.261–276, 2006.

[2] Koren Y, Heisel U, Jovane F, Moriwaki T, Pritschow G, Ulsoy G, Van Brussel H: "Reconfigurable Manufacturing Systems", CIRP, Vol.48, No.2, pp.527–540, 1999.

[3] Fujii S, Morita H: “Highly productive and reconfigurable manufacturing system”, Proc Pacific Conf Manuf, Vol.2, pp.970–980, 2000.

[4] Sethi AK, Sethi SP: “Flexibility in manufacturing”, Int J Flexible Manuf Syst, Vol.2, pp.289–328, 1990.

[5] Stecke KE: “Flexibility is the future of reconfigurability”, 3rd Conference on Reconfigurable Manufacturing, Paradigms of Manufacturing-A Panel Discussion, Michigan, USA, 2005.

[6] B. Vogel-Heuser, G. Kegel: “Global Information Architecture for Industrial Automation”, atp, 2009

[7] Nadine Keddis, Gerd Kainz, Christian Buckl, Alois Knoll: "Towards Adaptable Manufacturing Systems", 2013 IEEE International Conference on Robotics and Automation, pp. 1410–1415, 2013.

[8] M. Naumann, K. Wegener: “Control Architecture for Robot Cells to Enable Plug’n’Produce”, IEEE International Conference on Robotics and Automation, pp.287–292, 2007.