

Implementation of Context-Aware Based Robot Control System for Automatic Postal Logistics

Moon Sun SHIN¹, Myeong Cheol KO¹, Yong Wan JU², Young Jin JUNG³, Bum Ju LEE⁴

¹ Department of Computer Engineering, Konkuk University,
Danwol-dong, Chungju-si, Chungbuk, 380-701, Korea,
msshin@kku.ac.kr, cheol@kku.ac.kr

² Division of Industry Development, Korea Internet & Security Agency,
Seoul, 138-950, Korea,
ywj@kisa.or.kr

³ Department of Cyber-infrastructure Development, KISTI,
Yuseong-gu, Daejeon, 305-811, Korea,
yjjung@kisti.or.kr

⁴ Division of Constitutional Medicine Research, Korea Institute of Oriental Medicine,
Yuseong-gu, Daejeon, 305-811, Korea,
jupiter-lee@hanmail.net

Abstract: These days, the size of the Internet market, the use of parcels increases every year. Currently, in local delivery centers, the sorting of parcels has been processed manually. The manual process requires much time and high cost. Also it causes occupational injury and musculoskeletal diseases. Therefore it is necessary that the current method should be replaced with automatic smart robot manipulators. In this paper, we propose a robot control system with a context aware mechanism for generating information to control the movement of a robot, which automatically sorts parcels. The context aware mechanism generates intelligent information to control a robot using context data such as the parcel shape, weight, location and barcodes. Our robot control system collects context data of the parcel and carries out the reasoning of robot control data to pick up and drop parcels according to contexts. It also generates optimal delivery path using GIS engine. Then a robot manipulator, which receives control data of picking-up and dropping, processes the automated sorting of parcels according to delivery persons and delivery routes. It will be useful to save much time and cost by automating the parcel delivery sorting process, thus reducing frequency of industrial accidents.

Keywords: Robot Manipulator, Context Aware System, Robot Control System, Delivery Logistics, Parcel Sorting

1. Introduction

These days, the number of people using e-commerce and the mobile Internet increases rapidly. Therefore, the quantity of delivery logistics increases accordingly every year. The parcel sorting operation in the current logistics industry is carried out mechanically using an automated sorter installed at a large distribution center [1]. However, the sorting process, which classifies parcels according to delivery orders, is manually done by deliverers working at local delivery centers such as local post offices. The parcel delivery sorting operation consists of two steps. This first step is the initial sorting, where arrived parcels are sorted according to deliverers. The second step involves sorting the parcels according to the delivery order. The two steps mean that the parcels are sorted within each delivery person's assigned area. Due to the volume and weight of parcels, the issues of excessive labor and industrial accidents can arise.

A robot manipulator can be applied to those working conditions to automate parcel sorting.

Robot manipulators are widely used to efficiently conduct products distribution in many other industries such as automobile and electronic industries [4]. Generally a robot manipulator only conducts the same operation repeatedly [2]. It usually operates a process, which is a pre-programmed process according to standard size of products [3]. However, we cannot use the existing robot manipulator for automated parcel sorting because parcels have various shapes and different sizes. The use of context is important in interactive applications [5].

A new robot manipulator, which is able to recognize information concerning the different size and weight of parcels, is needed in the delivery logistics industry [9].

We proposed context aware model to automate parcel sorting in delivery logistics in our previous work [9]. Here, we design and implement a robot control system, which creates 'pickup and drop' control information in order to control the robot operation for each parcel and figures out optimal shortest path for delivery. After it receives parcel information such as the shapes of parcels or transport

information from the sensors, it generates control information and sends it to a robot manipulator to carry out parcel sorting according to deliverymen and delivery routes. To do so, it is necessary to develop a robot control system consisted with a context aware system and a sorting plan system with GIS engine regarding parcels accordingly.

The rest of the paper is organized as follows. Section 2 describes the process of the automatic parcel sorting and Section 3 describes in detail the implementation of the proposed robot control system. Section 4 explains how the proposed system works and Section 5 contains a brief conclusion.

2. Automatic Sorting Process of Postal Logistics

In the various areas of industry, robot manipulators are used for an automation of the production process [6].

They do only pre-programmed work so that they cannot be used in more delicate work such as sorting parcels according to different size and weight. The current robot manipulators used in the distribution industry target same-sized and same-shaped boxes or small pallets due to the technical limitation [7]. Figure 1 shows a configuration of the smart robot model for the logistics industry.

It is required to change control orders frequently according to the contexts given to robot manipulators by analyzing the context of parcels such as shape, weight and location [8]. The various parcels are moved on the conveyor and 3D sensors recognize the context data of

each parcel such as weight, size, address etc. Then the robot control system receives the context data of the parcel and generates pickup control information and sends it to robot manipulator. The robot manipulator picks up the parcels and drops it to the appropriate delivery box. An automatic process of sorting the parcel applying robot control system is shown in Figure 2. A smart robot control system is essential to enable robots to provide an active process for automatic parcel sorting. It is able to create control information. The automatic process for delivery sorting system goes through the following processes:

- The process of recognizing image, weight and location. The barcode recognition sensor sends context information and barcode information. A robot control system consists of two components. The one is a generation module for optimal sorting plan. It finds out delivery sorting plan using address barcode. The other module generates robot control data. Robot pick-up control information is created. More sophisticated robot pick-up control information is created and modified using the rule base. Through the process, the context aware based reasoning is needed.
- Delivery addresses of parcels are read by using barcode information and the delivery orders are decided by addresses.
- The delivery information is transferred to the conveyor according to the delivery orders of parcels. According to the delivery orders, locations of delivery sorting boxes are sent to robot manipulators.



Figure 1. Smart Robot Model for Logistics Industry

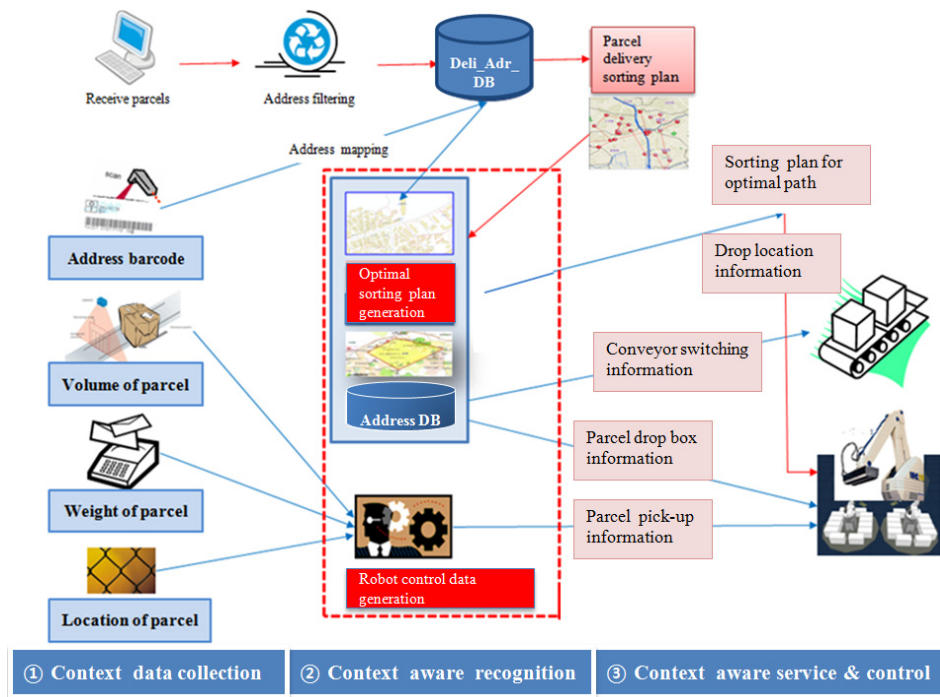


Figure 2. Automatic process for the context aware-based robot control system

- Robot control system is monitoring the conveyors and robot manipulators.

For the context aware service, first we define the domain ontology and the context model [8]. Then we construct the rule base, which performs reasoning for the generation of robot control data according to the context of each parcel. The general components of the context aware system based on ontology are

context acquisition module, ontology based context model, context knowledge base and context reasoning engine [6]. Figure 3 presents the class hierarchy of robot control data for defining domain ontology. It is used in order to construct the ontology based context model. The context acquisition module collects contexts from various sensors. We define domain ontology using

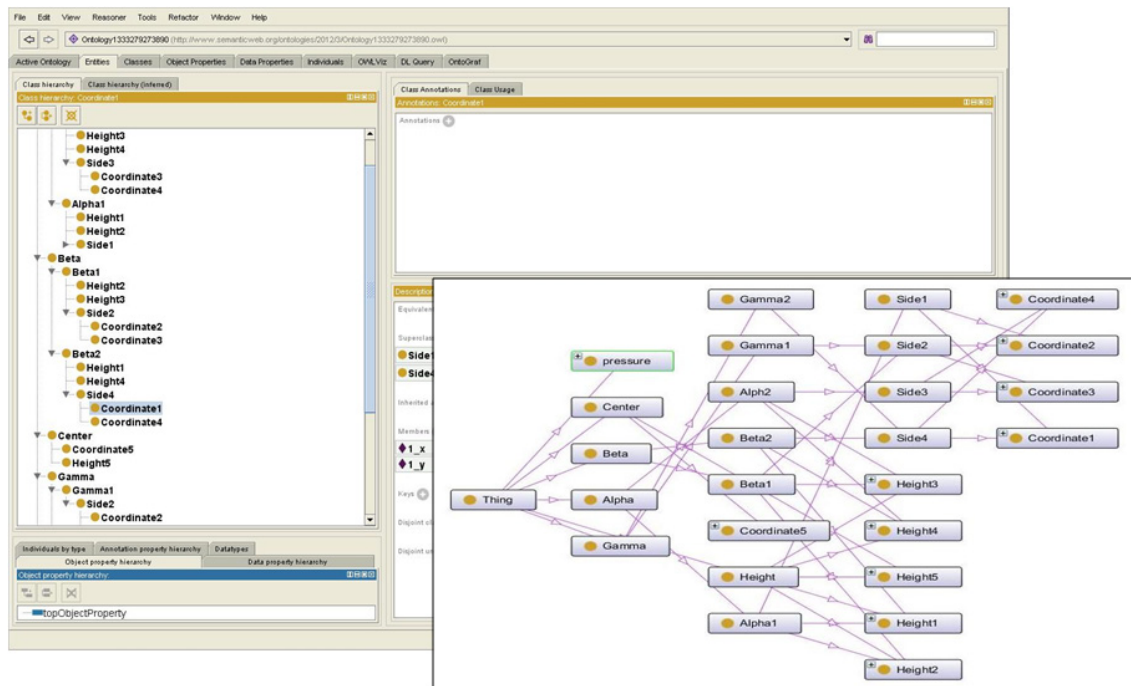


Figure 3. Context Model for Robot Control Information of Parcels

owl [10] and construct an ontology based context model for a robot manipulator to perform the sorting of parcels automatically.

The context knowledge base stores rules for context service. Sometimes the context data is invalid, so the context reasoning engine figures out the context data, and enables to generate revised contexts for proper context service [5].

In section 3, we describe how our system controls a robot manipulator for automatic parcel sorting and implementation of our system.

3. Architecture of the Robot Control System

The framework of the robot control system is shown in Figure 4. The robot control system performs context service and reasoning in order to generate robot control data.

into a rule database. The validation rule is used to validate context data. An invalid context data will be initialized by the validation rule and replaced with a 'null' value. The invalid context data is revised again by the revision rule. The pick-up rule is used to generate robot control information. The process of the context aware service to provide pick-up information is shown in figure 6. As shown in figure 6, the final robot control information is created through three-step rules. In the first step, the validation rules are used in order to validate the context data of parcels and to decide initial data that is classified as an error. It is a very important step since context information coming from sensors often has errors.

After applying the validation rule, the invalid information or the initialized information is restored. If there is no item applied to the validation rule, the next step will be the

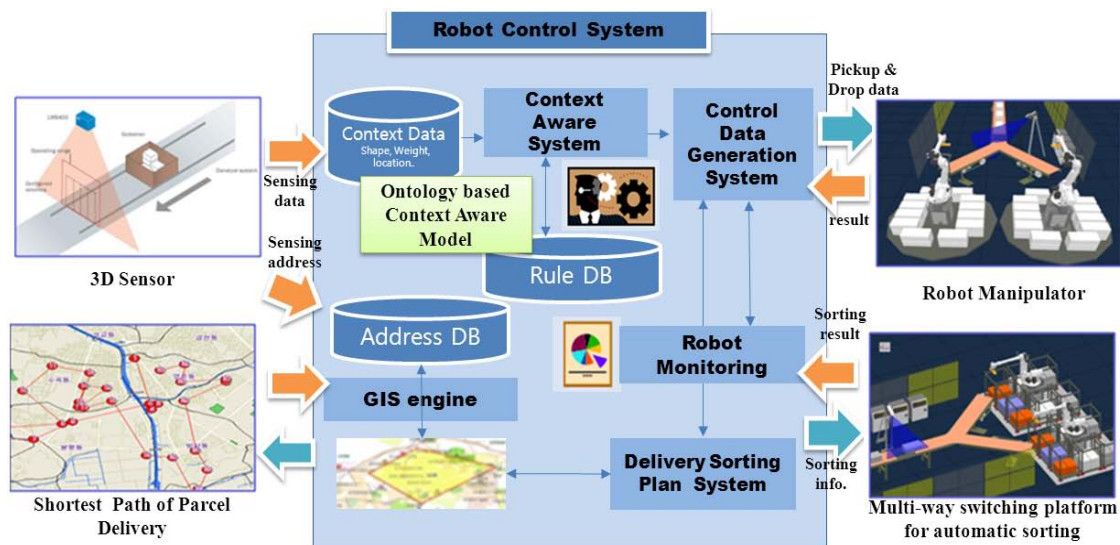


Figure 4. Framework of a Robot Control System for automatic parcel sorting

The raw data from the sensor is transferred and stored in the context information database going through the process of abstraction and generalization in the context collector. Then the context manager, which provides robot control information, creates pick-up data from the context of the parcel using a rule database.

Then it recognizes the context by relational database ontology. It also creates robot control information and transfers it to the robot utilizing the knowledge base. An example of the context acquisition process is shown in Figure 5.

We construct a relational database to store the ontology rules. Three types of rules are stored

pick-up rule process, which creates robot control information.

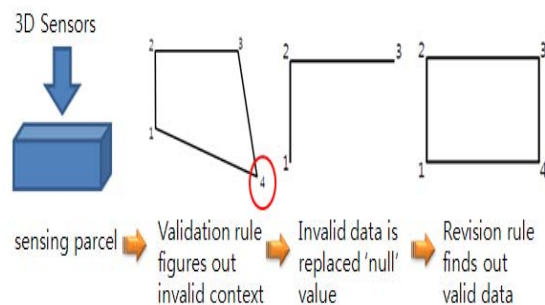


Figure 5. An example of the context acquisition process

The pickup rule creates the final robot control information. If there is not enough context information to be processed, the pickup rule cannot be applied. In this case, the context aware system will give a rejecting order to the robot, and parcels will be sent to an unclassified box.

beta, gamma and weight of the parcel in order to pick up a parcel.

We designed software architecture of the robot control system. The main system operates an each subsystem and device environments control.

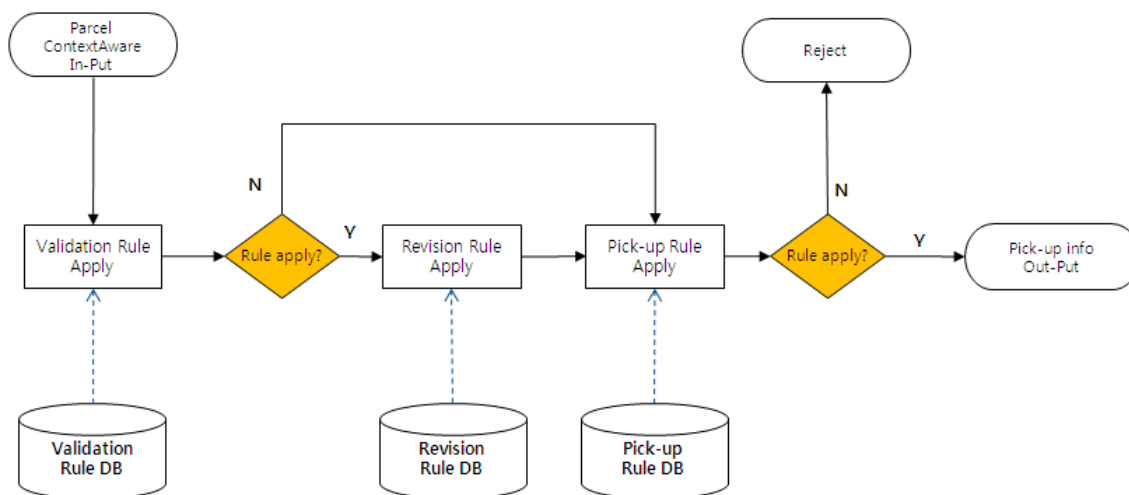


Figure 6. Process of the context aware mechanism for the robot control system

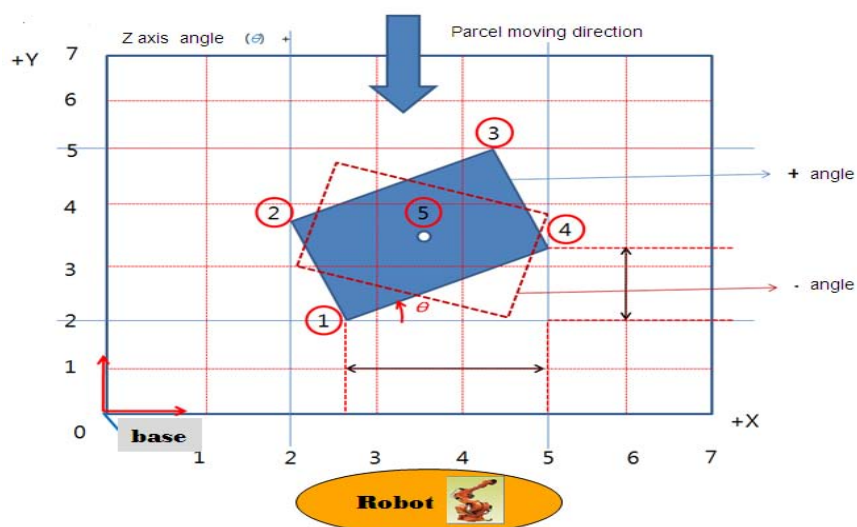


Figure 7. 5 coordinates of parcel

In order to generate robot control data, we need to obtain five coordinates: four from the angular points of the upper surface of a parcel and one from the center point as shown in Figure 7. The four coordinates are numbered as 1 to 4 and center 5. In addition, we can calculate three values such as alpha, beta and gamma, as shown in Figure 8. The robot needs to get the values such as alpha,

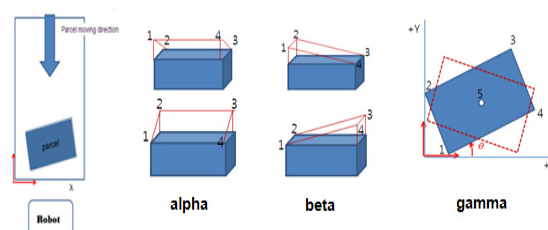


Figure 8. 3 values: alpha, beta, gamma

Figure 9 shows the architecture of the robot control system. Address manager filters address barcode information into delivery address using address database.

Switching & drop data generation system creates conveyer switching information that is which robot is activated. And then it generates drop data of which box is available for the parcel.

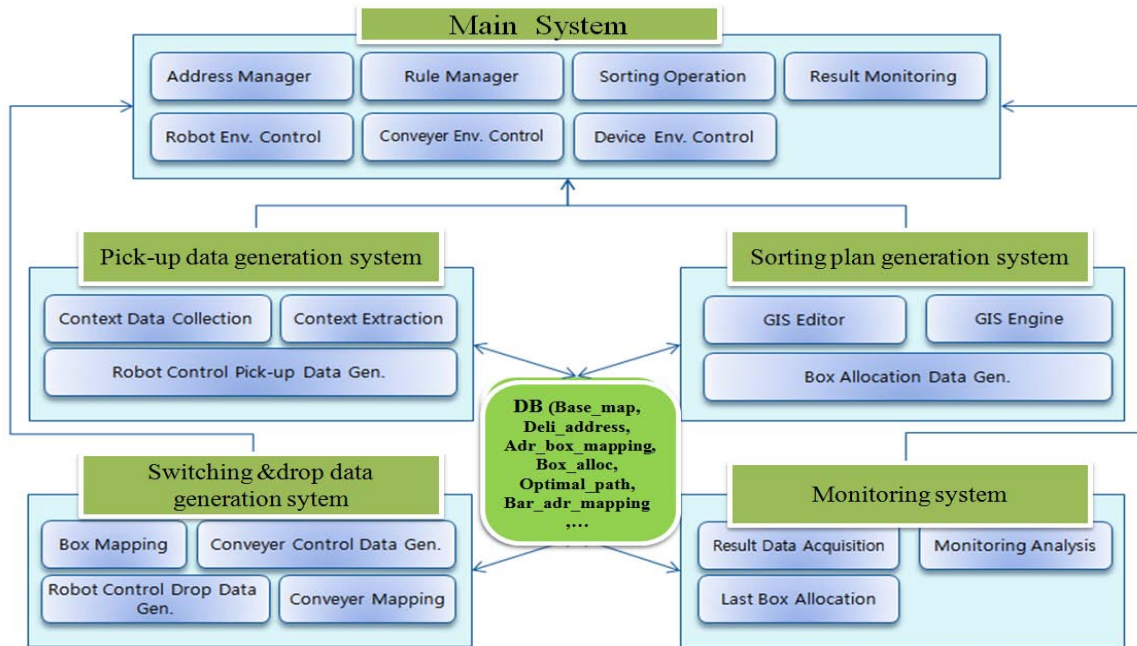


Figure 9. Software architecture of a robot control system

Rule manager creates rules and manages rule database. Monitoring system receives the results of parcel sorting from the robot and analyzes the statistics of automatic parcel sorting.

These systems use the databases such as address database, context database, rule database and so on.

Pickup data generation system generates pick-up control information using context data of a parcel.

Before we designed and implemented a robot control system with a context aware mechanism, we designed database schema and constructed a database and made rules of reasoning for pick-up control data according to context data. We need various inference rules and we are able to append inference rules incrementally from the experiments.

Sorting plan generating system constructs optimal delivery path using taboo search.

ID	ITEM	RULE_COND	VAL
0001	size_weight	size_weight>30	size_weight=null
0002	size_weight	size_weight<0	size_weight=null
0003	cordnats1_x	cordnats1_x<=0	cordnats1_x=null
0004	cordnats1_y	cordnats1_y<=0	cordnats1_y=null
0005	cordnats2_x	cordnats2_x<=0	cordnats2_x=null
0006	cordnats2_y	cordnats2_y<=0	cordnats2_y=null
0007	cordnats3_x	cordnats3_x<=0	cordnats3_x=null
0008	cordnats3_y	cordnats3_y<=0	cordnats3_y=null
0009	cordnats4_x	cordnats4_x<=0	cordnats4_x=null
0010	cordnats4_y	cordnats4_y<=0	cordnats4_y=null
0011	cordnats_cntr_x	cordnats_cntr_x<=0	cordnats_cntr_x=null
0012	cordnats_cntr_y	cordnats_cntr_y<=0	cordnats_cntr_y=null
0013	cordnats1_height	cordnats1_height<=0	cordnats1_height=null
0014	cordnats2_height	cordnats2_height<=0	cordnats2_height=null
0015	cordnats3_height	cordnats3_height<=0	cordnats3_height=null
0016	cordnats4_height	cordnats4_height<=0	cordnats4_height=null
0017	cordnats_cntr_height	cordnats_cntr_height<=0	cordnats_cntr_height=null

Figure 10. Validation rules

ID	ITEM	RULE_COND
0001	size_weight	size_weight is null
0002	tag_no	tag_no is null
0003	cordnats1_x	cordnats1_x is null and cordnats2_x is null and cordnats3_x is not null and cordnats4_x is not null and cordnats_
0004	cordnats1_y	cordnats1_y is null and cordnats2_y is null and cordnats3_y is not null and cordnats4_y is not null and cordnats_
0005	cordnats2_x	cordnats1_x is null and cordnats2_x is null and cordnats3_x is not null and cordnats4_x is not null and cordnats_
0006	cordnats2_y	cordnats1_y is null and cordnats2_y is null and cordnats3_y is not null and cordnats4_y is not null and cordnats_
0007	cordnats1_x	cordnats1_x is null and cordnats4_x is null and cordnats2_x is not null and cordnats3_x is not null and cordnats_
0008	cordnats1_y	cordnats1_y is null and cordnats4_y is null and cordnats2_y is not null and cordnats3_y is not null and cordnats_
0009	cordnats4_x	cordnats1_x is null and cordnats4_x is null and cordnats2_x is not null and cordnats3_x is not null and cordnats_
0010	cordnats4_y	cordnats1_y is null and cordnats4_y is null and cordnats2_y is not null and cordnats3_y is not null and cordnats_
0011	cordnats3_x	cordnats3_x is null and cordnats4_x is null and cordnats1_x is not null and cordnats2_x is not null and cordnats_
0012	cordnats3_y	cordnats3_y is null and cordnats4_y is null and cordnats1_y is not null and cordnats2_y is not null and cordnats_
0013	cordnats4_x	cordnats3_x is null and cordnats4_x is null and cordnats1_x is not null and cordnats2_x is not null and cordnats_
0014	cordnats4_y	cordnats3_y is null and cordnats4_y is null and cordnats1_y is not null and cordnats2_y is not null and cordnats_
0015	cordnats2_x	cordnats2_x is null and cordnats3_x is null and cordnats1_x is not null and cordnats4_x is not null and cordnats_
0016	cordnats2_y	cordnats2_y is null and cordnats3_y is null and cordnats1_y is not null and cordnats4_y is not null and cordnats_
0017	cordnats3_x	cordnats2_x is null and cordnats3_x is null and cordnats1_x is not null and cordnats4_x is not null and cordnats_

Figure 11. Revision rules

The real rules are shown in the following figures. Figure 10 shows the validation rules. The revision rules are shown in Figure 11, and the pick-up rules are shown in Figure 12.

the upper side of parcels, and the central point, and entered into the system. Robot control information is also created by a knowledge based reasoning engine, and sent to a robot.

ID	ITEM	RULE_COND	VAL
0001	gamma2	size_width>size_length and cordnats1...	size_gamma=asin((cordnats4_y-cordnats1_y)/sqrt((cordnat...
0002	gamma2	size_width>size_length and cordnats1...	size_gamma=360-asin((cordnats1_y-cordnats4_y)/sqrt((cor...
0003	gamma2	size_width>size_length and cordnats1...	size_gamma=0
0004	gamma2	size_width<size_length and cordnats1...	size_gamma=asin((cordnats4_y-cordnats1_y)/sqrt((cordnat...
0005	gamma2	size_width<size_length and cordnats1...	size_gamma=asin((cordnats1_y-cordnats4_y)/sqrt((cordnat...
0006	gamma2	size_width<size_length and cordnats1...	size_gamma=90
0007	gamma1	size_width>size_length and cordnats2...	size_gamma=asin((cordnats3_y-cordnats2_y)/sqrt((cordnat...
0008	gamma1	size_width>size_length and cordnats2...	size_gamma=360-asin((cordnats2_y-cordnats3_y)/sqrt((cor...
0009	gamma1	size_width>size_length and cordnats2...	size_gamma=0
0010	gamma1	size_width<size_length and cordnats2...	size_gamma=asin((cordnats3_y-cordnats2_y)/sqrt((cordnat...
0011	gamma1	size_width<size_length and cordnats2...	size_gamma=asin((cordnats2_y-cordnats3_y)/sqrt((cordnat...
0012	gamma1	size_width<size_length and cordnats2...	size_gamma=90
0013	alpha1	size_width>size_length and cordnats1...	size_alpha=asin(abs(cordnats1_height-cordnats2_height)/s...
0014	alpha1	size_width>size_length and cordnats1...	size_alpha=asin(abs(cordnats1_height-cordnats2_height)/s...
0015	alpha1	size_width>size_length and cordnats1...	size_alpha=asin(abs(cordnats1_height-cordnats2_height)/s...
0016	alpha1	size_width>size_length and cordnats1...	size_alpha=asin(abs(cordnats1_height-cordnats2_height)/s...
0017	alpha1	size_width>size_length and cordnats1...	size_alpha=0

Figure 12. Pick-up rules

4. Implementation and Experimental Results

We implemented the robot control system under the development environment shown in Table 1. Also we carried out two experiments. First experiment was an examination of the process of collecting and storing information from sensors. We also figured out robot control information regarding several types of parcels. Figure 13 demonstrates the implementation of our robot control system to automate parcel sorting. The following table shows that parcel context information such as weight and barcode is calculated by both the four coordinators on

Table 1. Development Environment

H/W	<ul style="list-style-type: none"> - CPU: Intel Core(TM) i7 CPU 950 3.07GHz - RAM: DDR3 4GB PC3-10600 (3EA) - VGA: Radeon HD 5770 D5 1GB Vapor-X - HDD: WD 2TB Carviar Green WD20EARS (SATA2/64M) - LAN: Marvell Yukon 88E8056 PCI-E Gigabit Ethernet Controller
S/W	<ul style="list-style-type: none"> - OS: Windows Server 2008 32bit - Language: VB .Net (VS 2005), VC++ - Platform: .NET Framework - DB: MySql 5.1 - GIS Engine: GeoBusObject v3.0

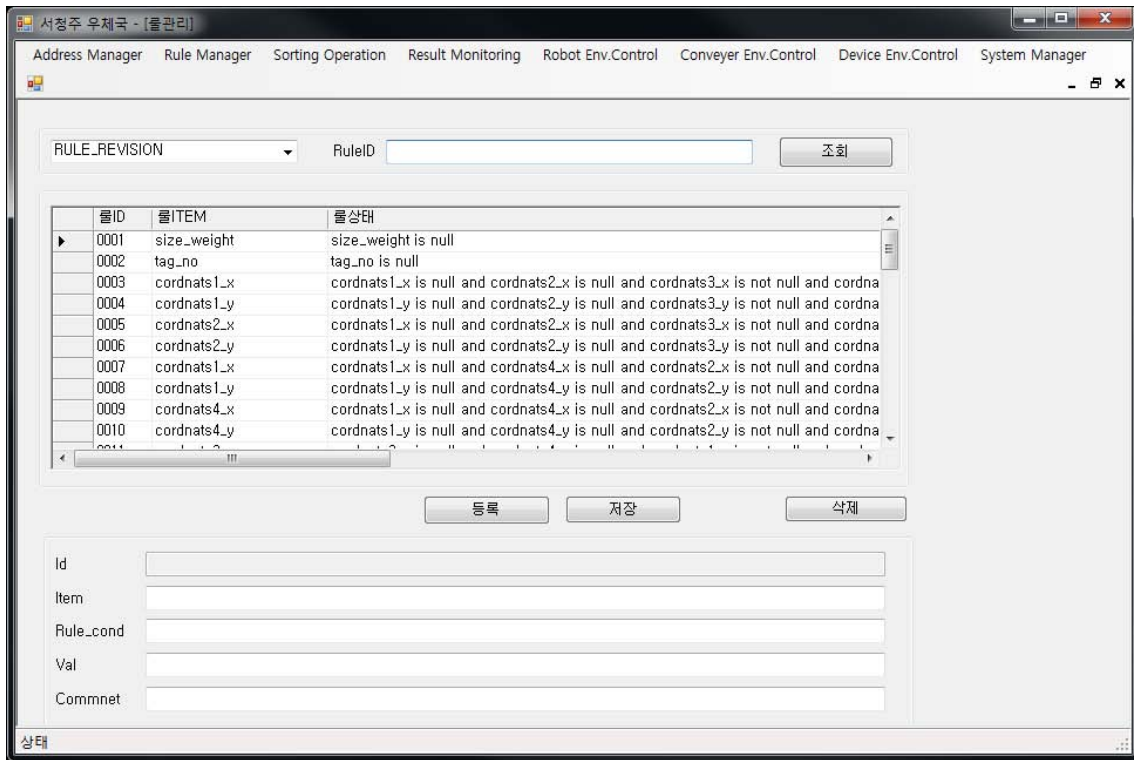


Figure 13. Implementation of the robot control system

Table 2. Context Data of Parcels

item set	CASE A	CASE B	CASE C
Tag_No	0001A	0001B	0001C
Size_Width	NULL	NULL	NULL
Size_Length	NULL	NULL	NULL
Size_Height	15	34	20
Size_Weight	13	25	17
CORDNATS1_X	5	-	4
CORDNATS1_Y	3	-	6
CORDNATS1_HEIGHT	15	33	20
CORDNATS2_X	5		
CORDNATS2_Y	21		
CORDNATS2_HEIGHT	13	33	20
CORDNATS3_X	32		
CORDNATS3_Y	21		
CORDNATS3_HEIGHT	13	34	20
CORDNATS4_X	32		
CORDNATS4_Y	3		
CORDNATS4_HEIGHT	12	34	20
CORDNATS_CNTR_X	22.5		
CORDNATS_CNTR_Y	15		
CORDNATS_CNTR_HEIGHT	19		

Here is an example scenario. As shown in Table 3, the context data of the parcel in case A was normal. The robot control data is generated and the result of the reasoning of CASE A is shown in the Table3. So the robot can pick up the parcel and drop it to the appropriate box.

Table 3. Robot Control Data Generation

pick-up control data set	RESULTS		
	A	B	C
Tag_No	0001A	0001B	0001C
Alpha	--	0	0
Beta	0	--	0
Gammar	0	0	--
CORDNATS_CNTR_HEIGHT	19	--	--
CORDNATS_CNTR_X	22.5	--	--
CORDNATS_CNTR_Y	15	--	--
Pressure	22	--	--

CORDNATS1_X and CORDNATS1_Y are missed in CASE B. Therefore, the rule reasoning engine must find out the values of two coordinates. In this case, the parcel can be picked up by a robot, which receives pickup control data from the context aware system. The parcels can be sorted.

The values of CORDNATS2_X and CORDNATS2_Y are invalid in CASE C. It is caused by the excessive difference of the two values. The context reasoning engine recognizes that values are wrong. The revision rules can be applied by the reasoning engine. Then, the pickup control data are revised and sent to a robot. And the parcel can be sorted.

Figure 14 shows the results of generation of robot control data. Through the experiments, we only verified that the robot control system is working properly.

Secondly, we performed an experiment of the time for parcel sorting in the case of using a robot manipulator compared to a manual method. Six hundreds of parcels were sorted automatically by two robots. The simulation was carried out 10 times. The average time of automatic parcel sorting by the two robots was 84 minutes. It took 120 minutes that a delivery man sorted the same amount of parcels manually. We can verify an automatic parcel sorting by the robot saves much time. We need to get more rules to reason through practical operation after integrating more than two robots with our robot control system.

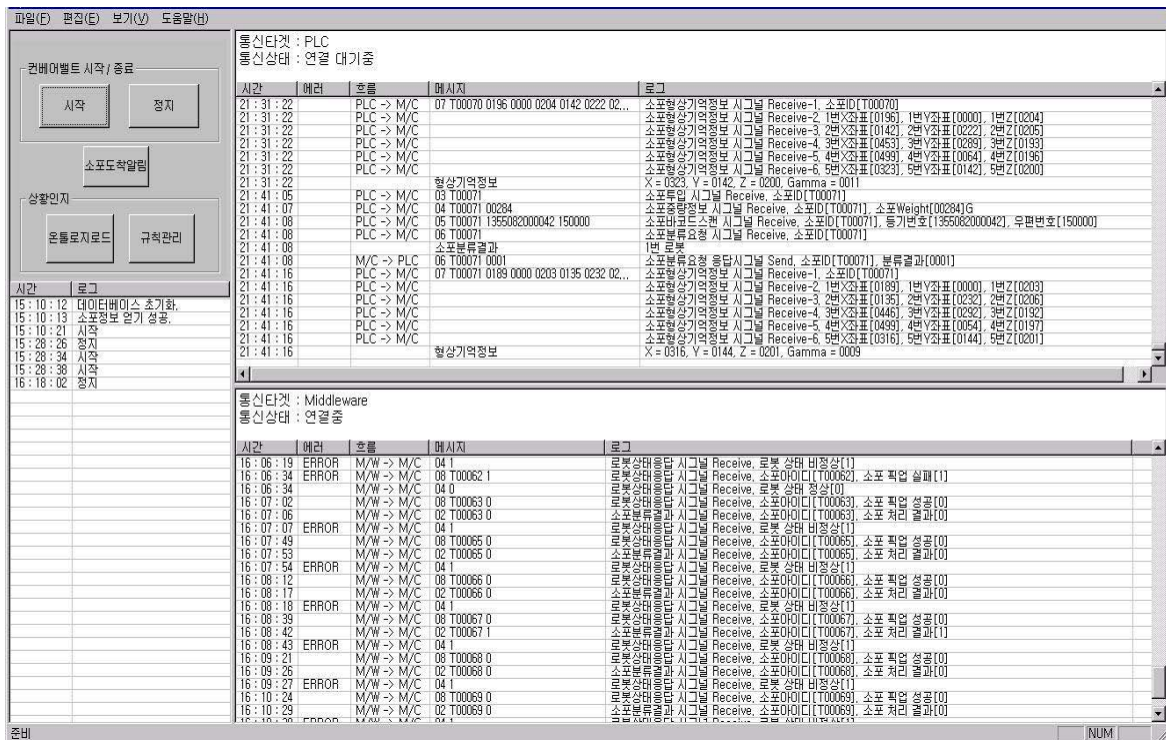


Figure 14. Generation of robot control data

5. Conclusion

The context aware service enables a robot manipulator to sort parcels automatically.

In this paper, we designed and developed a robot control system based on the context aware mechanism in order to automate parcel sorting. The implemented robot control system consisted with optimal sorting plan generation system, pick-up data generation system, switching & drop generation system and context aware system. Our robot control system performed the following activities. First, context data obtained from different types of sensors were analyzed for the robust motion control of robotic manipulators. Second, the rule databases, which stored and managed various inference rules, were built and managed. Third, pick-up & drop information to control the robot manipulator was generated from the rules. Fourth, it controls a robot manipulator to carry out parcel sorting according to deliverymen and delivery routes.

We verified our system by conducting experiments with various post boxes. The experiments worked well in sorting parcels and proved that the same can be done in a real situation. We also created context data, various inference rules from the test data and various types of cases. Our experiments proved that the robot control information was correctly created from context data and inference rules. Also, the robot manipulator automatically carried out parcel sorting. Our research will contribute to reducing time, cost and industrial accidents. In the future, the proposed robot control system will be installed in conveyors and machine systems with more than two robot manipulators.

Acknowledgements

This paper was supported by Konkuk University.

REFERENCES

1. JEONG, I.-C., H.-S. KIM, H. KIM, **Future Robotic Terminal based IT Convergence Analysis of Electronics and Communications Trends**, Vol. 25(4), August 2010, pp.16-17.
2. LEE, J.-J., **Robust Adaptive Control for Underactuated Robot Manipulators** Doctoral thesis -- Korea Advanced Institute of Science and Technology: Department of Electrical Engineering, 1999, p.19.
3. SUSNEA, I., G. VASILIU, **On Using Passive RFID Tags to Control Robots for Path Following**, Studies in Informatics and Control, vol. 20(2), 2011, pp. 157-162, ISSN 1220-1766.
4. CHO, J.-H., **A Context Awareness and Prediction Support System for Efficient Management of u-City** The Graduate School of Chung-Ang University, PhD thesis, 2011, pp. 55-85.
5. LEDEY, A., G. ABOWD, **Towards a Better Understanding of Context and Context-Awareness**, Workshop on the what, who, where, when and how of context-awareness at CHI 2000, 2000.
6. RANGANATHAN, A., R. H. CAMPBELL, **A Middleware for Context-aware Agents in Ubiquitous Computing Environments**, In: ACM/IFIP/USENIX International Middleware Conference, Rio de Janeiro, Brazil, 2003.
7. BRASSAI, S. T., B. IANTOVICS, C. ENACHESCU, **Optimization of Robotic Mobile Agent Navigation**, Studies in Informatics and Control, vol. 21(4), 2012, pp. 403-412, ISSN 1220-1766.
8. YAU, S. S., F. KARIM, Y. WANG, B. WANG, S. GUPTA, **Reconfigurable Context-sensitive Middleware for Pervasive Computing**, IEEE Pervasive Computing, vol. 1(3), 2002, pp.33-40.
9. KIM, T. H., D. M. SHIN, M. S. KIM, J. S. KIM, K. H. RYU, M. S. SHIN, **An Ontology based Context Aware System to Control the Robot Manipulator in Delivery Logistics Industry**, In Proceedings of the 2nd International Conference on Information Science and Industrial Applications, ISSN 2287-1233, pp. 221-226, Cebu, May 2012.
10. **W3C, OWL2 Web Ontology Language**, W3C Recommendation 27 October, 2009, <http://www.w3.org/TR/owl2-overview/>