2012 International Conference on Advanced Computer Science Applications and Technologies

Using Information Management Architecture to Support Flexible Supply Chain Management

Mohd Izzuddin Mohd Tamrin
Department of Computer Science
International Islamic University Malaysia
Selangor, Malaysia
izzuddin@iium.edu.my

Tengku Mohd Tengku Sembok
Department of Computer Science
International Islamic University Malaysia
Selangor, Malaysia
tmts@iium.edu.my

***Abstract*—Integration between the members in complex network of the supply chain group has been an issue for flexible supply chain management. Restructuring of pre-planned activities across the group to react to known deviation may not be possible. The authors decided to introduce an Information Management Architecture (IMA) as the intermediary process which can provide semi-automated assistance for tighter integration. The IMA sits between the RFID infrastructure and the existing supply chain systems and continuously monitors the current activities and wraps services from existing systems to increase the flexibility for immediate process restructuring. A comparison was made between two types of deviations that occurred at the early and final stages of a customization process with and without IMA support. The results showed that the IMA was more stable compared to other supports and significantly performed better in completing joint tasks.**

Keywords-Intermediary process, RFID services, component systems, flexible process, case-based reasoning systems.

I. INTRODUCTION

The supply chain is a very complex network of entities working together to the deliver services or products to consumers. The application scenario is based on the integration architecture, whereby the supply chain consists of the manufacturer, service integrator, service providers and the client. Careful management of the supply chain activities to act as a group will determine whether the right services are delivered to the right consumer at the right time. Acting as a group requires them to focus on the same goals and clearly defined process flows between the members in the group in order to coordinate their joint activities. This illustrates how important the supply chain management is due to its direct impact on consumer satisfaction and the cost of operations.

Integration issues tend to hinder a more flexible supply chain management. Most joint activities have been pre-planned and restructuring these activities to react to known deviation [12] may not be possible. This is because the ongoing status of the services is not visible [5] to the other members in the same group once they are forwarded to next member in the chain. Furthermore, the systems that are employed for each member to carry out these activities operate in isolation and other members have limited control over them. The quality of information generated from these systems may be compromised by the manual data retrieval from the workers [8] performing the supply chain activities and the information gathered are not uniquely associated at the individual level. An example of supply chain that focuses on delivering services with the client involved in its creation can be found in [7].

The authors believed that creating Radio Frequency Identification (RFID) services as the intermediary process can provide the necessary semi-automated assistance to the members for tighter integration. The RFID services can limit retrieval of the manual erroneous data from the workers, and continuously monitor as well as share the information on the ongoing business process across the supply chain group between the members. An Information Management Architecture (IMA) is introduced to detect any unintended but known deviation that can occur in the current supply chain activities to try to overcome it. By wrapping the functions from the existing supply chain systems into services allows the IMA to remove and add relevant services into the main process; and react accordingly to the problems that impinge the current business activities.

Sections II and III describe examples of related works and an overview of the IMA respectively. Section IV describes the collaboration algorithm that governed the interaction of the IMA components. Section V presents the evaluation of the IMA prototype via simulated environments. The final section shows the results and analysis.

II. RELATED WORKS

The RFID mean time architecture [4] captures the average time between the movements of the item from one point to the next point based on the working schedule. Discovery of high average time triggered the investigation team to study the cause of the problems for the high latency time and suggest ways in improving it. The average time does not tell us much on the activities of the supply chain. For this reason, the authors' approach was to associate the tag identification with the operations from the supply chain.

The traceability system architecture [6] captured the tag identification by the readers installed across the supply chain group. This information is associated with other tag identification in the same batch and the operations it has to go through to get high level overview of the supply chain process. In comparison, the authors included additional

978-0-7695-4959-0/13 $25.00 © 2013 IEEE
DOI 10.1109/ACSAT.2012.10

information such as the profile of the members, the services offered and information accessibility to facilitate dynamic interaction for process restructuring.

The RFID-based resource management system [2] utilized the case in the database to optimize the selection for resources to be used in every order triggered. The approach the authors took differed from this approach in that we employed the XML to model our case solutions. The reasons for this are to enable faster navigation through the document and to simplify the process of modification in the case model to reflect additional known deviations in the future.

The service-based shop floor architecture [9] is designed to support agile shop floor process. As the requirements from the client keep on changing, the current shop floor process is reconfigured to meet those requirements. It employed the manufacturing device services to orchestrate the right process. However, it has a narrowed focus on the process of the shop floor. In comparison, the authors employed services across the members in the group in order to have full control over the entire process.

The collaborator system [1] is designed based on the service oriented architecture. For this reason, it has the ability to generate global planning across the collaborating members to create the final process for the entire group. The services are presented as web portlet and the module administrator is responsible to define the tasks of the web portlet. Our approach differed in that we wrapped the functionalities from the existing systems and make them available as services.

The real time collaboration architecture [10] relied on agent technologies to provide real time collaboration in the supply chain activities; techniques that are similar to [3]. Semantic of the information exchanged and the reasoning technique are defined in order to understand the message and behavior of the agents respectively. The authors followed a simpler approach, with the architecture to merely assist the team members appointed from the group to make decisions on process restructuring through case recommendation and validation.

III. Architecture Overview

The IMA sits between the existing systems and the RFID readers. It is an intermediary layer that are composed of six main components in two main sections as shown in Figure 1: the main system and the subsystems. The system communicator, RFID service provider and service handler come in as a set and run on the fixed network of every member in the supply chain group. On the other hand, the service integrator, case handler and information handler run on a central server which communicates with the rest of the subsystems across the members in the group.

The following are the responsibilities of the components in the subsystems:

- System Communicator – interacts with the existing supply chain system and provides their functionalities as services to the IMA
- RFID Service Provider – transforms basic data retrieved from the RFID readers such as tag identification, time stamp and reader identification. By associating it with business rules allows the IMA to make sense of it at activities level.
- Service Handler – serves as the middleman to the subsystem from the main system. Any services that are scheduled from the main system will have to be passed to the respective service handler for execution. It is also responsible to detect deviation in the current activities by comparing the information passed by the RFID service provider with the instruction from the main system.

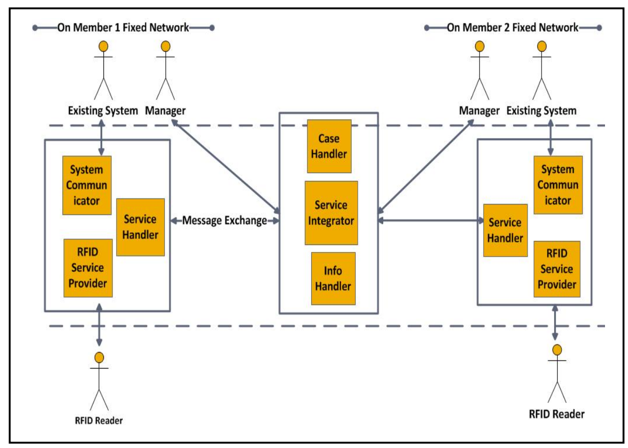

Figure 1. The Organization of the IMA.

The following are the responsibilities of the components in the subsystems:

- Service Integrator – serves as the middleman to the main system from the subsystems. It is responsible for restructuring the supply chain process in order to react to the deviation detected by the service handler. However, it requires the case handler and the information handler to give insight into new processes before it can proceed further for execution.
- Case Handler – responsible to generate recommendation on the tasks that should be executed by the IMA in response to the deviation detected. It holds many potential cases that are known to be the solutions for the deviation in the supply chain activities.
- Information Handler – responsible to verify the possibility for the execution of the recommended tasks by the case handler. It mainly searches for the status of the member that will be executing the services. These services to be executed are within the job scope of that member and accessibility of the information required by the services.

IV. Collaboration Algorithm

In this section, we will describe the collaboration algorithm that governs the interaction between six main entities mentioned in the previous section. The algorithm is designed to undergo six main phases: vigilant, deviation, alteration, validation, switch over and registration. These algorithms are presented at a high level description in the form of a pseudo-code.

A. The Vigilant Phase

Figure 2 describes the algorithm in the vigilant phase. The subsystem denotes the rfidServer as continuously listening to the data pushed by the RFID readers in its vicinities. The sequence of data (LR) will be forwarded to the sub-component of the RFID service provider, namely readerInterface. It then invokes the "startMonitor" method call to initiate the transformation process of the LD into the list of event (LE) which describes the activities. However, before that the RFID service provider will first check whether the vicinity of the current member is recorded in the link. The reason for this is to enable the IMA to track the history of the supply chain activities across the group. The link is updated by invoking the "writeLink" method call by supplying the memberNum as the argument.

```
Variables: socket, rfidServer, LE, LOR, memberNum
while(true):
        try :
                Socket socket = rfidServer.accept();
                SubSystem(socket);
SubSystem:
spObj = new ServProvider();
LOR = spObj.getCurLink();
if  (LOR != memberNum), then:
        LOR = spObj.writeLink(memberNum);
else:
        riObj = new ReaderIntr();
        LE = riObj.startMonitor(socket);
```

Figure 2. High Level Algorithm Description for Vigilant Phase.

B. The Deviation Phase

```
Variables: LE, LOR, LP, clientMain, pw, hostName, portNum
SubSystem:
while(curEvent == false):
     evmObj = new EventMgr();
     LOR = evmObj.detDeviation(LE);
  if (LOR == true), then:
     clientMain = new Socket(hostName, portNum);
     pw = new PrintWriter(clientMain.getOutputStream(), true);
     pw.println(LP);
  else:
     evmObj.insertEv(LE);
```

Figure 3. High Level Algorithm Description for Deviation Phase.

Figure 3 describes the algorithm in the deviation phase. The eventManager, a sub-component of the service handler utilizes the LE from the previous phase and compares it with the list of instruction (LI) retrieved from the existing system to determine any deviation from the current activities. The outcome of the analysis will be recorded into the list of result (LOR). Based on the result, if deviation has occurred, the main system will be contacted via the client socket, namely clientMain. The host name and port number of the main server need to be specified before the list of problem (LP) can be forwarded for process restructuring. On the other hand, if there is no deviation detected, the event manager will invoke the "insertEv" method call to record the current LE into the central repository.

C. The Alteration Phase

Figure 4 describes the algorithm in the alteration phase. The main server will pass the LP to the caseInterface, one of the sub-components of the service integrator. It will invoke the "initiateChange" method call to initiate process change. At the same time, running in parallel is the subServer, pairing with the rfidServer to establish the entire intermediary process for the subsystem in the local vicinity. Due to the deviation in the current activities, the IMA needs to stop the current process to make way for the new process. For this reason, every service meant to be executed next from the list of scheduled services (LSS), the event manager will have to first check whether the LE for the previous service has been successfully generated.

```
Variables: socket, mainServer, LP, LC, LSS, LOR
while(true):
        try :
                Socket socket = mainServer.accept();
                MainSystem(socket);
MainSystem:
ciObj = new CaseIntr();
LC = ciObj.initiateChange(LP);
SubSystem(Running in Parallel):
for (int i=0; i<LSS; i++) :
// Based on the insertEv(LE) method execution in deviation
// phase
        evmObj = new EventMgr();
        LOR = evmObj.verifyEv(LSS.get(i));
        if (LOR == true) , then:
                exmObj = new ExecMgr();
                exmObj.prepExec(LSS.get(i));
        else:
                break;
```

Figure 4. High Level Algorithm Description for Alteration Phase.

This is accomplished by invoking the "verifyEV" method call and the outcome will be recorded in the LOR. If the LE is generated, it will then continue to proceed with the execution by invoking the "prepExec" method call. Otherwise, it will seize the process from running. Going back to the main system, the "initiateChange" method call will trigger an alert to the relevant team members that have been assigned to be responsible for process restructuring across the supply chain group. Decision will be made with the help from case manager.

D. The Validation Phase

Figure 5 describes the algorithm in the validation phase. The caseInterface, the sub-component of the service integrator is responsible for passing the list of case (LC) or the potential solutions to the information handler for verification. These solutions are made from relevant services which could overcome the problem of deviation. The information handler is responsible for checking the status of the members that are supposed to execute those services, the accessibility of the information required by the services, and those services within the job scope of the executors. The outcome from the verification process will be kept in the LOR. If the result is positive, the validation phase has ended and the switch over phase will begin. Conversely, if the result is negative, the team member will have to make amendments to the LC and undergo the validation process again until the LOR is positive.

```
Variables: LC, LOR
MainSystem:
while(LOR == false):
// While loop is located before entering the alteration phase
ciObj = new CaseIntr();
LOR = ciObj.verifyCase(LC);
```

Figure 5. High Level Algorithm Description for Validation Phase.

E. The Switch Over Phase

```
Variables: C, LC, LC^partial, probIndex, LSS, refNum, memberNum, LOR
MainSystem:
LSS.remove(probIndex);
for (int i=probIndex+1; i<LSS; i++) :
        LC.add(LSS.get(i));
while(LC != null):
        for (int j=0; j<LC; j++) :
                if(j=0), then:
                        C = LC.get(j);
                        refNum = C.getMemberNum();
                        LC^partial = LC.get(j);
                else:
                        C = LC.get(j);
                        memberNum = C.getMemberNum();
                if (memberNum == refNum), then:
                        LC^partial = LC.get(j);
                        LC.remove(j);
        pmObj = new ProcessMgr();
        pmObj.execPartialCase(LC^partial);
        while(LOR==false):
                evmObj = new EventMgr();
                LOR = evmObj.verifyPartialCase(LC^partial);
```

Figure 6. High Level Algorithm Description for Switch Over Phase.

Figure 6 describes the algorithm for the switch over phase. The service integrator is responsible for preparing the new list of the services for the entire supply chain group. Firstly, it will remove the deviated service from LSS by pointing to the probIndex. Next, the services that have not been executed yet from the LSS are appended to the LC. LC contains the recommended services that are supposed to overcome the deviation in the current activities. However, the service integrator needs to break the LC that is composed of the new services for the entire group into a list of partial cases ($LC^{partial}$). The segregation is based on services that belong to the same member identification.

This is done to make it easy to delegate the $LC^{partial}$ to their respective subsystems across the supply chain group. The process manager, the sub-component of the service integrator is responsible for the delegation to the respective service handler by invoking the "execPartialCase" method call. Later, it will have to monitor that the first $LC^{partial}$ has finished the execution before proceeding with the next $LC^{partial}$ by invoking the "verifyPartialCase" method call. Only if the LOR is positive from the verification will it continue with the next partial case. Otherwise it will keep on checking the status of the first partial case.

F. The Registration Phase

```
Variables: LC^partial, LOR, C, serviceList, tagList
SubSystem:
for (int i=0; i<LC; i++) :
        C = LC^partial.get(i);
        serviceList.add(C.getService());
        tagList.add(C.getTag())
for (int j=0; j<tagList; j++) :
        for (int k=0; k<serviceList; k++) :
                spObj = new ServProvider();
                LOR = spObj.write(tagList.getTag(j),
                        serviceList.getService(k));
```

Figure 7. High Level Algorithm Description for Registration Phase.

Figure 7 describes the algorithm for the registration phase. After receiving the $LC^{partial}$ from the main system, the service handler begins the process of recording the relevant services into the memories of every tag that is involved in the newly scheduled activities. This is accomplished by separating the service identification and tag identification from every case in the $LC^{partial}$ into the serviceList and tagList respectively. Since the tag comes in batches, for every tag extracted from the tagList, it will be associated with every service in the serviceList. These values will be forwarded to the RFID service provider, and by invoking the "write" method call, the list of services will be recorded into that tag's memories.

V. SIMULATION

In order to prove that the IMA is able to address the integration issue between members in the supply chain group, the authors created a prototype of the IMA that had been tested in simulated environments. Four different simulation settings were designed to compare the performance of the IMA with three other related systems. The first system has the ability to detect ongoing supply chain activities but has no control over the pre-planned process once it has been carried out. Conversely, the second system has the ability to manipulate the flow of the current activities after it has been executed but not able to continuously monitor their progress. For this reason, a three-day delay was introduced before the current process can be restructured for this setting. The third system has the same capabilities as the second system but with a longer delay time of five days.

In each of the simulated settings, two types of disruptions were inflicted in the current process: transportation delay and misplaced items. The purpose of designing these disruptions was to assess the capabilities of the four different systems at the early and final stages of the process. The processes undertaken in each of these simulation settings were also defined. Three types of customization processes were designed to produce IT solutions for the client, namely, basic, focused and full-fledged IT solutions. For each type of customization process, two different process tracks for the creation of IT solutions correspond to two different server capabilities. The differences between the high and low end servers lie with the CPU multiplicity, the CPU speed, the power usage, the storage memories and better controller for reading/writing data.

The authors created two servers to run the three main components of IMA (RFID service provider, system communicator and the service handler) on the subsystems. These servers were installed on every member in the group composed of the manufacturer, main service integrator, specialized developer for supplier management module, specialized developer for customer management module, the client and third party logistics. Based on this, 12 servers were needed plus two more as the central server and central repository. The central server housed the service integrator, case handler and information handler; whereas the central repository was used to record the events generated from the six main components of the IMA.

VI. RESULTS AND ANALYSIS

The number of processes completed by the IMA and the three other systems were not much different when there are items misplaced in customization activities. This type of disruption occurred at the final stage of the process and only one remaining process had to be carried out in order to complete the customization activities. For this reason, there was not much difference in the performance of these systems when the disruption occurred at the final stage of the customization activities. Figure 8 demonstrates entirely contrasting pictures when the type of disruption occurs at an early stage of the customization activities. In this case, there were delays in the transportation to deliver the servers to the right service providers for customized IT solutions.

The number of processes completed by system 1 is one task for the all types of customization processes. This is because it is unable to make different arrangements to the current activities and waits patiently for the servers to arrive at the destination. As for system 2 with a three-day delay, the number of completed processes is the same for basic and focus customization type. However, as the number of tasks involved in the full-fledged increases, and proportionately, the safety buffer (in days) decreases. It was not able to complete the last two tasks due to the delay in detecting the deviation. On the other hand, five days delay is too long for system 2 to react within the safety buffer given to the supply chain group and therefore it only managed to complete only half of the tasks.

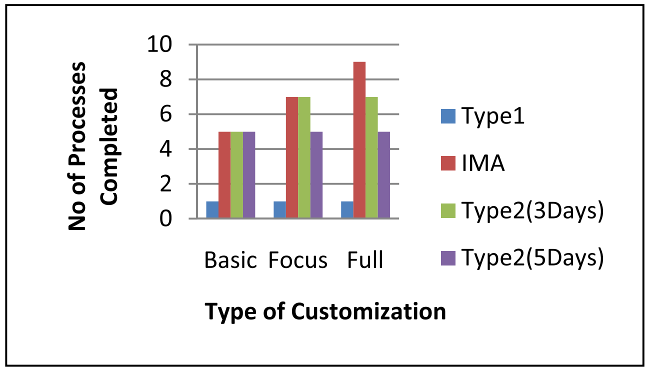

Figure 8. Delay: Number of Completed Processes.

Figure 9 demonstrates the percentage of outperformed tasks by IMA over the other three systems. The performance of the IMA is better compared to the other systems as it moved to full-fledged type of customization. The IMA completed more tasks between the ranges of 80 to 89 percent compared to system 1 as the customization activities increased. Conversely, system 2 with three days delay is doing well in the basic and focus customization types but as the number of customization activities increased in full-fledged, the IMA performed better by 20 percent. As for system 2 with five days delay, the IMA outperformed it in the earlier stage, namely in the focus stage by 29 percent and peaked to 44 percent by the time it reached the full-fledged stage.

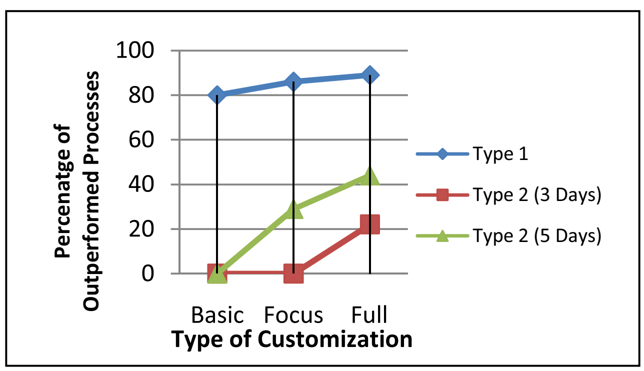

Figure 9. Delay: The Percentage of Outperformed Processes by IMA.

The Mann-Whitney test was employed to prove statistically that the IMA had performed significantly better than system 2 with three days delay. The two samples were based on only the recorded number of tasks completed in full-fledged customization process. Figure 10 demonstrates the average rank and test statistic between the IMA support and system 2 with three days delay respectively. Their average ranks were 30.5 and 10.5 respectively with the IMA having a higher value. This showed that with IMA support, more tasks were completed. In addition, the small p value which is less than 0.05 statistically proves that they are significantly different from each other.

	Systems	N	Mean Rank	Sum of Ranks
Tasks	3.00	20	10.50	210.00
	4.00	20	30.50	610.00
	Total	40		

(a)

	Tasks
Mann-Whitney U	.000
Wilcoxon W	210.000
Z	-6.245
Asymp. Sig. (2-tailed)	.000
Exact Sig. [2*(1-tailed Sig.)]	.000[b]

(b)

Figure 10. Mann-Whitney U Test between IMA and System 2 (3 Days Delay): (a) Average Rank (b) Test Statistic.

VII. CONCLUSION

The customization processes require good coordination between the relevant members in the complex network of the supply chain. The IMA provides an intermediary process that works in the background to support the members to overcome unknown deviations that could halt the current activities. The abilities of the RFID service provider and service handler to continuously monitor and detect deviation across the group respectively allowed immediate reaction. This is a very important ability as we have observed from the support of system 2. As the delay in days to react to the known deviations increased, the performance deteriorated in the completion of the number of tasks.

Prompt reaction alone is useless without having the ability to restructure the current process and overcoming the deviations in order to complete the objectives of the previous process. The authors observed the importance of this ability from the performance of system 1. For this reason, the IMA which comprises the system communicator wraps [11] the functionalities from the existing systems and allows it to have more flexibility in the process by adding and replacing services. However, the service integrator is responsible in managing the change process by employing the case handler and information handler for case recommendation and validation respectively before allowing the IMA to execute the new processes.

With these abilities, we believed that other supply chain activities can be supported to increase the integration between the members in the group and together overcome the known deviations.

REFERENCES

[1] J.C. Cheng, K.H. Law, H. Bjornsson, A. Jones, R. Sriram, "A service oriented framework for construction supply chain integration", Automation in Construction, vol. 19, no. 2, 2010, pp. 245-260

[2] H.K.H. Chow, K.L. Choy, and K.C. Lau, "Design of a RFID case-based resource management system for warehouse operations", Expert Systems with Applications, vol. 30, no. 4, 2006, pp. 561-576.

[3] H.K. Chow, K. Choy, and W. Lee, "A dynamic logistics process knowledge-based system: An RFID multi-agent approach", Knowledge-Based Systems, vol. 20, no. 4, 2007, pp. 357-372.

[4] D. Delen, B.C. Hardgrave, and R. Sharda, "RFID for better supply chain management through enhanced information visibility", Production and Operation Management, vol. 16, no. 5, 2007, pp. 613-624

[5] L. Gadde, and K. Hulthen, "Improving logistics outsourcing through increasing buyer-provider interaction", Industrial Marketing Management, vol. 38, no. 6, 2008, pp. 633-640.

[6] T. Kelepouris, K. Pramatari, and G. Doukidis, "RFID-enabled traceability in the food supply Chain", Industrial Management and Data Systems, vol. 107, no. 2, 2007, pp. 183-200.

[7] R.F. Lusch, S.L. Vargo and M. O'Brien, "Competing through service: Insight from service-dominant logic", Journal of Retailing, vol. 1, no. 83, 2007, pp. 5-18

[8] Y. Rekik, "Inventory inaccuracies in the wholesale supply chain", International Journal of Production Economics, vol. 133, No. 1, 2010, pp. 172-181.

[9] L. Ribeiro, J. Barata, and A. Colombo, "Supporting agile supply chains using a service-oriented shop floor", Engineering Applications of Artificial Intelligence, vol. 22, no. 6, 2009, pp. 950-960

[10] J. Soroor, M.J. Tarokh, and A. Shemshadi, "Initiating a state of the art system for real-time supply chain coordination", European Journal of Operational Research, vol. 196, no. 2, 2009, pp. 635-650.

[11] C. Tarantilis, C. Kiranoudis, and N. Theodorakopoulos, "A web-based ERP system for business services and supply chain management: Application to real-world process scheduling", European Journal of Operational Research, vol. 187, no. 3, 2008, pp. 1310-1326.

[12] Y.Y. Yusuf, A. Gunasekaran, E.O. Adeleye, and K. Sivayoganathan, "Agile supply chain capabilities: Determinants of competitive objectives", European Journal of Operational Research, vol. 159, no. 2, 2004, pp. 379-392.