UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Federated Information Management for virtual enterprises

Garita Rodriguez, C.O.

Publication date
2001

Link to publication

Citation for published version (APA):
Garita Rodriguez, C. O. (2001). *Federated Information Management for virtual enterprises*.

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:27 Jun 2022

Appendix C

Description of DIMS Implemented Services

This appendix contains a brief description of the DIMS services that were developed to support the PRODNET VE infrastructure, as described in Chapter 4. For a more detailed description of the service implementations, please see [76].

C.1 Support Services for LCM

Service DeleteDataElements: the DeleteDataElements service is requested in order to remove data items that have been stored in the DIMS database, along different workflow executions. The data items include for instance, purchase orders, blobs, and attached file names. This service is invoked at the end of the workflow, once the data items are no longer needed for the VCL modules involved in a given workflow.

Service AttachFilesRef: this service is requested by LCM when executing the workflow to send an EDI CONDRA message. The purpose of the AttachFilesRef service is to attach the physical file names to the CONDRA message that was generated by EDI, so that the PCI module is able to send the CONDRA message as a normal VCL message together with the attached files.

Service GetDBP: LCM uses this service, in order to access the general information of the virtual enterprise associated with a given distributed business process.

C.2 Support Services for DBPMS

Service GetRequestedOrderTree: the DBPMS module needs to ask distributed queries to the DIMS to gather information that is spread over the network, so that it can monitor the status and performance of VE nodes. This service is used by

DBPMS to get the data key of the most up-to-date information about orders in a tree structure, composed of the Requested Order, Requested Items, Production Orders, and VE Intra information (see Chapter 4).

Service GetRequestedOrder: this service is used by DBPMS to get the data key of the most recently updated information of requested order from the local or remote node whose identification is given in the input parameters.

Service GetRequestedItem: this service is used by DBPMS to get the most recently updated information of a requested item from the local or remote node whose identification is given in the input parameters.

Service GetProductionOrder: this service is used by DBPMS to get the data key of the most recently updated production order information from the local or remote node whose identification is given as input parameters.

Service GetResultRequestedOrderTree: this service receives the data key retrieved using the GetRequestedOrderTree service, in order to get the actual requested order tree data from DIMS (see Chapter 4).

Service GetResultRequestedOrder: this service receives the data key retrieved using GetRequestedOrder, in order to get the actual requested order data from DIMS.

Service GetResultRequestedItem: this services receives the data key retrieved using GetRequestedItem in order to get the actual requested item data from DIMS.

Service GetResultProductionOrder: this services receives the data key retrieved using the GetProductionOrder service, in order to get the actual production order data from DIMS.

Service GetDBPMSNotification: legacy systems of the VE node may need to inform the DBPMS when they generate local production information of the VE orders, so that DBPMS knows that the information is available for a specific order of this specific partner. This service is used by DBPMS to read the DBPMS notification from DIMS so that it can get the information about the available order.

Service PutSupClause: DBPMS uses this service to store the information about a supervision clause. This information includes the supervision clause itself and the information of the requested order, requested items and the production orders.

Service GetResultDpsSupClause: this service is used by DBPMS to access the information of the partner and the supervision clause associated with it.

Service GetDPId: DBPMS uses this service to get a list of all the partners that are participating in the specified virtual enterprise.

Service GetDBPId: this service is used by DBPMS, in order to get a list of all the virtual enterprises in which the partner is participating.

C.3 Support Services for PPC Module

Service PutVclOrder: this service is used by PPC to temporarily store the purchase order information in the DIMS database before it is sent out to another enterprise. This information is actually composed of one order definition connected with several order lines, and it is deleted from the DIMS later by another DIMS service when the sending order workflow finishes.

Service GetVclOrder: this service is used by PPC to read the purchase order information, which is received from another enterprise. This received data is kept temporarily in a DIMS table until the receiving order workflow finishes.

Service GetPPCMessageContent: this generic service is used by PPC to read any message information received from another enterprise. For instance, in the receiving delivery record workflow, PPC reads the message blob from DIMS and transforms it into delivery record structure. Also, when receiving an order acceptance, the transformation will be done into an order acceptance structure.

Service PutPPCMessageContent: this generic service is used by PPC to store any VCL message information, which will be sent out to another enterprise. Similarly to the GetPPCMessageContent service, this service is used for instance, when sending a delivery record and when sending an order acceptance to another VE partner.

Service PutDBPMSNotification: when the PPC generates local production information about VE orders, it informs the VE coordinator. Therefore, this service is used by PPC to store the DBPMS notification structure into DIMS and to determine the location of VE coordinator.

Service PutReqOrdTree: the DIMS services for PPC regarding the storage of DBPMS input data are invoked by PPC when it needs to push up-to-date data for DBPMS. For instance, this particular service aims at inserting the requested production order tree into the DIMS database (see Chapter 4).

Service PutRequestedOrder: this service aims at inserting a specific Requested Order into DIMS.

Service PutRequestedItem: this service aims at inserting one specific requested item into DIMS.

Service PutProductionOrder: this service aims at inserting one specific production order into the DIMS database.

C.4 Support Services for EDI Component

The services described in this section were actually implemented using database stored procedures and functions defined in a DIMS package (see [76] for more details).

Service PutOrder: this service is used by the EDI module to store purchase order data into DIMS so that PPC can read the order. This service is called after EDI interprets the order information contained in the EDI message from another enterprise.

Service PutOrderLines: this service is used by EDI to store purchase order lines data into DIMS so that PPC can read the order lines. This service is called after EDI stores the order in the DIMS and gets the DIMS data key for it.

Service GetOrder: this service is used by EDI to read purchase order data from DIMS so that it can process the data and build an EDI message. After this service invocation, the EDI message is sent to other enterprise.

Service GetOrderLines: this service is used by EDI to read purchase order lines from DIMS after it gets the order with the GetOrder service.

Service FetchOrderLine: this service is used by EDI to fetch purchase order lines one by one from the cursor given in the GetOrderLines Service.

Service PutBlob: this service is used by EDI to store and EDI message, which will be sent to other enterprise.

Service GetBlob: this service is used by EDI to read EDI message received from other enterprise.

Service GetStepDataExchange: some DIMS services for EDI are related to the exchange of STEP information using CONDRA messages. This service is used by EDI to read the STEP data exchange information that was stored in DIMS by the STEP module. The STEP data exchange information is basically used by EDI to build the main body of the CONDRA message. This information is common to all the files being exchanged (the specific file information is retrieved using GetStepDataElement).

Service PutStepDataExchange: this service is invoked by EDI when a received CONDRA message has been parsed and interpreted, and needs to be stored in a table in DIMS so that the STEP module can read it.

Service GetStepDataElements: a "step data element" represents the information about a specific STEP file that needs to be transferred as part of a "STEP data exchange". There can be several STEP data elements per STEP data exchange. This service is used by EDI in order to read all the file information that was stored by STEP. This information is used in order to build the EDI CONDRA message.

Service PutStepDataElements: this service is called by EDI after parsing the received CONDRA message. When the specific STEP data elements (file descriptions) are decoded from the CONDRA message, they are stored in a DIMS table so that the STEP module is able to retrieve the information.

C.5 Support Services for STEP Component

This section describes the DIMS services that have been developed in order to support the workflows for CONDRA messages exchange among two VCL nodes [76]. The specific DIMS services for the STEP module involved in these workflow definitions are described below.

Service PutStepDataExchange: this service is used by the STEP module in order to store in DIMS the information regarding a "STEP data exchange". This information includes all the data items that are required by EDI in order to create a CONDRA message, such as sender/receiver node ids, STEP file names, etc.

Service GetStepDataExchange: this service is used by the STEP module in order to read the STEP data exchange information (together with the STEP file descriptions) after the EDI module has parsed an incoming CONDRA message.

Service PutAttachedFileNames: this service is called by the STEP module in order to store in DIMS the physical file names of the STEP files that will be sent as attachments by PCI, together with the CONDRA message that EDI will generate.

Service GetAttachedFileNames: this service is used by the STEP module in order to read the local file names of the STEP files attached to the received CONDRA message. The PCI module is in charge of transferring the attached files from one node to another node, and store them in a given directory. The PCI module stores this information into DIMS when saving the VCL message (that contains the embedded CONDRA message). The STEP module needs this information in order to process the actual STEP files that have been transferred to the receiving node.

C.6 Support Services for PCI

Service PutPciMessageContent: this service is used by PCI when it receives an incoming VCL message. The message is then stored in DIMS through this service. VCL messages are basically seen as "blobs" that contain the actual information that will be interpreted by the appropriate VCL module. For instance, the blob may represent an EDIFACT order message that will be parsed by the EDI module. The VCL messages may have attached files as well.

Service GetPciMessageContent: this service is used by PCI in order to read a VCL message that has been stored in DIMS by another module. PCI will send the VCL message to another node according to the specified parameters. For instance, when EDI builds an EDIFACT message to send an EDI order, the EDIFACT message itself will be read by PCI using this service. The message stored in DIMS may contain attached files as well, such as in the case of sending a CONDRA message.

Service PutVEPartner: this service is used by PCI in order to store in DIMS the information of the partner and the communication configuration needs.

Service GetVEPartner: this service is used by PCI in order to read from DIMS the VE Partner general information and its communication configuration information.

Service PutPrivateKey: this service is used by PCI in order to store the private key of the partner that is used as an encryption key value.

Service GetPrivateKey: PCI uses this service to access the value of the private encryption key of the node.

Service PutPublicKey: this service stores the public key of the specified partner. The PCI module uses this public key for data encryption purposes.

Service GetPublicKey: the service gets the public key of the specified node for the encryption algorithm used by the PCI module.

C.7 Support Services for VE Configuration

Service DIMSLoadVEConfiguration: this service is used to load the information of the VE and the VE partners in the corresponding database tables. The data is loaded from an external file that is generated by the VEND tool. In order to load the data, this service uses the Oracle SQL Loader utility.