Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium, Volume 23, No.1, ISSN 2304-1382
ISBN 978-3-901509-91-9, CDROM version, Ed. B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2012
Make Harmony between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals & Proceedings of DAAAM International 2012

SOLUTION FOR REAL TIME DESIGN DATA TRANSPORT FOR THE CAD PARAMETRIC FEATURES

POMAZAN, V[alentina]; SINTEA, S[orin]; BORDEA, N[icolae] & BORDEA, V[asilica]

***Abstract**: Designs that work in many models can be included in parametric parts defined in Autodesk Inventor® as iParts. Parameter values are to be selected from a list, or can be specified, when the iPart are placed in an assembly. There are many situations when, the custom parameters table can be remote accessed and filled-in with data, needed in the real situation. This paper offers a solution for data transportation and import into the appropriate format, so that the designer that builds the digital prototype of a part or assembly has constantly the newest data provided from the remote industrial context (real time in-situ measurements, dimensional modifications, geometric relationships).*
***Keywords:**Parametric solid model/ external parameter/OPC server*

1. INTRODUCTION

Designs that work in many models can be included in parametric parts defined in Autodesk Inventor® as iParts that are part families, driven by tables of geometric data. Each variation is an iPart member, defined by an entrance in the table [1]. One of the benefits of such features is predictable sketches, with fully constrained geometry with dimensions, dimensions driven by parameters and geometric constraints. Parameter values are to be selected from a list, or can be specified, when the iPart are placed in an assembly. One can create custom parameters, for specific members of the part family. Inventor offers a selectable list for custom values of the parameters.

There are many situations when an assembly design requires direct guidance from the assembly site, with specific measurements or types of mounting. In such cases, the custom parameters table can be remote accessed and filled-in with data, needed in the real situation. The designer will be able to provide the correct specifications and the adequate design for the assembly, using data that reflect the customer needs.Most common approach is to create an external table of parameterslinked to an existing part file to make the part a parametric table-driven model. Assigning parameters to existing dimensions will rezise the part when changing the parameters value.

The data can be manually filled-in using a web interface, or sent in specific data packages, via specific connections, to the designer's location, in simple, formats that can be easily read as table entries by the CAD program. One of the issues faced is the reliability of the external information captured from the industrial devices, as the solid model will reflect the assembly, according to specific measurements in real industrial environment.

This paper approaches the external link between the support data table used by the CAD system and the readings provided in a remote location, to be used as base or independent parameters for the solid model.The iPart is altering the width of the complete assembly. In the assembly one part fits into the array that needs to be table driven & linked to the iPart array. The assembly part pattern needs to also incorporate part suppression, so that a different part can be shown in some of the part feature pattern positions instead.

2. IPARTS DEFFINITION

Apart from the geometric parameterization and thorough coordination with external data, one of the important advantages offered by the iPart concept is the color and material control, feature suppression, multiple keys, and the possibility to edit the external table using Excel functions that via Spread Sheet.

When an iPart is created, both named parameters and features that have been suppressed are automatically added to the table. In this paper, an iPart concept is used in order to verify and access external data built from remote readings on a controlled coordinate measurement device.

2.1 The parameters

This type ofpart has to be mounted in an assembly in which the holes, radius of the support and distance between the holes vary.These five parameters are driven by external readings, measured in a different location. The part is represented in figure 1.

Parts created from custom iPart factories can have designated parameters, modified when placed into the digital model. Custom iPart factories are not edited directly. The value for custom parameters can be chosen at the placement of a member from the factory.

For custom parameters, any value can be specified. Supplementaryparameterscan be selected from a list.

The steps of custom iPart definitions are [2]:

- Determine the feature that changes inthe design.
- After the basic structures creation, rename the system parameters with unique names and author the iPart Factory, so that named parameters will be added to the iPart table.
- Decide and make significant changes between members, adding features to the table and specifying the suppression for the features in each row.

- Designate the keys that determine the nesting hierarchy: sizes, specific values, materials
- Verify the documents units and materials properties for current properties (included in the iPart table, even if constant among members).

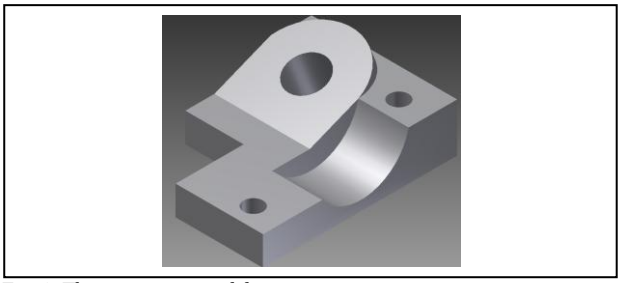

Fig. 1. The parametric model

2.2 The table

The data that drive the iPart are obtained populating the lines in the table, each row corresponding to a different member in the part family. Each column means a significant parameter which will differentiate the geometry of the members (figure 2).

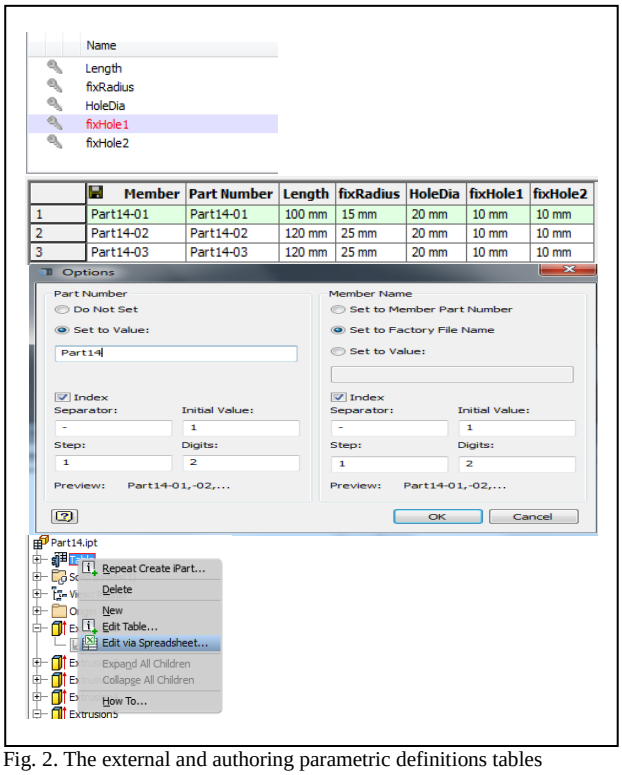

Fig. 2. The external and authoring parametric definitions tables

The table is shown in the first hierarchic position in the Desktop Browser and is available for editing.

More, an iFeature can be saved and reused in other designs [2]. Features dependent on the selected feature are included in the iFeature. In order to facilitate the iFeature extraction, it is useful to create the parameters along with the dimensions creation, with comprehensive names. The next step is to create a table driven iFeature from a table driven iPart. This table can be remotely accessible and filled-in with the same mechanism presented below.

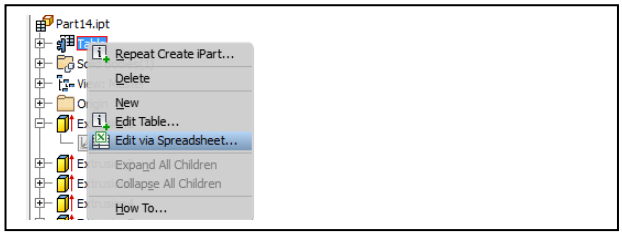

Fig. 3. The iPart hierarchy

3. DATA ACQUISITION

Process data acquisition is usually done with advanced, nonhomogeneous data transfer structures. A process server usually acquires the data from a variety of equipment, in various formats.

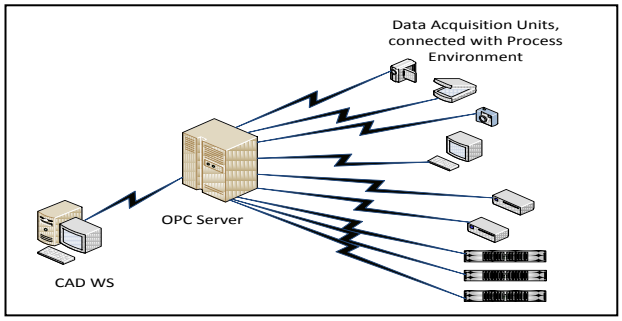

Fig. 4. The data acquisition and usage general scheme

3.1 The OPC server

In order to process this information, the data must be presented under a standard format. This function is performed by OPC servers (OLE for Process Control), an interface that allows software applications to capture dynamic data from industrial devices.

These servers can crop data from different control process equipment connected with the server with various communication networks (Ethernet IEEE 802.3, Wi-Fi, wireless IEEE 802.11, RS232, RS485 etc.), and protocols (Field Bus, Mod Bus etc.).OPC is implemented in server/client pairs. The OPC server is a software program that converts the hardware communication protocol used by a PLC [3] into the OPC protocol. The OPC client software is any program that needs to connect to the hardware, such as an HMI [4].

The OPC client uses the OPC server to get data from or send commands to the hardware.

The OPC is an open standard, which means lower costs for manufacturers and more options for users. Hardware manufacturers need only provide a single OPC server for their devices to communicate with any OPC client. Software producers simply include OPC client capabilities in their products and they become instantly compatible with thousands of hardware devices.

The typical OPC connection scenario is a single server-client connection on a single computer [3, 4].

The devices can primarily process the information and cram it in a data base installed on the OPC server.The OPC protocol uses the concept of an item as a way to structure data. Every item has six required properties: Value, Timestamp, Quality, Access Rights,

Scan Rate, and Canonical Type. DDE (Dynamic Data Exchange) is a well-established mechanism for exchanging data among processes in MS-Windows.The OPC Data Hub maintains an item and all of its optional properties as separate data points. The item's 6 required properties are maintained internally, but the Data Hub displays the information corresponding to the Value, Timestamp, and Quality in the Data Browser under the columns Value, Date and Quality.

There are three DDE commands for establishing communication with Windows programs such as Excel, easy accessible by the CAD systems [4].

These data can be accessed using a standard OLE (Object Linking and Embedding) interface, available on Windows, Linux systems as well as in Java environment.

The OPC Data Hub maintains a queue of old values for all registered points for each client. The depth of this queue is variable. The purpose of queuing old values is to reduce the chance that a data change will be missed during bursts of abnormally high data flow.

The OPC Data Hub buffers data that will be transmitted to a client such that if it knows that more incoming data is available, it will hold off outgoing transmissions until it has a complete data set to send onward to the client. This does not introduce extra latency because the DataHub will only accumulate data destined for a client that arrives together in an incoming message [5].

This buffering greatly increases efficiency by reducing thread context switching and by giving its protocol-specific data transmitters an opportunity to collect more than one data change into a single outgoing message.

TCP/IP connections to the OPC Data Hub can be either ASCII or binary mode. The binary mode is more efficient in both network bandwidth and CPU usage for both the sender and the receiver. Binary mode requires that the CPU architecture of the sender and the receiver agree [6].

3.2 The programmable logic controllers

For this application we used an automation structure consists of two PLCCJ2M-CPU15 (from OMRON) connected,using a FieldBus (IEC 61158) network, with a RS485/RS232 convertor. The convertor links the FieldBus network with the OPC server. The programmable logic controllerPLC1 address is 0x00, while for the PLC2 address is 0x01. PLC is a small industrial computer that measure and controls one or more hardware devices.

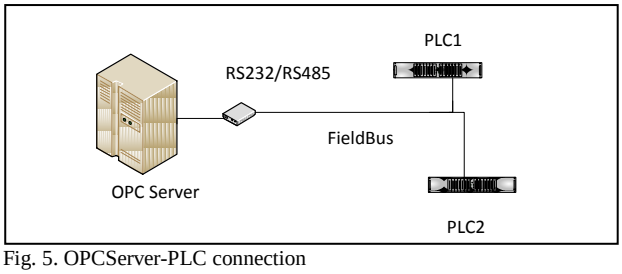

Fig. 5. OPCServer-PLC connection

The process data are brought to PLC by sensors, integrated in the automation and control system supposed to acquire data for the CAD system (coordinate measurement device). Further, the PLCs conduct the data to the OPC server (a “Matrikon Universal PLC”). Using “Matrikon Universal Connectivity Server”, the OLE interface gives access to the monitorized data sent by the PLCs. The server is configured to get these data through FieldBus, using DCOM function (Distributed Component Object Model) CoGetObject.

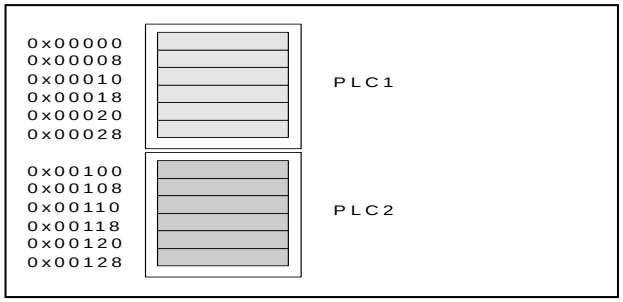

Fig. 6. PLC Data batch

3.3 The interface service

The OPC Server supports both real-time and historical data access to ODBC, MS SQL, MySQL, and Oracle compliant databases. Users may map the OPC point name, value, quality and timestamp by ODBC source table column, custom queries or through pre-configured stored procedures. CAD applications that require connectivity to databases can use this MatrikonOPC Server to access real-time and historical process data that is archived [7].

In order to transfer the data to the CAD system, the server uses a specific implemented service which will drive the process towards the exterior table, linked with the iParts parametric geometry.

The method of connection causes that only one application (only one system process) may be connected to that database; therefore it is much better to connect the service file to the MS SQL or MS SQL Express database server and use the database by many applications [8].

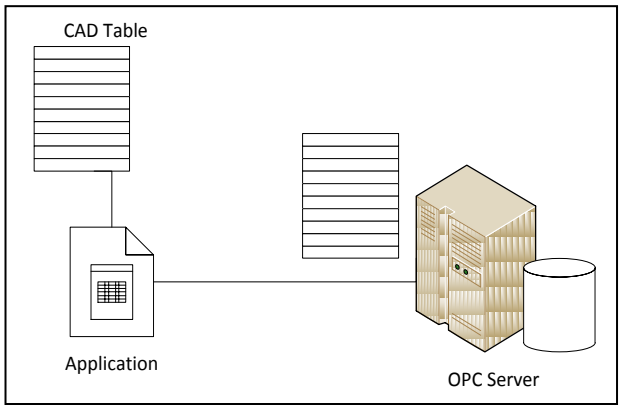

Fig. 7. Structured application for CAD external table

Further, an important aspect of large multi-site implementations is the database [9]. This is usually done using the Database Mail to send email from the SQL Server Database Engine. It is important to monitor

exceeding the specified synchronization time, falling short of processing the specified number of rows in a given amount of time [10].

The algorithm that structures the data in in the external "file"-the table used by the CAD systemis using the structure represented in the figure 8.

```
typedef struct {
        ....
} plc_structure_t;
plc_structure_t plc0, plc1;
Service()
{
try {
                ....
                // read data
                if( CoGetObject( ... ,& plc0, ...))
                        return -1;
                if( CoGetObject( ..., & plc1, ...))
                        return -1;
                // process data from plc0, plc1
                ...
                // write data into table file
                write( file, plc0, sizeof(plc_structure_t));
                write( file, plc1, sizeof(plc_structure_t));
                ...
        }
        catch( Exception e) {
                ...
        }
}
```

Fig. 8. Interface service algorithm

Its main advantage is that the SQL server can stay free of any custom, OPC-related code.

The service file is installed with a sample file of MS SQL Express database, which can easily be used to initiate storing into the database.The following requirements must be met:

- MS SQL database is installed on the test PC. Access to the database is configured so that an integrated MS Windows authentication is used and the selected user has relevant privileges in the database.
- An OPC server is installed on the PC
- The service file is installed on the PC

The application that connects to the SQL server can use traditional data exploitation means and give great flexibility in selecting technologies and application types.

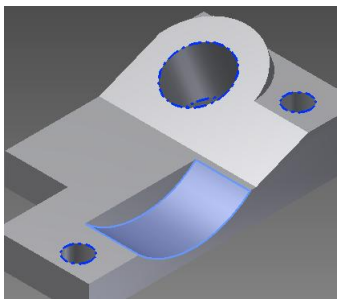

Fig. 9. Modified custom ipart

4. CONCLUSION

The steps to take to control the part using parameters (link an external table, assign parameters to existing dimensions, read/ modify the value of parameters) can be enhanced for remote data acquired from real environment. Each attempt to read the table conducts to modifications in the model geometry. This can be achieved using table driven parts, better families of part, reflecting that the design intent has repetitive content.

A data transportation structure, using OPC server and standard structured network is able to bring the real-time information to an interface application [11], built to structure the data batch according to the format required by the CAD system.

For future research, the new OPC Unified Architecture can be used, as it enhances more the reliability of the real-time data. It focuses on cross-platform interoperability with a much richer set of data and information. OPC UA is intended to provide a platform for secure, reliable interoperability based on Web Services [11].

CAD systems have strong embedded facilities for networking and multi-user design and the perspective to augment the design with more capabilities interacts withina common services-oriented architecture (SOA) environment.

5. REFERENCES

[1] Thom Tremblay, *Autodesk Inventor 2013 and Autodesk Inventor 2013 LT*, Sybex, ISBN 978-1-118-24479-1, John Wiley & Sons, 2012

[2] Rhandi Shih, *Parametric Modeling with Autodesk Inventor 2013*, ISBN 978-1-58503-726-1, SDC Publications, 2012

[3] Nitaigour P. Mahalik, *Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control*, Engineering Online Library, ISBN 9783540401834, Springer, 2003

[4] Deon Reynders, Steve Mackay, Edwin Wright, *Practical Industrial Data Communications: Best Practice Techniques*, ISBN 0 7506 6395 2, Ed. Elsevier, 2005

[5] Stjepan Bogdan, Frank L. Lewis, Zdenko Kovacic, Jose Mireles, *Manufacturing Systems Control Design: A Matrix-based Approach (Advances in Industrial Control)*, Ed. Springer, 2010

[6] Deon Reynders, Steve Mackay, Edwin Wright, *Practical Industrial Data Communications: Best Practice Techniques*, ISBN-10: 0750663952, Elsevier, 2005

[7] Alina Girbea, Septimiu Nechifor, Francisc Sisak, Liviu Perniu, *Efficient address space generation for an OPC UA server*, J. of Software—Practice & Experience archive, Volume 42 Issue 5, May 2012 , pp. 543-557, John Wiley & Sons, Inc. New York, NY, USA, 2012

[8] F. Perez, D. Orive, M. Marcos, E. Estevez, G. Moran, I. Calvo, *Access to process data with OPC-DA using IEC61499 Service Interface Function Blocks*, *Dep. Autom. Control & Syst.*, DOI:10.1109/ETFA.2009.5347024, ETSI of Bilbao, Bilbao, Spain, 10/2009

[9] Jianchun X., Qiling Y et. al., *Design and application of Fieldbus OPC DA Server*,ICCE2011 Proceedings, pp. 337-344, ISBN 978-3-642-25184-9, Ed. Springer, Hilderberg, 2011

[10] Frank Iwanitz, Jrgen Lange, *OPC Fundamentals, Implementation and Application*, ISBN 9380386370, Laxmi Publications Pvt Limited, 2010

[11] Wolfgang Mahnke, Stefan-Helmut Leitner, Matthias Damm, *OPC Unified Architecture*, ISBN-10: 3540688986, Ed. Springer, 2009