Sensor Array Based Distributed Instrument System

http://dx.doi.org/10.3991/ijoe.v10i3.3546

D.P. Zhang and H.L. Zhu
Henan Polytechnic University, Jiaozuo, China

Abstract—**The dynamic integration of sensors to design distributed instrument system is absorbing more and more research focus for its requirements in modern producing enterprises for low-cost products. In this paper, the distributed instrument based on sensor array is proposed, and the data exchange model is discussed which implemented the communication of the modules of the measurement instrument. And then the control mechanism of the data exchange among sensors is studied which manages the information of the measurement components and schedules the data process in the measurement cases. In this way, a group of measurement components and the distributed measurement sensors can be connected together dynamically. Based on the hardware concepts, the instrument components are proposed for data acquisition, data management, data processing etc.. The data exchange among instrument components can be configured according to the communication software bus. The measurement data from the distributed sensor array can propagate among the configured data logical flow ways asynchronously, and users can plug and unplug any components to the online instrument cases. Finally, the proposed application case is present which shows that the proposed way can accelerate the collaboration of production for different enterprise and improve the producing efficiency.**

Index Terms—**components, distributed instrument, sensor array, virtual bus.**

I. INTRODUCTION

Measurement technologies that serve as the important parts of industrial production tend to be updated and changed many times during their lifetime [1]. Object-oriented model has been addressed to describe, analyze, and design distributed measurement system and the object model of measurement domain are presented to support the reuse of component, but the object model tend to be simple without considering the evolution of component and the coupling interactions of components [2-5].

The reconfiguration of the measurement tasks is very important for the evolution of the software platform because the maintenance of which is favored when distinguishing the measurement pattern from the programming details. The platform can carry out the operation to unplug components and reconfigure the needed parts [6,7]. Based on the modern production situation, the running measurement systems must be reconfigured without interruption of their execution to achieve automatic reconfiguration and the process only works in the current application as the result of some computation performed by the platform working mechanism.

To assemble the components together to enjoy the needs of the measurement tasks, the configuration protocols must be used to allow the specification of the measurement system and the definition of instrument component interaction [8-10]. In the given patterns, the instrument components can be divided into interface components, communication components and data processing components, and the platform determines which components will be used in the current measurement tasks.

The novel idea of this paper is put a high premier on sensor array distributed in the web environment, instrument components and virtual bus, which can integrate the measurement instrument components dynamically according to the measurement tasks. Based on the proposed way, all the instrument components can exchange data on the uniform virtual data bus sequence asynchronously. In this way, the measurement tasks can be reconfigured on the executing interfaces which change the measurement structure of the system dynamically and flexibly.

The rest of this paper is organized as follows, Section 2 describes the basic architecture of the architecture of the instrument system; Section 3 describes The Mechanism of the Data Exchange in the Distributed Instrument System; Section 4 discusses Data Exchange Protocol and Implementation; Section 5 gives the implementation and application. Section 6 draws some conclusions.

II. THE ARCHITECTURE OF INSTRUMENT SYSTEM

A. The Mechanism of the Distributed Instrument System

The architecture of the instrument system includes two levels, the first level is called application level which provides users with virtual instrument and operation functions in measurement process, the second level is composed of distributed local area measuring network which include all kinds measuring sensors and data processing components or databases.

The mechanism is implemented as virtual bus patterns. Virtual bus model provides the measurement cases with data exchange channels among system components. With virtual bus, the distributed measurement data can be exchanged in series flexibly and asynchronously, as is shown in Fig. 1. According to the functions, the virtual bus falls into two categories, information bus and data bus. The distributed sensors can collaborate based on the local bus with information bus and data bus, and then come into sensor array.

Virtual bus separates the interaction specifications of measurement services implementation and dynamically couples for the measurement cases, and the measurement system can easily add or release measurement sensors or instrument components at run-time. This makes the collaborative measurement pattern be different from traditionally distributed measurement tasks.

The virtual bus can be classified into global and local virtual bus controllers. The global controller is located at user end, which implements the data exchange among instrument components and virtual bus, and control of all the distributed local virtual bus controllers. When the measurement cases are reconfigured, the instrument modules can be connected to the data bus, and the routing table records the connecting information, in addition, the global controller sends configuring information to all the measurement units to carry out the tasks of connecting to the bus, and the local controllers record the information and refresh routing table of the current processes, and then send back the successful information. If the global controller sends instructions to the measurement objective units, the data can propagate according to the prearranged orders.

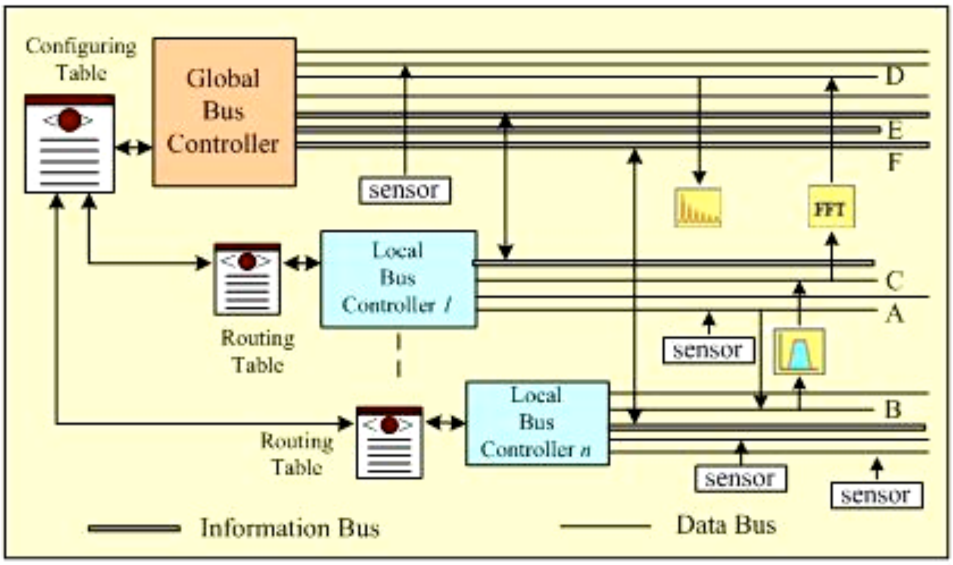


Figure 1. The Control Mechanism of the Data Exchange

B. *Encapsulation of Instrument Components*

The instrument components, as are called software components with some special functions and special terminals for connecting to the data bus and collaborated with other components, as is shown in Fig. 2. All kinds of components make up the instrument component library which is distributed in the local nodes of the instrument system. So the distributed sensors are encapsulated as sensor components with some intelligence according to functions based on information and data exchange protocols.

The instrument components play certain roles in the data flow of the measurement cases. To implement the special function exchange data with other components, the auxiliary functions must be defined to carry communication tasks. The function structure of the instrument component is shown in Fig. 3.

From Fig. 2 and Fig. 3, an instrument component includes main functions, Activities, Trigger, work and a series of attributes. Attributes are used to define the work parameters of the components, such as color, shape, work pattern, address parameters, etc. If the component receives the commands of read data, it will call the relative APIs to read the data from the connecting data bus, and save the data to the data buffer of the component. If the component finished the current task, the trigger event will be start and the data will be wrote on the configured data bus, at the same time, sending the message to the software bus controller. About software component, the work is the key function, which plays the role of data process.

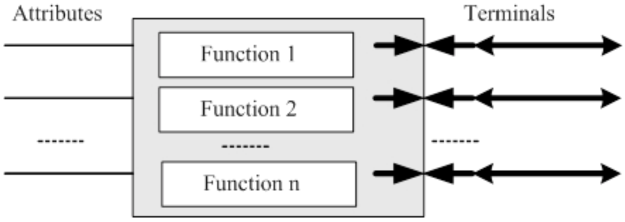


Figure 2. Measurement Components

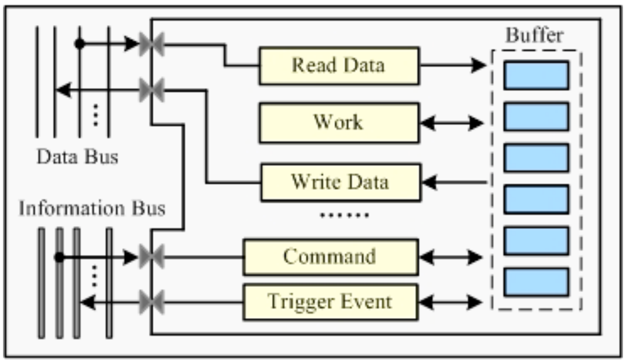
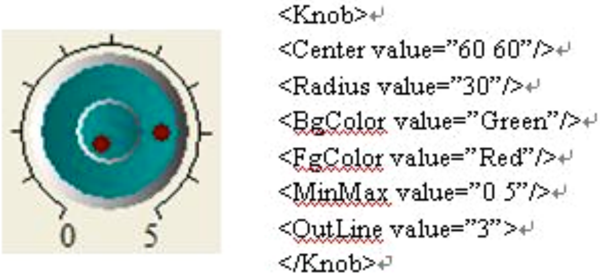


Figure 3. The function structure of instrument component

When the instrument component is scheduled, the instrument platform interprets the attaching language of the related component. For example, a Knob component can be given follow configuration, as is shown in Fig. 4.



```
<Knob>
<Center value=”60 60”/>
<Radius value=”30”/>
<BgColor value=”Green”/>
<FgColor value=”Red”/>
<MinMax value=”0 5”/>
<OutLine value=”3”>
</Knob>
```

Figure 4. The information of the Knob component

The Knob component is used to receive user's operation information which only writes data to data bus. The language of the component includes attributes that are used to define its size, position, color and interface that are used to send user's operation data. According to Fig. 4, the center of the knob is (60, 60), the radius of the knob is 30, the background color of the component is green, and the pin color is red. The range of user's input is (0, 5). And user can send the operation value to the line ID 3.

The instrument components have the properties of self-organizing and self-running. In the given platform, the data bus is designed as a global object that can be access by instrument components which can exchange executing information and data processing information. With measurement components, the interfaces are designed as the API to implement the scheduling commands and assigning the address of the data bus ID to the interface of the component, and then component can be connected to the pipeline. Only the data formats between output interface and input interface of two different components, the components can be linked together by data bus, as is shown in Fig. 4.

In Fig. 5, component A and component B can exchange data in the data flow sequence, so the output interface of component A and the input interface of component B can then plug in the same data bus line(with the same line ID).

The data bus controller exchange data by routing table of the instrument component which is configured with the process creating. When the component finishes the current tasks, the controller checks the routing table to get the configured output address information of the virtual bus, and then packs the data according to the data exchange protocols and then writes the package on the data bus based on the address.

The implementation of the data exchange is specified as informing events and writing events. When the exchange controller gets the information of the data bus, it activates the informing events and then the controller reads the data from data buffer and invokes measurement service. When the data processing task is finished, the controller activates the writing events, and then informs the next service connecting to the bus according to routing table of current component. Specially, there are two kinds of instrument components needing to be introduced, data end components and data acquisition component which encapsulates the distributed sensors. Data acquisition component provides data acquisition services based on the configuration of users, which only writes data on the virtual bus, but the data end component only reads data from data bus to show on the instrument window or save to the databases.

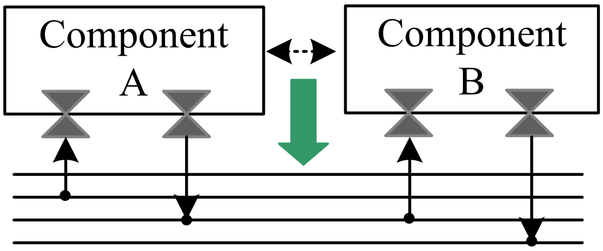


Figure 5. The connection of components

III. THE MECHANISM OF THE DATA EXCHANGE IN THE DISTRIBUTED INSTRUMENT SYSTEM

A. *Sequential Model*

Sequential model carry out the data exchange according the order of the functional components. As is shown in Fig. 6, the symbol of ***P*** is buffer between the components and the symbol of S is data processing components which implements the special function. In this model, the buffer ***P*** implements the data exchange during the measurement process.

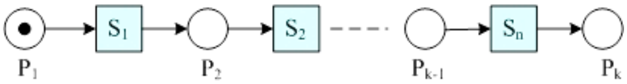


Figure 6. Sequential model for data exchange

B. *Parallel Model*

The measurement process usually many parallel processing to implement the scheduled measurement tasks. The sub-task of the measurement case can always be divided into many processing ways according to the measurement targets, as is shown in Fig. 7.

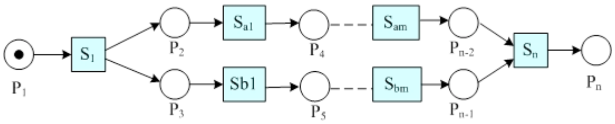
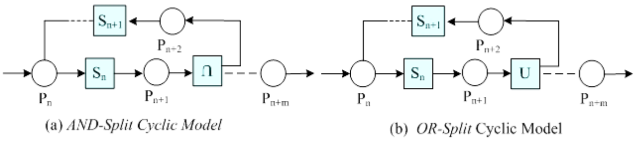


Figure 7. Parallel model for data exchange

C. *Cyclic Model*

Usually, the measurement process needs repeated process based on the results and the input data. This kind of data exchange model is called cyclic model, which can be divided into AND-Split Cyclic Model and OR-Split Cyclic Model. All measurement processes must be executed in the AND-Split model, but the OR-Split model executes the measurement task based on the conditions. This kind of model is shown in Fig. 8.



(a) *AND-Split Cyclic Model*

(b) *OR-Split* Cyclic Model

Figure 8. Cyclic model for data exchange

To all data exchange model, the measurement module and the buffer are designed as components, and the instrument platform can manage all the distributed components or resources and schedule the components according to the measurement tasks.

The distributed instrument system can be viewed as collaborative measurement platform with global uniform protocols for information and data exchange integrating all the distributed sensors dynamically, which include three main sections: I/O interface, data processing and virtual bus, as is shown in Fig. 9. The protocols can be divided into two sections: information protocols and data exchange protocols. The information protocols carry out the operational functions and schedule the data channels among instrument components. The data exchange protocols carry out the data exchange among components which is implemented between component and data bus which will be discussed in the following sections.

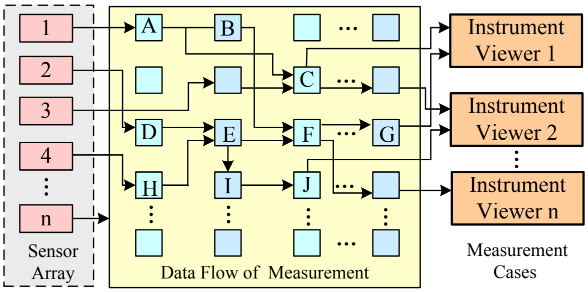


Figure 9. Data exchange patterns of distributed instrument

The I/O interface can implement three kinds of tasks: data acquisition from device or databases, accepting instruction of users to carry out reconfiguration of measurement system, propagating data among current service units to the instrument end.

The data bus can be called as virtual address space, which provides data exchange among current measurement units, and the instrument components of user interface have the uniform virtual address series with web address series. All the data exchanging address of measurement resources are ordered when the resources are integrated on the given measurement platform, so the data can be propagated according to virtual address.

The architecture can carry out configuration of the measurement cases dynamically for the special tasks by definition of interfaces of components.

IV. DATA EXCHANGE PROTOCOL AND IMPLEMENTATION

The type of modern measurement includes online measurement, offline measurement, and the components of data exchange including single value and data array model. If the instrument is designed to analyze the historical data, the data resource will be database or data file.

A. Protocol Format

In order to improve the reliability of data exchange, the data is encapsulated as the protocols according to special format and the data flow. The Format can be expressed in Fig. 10.

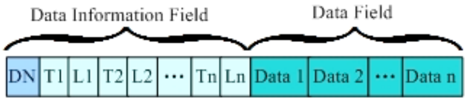


Figure 10. The data exchange package

The data package includes two sections, data information field and data field. Data information field is used to define the data information about the data groups encapsulated in the data package. Data information field include information of all the data or data array, including data type (T), data length (L). The data including in the data field can be single value or data array, as is shown in Table 1.

TABLE I.
DATA STRUCTURE OF THE DATA EXCHANGE PACKAGE

Package Structure	***Information Type***	***Statement***
Number	int	Data number in the package
Data Type1	Defined data type	Type of the first parameter
Data Length1	int	Length of the first parameters
Data Type2	Defined data type	Type of the second parameter
Data Length2	int	Length of the second parameter
…	…	…
Data field 1	Defined data type	The data in data field 1
Data field 2	Defined data type	The data in data field 2
…	…	…
Data field n	Defined data type	The data in data field n

B. Protocol Implementation

The process of measurement is carried out by protocol to implement data exchange. The process includes the whole global virtual instrument platform. The virtual instrument controller must be started and initialized the functions and all the components to connect to the data bus and information bus, and when the working status has been updated, the controller begins monitoring the event of adding the components. If the waiting time is too long, the controller will be closed automatically.

The controller allocates the current component with inputting and outputting address of virtual bus and the sequential code of the identification. The attributes of the component need to be configured according to the measurement tasks of the instrument. If the configuration has been finished, the initial state of the component is waiting the command and data.

According to the measurement tasks, if there are enough components to carry out the preset tasks. To implement the measurement target, the data source components will be started, and the subsequent components will read the data from the assigned data bus address according to the logical analysis flow of the measurement tasks. The virtual bus will monitor all the current components of the measurement cases, and scheduling the components based on the functions. The software bus controller can delete any component on the basis of the measurement tasks.

V. IMPLEMENTATION AND APPLICATION

To better highlight the obtainable improvements with the proposed sensor array based distributed instrument system for the manufacturing production of the large scale equipment. The application scene is described, which carries out the collaborative measurement and scheduling the data flow among geographically distributed measurement targets in the distributed measurement environment, as is shown in Fig. 11.

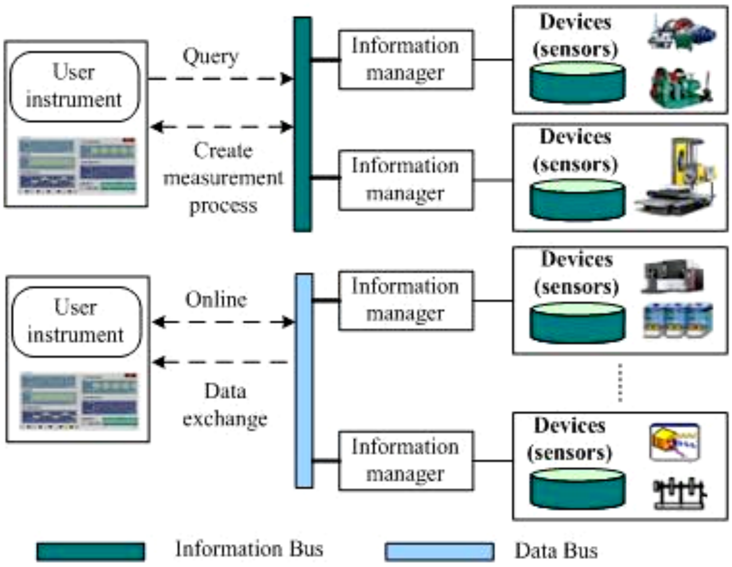


Figure 11. The measurement application

In the given example, all the measurement objects which are distributed in the producing environment are managed by the application software being called information manager. All the sensors and databases are configured in the manager platform. When users use the measurement client to carry out the measurement tasks, the Query information will be sent to the relative information manager. The manager creates a measurement process to enjoy the measurement requirements and configures the relative measurement components, and then users can reconfigure the measurement instrument dynamically according to the device information. During the measurement process, the user-end platform keeps the state of "Online", and then all the relative devices, data processing modules and data resources can exchange data with user-end dynamically.

VI. CONCLUSIONS

In this paper, we presented the distributed instrument design methods for the dynamic integration measurement, which uses distributed sensor array, instrument components and virtual bus, etc to configure measurement cases dynamically. The proposed the method could solve the measurement problems of online configuring operation without stop the current measurement tasks.

In this paper, we have discussed all kinds of traditional way in data exchange of modern modularized measurement instrument. For this object, we introduced the structure of measurement components and its definition. The virtual bus is proposed which shield the IP address in the distributed environment and memory address in the user terminal by using uniform address lists.

Based on the given methods, all the measurement components can carry out data exchange and all kinds of instrument functions were encapsulated as measurement functional components, and the functions could be configured on the data bus by the measurement reconfiguration technologies.

The required measurement case was carried out from prearranged data flow sequences, and the data could be transferred asynchronously like the propagation of the optical waves, which improved the efficiency and flexibility of the measurement tasks.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China (Grant No.50275061) and the National High Technology Research and Development Program (Grant No. 2008AA04Z133).

REFERENCES

[1] P. Spicer and H. J. Carlo, "Integrating reconfiguration cost into the design of multi-period scalable reconfigurable manufacturing systems," *J. Manuf. Sci. Eng.-Trans. ASME*, vol. 129, no. 1, pp. 202-210, Feb 2007.

[2] Q. P. Yang and C. Butler, "An object-oriented model of measurement systems," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 1, pp. 104-107, Feb 1998. http://dx.doi.org/10.1109/19.728800

[3] S. Kohout, J. Roos and H. Keller, "Automated operation of a homemade torque magnetometer using LabVIEW," *Meas. Sci. Technol.*, vol. 16, no. 11, pp. 2240-2246, Nov 2005. http://dx.doi.org/10.1088/0957-0233/16/11/015

[4] D. Hyun and J. Kim, "Study of external humidification method in proton exchange membrane fuel cell," *J. Power Sources*, vol. 126, no. 1-2, pp. 98-103, Feb 2004. http://dx.doi.org/10.1016/j.jpowsour.2003.08.041

[5] W. Kozaczynski and G. Booch, "Component-based software engineering," *IEEE Softw.*, vol. 15, no. 5, pp. 34-36, Sep-Oct 1998. http://dx.doi.org/10.1109/MS.1998.714621

[6] L. S. He and D. P. Zhang, "XVIML: an extensible virtual instrument markup language," in *AUTOTESTCON 2005, IEEE System Readiness Technology Conf.*, Orlando, USA, Sep 2005, pp. 36-42.

[7] D. N. Gray, J. Hotchkiss, S. LaForge, A. Shalit and T. Weinberg, "Modern languages and Microsoft's Component Object Model," *Commun. ACM*, vol. 41, no. 5, pp. 55-65, May 1998. http://dx.doi.org/10.1145/274946.274957

[8] I. Crnkovic and M. Larsson, "Challenges of component-based development," *J. Syst. Softw.*, vol. 61, no. 3, pp. 201-212, Apr 2002. http://dx.doi.org/10.1016/S0164-1212(01)00148-0

[9] D. B. Stewart, R. A. Volpe and P. K. Khosla, "Design of dynamically reconfigurable real-time software using port-based objects," *IEEE Trans. Softw. Eng.*, vol. 23, no. 12, pp. 759-776, Dec 1997. http://dx.doi.org/10.1109/32.637390

[10] M. Moallem, "Design and implementation of computer control software," *IEEE Control Syst. Mag.*, vol. 25, no. 1, pp. 26-29, Feb 2005. http://dx.doi.org/10.1109/MCS.2005.1388796

AUTHORS

D. P. Zhang is with the School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo 454000, Henan Province, China (e-mail: albert_12@ 126.com).

H. L. Zhu is with the School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo 454000, Henan Province, China (e-mail: wfzhuli@ hpu.edu.cn).

This work was supported in part by the National Science Foundation of China (Grant No. 50275061) and the National High Technology Research and Development Program (Grant No. 2008AA04Z133). Submitted 07 February 2014. Published as re-submitted by the authors 28 April 2014.