AUTOMATED CONFIGURATION AND VALIDATION OF INSTRUMENTATION NETWORKS

Timothy Darr, Ronald Fernandes, Michael Graul, John Hamilton
Knowledge Based Systems, Inc
1408 University Drive East
College Station, TX 77840

Charles H. Jones
812TSS/ENTI
Edwards AFB, CA

ABSTRACT

This paper describes the design and implementation of a test instrumentation network configuration and verification system. Given a multivendor instrument part catalog that contains sensor, actuator, transducer and other instrument data; user requirements (including desired measurement functions) and technical specifications; the instrumentation network configurator will select and connect instruments from the catalog that meet the requirements and technical specifications. The instrumentation network configurator will enable the goal of mixing and matching hardware from multiple vendors to develop robust solutions and to reduce the total cost of ownership for creating and maintaining test instrumentation networks.

KEYWORDS

Instrumentation configuration, IHAL, sensor networks, instrumentation system, telemetry system, range applications, XML applications.

INTRODUCTION

This paper presents preliminary results for the development of a Java-based test instrumentation network configuration and verification application. Given an instrumentation part catalog, a high-level functional description of the desired systems, in terms of required measurements, this application will select and connect the necessary instruments. The output of the configurator is a list of instruments and instrument connection list.

In conjunction with the Instrumentation Hardware Abstraction Language (IHAL) [1], this configurator will re-engineer how instrumentation and flight test engineers perform various design, validation, development, verification, and management activities for aircraft T&E instrumentation subsystems. This configurator provides the ability to automatically configure instrumentation networks across applications and hardware systems. In the absence of representations, such as IHAL and configurator applications such as the one described in this paper, military and commercial enterprises will continue to invest increasing amounts of time, resources, and technology to manually configure networks or to custom develop brittle configuration and verification applications that are hostage to a limited number of vendors' hardware offerings [2,3].

INSTRUMENTATION NETWORK CONFIGURATION

The instrumentation network configuration problem consists of at least two subproblems that will be the focus of this paper: *synthesis* and *verification* [4]. In synthesis, an instrumentation network is created to satisfy user requirements and domain constraints. In verification, an existing instrumentation network is evaluated against user requirements and domain constraints.

The synthesis subproblem can be stated as follows:

- Given:
  - An instrumentation catalog, consisting of sensors, actuators, transducers, data acquisition units, signal conditioners, etc.
  - A set of user requirements in terms of desired functions and functional characteristics.
  - A set of domain constraints such as instrument port-to-cable compatibility, electrical input and output characteristics (input voltage and impedance, etc.).

- Find:
  - A set of instruments that satisfy the user requirements and domain constraints.
  - The connections among the instruments that do not violate the user requirements or domain constraints.

The verification subproblem can be stated as follows:

- Given:
  - A fully configured instrumentation system.
  - A set of user requirements as defined in the synthesis subproblem.
  - A set of domain constraints as defined in the synthesis subproblem.

- Find:
  - A Boolean value (true / false) that indicates if the configured instrumentation systems satisfies the user requirements and domain constraints.
  - A list of constraints that are not satisfied by the configured instrumentation system.

The instrumentation configuration system described in the remainder of this paper can solve both the synthesis and verification subproblems.

INSTRUMENTATION CATALOG REPRESENTATION

This section presents the default instrumentation catalog eXtensible Markup Language (XML) representation. The configurator can be easily adapted to import other representations, such as the IHAL. The catalog consists of a collection of part definitions, one of which is shown in Figure 1. Each part is described by a name ("TR-A") and a recursively collection of function definitions. The top-level function(s) are implicitly the "primary" functions of the part. For the part shown in Figure 1, the primary function is "TRANSDUCER."

Figure 1 – Part Catalog Representation

The configurator application assumes a functional decomposition of the instrumentation system parts. Each part is described by possibly multiple functions, some of which are subfunctions. Each function is described by one or more functional attributes. The configurator application constraints are described in terms of functional attributes.

In the example in Figure 1, the "Port1" function is a subfunction of the top-level "TRANSDUCER" function. This port function is described by the functional attributes `name = TRANSDUCER Port1 Type` and `name = TRANSDUCER Port1 Name`. These attributes are used in the connection logic described below to connect the ports of other parts via cables.

CONFIGURATION CAPABILITIES

There are two key capabilities that must be provided by any instrumentation configuration system: selection and connection. In selection, one or more parts are selected to satisfy one or more user requirements. In connection, one or more selected parts are connected to each other, to a cable, or to a bus, to satisfy domain constraints or one or more user requirements. The connection capability includes logic to verify that the physical port connection points are compatible, the port types are compatible, the electrical characteristics are compatible, etc. This section describes the instrumentation network selection and connection capability.

INSTRUMENT SELECTION

The device selection constraint represents the logic to select a single device to implement a specified function instance. Figure 2 shows an example constraint specification for selecting a device to implement the TRANSDUCER function.

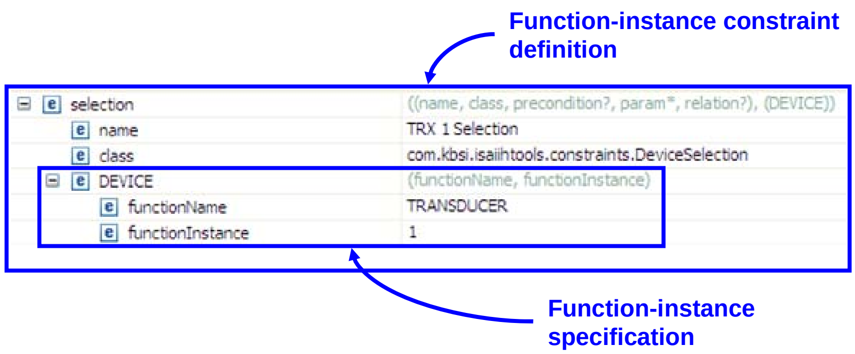

Figure 2 – DeviceSelection Constraint Schema

The constraint is described by a unique name (`name = TRX 1 Selection`) and a Java class that implements the selection logic. This constraint requires a single device specification that includes the name of the function and the function instance. Each functional requirement is specified by the name and a unique instance identifier. This allows the configurator to select instruments to satisfy more than one instance. For example, the user requirements may be such that five accelerometers are needed. Each accelerometer needs to be uniquely identified for connection and reporting purposes.

POINT-TO-POINT CONNECTIONS

The point-to-point connection constraint represents the logic to connect two devices by specifying the function instances and *specific ports* for the source instrument, destination instrument, and the cable to connect the two. Figure 3 shows an example constraint specification to connect a transducer to a Data Acquisition Unit (DAU) via a cable.

The constraint is described by a unique name (`name = TRX to DAU connection`) and a Java class that implements the connection logic. This constraint requires three function-instance port specifications that include the function name and instance and the port name(s). The source (FROM) and destination (TO) devices require a single port name specification. The cable

(MEDIUM) device requires two port name specifications; one to connect to the source device and one to connect to the destination device.

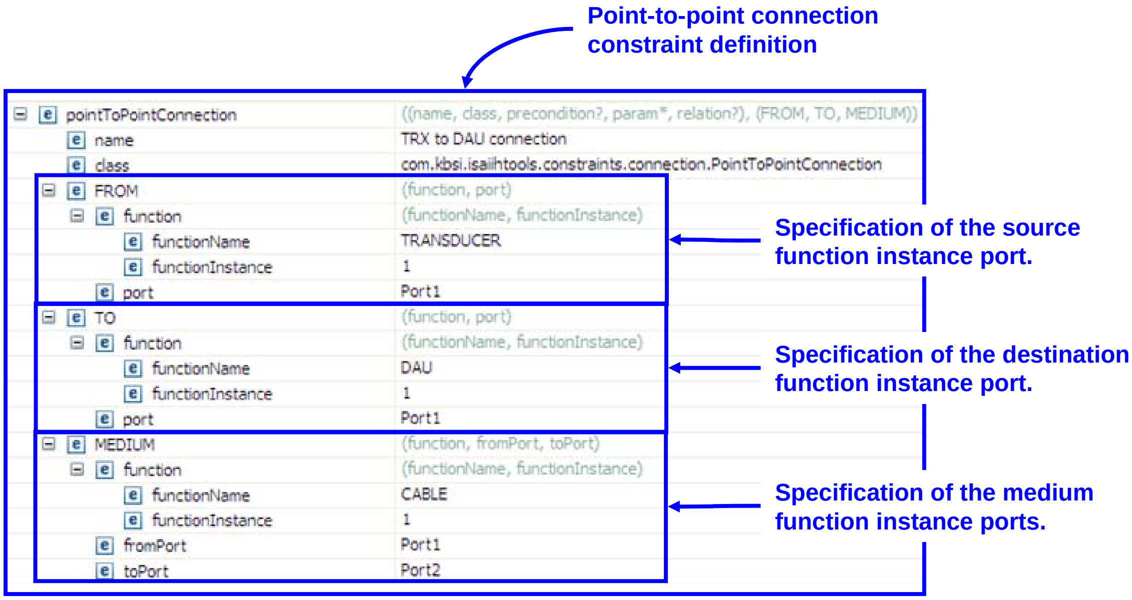

Figure 3 – Point-to-Point Connection Constraint Representation

INSTANCE-TO-INSTANCE CONNECTIONS

The point-to-point connection constraint is not very flexible in that specific ports are specified for the connection points. In some cases, this may be desirable from an engineering or user requirements perspective. In other cases, it is desirable to allow the configurator to determine which ports to use to connect instruments to one another.

The instance-to-instance connection constraint represents the logic to connect two devices by specifying only the function instances for the source instrument, destination instrument, and the cable to connect the two. Figure 4 shows an example constraint specification to connect a transducer to a DAU via a cable. This is a similar constraint as shown in Figure 3 except that the specific ports are not specified.

The constraint is described by a unique name (`name = TRX 1 to DAU 1 via CABLE 1`) and a Java class that implements the connection logic. This constraint requires three function-instance specifications that include the function name and instance. The configurator will automatically determine the ports to connect the source instrument to the destination instrument via the cable.

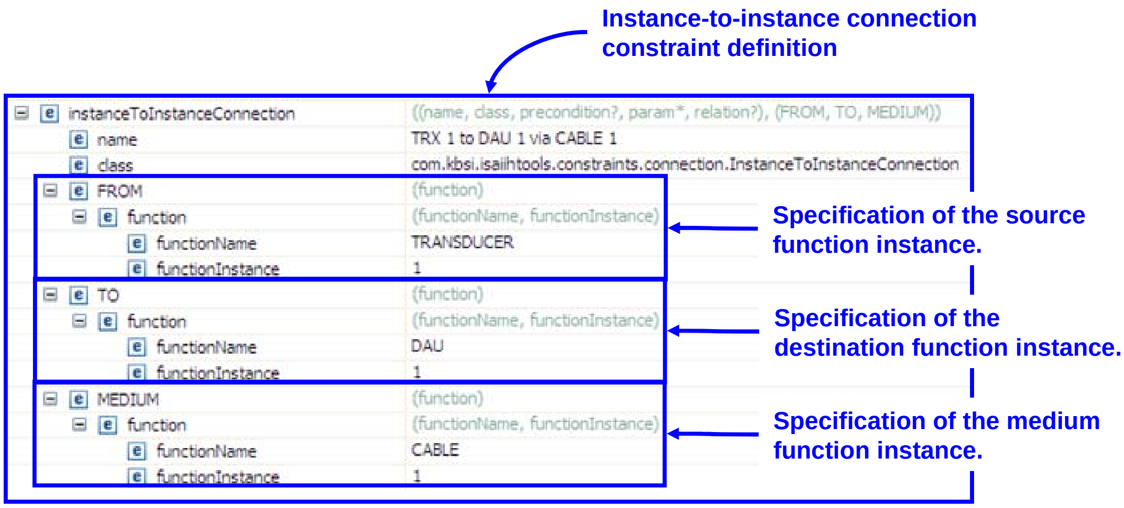

Figure 4 – Instance-to-Instance Connection Constraint Representation

SOLUTION

Figure 5 shows a solution to an instrumentation configuration problem that includes the catalog shown in Figure 1 and the constraints shown in Figure 2 and Figure 4. The solution consists of the results of the connection constraints and the results of the device selection constraints. The connection constraint results list the ports that connect the source device to the cable and the ports that connect the cable to the destination device. The device selection constraint results list the parts that are selected to satisfy the function-instance specifications.

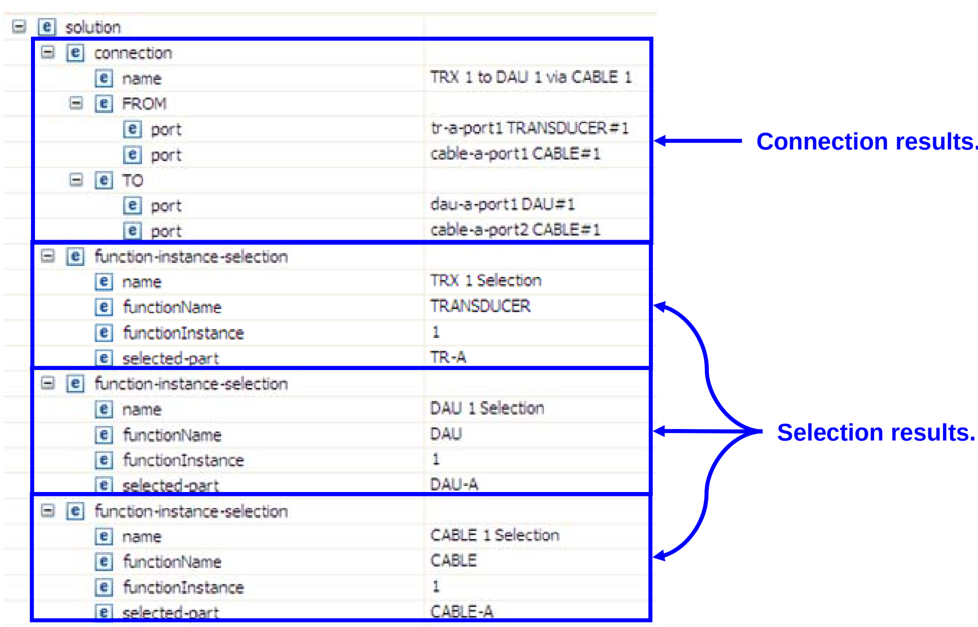

Figure 5 – Solution Representation

COMPUTATIONAL MODEL

The instrumentation network configuration and validation capability described in this paper is based on a constraint-satisfaction (CSP) computational model [5]. A CSP is a computational model that can be used to model and solve a variety of computationally intractable problems. A CSP consists of a set of variables, variable domain values that can be assigned to the variables, and constraints that restrict the assignment of values to variables.

In this computational model, the instrumentation network functions (physical property measurement, signal conditioning, data recording, etc.) map to CSP *variables* and instrumentation network requirements (sensor selection, instrument selection, sensor-to-instrument connection, etc.) map to CSP *constraints*. Using a combination of CSP *inference* and *search* [7, 8], the configurator will rapidly and efficiently converge to a complete and valid instrumentation network.

SUMMARY

This paper has provided an overview of the capabilities of an instrumentation network configurator. This application can perform both synthesis and validation of instrumentation systems, given a set of user requirements, technical specifications, and domain constraints. The configurator accepts input in the form of an XML document that describes the functional characteristics of the instrument part catalog and an XML document that describes the desired functions and connections among the instruments. The output is in the form of an XML document that lists the selected parts and their connections. The configurator is based on a constraint-satisfaction problem computational model, providing both inference and search capabilities to rapidly and efficiently converge to a complete and valid instrumentation network.

ACKNOWLEDGEMENTS

The configurator application described in this paper has been developed by KBSI as part of a Phase I and Phase II Small Business Innovative Research (SBIR) project funded by Edwards Air Force Base.

REFERENCES

[1] Hamilton, J., R. Fernandes, P. Koola, and C. Jones, *An Instrumentation Hardware Abstraction Language*, Proc. International Telemetering Conf., Vol. XXXXII, (2006) Paper 06-10-02, San Diego, California.

[2] Hamilton, J., Fernandes, R., Graul, M. and Jones, C., *IHAL-Based Instrumentation Configuration Management Tools*, Proc. International Telemetering Conf., Vol. XXXXIII, (2007) Paper. 07-12-01.

[3] Hamilton, J., R. Fernandes, M. Graul, and C. Jones, *Applications of a Hardware Specification for Instrumentation Metadata*, Proc. International Telemetering Conf., Vol. XXXXIII, (2007) Paper. 07-12-02.

[4] Darr, T. P., M. Klein, and D. L. McGuinness (eds.), *Configuration Design*, AI EDAM Special Issue: Configuration Design, Vol. 12, No. 4, 1998, pp. 293-294.

[5] Dechter, R., *Constraint Processing*, Morgan Kaufmann, 2003.

[6] Darr, T. P., and W. P. Birmingham, *Part-selection triptych: A Representation, Problem Properties and Problem Definition, and Problem-solving Method*" Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 14, No. 1, 2000, pp. 39-52.

[7] Dechter, R. and P. van Beek, *Local and Global Relational Consistency*, Theoretical Computer Science, 283-308.

[8] Mackworth, A. K. and E. C. Freuder, *The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems*, Artificial Intelligence, Vol. 25.