Integration of CSP and CBR to Compensate for Incompleteness and Incorrectness of Models *

Mohammed H. Sqalli and Eugene C. Freuder
Department of Computer Science
University of New Hampshire
Durham, NH 03824 USA
msqalli,ecf@cs.unh.edu

Abstract

Modeling is very useful in many domains. A model can be very simple such as a mathematical equation, or be very complex. But, models are not always perfect and may not represent all the information about the system. In this paper, we suggest compensating for incompleteness and incorrectness of models by integrating Constraint-Based and Case-Based Reasoning. We model the problem as a *Constraint Satisfaction Problem* (CSP), then *Case-Based Reasoning* (CBR) is used to compensate for what is missing in this model. CBR supports the process of learning by supplying the case-base with new cases that can be used to solve future similar problems. CBR is also used to update the CSP model, and make it more robust for solving more problems. The domain we are using is *InterOperability Testing* of protocols in ATM (Asynchronous Transfer Mode) networks.

Introduction

In this work, we suggest compensating for incompleteness and incorrectness by integrating two modes of reasoning: constraint-based and case-based. The first step is modeling our system as a *Constraint Satisfaction Problem* (CSP). CSP has proven very useful in many applications including diagnosis of protocols, and interoperability testing. We propose to deal with incompleteness and incorrectness by using the expert's knowledge (case-base) of the specific system, and taking into consideration its external interactions and the flaws it may have.

One example of these systems is interoperability testing of protocols. Protocol specifications are written by experts but may have flaws and be incomplete, which may lead to non-interoperability of devices. In addition, if many protocols are running at the same time between two devices, they may cause the wrong behavior of one protocol due to the external interactions with the other.

Models can be incomplete because they represent the behavior of a specific system and may not include all the interactions with the external world. In addition, models may represent systems that are not well-defined or contain flaws (mistakes, bugs, errors, ...), because these systems are the outcome of imperfect human thinking.

*This work is supported in part by the NSF under Grant No. IRI-9504316.

Related Work

(Karamouzis & Feyock 1992) show that the integration of CBR and MBR enhances CBR by the addition of a model that aids the processes of matching, and adaptation; and it enhances MBR by the CBR capacity to contribute new links into the causality model. They talk about the expansion of the available knowledge to the model; thus the idea of an incomplete model was implicitly mentioned in this paper.

In (Huang & Miles 1996), CBR was used to enhance CSP solving in problems characterized by large cardinality, and heavy database searches. In this paper, CBR was mainly used to reduce the search space.

(Bartsch-Sporl 1995) presents a way to bridge CBR and MBR by using SBR (Schema-Based Reasoning). A case is enhanced by adding to it generic knowledge (rules and constraints).

In (Purvis & Pu 1995), case adaptation process in assembly planning problems was formalized as a CSP. Each case is represented as a primitive CSP, and then a CSP algorithm is applied to combine these primitive CSPs into a globally consistent solution for the new problem. CBR is used to fill the values (incompleteness) of the problem, then CSP is used to make the problem consistent.

(Bilgig & Fox 1996) present the case-based retrieval for engineering design as a set of constraints. They state that knowledge, constraints and goals change over time.

In (Portinale & Torasso 1995), it is stated that approaches combining MBR and CBR can be roughly classified into two categories: approaches considering CBR as a speed-up and/or heuristic component for MBR, and approaches viewing CBR as a way to recall past experience in order to account for potential errors in the device model. Their proposal was in the first category by means of the development of ADAPtER, a diagnostic system integrating the model-based inference engine to AID (a pure model-based diagnostic system), with a case-based component intended to provide a guide to the abductive reasoning performed by AID.

In (Someren, Surma, & Torasso 1997), CBR is used as a form of "caching" solved problems to speedup later problem solving. The approach taken is to construct a "cost model" of a system that can be used to predict the effect of changes to the system. The CBR-MBR architecture presented is essentially the one adopted in ADAPtER. They state that in general model-based diagnosis is very expensive from a computational point of view since the search space is very large.

(Lee *et al.* 1997) developed a case and constraint based project planning expert system for apartment domain. This large scale, case based and mixed initiative planning system integrated with intensive constraint-based adaptation utilizes semantic level meta-constraints and human decisions for compensating *incomplete* cases embedding specific planning knowledge. The case and constraint based architecture inherently supports cross-checking cases with constraints during the system development and maintenance.

(Hastings, Branting, & Lockwood 1995) describe a technique for integrating CBR and MBR to predict the behavior of biological systems characterized both by *incomplete* models and *insufficient* empirical data for accurate induction. They suggest the exploitation of multiple, individually incomplete, knowledge sources to get an accurate prediction of the behavior of such systems. They state that precise models exist for the behavior of many simple physical systems. However, models of biological, ecological, and other natural systems are often incomplete, either because a complete state description for such systems cannot be determined or because the number and type of interactions between system elements are poorly understood. In this paper, MBR is mainly used to determine values for variables in cases, and compute new values from old cases' values. MBR is used for the adaptation of cases (MBR is used within the CBR formalism).

In (Marrero, Clarke, & Jha 1997), Model Checking is used for verifying hardware designs, security protocols,... By modeling circuits or protocols as finite-state machines, and examining all possible execution traces, model checking is used to find errors in real world designs. This work uses finite-state machines for representation, which we have shown in (Sqalli & Freuder 1996) to be less expressive than CSPs. The way the model is checked is also different from what we do; we take an instance and check whether it is consistent, while in model checking the whole space is searched to check if there is an inconsistent instance.

Our focus in this paper is compensating for incompleteness and incorrectness. The latter did not get much attention in previous work. First, CSP is used to solve the problem. If the CSP model is incomplete or incorrect, then CBR is used. This way CBR will not be used unless CSP fails. The result obtained from the CBR is then used to update the model. This is similar to what has been done in integrating CBR and MBR to update causality models. The difference is that we are using CSP models, taking advantage of the CSP representation and applying that to compensate for incompleteness and incorrectness.

Modeling

We define a few terms which will be used extensively in this paper:

- CBR: Case-Based Reasoning
- CSP: Constraint Satisfaction Problems
- MBR: Model-Based Reasoning
- RBR: Rule-Based Reasoning

The following tables categorize problem modeling along two axes.

Modeling	**Complete knowledge**
Complex Systems *Methods*	Planning, Scheduling *Advanced models (CSP)*
Simple Systems *Methods*	Elec. circuits, Physical systems *Precise models (MBR)*

Modeling	**Incomplete knowledge**
Complex Systems *Methods*	Natural Systems (biological,...) *CBR, RBR*
Simple Systems *Methods*	Interoperability testing *CBR, MBR, CSP, RBR*

We are mainly interested in incomplete systems. We note some differences between simple and complex incomplete systems.

Complex systems with incomplete knowledge usually have the following characteristics:
- modeling is hard = very expensive,
- predicted behavior,
- general domain knowledge,
- no complete state description,
- interactions between elements poorly understood,
- insufficient empirical data for accurate induction.

Simple systems with incomplete knowledge usually have the following characteristics:
- modeling is easy = less expensive,
- actual behavior,
- system not well defined (people's mistakes, incorrect model),
- complete state description of what is defined or specified,
- interactions with the external world poorly understood,
- insufficient empirical data for accurate induction.

The interoperability problem is one example of simple systems with incomplete and/or incorrect knowledge. We propose to use CSP enhanced with CBR to solve interoperability problems.

A model is *incomplete* if it is missing some knowledge about the system's behavior. This means that this incomplete model will be sufficient to answer questions not involving the missing knowledge. Otherwise, the behavior will be unpredictable; either no answer is possible or a wrong/correct answer is given. A model is *incorrect* if it represents wrong knowledge. This model will be sufficient for and will answer correctly questions that do not involve the incorrect knowledge. Otherwise, the answer given might be wrong. The problem in all these scenarios is that it is hard to know where the missing or the incorrect information is, so it may not be possible to tell whether the answer is correct or not. An example of an incomplete model is a CSP problem where a constraint or a variable is missing. An example of an incorrect model is a CSP problem where a constraint is incorrect.

A model can be incomplete or incorrect because:
- the interactions with the external world are unknown,
- the definition of the system is done by a human being, who can miss or describe wrongly some knowledge.

Integration of CSP and CBR

CSP is a powerful and extensively used artificial intelligence paradigm (Freuder & Mackworth 1992). CSPs involve finding values for problem variables subject to restrictions on which combinations of values are acceptable. A constraint graph is a representation of the CSP where the vertices are variables of the problem, and the edges are constraints between variables. Each variable has labels which are the potential values it can be assigned. CSPs are solved using search (e.g. backtrack) and inference (e.g. arc consistency) methods. CSP representations and methods will be used for modeling our interoperability problem since they provide a powerful tool in this case.

CBR is very useful when there is enough empirical data for accurate induction. It is composed of four main steps: case matching, case retrieval, case adaptation, and case storage. CBR uses a case-base where it stores learned cases. A case is usually composed of a description of a problem, and a solution to it. Whenever there is a new problem, it is matched to what is already in the case-base using similarity metrics. Then, the useful cases are retrieved, adapted to the new problem to provide a solution. The new case (problem and its solution) will be stored in the case-base if it provides new information.

Approach

Although there has been a lot of work done combining CBR and MBR including CBR with CSP, our approach to this integration is novel. We propose to represent our system as a CSP model supported by a case-base to deal with incompleteness and incorrectness.

In figure 1, we show how CBR and MBR are combined to solve these problems. The first step is to model the system as a CSP. Then, the actual behavior of the system is checked against this model. If we can get the answer as to whether the actual behavior matches what is expected by the model, then the result is reported. If the model does not provide enough information to give the answer, then CBR is used. CBR checks if there is a similar case in the case-base. If one or many are found, then they are retrieved and used to solve the new problem. The new case, consisting of the problem and solution, is eventually stored in the case-base. The new solution can also be used to update the CSP model, and make it more adaptable to new situations. In the next section, more explanation will be given on how to combine these two modes of reasoning in a practical problem.

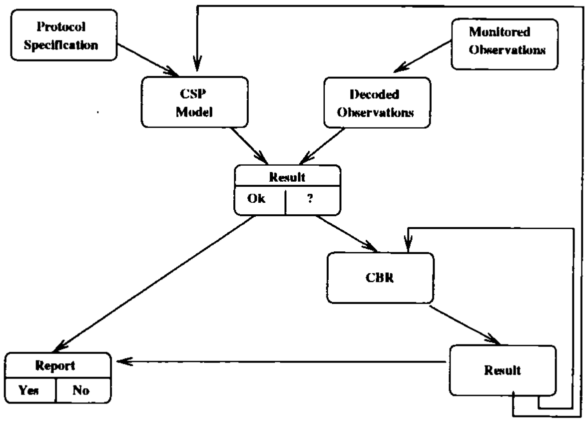

Figure 1. Integration of CSP Model and CBR for InterOperability Testing.

The advantages of our approach can be summarized as follows:

- If we use only CBR then we will need to store many cases. Instead, we choose to reduce the number of cases by using the CSP model. The CSP model represents the core of the system, and CBR adds the missing elements in this model.
- There is no need for CBR use at first but only after CSP fails.
- CSP is enhanced by the CBR results. The effectiveness of the model increases as more problems are solved.
- The representation of cases is done using CSP. This assures uniformity of representation.
- The system is open to new expertise and easily updated. The expert can add cases as needed by the system.
- One case can be used to update different parts of the model. This will assure that the expert is only consulted when CBR fails.

The model will be represented as a set of sub-models. The combination of these sub-models represents the system specification. The advantage of this is to simplify the representation and be able to pinpoint problems at a smaller scale.

InterOperability

InterOperability Testing of the PNNI Protocol in ATM Networks

Asynchronous Transfer Mode (ATM) has emerged as a networking technology capable of supporting all classes of traffic (e.g. voice, video, data). ATM is a connection-oriented technology that uses fixed-size cells, and can guarantee certain quality of service (QoS) requested by the user.

PNNI (Private Network Network Interface) protocol provides dynamic routing, supports QoS, hierarchical routing, and scales to very large networks (ATM 1996). Two switches running PNNI are able to send data to each other either via direct link or by using a route. The PNNI protocol is composed of PNNI routing that includes discovery of the topology of the network and becomes ready to route to different points in the network, and PNNI signaling which is responsible for dynamically establishing, maintaining and clearing ATM connections between two ATM networks or two ATM nodes (ATM 1996). The PNNI routing protocol starts when the link is up. Every switch should send HELLO packets (information about itself) during the Hello Protocol phase.

InterOperability Testing in networks is used to ensure that a device does what it is intended for. It is meant to supplement conformance testing by verifying that the end-to-end behavior of devices is compatible with the protocol specifications. This work is focused on testing protocols that run over ATM networks, and the examples used are taken from the PNNI protocol.

For our purposes, interoperability testing of PNNI allows us to detect any problems that arise when two switches supporting the PNNI protocol are connected. The network can be large with many switches connected. But, for simplicity we propose to work on a two-switch network and perform interoperability testing on them. We suppose that the two switches have passed conformance testing. We base our work on the BTD-TEST-pnni-iop.000.000 document which provides the test suite for performing PNNI interoperability testing (ATM 1998).

Representation

Modeling the entire protocol may be a costly way of approaching this particular problem (Sqalli & Freuder 1996), since the model must include all the information found in the protocol specifications. In addition, interoperability testing is usually presented as a test suite (ATM 1998). A test suite is a collection of tests. Each test provides the mechanisms for testing a particular phase or component of the protocol. We propose a simple way of modeling the protocol by using sub-models where each sub-model represents one test. Tests are written in an incremental way, and some are subsets of others. Running all tests then would be the same as testing the whole protocol.

Example 1: Incompleteness

We are interested in phase one of the PNNI routing protocol (Hello protocol) to demonstrate the advantages of integrating CBR and MBR to interoperability testing. The nature of the problem makes it suitable for our purpose, because either the specification or the implementation can be incomplete or incorrect.

The following is an example of a test (Test Case ID: V4201H_001) from the BTD-TEST-pnni-iop.000.000 document (ATM 1998) that we are going to focus on:

```
Test Case ID: V4201H__001
Test Purpose: Verify that the Hello Protocol
  is running on an operational physical link.
Pre-requisite: Both SUTs are in the
  same lowest level peer group.

Test Configuration: The two SUTs (e.g., ATM
  switches (PNNI capable)) are connected.
Test Set-up: Connect the two SUTs with one
  physical link.
Test Procedure: Monitor the PNNI
  (VPI/VCI=0/18) between SUT A and SUT B.

Verdict Criteria: Hello packets shall be
  observed in both directions on the PNNI.
Consequence of Failure: The PNNI protocol
  can not operate.
```

The following is a model of this test:

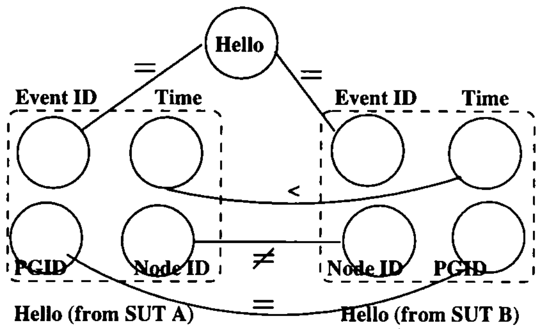

Figure 2. CSP Model of Test Case ID: V4201H_001

These are some of the results we may observe:

Observation 1	Observation 2	Observation 3
(Bad)	*(Good)*	*(Bad)*
Nothing	Hello(A)	Hello(A)
	Hello(B)	Hello(A)
	Hello(B)	Hello(A)

Fact: ILMI (Interim Local Management Interface) should be able to run at the same time as PNNI. This means that ILMI should be part of the model, but it is not because it is a different protocol than PNNI, which is running concurrently. This is an example of incompleteness in the model.

An example of a problem that may happen in this case is the following:
Because of an implementation problem, when PNNI is enabled in both devices, and ILMI is enabled in SUT B an disabled in SUT A, Hello packets are not sent by the device (SUT B). Observation 3 messages are seen.

By checking the model, we can detect the problem but cannot solve it. So we need to ask the expert and get the solution for this particular problem. This problem and its solution will then be stored in the case-base. The expert's solution in this case is: *"Disable or enable ILMI in both devices and do the test again. If it works, then it is an implementation problem that should be reported to the vendor."* This solution can also be used to update the model, so that when a similar problem happens the model will be sufficient to solve it. This will allow us to have a more robust model that has learned from experience. In this example the model will have to include somehow the fact that ILMI should be able to run concurrently with PNNI protocol. This means adding to the model information about a different protocol.

Example 2: Incorrectness

Here, we are interested in the PNNI routing protocol to demonstrate the advantages of integrating CBR and MBR to interoperability testing. In this example we will show how we can deal with an incorrect model.

The following is an example of a test (Test Case ID: V4202H_004) from the BTD-TEST-pnni-iop.000.000 document (ATM 1998). that we are going to focus on:

```
Test Case ID: V4202H__004
Test Purpose: Verify that the first Hello
  sent from both sides contains Remote node
  ID and Remote port ID set to zero.
Pre-requisite: Both SUTs are in
  different lowest level peer groups.

Test Configuration: The two SUTs (e.g., ATM
  switches (PNNI capable)) are connected.
Test Set-up: Connect the two SUTs with one
  physical link.
Test Procedure:
  1- Monitor the PNNI (VPI/VCI=0/18)
     between SUT A and SUT B.

Verdict Criteria: The first Hello
  packet observed from each SUT will have
  the Remote node ID field and Remote
  port ID field set to zero.

Consequence of Failure: The old PNNI
  information was retained causing the
  protocol not to operate.
```

The following is part of the CSP model of this test, representing the time variables and constraints:

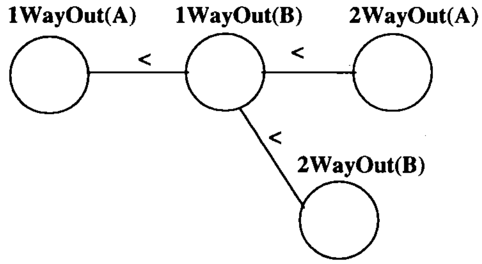

Figure 3. Initial CSP Model of Test Case ID: V4202H_004

These are some of the results we may observe:

Observation 1	Observation 2	Observation 3
(Bad)	*(Good)*	*(Bad?)*
Nothing	1WayOut(A)	1WayOut(A)
	1WayOut(B)	2WayOut(B)
	2WayOut(A)	2WayOut(A)
	2WayOut(B)	

A Hello packet is a 1WayOut if the Remote node ID field and Remote port ID field are set to zero. Otherwise, it is a 2WayOut.

This is the case stated by the expert when we encountered an earlier problem: *If a device receives 1WayIn before sending one, then it can skip sending 1WayIn and send 2WayIn.*

This case is retrieved and reused in this example. The new case we obtain is: *If a device receives 1WayOut before sending one, then it can skip sending 1WayOut and send 2WayOut.*

Using this case, the model becomes as follows:

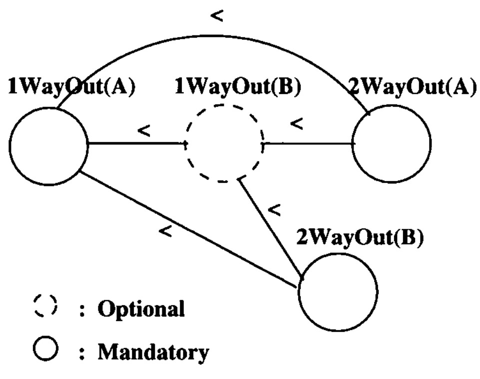

Figure 4. Corrected CSP Model of Test Case ID: V4202H_004

This problem happened because of misinterpretation of the specifications that caused an incorrectness in test V4202H_004 of the test suite.

So, when the model was corrected by taking out one inequality constraint, the two observations 2 and 3 are shown to be correct.

Conclusion

Models can be incomplete or incorrect because they cannot represent all the interactions with the outside world and because systems are not always well-defined. We propose to solve this problem using multimodal reasoning that integrates Constraint-Based and Case-Based Reasoning. Models are constraint-based supported by a case-base where special problems with their solutions are stored for future use. The model is used first, and if it is not sufficient then similar cases are retrieved and CBR is used. The solution obtained can be used to update the model and compensate for its incompleteness and incorrectness.

Acknowledgments

This material is based on work supported by the InterOperability Laboratory (IOL). at the University of New Hampshire, and by the National Science Foundation under Grant No. IRI-9504316. The first author is a Fulbright grantee sponsored by the Moroccan-American Commission for Education and Cultural Exchange. Special thanks to Robert Blais and Scott Valcourt from IOL, and David Cypher from NIST for discussions, comments, and ideas on InterOperability Testing and the PNNI protocol. This work has benefited from discussions with members of the Constraint Computation Center at UNH, and from the ideas suggested by the reviewers.

References

The ATM Forum, Technical Committee. 1996. *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0).* af-pnni-0055.000.

The ATM Forum, Technical Committee. 1998. *Interoperability Tests for PNNI v1.0.* BTD-TEST-pnni-iop.000.000.

Bartsch-Sporl, B. 1995. Towards the Integration of Case-Based, Schema-Based and Model-Based Reasoning for Supporting Complex Design Tasks. In Veloso, M., and Aamodt, A., eds., *Topics in Case Based Reasoning, Proceedings of the First International Conference on Case Based Reasoning, LNAI Series*, 145–156. Springer Verlag.

Bilgig, T., and Fox, M. S. 1996. Constraint-Based Retrieval of Engineering Design Cases: Context as constraints. *Artificial Intelligence in Design* 269–288.

Freuder, E., and Mackworth, A. 1992. Constraint-Based Reasoning, Special Volume. *Artificial Intelligence* 58.

Hastings, J. D.; Branting, L. K.; and Lockwood, J. A. 1995. Case Adaptation Using an Incomplete Causal Model. In Veloso, M., and Aamodt, A., eds., *Topics in Case Based Reasoning, Proceedings of the First International Conference on Case Based Reasoning, LNAI Series*, 181–192. Springer Verlag.

Huang, Y., and Miles, R. 1996. Using Case-Based Techniques to Enhance Constraint Satisfaction Problem Solving. *Applied Artificial Intelligence, an International Journal* 10(4).

Karamouzis, S. T., and Feyock, S. 1992. An Integration of Case-Based and Model-Based Reasoning and its Application to Physical System Faults. In Belli, F., and (Eds.), F. R., eds., *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Lecture Notes in Artificial Intelligence 604*. Springer-Verlag.

Lee, K. J.; Kim, H. W.; Lee, J. K.; Kim, T. H.; Kim, C. G.; Yoon, M. K.; Hwang, E. J.; and Park, H. J. 1997. Case and Constraint Based Apartment Construction Project Planning System: FASTrak-APT. In *Proceedings of IAAI-97.*

Marrero, W.; Clarke, E.; and Jha, S. 1997. Model Checking for Security Protocols. Technical Report CMU-CS-97-139, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA 15213.

Portinale, L., and Torasso, P. 1995. ADAPtER: An Integrated Diagnostic System Combining Case-Based and Abductive Reasoning. In Veloso, M., and Aamodt, A., eds., *Topics in Case Based Reasoning, Proceedings of the First International Conference on Case Based Reasoning, LNAI Series*, 277–288. Springer Verlag.

Purvis, L., and Pu, P. 1995. Adaptation Using Constraint Satisfaction Techniques. In Veloso, M., and Aamodt, A., eds., *Topics in Case Based Reasoning, Proceedings of the First International Conference on Case Based Reasoning, LNAI Series*, 289–300. Springer Verlag.

Someren, M. V.; Surma, J.; and Torasso, P. 1997. A Utility-based Approach to Learning in a Mixed Case-Based and Model-Based Architecture. In *Proceedings of the Second International Conference on Case Based Reasoning.*

Sqalli, M., and Freuder, E. 1996. A Constraint Satisfaction Model for Testing Emulated LANs in ATM Networks. In *Proceedings of Dx-96*, 206–213.