An Application of the Constraint Programming to the Design and Operation of Synthetic Aperture Radars

Michael Holzrichter
Sandia National Laboratories
Albuquerque, USA

***Abstract*— The design and operation of synthetic aperture radars require compatible sets of hundreds of quantities. Compatibility is achieved when these quantities satisfy constraints arising from physics, geometry etc. In the aggregate these quantities and constraints form a logical model of the radar. In practice the logical model is distributed over multiple people, documents and software modules thereby becoming fragmented. Fragmentation gives rise to inconsistencies and errors. The SAR Inference Engine addresses the fragmentation problem by implementing the logical model of a Sandia synthetic aperture radar in a form that is intended to be usable from system design to mission planning to actual operation of the radar. These diverse contexts require extreme flexibility that is achieved by employing the constraint programming paradigm.**

***Index Terms*—synthetic aperture radar, constraint programming, propagation networks.**

I. Introduction

The design and operation of synthetic aperture radars (SARs) requires compatible sets of values for the hundreds of quantities that govern the operation of a radar. Quantities are compatible when they satisfy constraints in the form of quantitative relationships arising from physics, geometry, etc.

The quantities and constraints form a large web. In aggregate, this web is a logical model of the radar. For complex systems, the aggregate is large enough that the model is not tractable to work with *in toto*. The logical model becomes fragmented into numerous partially overlapping pieces distributed over numerous people, documents and software modules. Fragmentation of the logical model gives rise to serious problems. Fragments tend to become inconsistent and incompatible. Resolving inconsistencies is expensive and unresolved inconsistencies can result in errors.

The remainder of this discussion presents an effort underway at Sandia National Laboratories to address the fragmentation problem. This effort is called the SAR Inference Engine and its goal is to implement a common logical model in software that can be used throughout the lifecycle of the radar. Such a model needs to be comprehensive, extremely adaptable and very well validated and verified. The SAR Inference Engine is intended to be usable in diverse contexts ranging from system design, mission planning as well as the actual operation of the radar. The constraint programming paradigm is a key enabler to meeting these goals.

II. Prior Work

The following combination makes this work unique:

- usage of the constraint programming paradigm,
- application in the field of synthetic aperture radar, and
- usage throughout the whole system lifecycle, from design to fielded operation.

Walser [1] discusses applying a propositional satisfiability algorithm to optimizing radar surveillance. The scope of its application was restricted to determining the location of radar installations. Beaumet et al. [2] discuss the usage of the constraint programming paradigm in an operational/embedded context, i.e. computing adjustments to satellite trajectories onboard those satellites. Chenouard et al. [3] discuss the application of constraint satisfaction techniques in the design phase of a complex system – the air conditioning system of an aircraft. The SAR Inference Engine's computational framework consists of a propagation network having much in common with the propagation networks which form the subject of the dissertation of Radul [4].

III. The Constraint Programming Paradigm

Most software follows the imperative programming paradigm. Imperative programs have fixed sets of inputs and outputs and explicitly define the sequences of actions that transform the former into the latter. The declarative programming paradigm (which includes the constraint programming paradigm) shifts the focus from defining control flow to defining the desired result. For constraint programming, the desired result is to satisfy a set of constraints.

In the SAR context, constraints come from physics, geometry, signal processing theory, system design, etc. For example, physics constrains a travelling wave's frequency, wavelength and propagation velocity. Once values are assigned to any two of them, this constraint dictates the value of the third. In this example, there is no *a priori* determination of which quantity would be calculated from the other two by virtue of imposing the constraint.

The SAR Inference Engine benefits greatly from constraint programming paradigm. The fact that the set of inputs and outputs is not fixed *a priori* enhances its potential to be used in multiple parts of the system. Separation of problem domain logic from implementation is enhanced by the constraints' existence as discrete entities. This separation promotes visibility. It is explicit which constraint is the basis for every assignment of a value to a quantity. Finally, the SAR Inference Engine is a useful medium for preserving and disseminating knowledge related to synthetic aperture radar systems.

IV. SAR Inference Engine

The SAR Inference Engine is implemented as a library that can be linked into other programs. There are two phases of operation. Initialization takes place in the first phase. During initialization the model is loaded into a model-independent core which performs the inferences. After initialization, the SAR Inference Engine is informed, one at a time, of values to assign to quantities. After each assignment the SAR Inference Engine infers values of other quantities until no more inferences can be made. Between assignments, the status of any quantity may be queried. In addition, through careful bookkeeping, the SAR Inference Engine provides the ability to selectively backtrack by invalidating prior assignments of values to quantities.

A. Usage Throughout the Radar Lifecycle

The SAR Inference Engine is envisioned to be useful throughout the lifecycle of SAR radars.

- System design: System designers can benefit when they explore the design space because the SAR Inference Engine has the ability to backtrack as well as it allows for any quantity to be an input or output. Because the SAR Inference Engine can be used in later phases, system designers are assured that calculations performed in later phases will match theirs.
- Mission planning: Sandia's current mission planning tools are difficult to use in ways that were not envisaged when they were written. The flexibility of the SAR Inference Engine addresses this. As an example, one mission may require a minimum signal-to-noise ratio and allow range to be adjusted to achieve this minimum SNR. Another mission may need to satisfy a minimum range and will try to maximize SNR possible subject to that lower bound on range.
- Radar operation: It is envisioned that the SAR Inference Engine can prove highly beneficial during the operation of the radar. Incorporating the SAR Inference Engine into the radar operator interface would provide the operator with greater flexibility. It is also envisioned that the SAR Inference Engine could be embedded in the radar. Since the radar is using the same code as used during system design and mission planning, errors due to inconsistencies are less likely.

B. Full-scale SAR Model

The current full-scale model has approximately 250 constraints of physical interest involving approximately 300 quantities of physical interest. (There are actually considerably more constraints and quantities. These numbers exclude quantities and constraints which were introduced to express the complex constraints as a larger number of simpler ones.) The scope of the SAR model is comprehensive. It includes signal-to-noise ratios, antenna characteristics, atmospheric loss, image characteristics, various electronic delays, and many others.

The model in the SAR Inference Engine is a port of a Matlab model developed by system engineers during the design phase of an existing Sandia SAR. The SAR Inference Engine is fully compatible with the Matlab model and it has been validated to produce the same results to one part in a trillion.

While the SAR Inference Engine has been verified to be functionally compatible with the original Matlab model extensive modifications had to be done to make the model conducive to implementation under a constraint programming paradigm. The Matlab model was restructured so that every variable is assigned to only one time. Redundant common sub-expressions were eliminated. Complicated equations and logic were decomposed into multiple simpler equations that translate into a small number of types of basic constraints. Most importantly, rigorous definitions were given to all quantities and the rationale behind all constraints was documented.

V. Future Work

The generality and flexibility of the SAR Inference Engine come at a price. One price is its resource usage in terms of both execution time and memory. Another is that it brings some theoretical issues to the forefront. One theoretic issue is the completeness of the model itself. This can be summarized by the question "Are there any situations where nature will fix the value of some quantity in the model but the SAR inference Engine will not?" Self-interactions are another class of difficulty that requires further research to ensure correctness.

The generality of the SAR Inference Engine imposes a penalty on execution time and the amount of memory required. In most cases when speed is essential in a SAR, the same calculations are performed many times. Two strategies that have been prototyped for speeding up the SAR Inference Engine take advantage of this fact: preplanned execution in the manner of FFTW and generating C code that can be compiled and executed at the native speed of the CPU. All the bookkeeping that the SAR Inference Engine keeps to support selective backtracking requires memory. This has not been observed to be a problem but further investigation is required to characterize how memory usage scales with model size.

VI. Summary

By being distributed over multiple people, documents and software, the logical model of a synthetic aperture radar becomes fragmented. Fragmentation gives rise to errors and inconsistencies. The SAR Inference Engine addresses this by implementing a comprehensive model of a Sandia synthetic aperture radar. To be a viable solution to the fragmentation problem the model needs to be accessible and useful in a wide range of contexts. The constraint programming paradigm gives the SAR Inference Engine this flexibility.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] J. Walser, "Solving linear pseudo-Boolean constraint problems with local search", Fourteenth National Conference on Artificial Intelligence, pp. 269-274, 1997.

[2] G. Beaumet, G. Verfaillie, M. Charmeau , "Estimation of the minimal duration of an attitude change for an autonomous agile earth-observing satellite", Proc. 13th International Conference on Principles and Practice of Constraint Programming, pp. 3-17, 2007.

[3] R. Chenouard, P. Sébastian, L. Granvilliers, "Solving an air conditioning system problem in an embodiment design context using constraint satisfaction techniques", Proc.13th International Conference on Principles and Practice of Constraint Programming, pp. 18-32, 2007.

[4] Alexey Radul, "Propagation Networks: A Flexible and Expressive Substrate for Computation", PhD Thesis, MIT, 2009.