A Hierarchical Requirements Reference Model

Anitha Murugesan
Department of Computer Science, University of Minnesota
200 Union Street, Minneapolis, MN 55455
anitha@cs.umn.edu

1. INTRODUCTION
On one hand, the advances in technology has increased and improved the capability of systems (such as devices). On the other hand, this has increased the size and complexity of such systems, thereby complicating their development. In practice, iterative decomposition of a system into sub-systems is a common approach to manage complexity and maintain intellectual control during development[9]. In such cases, the overall system developed is a hierarchical composition of sub-systems.

Developing a system hierarchically, pose some interesting and unique challenges that warrants a need for systematic development and rigours reasoning techniques [2, 8]. Precisely capturing the decomposition information and ensuring if the overall composed system satisfies its requirements is a major challenge, due to the hierarchy and complexity of the sub-systems used to build it. Hence, when developing such complex systems, it is beneficial and also a common approach to use standard guidance, such as requirements reference models, to help systematically represent a system and reason about its correctness. Well known requirements reference models, such as WRSPM model [3] and Functional Documentation model [7], provide a conceptual framework to describe and reason about a system using its development artifacts such as requirements, assumptions (environment) and design. However, their conceptual view of the system lacks the generality required for representing and reasoning the modern multi-component, multi-hierarchical systems, thereby making them inadequate to be used as reference while developing such systems.

This paper introduces a Hierarchical Requirements Reference (HRR) Model that provides a generic framework for representing and reasoning about a hierarchically composed system. The HRR model provides a simple and consistent conceptual view and representation scheme for the system to capture the multi-component and multi-hierarchical system artifacts. In addition, the model also formalizes the properties of the artifacts, that are sufficient to hierarchically reason about the correctness of the system.

2. BACKGROUND
This section introduces two well-known requirements reference models, that are used as a guidance to document and reason about systems. The models described in this section serves as a background for understanding the HRR model explained in the following section.

WRSPM Model [3], illustrated in 1, provides a conceptual framework to represent the problem domain - the 'World', the solution domain - the 'Machine' and the interaction space between them - the 'Interface'. Based on this conceptual separation, the model defines various system development artifacts such as Requirements (R), Assumptions (W), Specifications (S), Program (P) and Programming Platform (M) using phenomena - such as states, events, or individuals - identified and categorized based on their visibility and control by the domains (e_w, e_s, e_p and e_m). The model also defines relationships between the artifacts that provide means to formally reason about the correctness of the system.

Similar to the WRSPM model, The Four Variable Model [7], illustrated in Figure 2, attempts to conceptualize a generic control system representation. In this model, a system is represented as five mathematical relations expressed over shared quantities/variables of interest: NAT- defines the assumptions, REQ- defines the requirements, IN- defines the sensors that observe problem domain, SOFT- defines the controller (software) and OUT- defines actuators that control the problem domain. This model also formally defines properties between these relations to verify correctness.

Figure 1: WRSPM Model

Figure 2: Functional Documentation Model

Both these models were useful for conceptualizing, discussing and reasoning artifacts of a simple system. However, they do not lend themselves well, when attempted to use them for a hierarchical system, since (1) the conceptual partition of domains and artifact representation is not generic enough to be applicable to all layers and components in the system hierarchy and (2) provision to reason about hierarchical composition of components is lacking. Hence there is a need for a much generic representation and reasoning framework.

---

*This work has been partially supported by NSF grants CNS-0931931 and CNS-1035715.
3. THE HRR MODEL

The Hierarchical Requirements Reference (HRR) model, primarily intended to address the inadequacies of the existing reference models, provides a generic and simple framework that captures any multi-component and multi-level hierarchically composed system, and defines rules to rigourously reason about the correctness of such a system.

When a system is decomposed (or composed), the components are structured - termed System Architecture- and each component is allocated with its function or its requirement - called Requirements flow down- in such a way that the overall system requirements are satisfied by composing its component requirements in the specified architecture [4].

![Figure 3: The HRR Model](image)

The HRR model, as illustrated in Figure 3, represents a system as a hierarchical composition of requirements (R). At the top most level (the system level), R represents the overall system requirements. When the system is decomposed, each component is allocated with its set of requirements. Hence in a hierarchical view, each component’s requirement is derived from a requirement in its parent component, which in turn is flown down from its ancestor components. The term R is repeated for both the system and component levels intentionally, since R always represents the requirements from the component perspective. However, in order to uniquely identify the component in the hierarchical architecture, the artifact is superscripted with the level of hierarchy (i) and subscripted with the decomposition component identifier (j)\(^1\).

These requirements are expressed over relevant quantities of interest or variables, monitored variables (m)\(^2\) and controlled variables (c)\(^3\). Hence at any level i, the system requirements are expressed as \(R^i\)\((m^i, c^i)\). At every level of decomposition (i), the composition of all, say n, component’s requirements is represented by \(\bigwedge_{j=1}^{n} R_{jn}^{i}(m_{jn}, c_{jn})\), where \(\bigwedge_{j=1}^{n}\) represents the architectural information that informs how each component \((jn)\) is composed; and \((m_{jn}, c_{jn})\) represents the set of monitored and controlled variables in that decomposition for each component. The set of m and c for every component is assumed to be disjoint sets for clarity and ease of representation and reasoning.

3.1 Artifact Rules

The following four rules, expressed using first order logic, are sufficient conditions to rigourously reason about a system represented using the HRR model.

I. Requirement Feasibility: For every component \((j)\) in any level of hierarchy \((i)\), it is essential to check if all the requirements are defined to have at-least one response (output) for every possible input. This ensures that the requirements are consistent and feasible (not trivially empty).

\[
\forall(m^i_j) \cdot \exists(c^i_j) \cdot R^i_j \quad (1)
\]

II. Architecture Well Formedness: The architecture \(\bigwedge_{j=1}^{n} R_{jn}^{i+1}(m_{jn}^{i+1}, c_{jn}^{i+1})\) describes the components connection among themselves and their parent by identifying common m and c variables. To ensure such an architecture has a well formed structure, architectural rules (2) between the common variables are essential. At every level \(i+1\) and its parent level \(i\), for every \(m_{jn}^{i+1}\) and \(c^i_j\)

\[
Fun_{comp} : m_{jn}^{i+1} \rightarrow m^i_j \land C \quad ; \quad Fun_{sys} : c^i_j \rightarrow m^i_j \land C \quad (2)
\]

where \(C = \{\bigcup_j c_{jn}^{i+1}\}\)

III. Requirement Satisfaction: The property (3) ensures that the composition of component requirements are sufficient to meet the system requirements.

\[
\forall(d) \cdot \bigwedge_{j=1}^{n} R_{jn}^{i+1}(m_{jn}^{i+1}, c_{jn}^{i+1}) \Rightarrow R^i_j(m^i_j, c^i_j) \quad (3)
\]

where \(d = \{\bigcup_j m^i_j, c^i_j, \bigcup_j (m_{jn}^{i+1}, c_{jn}^{i+1})\}\)

IV. Component Realizability: When an architecture is defined, if its realizability is asserted, the undesirable situation of trivially meeting Rule 3 when the decomposition is not realizable (antecedent equates to false), can be avoided.

\[
\forall(m^i_j) \cdot \exists(c^i_j) \cdot R^i_j \Rightarrow \exists(d^i) \cdot \bigwedge_{j=1}^{n} R_{jn}^{i+1}(m_{jn}^{i+1}, c_{jn}^{i+1}) \quad (4)
\]

where \(d^i = \{\bigcup_j c_{jn}^{i+1}/m^i_j\}\)

Using this model, the overall system can be reasoned by hierarchical satisfying rules (1) to (4). The hierarchical association is the link between the component’s decomposition with its parent. This model is particularly useful to capture and reason safety critical systems, such as medical devices, avionics etc. A practical implementation of this approach has been demonstrated using a compositional reasoning approach [6, 5] for a generic patient controlled analgesic infusion pump system [1].

4. DISCUSSION AND CONCLUSION

The HRR model can be considered as a generalization of the other reference models since it generalizes the system view conceptualized by the other models by simplifying the way of representing a system as a hierarchy of requirements. For example, the HRR model’s system decomposition structure is generic and the WRPSM model’s separation of artifacts as S-M-P or Functional documentation model’s IN-SOF-OUT can be considered as a special way of specifying system architecture using the HRR model.

Representing a hierarchically composed system and formally reasoning about it, is complicated due to the hierarchy of subsystems used to build it. The HRR model, introduced in this paper, provides a generic artifact representation scheme and rules to systematically capture system artifacts and rigorously reason about such systems.
5. REFERENCES


