# Reference Architectures; Why, What and How

## White Paper Resulting from Architecture Forum Meeting

## March 12 & 13, 2007 (Hoboken NJ, USA)

**Edited by:**

**Dr. Gerrit Muller**, Embedded Systems Institute

**Mr. Eirik Hole**, Stevens Institute of Technology

Input was provided by the following participants in the Architecture Forum:

| Name | Organization |
|---|---|
| Rob Cloutier | Stevens Institute of Technology |
| Teun Hendriks | Embedded Systems Institute |
| Ari Herva | Nokia |
| Eirik Hole | Stevens Institute of Technology |
| Jouko Junkkari | Nokia |
| Eric Kreuwels | FEI Company |
| Kenneth Kung | Raytheon |
| Jouko Kyllönen | Nokia |
| Bjørn V. Larsen | Kongsberg Defence & Aerospace |
| JP Leblanc | Nokia |
| Gerrit Muller | Embedded Systems Institute |
| Roshanak Nilchiani | Stevens Institute of Technology |
| Rolf Siegers | Raytheon |
| Andy Turner | Nokia |
| Dinesh Verma | Stevens Institute of Technology |
| | |

**Published on June 13, 2007**

## 1. Introduction

Architects active in the creation of complex systems frequently use the term *Reference Architecture*. However, these experienced architects may not have a consistent notion of what this actually is:

- What is a Reference Architecture?
- Why do we need Reference Architectures, what is their value, what is the benefit of creating and maintaining them?
- How do you capture a Reference architecture, how do you visualize it, what is the appropriate level of abstraction, how is it used?

These questions were discussed at this meeting of the System Architecture Forum, based on presentations of actual experiences of several architects from the defense domain and the equipment domain.

## 2. Why

The outcome of a chaotic discussion at this forum has been captured in a more structured graph; see Figure 1, which shows concurrent trends, such as increasing complexity and dynamics and the main objectives of Reference Architectures as reaction to these trends. These main objectives are achieved by more detailed objectives of Reference Architectures, shown at the right hand side of the graph of objectives. We will first discuss the trends and objectives, as shown in Figure 1, and then we will discuss the role of Reference Architectures as part of the business or enterprise strategy, as shown in Figure 2.

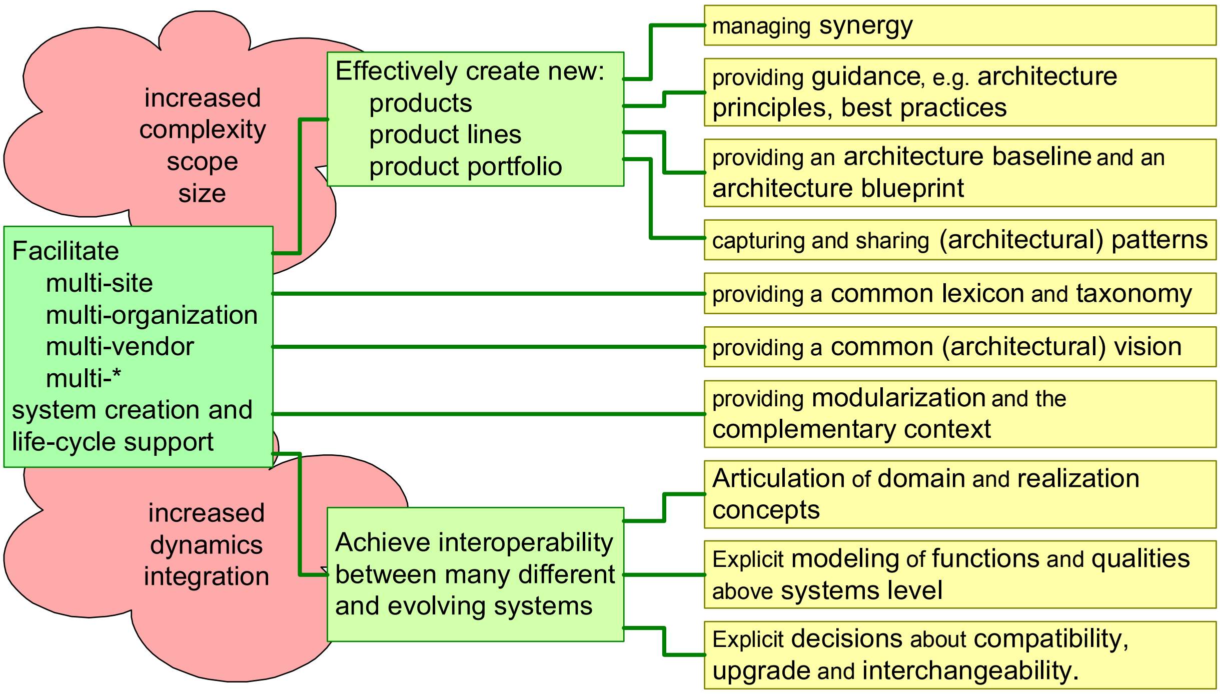
increased
complexity
scope
size
Effectively create new:
products
product lines
product portfolio
Facilitate
multi-site
multi-organization
multi-vendor
multi-*
system creation and
life-cycle support
increased
dynamics
integration
Achieve interoperability
between many different
and evolving systems
managing synergy
providing guidance, e.g. architecture principles, best practices
providing an architecture baseline and an architecture blueprint
capturing and sharing (architectural) patterns
providing a common lexicon and taxonomy
providing a common (architectural) vision
providing modularization and the complementary context
Articulation of domain and realization concepts
Explicit modeling of functions and qualities above systems level
Explicit decisions about compatibility, upgrade and interchangeability.

**Figure 1. Graph of objectives of Reference Architectures.**

### The magic *multi* word

In all domains we see two simultaneous trends:

- Increasing complexity, scope and size of the system of interest, its context and the organizations creating the system
- Increasing dynamics and integration: shorter time to market, more interoperability, rapid changes and adaptations in the field.

These trends cause a transition from "simple" closed system creation to distributed open system creation and evolution. In the simple and closed situation, a system could be created at one location, by one vendor, in one organizational entity. Many of today's systems are developed as distributed open development at multiple locations ("multi-site"), by multiple vendors, across multiple organizations.

In Figure 1 we also added multi-*, because the multiplicity is not limited to organizations, vendors and locations. Systems also become more multi-domain (e.g. security has military as well as civil applications), multi-application (e.g. electron microscopes are used for metrology in high volume applications and for material analysis in low volume applications), multi-cultural (global application, but customized for local cultural aspects), development and manufacturing is based more often on multi-sourcing, and so on.

Reference Architectures start to appear in organizations where the multiplicity reaches a critical mass triggering a need to facilitate product creation and life-cycle support in this distributed open world. The Reference Architecture provides:

- a common lexicon and taxonomy
- a common (architectural) vision
- modularization and the complementary context

The common lexicon and taxonomy facilitates communication across the multiple dimensions. The common (architectural) vision focuses and aligns efforts of multiple peoples and teams. Modularization helps to divide the effort, where the context information ensures later integration.

### Effective creation of products, products lines, and product portfolios

In this setting the goal is effectively create products, products lines, and product portfolios. The Reference Architecture improves the effectiveness by:

- managing synergy
- providing guidance, e.g. architecture principles, best practices
- providing an architecture baseline and an architecture blueprint
- capturing and sharing (architectural) patterns

Managing synergy is often the main goal of Reference Architectures from managerial perspective. It should be noted that maximization of synergy is *not* the goal of Reference Architectures. However, a good Reference Architecture helps in understanding where synergy can be harvested effectively and where harvesting of synergy might backfire. The insight that harvesting synergy is not always trivial has been formulated by Doug McIlroy at the 1968 NATO conference about Software Engineering.

Reflection of experiences can be captured in architecture principles and best practices. This condensed, somewhat abstract, know how provides guidance to later developments, hopefully preventing the re-occurrence of bad experiences over and over again. More concrete know how can be mined by looking for architectural patterns. A pattern is a well working solution for a common problem, where is described in what circumstances and context this solution is appropriate.

The effectiveness is also improved by providing an architecture baseline, a shared starting point to discuss future changes and extensions. The Reference Architecture serves as an architecture blueprint for future architectures. Again this hopefully prevents the re-invention and re-validation of solutions for already solved problems.

### Achieving interoperability between many different and evolving systems

In this multi-* world *interoperability* determines the usability, performance and dependability of user level applications. Reference Architectures must improve interoperability by:

- Articulation of domain and realization concepts
- Explicit modeling of functions and qualities above systems level
- Explicit decisions about compatibility, upgrade and interchangeability.

Decreased integration cost and time might also be an objective of Reference Architectures. Note that all interoperability considerations are also applicable to reduction of integration cost and time. Note also that for re-use to be effective it is required that integration effort must be small.

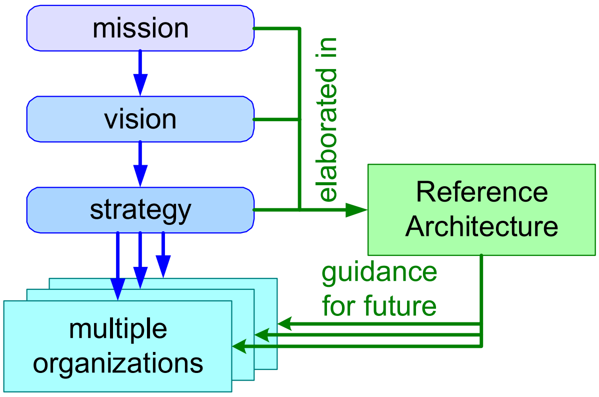

**Figure 2. A Reference Architecture elaborates mission, vision and strategy to provide guidance to multiple organizations.**

### Mission, vision, strategy

A Reference Architecture is strongly linked to company (or consortium, e.g. MIPI) mission, vision and strategy. The strategy determines what multi-dimensions have to be addressed, what the scope of the Reference Architecture is, what means, such as synergy, are available to realize mission and vision. In fact, a Reference Architecture is an elaboration of mission, vision and strategy.

**Principle 1:**

*A Reference Architecture is an elaboration of company (or consortium) mission, vision and strategy. Such Reference Architecture facilitates a shared understanding across multiple products, organizations, and disciplines about the current architecture and the vision on the future direction.*

### Linking past and future

A Reference Architecture facilitates a shared understanding across multiple products, organizations, and disciplines about current architecture(s) and future directions.

Architectures of the past are transformed in a Reference Architecture. However, the purpose of the Reference Architecture is future oriented. The mission, vision and strategy are needed to add the future direction to the wisdom of the past. Note that future directions are inherently unproven. Hence future directions might be conflicting with the experience as described in the "how" section that reference architectures should only contain proven concepts.

## 3. What

Many architectures that are labeled Reference Architecture describe the technical architecture only. The participants of the SAF meeting concluded that a Reference Architecture should address:

- Technical architecture,
- Business architecture, and
- Customer context.

In practice, business architecture and customer context are often missing, see (Rosen 2007). As a consequence these technical reference architectures represent solutions for unspecified problems in unspecified contexts.

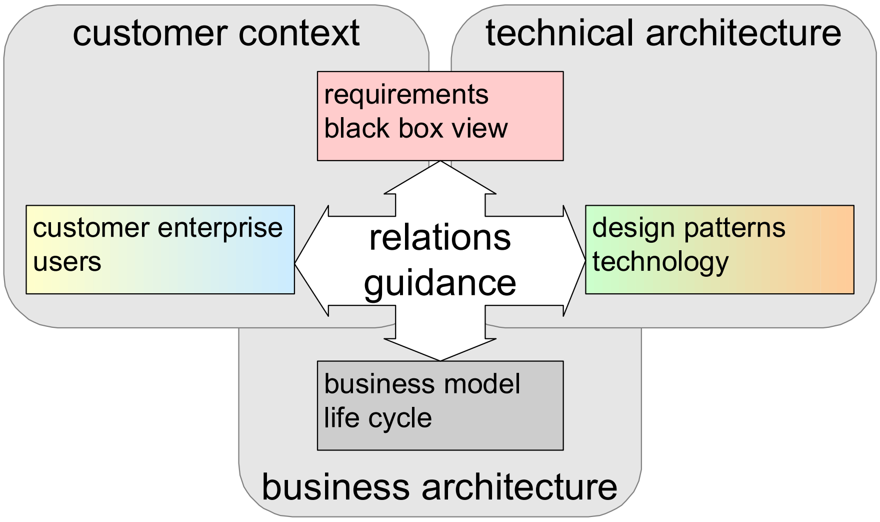

**Figure 3. Business architecture, technical architecture and customer context together form the Reference Architecture.**

Figure 3 shows the *business architecture*, the *technical architecture*, and the *customer context* as partially overlapping. The common denominator is the requirement or black box specification level, where the features and functions are modeled in a product independent way. The technical architecture provides solutions in technology, captured as design patterns. The business models and life cycle considerations in the business architecture guide decisions in the technical domain. The same holds for the customer context, where processes in the customer enterprise and user considerations will provide this guidance. Guidance from the Reference Architecture is largely based on the explicit understanding of the relations between the *business architecture*, the *technical architecture*, and the *customer context*.

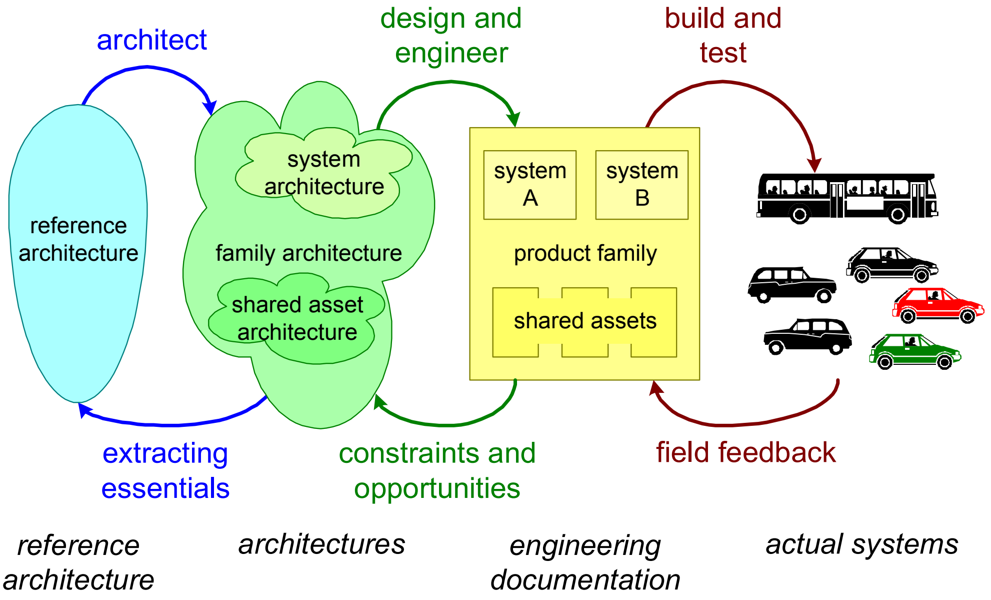

**Figure 4. Reference Architectures are very abstract. In several instantiation steps a Reference Architecture is transformed into actual systems.**

The level of abstraction of Reference Architectures makes it more difficult to understand their role. Figure 4 shows the instantiations that are needed to transform an abstract Reference Architecture into actual systems. The first step is to instantiate a system architecture based on the Reference Architecture. This system architecture is used to design and engineer the system, resulting in engineering documentation that describes how the system can actually be ordered, assembled and tested. Note that the creation and evolution of Reference Architectures is strongly feedback based. Field feedback from actual systems results in updates of the engineering documentation. The design and engineering effort provides constraints on architectures, but also opens opportunities. Finally the Reference Architecture itself is largely a mining and extraction effort of existing architectures.

The re-use or asset sharing dimension plays a role besides the instantiation dimension. If a product family is created, then we will instantiate a family architecture from the Reference Architecture. A family architecture describes the members of the product family and the mechanisms in the family to specialize members into the desired direction. The family architecture also describes the synergy within the product family and the associated rules for design, such as standardization. The shared assets often get a lot of focus, resulting in an architecture describing the shared assets (or platform).

In practice the term Reference Architecture is sometimes used for family architecture and/or shared assets architecture. The role of a Reference Architecture is related to the instantiation flow as shown in Figure 4. Some people see the Reference Architecture as an *asset* capturing generalized knowledge of systems. This view of Reference Architectures positions the Reference Architecture in the area of knowledge management. This specification *asset* is seen as essential to effectively deploy a re-use strategy of actual design artifacts. As discussed in section 2, the Reference Architecture contributes to communication effectiveness and provides guidance for future architecture instantiations.

These split views can be seen in today’s practice, where the term Reference Architectures is used for very different entities.

The IT world uses the term Reference Architecture often for the infrastructural concepts, for example for layering, communication and persistency. See for example (Simmons 2005) and (Reed 2002). The definition[1] provided by Wikipedia starts broader, then narrows to IT[2]. The examples are also limited to the infrastructural concepts, such as Java Enterprise Edition.

[1] A Reference Architecture provides a proven template solution for an architecture for a particular domain. It also provides a common vocabulary with which to discuss implementations, often with the aim to stress commonality.

At Philips Medical System the term Reference Architecture has been used for a much higher level domain understanding. In (Muller 2004) a Reference Architecture is discussed that provides guidelines for function allocation over products based on technology, life-cycle, safety and information characteristics.

A Reference Architecture is created with a certain scope in mind, e.g. a domain of a set of applications. In this scope the Reference Architecture links to relevant standards, legislation, domain constraints and mandatory frameworks.

## 4. How

### Process to create and maintain a Reference Architecture

A Reference Architecture captures previous experience, for instance by mining, or by generalizing existing architectures. To be of value for future architectures, a Reference Architecture is based on **proven** concepts. The validation of concepts in Reference Architectures is often derived from preceding architectures. Especially in cases where disruptive technologies or innovative applications are introduced it is challenging to have sufficient proof for a Reference Architecture. In these cases Reference Implementations and prototyping and an incremental approach might be an alternative for validation and proof. Note that flaws in Reference Architectures propagate to multiple architectures and actual systems and may damage or even destroy in that way entire enterprises.

**Principle 2**:

*A Reference Architecture is based on proven concepts. Most often preceding architectures are mined for these proven concepts. For architecture renovation and innovation validation and proof can be based on reference implementations and prototyping.*

[2] A reference architecture often consists of a list of functions and some indication of their interfaces (or APIs) and interactions with each other and with functions located outside of the scope of the reference architecture.

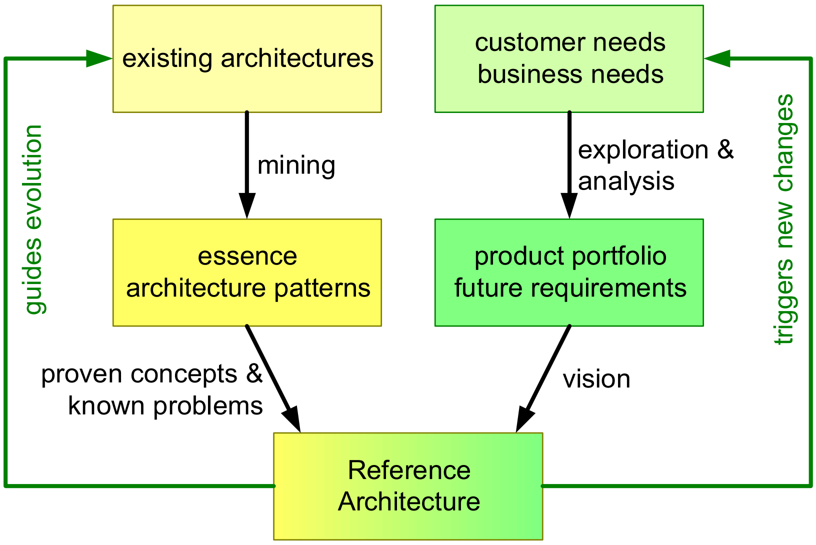

**Figure 5. Inputs of a Reference Architecture**

The future value of Reference Architecture depends on the vision going into it. This vision is based on (future) customer and business needs. These needs are explored and analyzed to be transformed into future requirements for the product portfolio.

Figure 5 shows this flow of proven concepts and known problems from existing architectures and vision derived from needs into the Reference Architecture. The Reference Architecture guides the evolution of existing architectures and influences the customers and business, which triggers new changes in their needs. Architectures, needs and Reference Architectures evolve continuously.

### Content of a Reference Architecture

A Reference Architecture must describe the essence of the architecture, the most significant and relevant aspects. A Reference Architecture can be supported by more detailed models, API's, standards et cetera. The challenge is to create a well-readable, accessible Reference Architecture that at the same time is non-ambiguous and effective for all stakeholders. The size of Reference Architecture is domain and objective dependent. Attempts to find a metric for the size resulted in lots of discussions. Many existing reference architectures are very big (hundreds or even thousands of pages). The risk of big Reference Architectures is that the essence is hidden instead of highlighted. A Counter measure for big Reference Architectures is to decompose it into a core Reference Architecture and more detailed models, interfaces, standards et cetera. Also navigational support can improve accessibility.

Reference Architectures are often associated with big and inflexible frameworks; see for example the nice contribution in (Martin 2003).

Content and structure of Reference Architectures is determined by domains and objectives. Typically a limited set (about 6) of aspects or cross cutting concerns or attributes are captured in the Reference Architecture. For example, one of the presented Reference Architectures captures the following 6 aspects:

- Timeliness/responsiveness of system (real time vs off line)
- Security considerations
- Degree of automation (machine vs man)
- Legacy retention/interoperability and integration/test
- Life cycle concerns (Costs, -ilities)
- COTS content

Domain specific tensions between cross cutting concerns determine the specific selection of architecture patterns.

## 5. When

A Reference Architecture has a life-cycle and *maturity*. As described in section 3 (see also Figure 4), Reference Architectures evolve continuously. Reference Architectures must be actively maintained and refactored. These observations trigger a set of questions:

- Is there an end-of-life for Reference Architectures? Can we create a list of criteria to assess the maturity or obsolescence of a Reference Architecture? Does the notion of generation break exist for Reference Architectures? Older Reference Architectures, well embedded and aligned in the organizations, may stifle changes and innovation. Disruptive technologies may kill the value of a Reference Architecture suddenly.

- How to do change control or management of the Reference Architecture? What is the level of formalization of a Reference Architecture? Which stakeholders are involved in the change management?

The life cycle of Reference Architectures is a clear opportunity for masters or doctoral research. Crucial to such research is the development of metrics to evaluate Reference Architectures.

## 6. Who

A Reference Architecture must be accessible and understandable for multiple stakeholders from engineers to business managers and customers. Therefore the Reference Architecture must be concrete and provide specific information. The challenge is to create a Reference Architecture that is generic for multiple architectures and that is concrete and contains specific information at the same time.

The Chief Architect (Business + Technical) owns the Reference Architecture. Ownership is a critical success factor for a Reference Architecture. Sponsorship of business managers for Reference Architectures is a prerequisite. Such sponsorship works only if the Reference Architecture provides value for the business, for example as a decision making tool for business leaders.

Reference Architectures are often related to organization structures. This relationship between organization and architecture is already a heated subject of discussion for actual architectures:

- Should the architecture follow the organization structure, or vice versa?
- Should we compromise architectural integrity to align better with the organization?
- Or should we adapt the organization to serve the desired architecture?

Note that Reference Architectures often span multiple organizations, complicating this issue even more. The maturity of the organization, the architects and the architecture is one of the selection criteria: the most mature entity will be normally be leading. Growth over time may result in changes in the leading entity.

The position of Reference Architectures as elaboration of mission, vision and strategy, suggests that Reference Architectures (ought to) play a significant role at enterprise level. This triggers a number of questions as well:

- What policy makers or boards should be involved with Reference Architectures?
- What policy processes and planning tools relate to Reference Architectures?
- Or, in summary, how are Reference Architectures embedded in the enterprise?

## 7. Value of Reference Architectures

Time is required of critical resources, chief architects, for the creation and maintenance of Reference Architectures. What is the value of Reference Architectures that justify the investment? No actual data on the value of Reference Architectures is available. The intuition of the participating architects is that in systems of significant size and complexity Reference Architectures provide value for all of the reasons discussed in section 2. The investment is justifiable if the scope of the Reference Architecture is sufficient: at least three[3] actual architectures are derived from the Reference Architecture.

If Reference Architectures are not perceived as valuable, then they are an easy target for cost reduction. Currently we don't have indicators to measure the value of Reference Architectures. Nevertheless, the sanity of Reference Architectures requires some kind of feedback loop, which in turn requires some indicators or metrics of Reference Architectures. An example of such a metric may be derived of an experienced benefit in project organizations is faster and better creation of bid proposals.

[3] A comparison was made with season passes of for instance Seaworld or Disney: cost break-even is 2.5 times for season passes.

## 8. Summary and conclusions

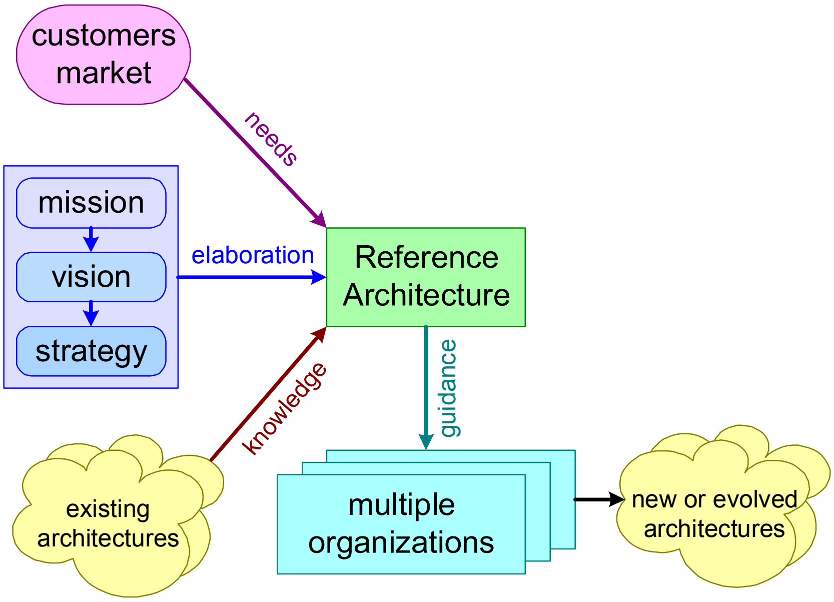

**Figure 5. Summary of the role of Reference Architectures, for simplicity feedback loops are not shown.**

Reference Architectures capture knowledge from existing architectures. Based on an elaboration of mission, vision and strategy, and on future customer needs the Reference Architecture is transformed in an architecture that provides guidance to multiple organizations that evolve or create new architectures.

Reference Architectures should address Technical and Business Architectures and the context. One of the main challenges is to make this inherently abstract Reference Architecture concrete and understandable by providing sufficient specific information and guidelines.

The value of Reference Architectures is foreseen in environments with a high multiplicity factor, creating social, organizational, business, application and technical complexity. This is a young area, where more questions are available than answers, ranging from proven value to lifecycle of Reference Architectures.

**Literature**

[Martin 2003] "Reference Architecture”; by Robert C. Martin, *Uncle Bob's Software Craftsmanship Corner*, http://www.artima.com/weblogs/viewpost.jsp?thread=17799 June 2004

[McIlroy 2004] "'Mass Produced' Software Components”; by Doug McIlroy, *proceedings of 1968 NATO Conference on Software Engineering*,

http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF October 1968, Brussels

[Muller 2004] "CAFCR: A Multi-view Method for Embedded Systems Architecting; Balancing Genericity and Specificity”; see Chapter 14. by Gerrit Muller, *PhD thesis*, http://www.gaudisite.nl/ThesisBook.pdf June 2004

[Reed 2002] "Reference Architecture: The best of best practices”; by Paul Reed *IBM developer works*, http://www-128.ibm.com/developerworks/rational/library/2774.html, September 2002

[Rosen 2007] "Enterprise Architecture Trends 2007: The year ahead”; by Michael Rosen, , *Cutter Executive Report*, http://www.cutter.com/offers/EAtrends.html, September 2002

[Simmons 2005] "Introducing the WebSphere Integration Reference Architecture”; by Scott Simmons, *IBM WebSphere Developer Technical Journal*,

http://www-128.ibm.com/developerworks/websphere/techjournal/0508_simmons/0508_simmons.html
August 2005