Transformational Government Framework Technical Committee

Guide to Working Draft 02

Recommended next steps

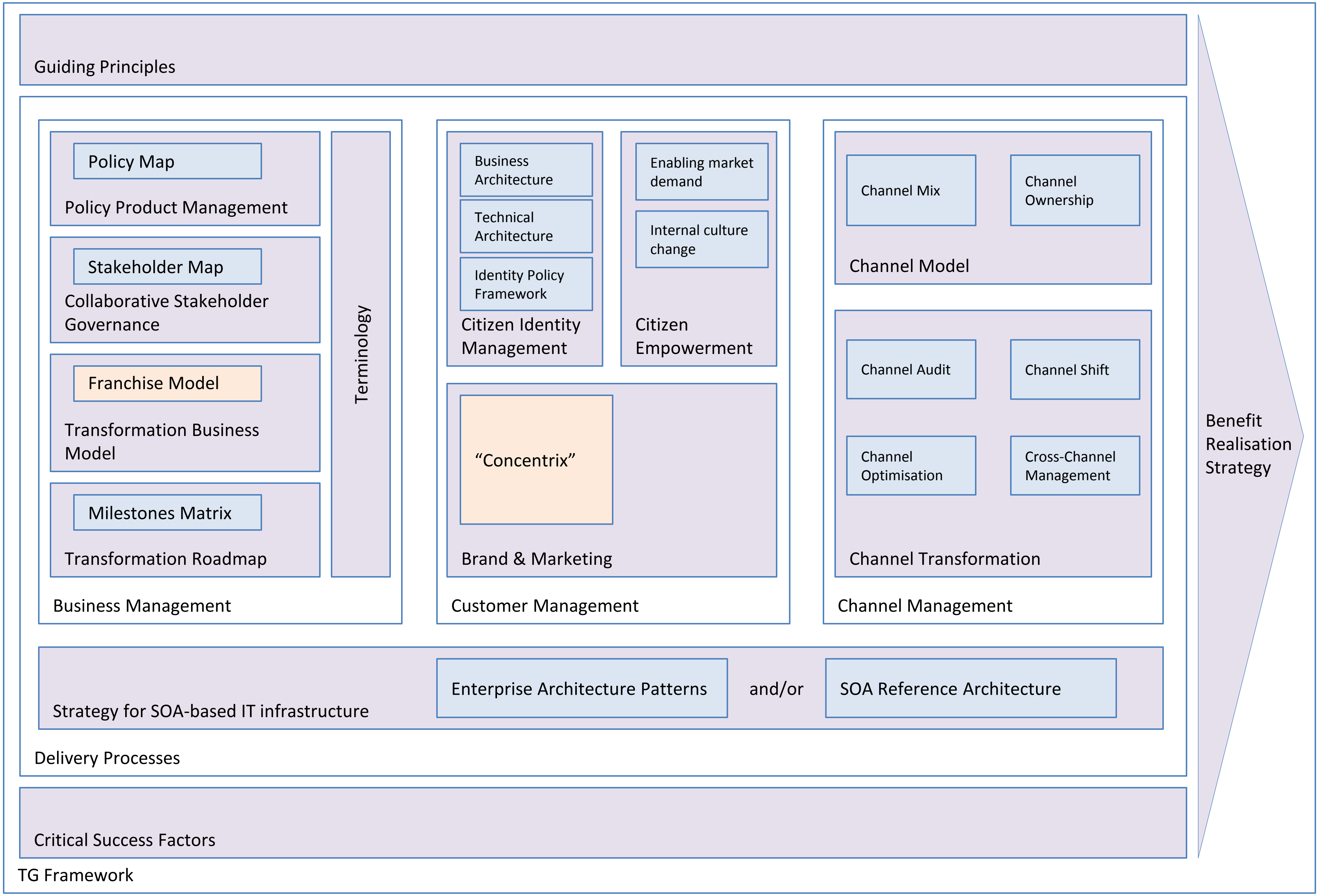
Peter F Brown
16 March 2011

Where we *were* in terms of objectives for deliverables….

OASIS Standard

Committee Specification

Committee Note



Guiding Principles

Business Management
- Policy Product Management
  - Policy Map
- Collaborative Stakeholder Governance
  - Stakeholder Map
- Transformation Business Model
  - Franchise Model
- Transformation Roadmap
  - Milestones Matrix
- Terminology

Customer Management
- Citizen Identity Management
  - Business Architecture
  - Technical Architecture
  - Identity Policy Framework
- Citizen Empowerment
  - Enabling market demand
  - Internal culture change
- Brand & Marketing
  - "Concentrix"

Channel Management
- Channel Model
  - Channel Mix
  - Channel Ownership
- Channel Transformation
  - Channel Audit
  - Channel Shift
  - Channel Optimisation
  - Cross-Channel Management

Strategy for SOA-based IT infrastructure
- Enterprise Architecture Patterns
- and/or
- SOA Reference Architecture

Delivery Processes

Critical Success Factors

Benefit Realisation Strategy

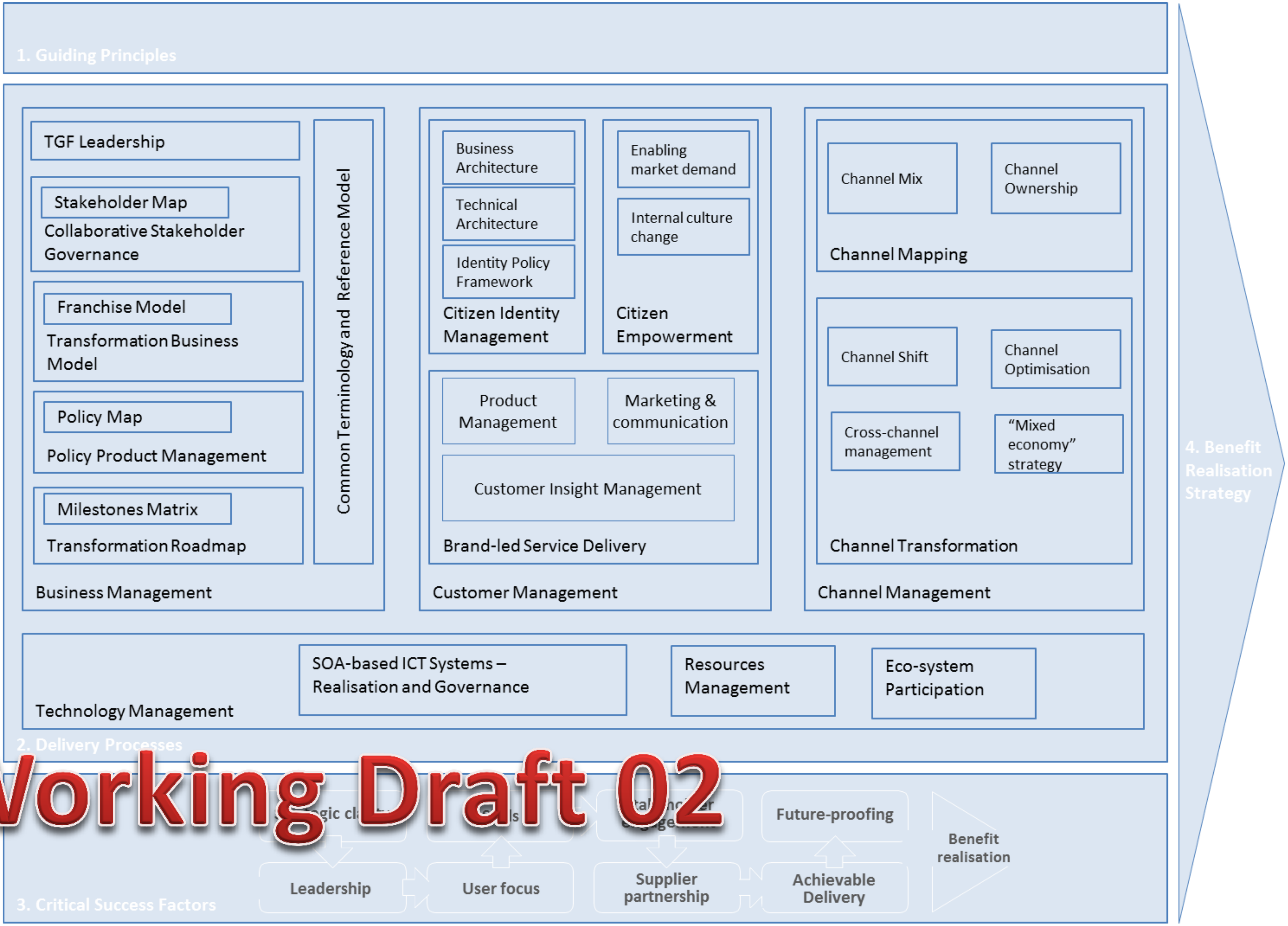
TG Framework

Where we *are* now….

OASIS Standard

Committee Specification

Committee Note



Working Draft 02

TC Process steps

- When the TC votes [1] to approve a Working Draft as a Committee Draft (CSD or CND), the Chair must submit a "Committee Specification Draft Creation and Upload Request"
  - Upon receipt of this form, the TC Administrator will process the Work Product for official publication in the OASIS Library as a Committee Draft (http://docs.oasis-open.org/), including addition of the requisite cover page metadata and other boilerplate information, as described in the TC Handbook

<- Today

- The decision by the TC to submit the draft for public review requires a Full Majority Vote, and must be accompanied by a recommendation from the TC of external stakeholders who should be notified of the review

<- Later

(optional)

[1] http://www.oasis-open.org/committees/process-2010-07-28.php#committeeDraft

Today

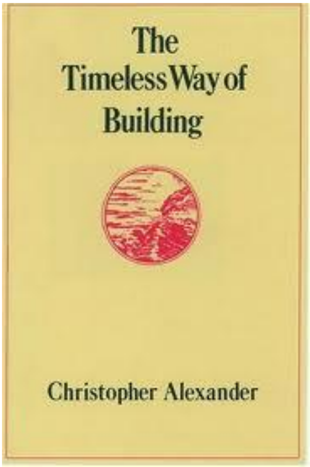
- **Vote** on Working Draft 02 (with minor edits)
  - To be published as ‘TGF Primer Committee Note Draft’
  - Remains an ‘internal’ document but can be cited and referenced
  - Processed and published by OASIS staff (url, id, etc)
- **Agree** to start the TGF Core
  - OASIS Standards Track document – new template
  - Mainly using Part II of Primer
  - But… need to discuss what ‘model’ to use to capture the core TGF standard

Options

1. Take Part II ‘as is’ and make better ‘standards-like’ phrases for elements concerning conformance
   - Maybe with additional diagrams, models?
2. Take Part II ‘as is’
   - add an Appendix with a formal model
3. Create a the whole TGF Core as a formal model
   - UML? Concept Map? Topic Map? RDF/OWL?
4. Or create TGF Core as a ‘Pattern Language’
   - What ?

A TGF ‘Pattern Language’

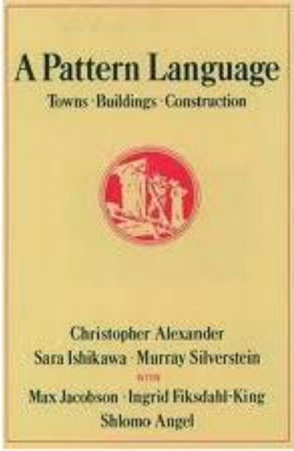
- ‘Pattern Language’ concept originally developed for architecture and building
  - Christopher Alexander “A Timeless Way of Building”
  - Subsequently adopted by many software architects, notably Grady and Booch (UML and Rational Rose fame)
    - Software Design Patterns; Requirements Patterns; etc…
- Formal structure possible
  - Alexander’s “pattern language” is only an example
  - Text/prose based
  - No hard-and-fast rules really ☺
  - Allows easy interconnections



The Timeless Way of Building

Christopher Alexander

Principles



A Pattern Language

Towns · Buildings · Construction

Christopher Alexander

Sara Ishikawa · Murray Silverstein

with

Max Jacobson · Ingrid Fiksdahl-King

Shlomo Angel

In his example pattern language,

- Each pattern is a three part rule:
  - Context
  - Problem to be addressed
  - Solution
- Each pattern contains 'essentials that cannot be avoided'
- Each pattern represents a 'current best guess'
- Patterns are not isolated, even if independent
- Patterns are ordered from broadest to most specific but…
- Patterns are related as a network, not completely linear
- Each pattern follows a determined structure
  - Archetypal example
  - Introduction, set context, includes cross-references to 'broader' patterns
  - Headline (bold) with essence of the problem
  - Body of the problem
  - Solution (bold) describing what needs to be done in what context, always expressed as an 'instruction', as a **policy** to be followed
  - Diagram of the solution, with labels for 'components' making up the pattern
  - Conclusion, tying the pattern to other related (narrower) patterns that either complete, refine, or embellish the current pattern

Fit for Purpose for TGF ?

- Pattern Language fits structure of TGF to date
  - Not too formal, prose based
  - Common repeatable structure
  - Allows cross-referencing and basic hierarchy of TGF parts
- Easy to use
  - Transposing existing TGF Primer Part II should be straightforward
  - Approx. 75 possible patterns identified
- Novel
  - Will give us all something to talk about in presentations ☺
- Proven
  - Pattern Language idea has already been transposed in IT-related areas (design patterns, requirements patterns)
- Easy to grasp
  - Easier to explain that UML, RDF, etc (especially for managers)

Supplementary arguments

- ‘Organic not totalitarian’:
  - No pattern language is ever complete
    - structure allows evolution as new issues and contexts arise
    - Allows ‘piecemeal growth without a piecemeal process’
  - Adopted ‘as a process not as a master plan’
- Participative
  - Rejects idea that ‘’participation = chaos’
  - Stakeholder engagement within clear process and language
- Patterns
  - are ‘general planning principles stating a clear problem, contexts of use and general features of any solution’
- Pattern Language
  - Constructed to adapt easily to any community
  - Includes process for adding new patterns
  - Different collections of patterns are possible for different cultures

Proposal

- Identify
  - the possible patterns (75 spotted so far)
  - inter-connections between patterns
  - supplementary TGF works that could be spun off which patterns
    - Committee Notes
    - White Papers
    - External resources
- Rapid ‘prototype’
  - with one section, such as “Business Management”
  - get feedback from TC
  - discuss at April TC meeting
- If +ve, proceed with full text
  - Aim for complete first draft for May meeting
  - Consider this TGF Pattern Language as fulfilling charter mandate to produce a ‘reference model’
  - Attribute editors/authors for identified and agreed additional work
- If –ve, re-visit alternatives, starting with (1) or (2)