



Orchestrating an SOA with Rules

Bright*Star

Service-Oriented Architecture & Web
Services Conference

Mark Norton

17 February 2004





The keyword is SERVICE

- but what does it mean??

- loosely coupled services, each of which perform a logical, discrete, business function. . *Services* are what you connect together using Web Services. A service is the endpoint of a connection. Also, a service has some type of underlying computer system that supports the connection offered.
 - <http://www.service-architecture.com/web-services/articles/service.html>
- Services are core business logic that are protocol-independent, location-agnostic and contain no user state. Services are coarse-grained, meaning the service can perform its logic and return the result in a single call. Services do not contain presentation logic, so they may be reused across diverse applications.
 - <http://searchwebservices.techtarget.com/ateQuestionNResponse>



The keyword is SERVICE (cont ...)

- but what does it mean??

- Services are published at a level of abstraction that corresponds to real world activity and recognizable business function. The really compelling aspect of this is the opportunity to implement comprehensive alignment and integration of the service life cycle with the business product and/or process life cycle.
 - <http://searchwebservices.techtarget.com/originalContent>

- What is unique about Web services-based SOA's is that in an SOA, a process is a service. Individual services can be composed into a process flow that itself can be described as a service using the same Web services standards.
 - <http://searchwebservices.techtarget.com/tip>



This presentation is about . . .

- How we approach service discovery and design to achieve optimal SOA implementations
- A decision-centric approach to service discovery for:
 - Better business alignment
 - Faster, more cost effective development
 - More intelligent, versatile processes
 - More complex/multi-party systems
- Learning from the business itself
 - Understanding decision-making behavior and it's promotion into day-to-day processing are already the core focus of business, but not of IT



What is a Service?

- A self-contained (meaning internally consistent and complete), context-independent, and stateless information processing task which is uniquely defined and addressable
 - Qualifies as an asset of the organization – it must be registered and owned by a business unit in the same way as any other asset
 - Is aggregated into larger units called **processes**



What is a Process?

- An aggregation of one or more services that when executed changes the state of the organization(s) in a predictable and repeatable way
 - State change is persistent and visible
 - Value creation is the primary goal of any process, therefore the state change has value
 - Must leave the business in a logically consistent state when finished



Intelligent processes

- Decision-making behavior is the key differentiator for many service-oriented businesses
- Automating decision making is a primary objective for many systems
- Intelligent processes are the result of automating decision-making behavior
- 'Intelligent Services' that actually implement the decision making are required for intelligent processes



Intelligent processes cont ...

- “Workflow and decisioning applications are top priorities of claims IT managers in 2005”
 - Published: 29 Dec 2004, Publisher: Datamonitor
- “Peter Drucker, the originator of the term ‘knowledge worker,’ tells us that in the Information Age, ‘the basic economic resource - the ‘means of production,’ to use the economist's term - is no longer capital. *It is and will be knowledge.*’ To these stakeholders, Web services are really knowledge products. Web services package up everything an organization knows about a process in a manner that is easiest for consumers (including software applications) to interpret and deploy for their own purposes”
 - <http://www.intelligenteai.com/showArticle.jhtml?articleID=57701431>



'Decisioning' or Intelligent services

- Can be used to automate business decision making for faster, more accurate, and more cost effective process outcomes
- Are always the drivers of intelligent processes – all other services are either dependent or optional
- Therefore, find the decision making → find the process
- Examples:
 - **Insurance**: underwriting; rating; claims adjudication
 - **Health**: clinical assessment; contract funding; process automation (e.g. admitting)
 - **Government**: compliance and enforcement; collection; inter-departmental processing
 - **Logistics**: rating; consignment routing and prioritization
 - **Finance**: reference data quality; contract definition compliance and reporting



Intelligent services in business

- The business unit that defines decision-making behavior is (almost) always a specialist unit
- Management of the process and management of “decisioning” are distinct and discrete
- Our systems rarely implement this distinction because the IT industry has ignored business rules
- The SOA must rediscover what the business already implements – separate “decisioning” from the processes that it drives



IT can also be a legitimate owner of intelligent services

at a meta level for IT, such as:

- Control process flow
 - which services are required and when
- Access control
 - user authorization, privacy management
- Determine quality of service
 - prioritization, optimized workflow
- Dynamic systems configuration
 - change the shape of the system to reflect the needs of each process



What is a decision

- “A Decision is an outcome that is derived by the business to qualify, quantify, or classify a domain object so that it can be further processed by the business in accordance with its declared business policies and practices”
- “A Formula is the logical expression that describes the transformation of named input variables into a Decision outcome”
- “Business rule” is a generic term that describes both/either decisions and formulas
- We intend to show that **Decisions** are the key to quickly and accurately defining modern business processes
 - Automation of decision making is the primary concern for most new systems
 - Decisions drive both data and process in systems construction



Examples of business decisions

- Gold card customers have a limit of 20000; Silver 10000
 - unless they have had an average daily balance of greater than 5000 for the previous 3 months, in which case 20000
- Imported turbo charged vehicles are a favorite of 'boy racers' and will be excluded from cover
 - only applies to second hand cars
- Patients who spend more than 5 hours in surgery recovery will be considered more serious cases and will attract 10 more points of funding
 - if mechanical ventilation is required



We can see that decisions . . .

- Are owned by the business
- Are dynamic and mutable – can change at any time
- Are technology and system independent
- To be automated the decision needs to have:
 - it's internal logic defined
 - the right input data available
 - a process to respond to the decision outcome
- When these conditions are met, decisions comply with our definition of service and can be implemented as a service
- A decision service can be updated constantly as the organization learns to make better decisions



IT has traditionally not viewed business rules as significant design artefacts

- Lack of an explicit concept to define and implement rules
- Lack of a single, unified method of expressing business rules
- Business rule agnostic class boundaries - business rules often apply to more than one requirement or class
 - e.g. Definition of business rules using UML class diagrams results in fragmented business rule definitions that are difficult to verify
- Future changes to rules are constrained by the 'SDLC' – wasting time and development effort
- Implementation of rules requires collaboration across many roles – leading to translation risk, increased workload and scheduling conflicts

This evidence suggests that traditional IT development does not give decisions the same importance that the business does!



Treat decisions as a discrete service

- don't bind them into system code

- A business' decision-making behavior is a critical business asset that provides greater differentiation than any other service, data or process
- The 'source of truth' and ongoing ownership of decisions always resides in discrete business units
- Decisions require no technical knowledge or input to be elicited and defined
- Decisions exist independently of any other service
- Decisions have a discrete and independent life cycle
 - The system does not need to know when or if a decision has changed or been added or deleted
 - The decision does not need to know how it will be implemented
- Implementing Decisions as services recognizes all of these characteristics



Benefits of discrete 'intelligent services'

- Visibility and focus of decision-making behavior
 - Places business policy and practice at the forefront
 - Transparent and auditable (Basel, Enron)

- Retention of decision ownership by the "source of truth"
 - No obfuscation
 - No translation risk

- A key requirements constraint is removed
 - Parallel and independent development of decisioning speeds development, reduces risk
 - No milestone dependencies, including go live

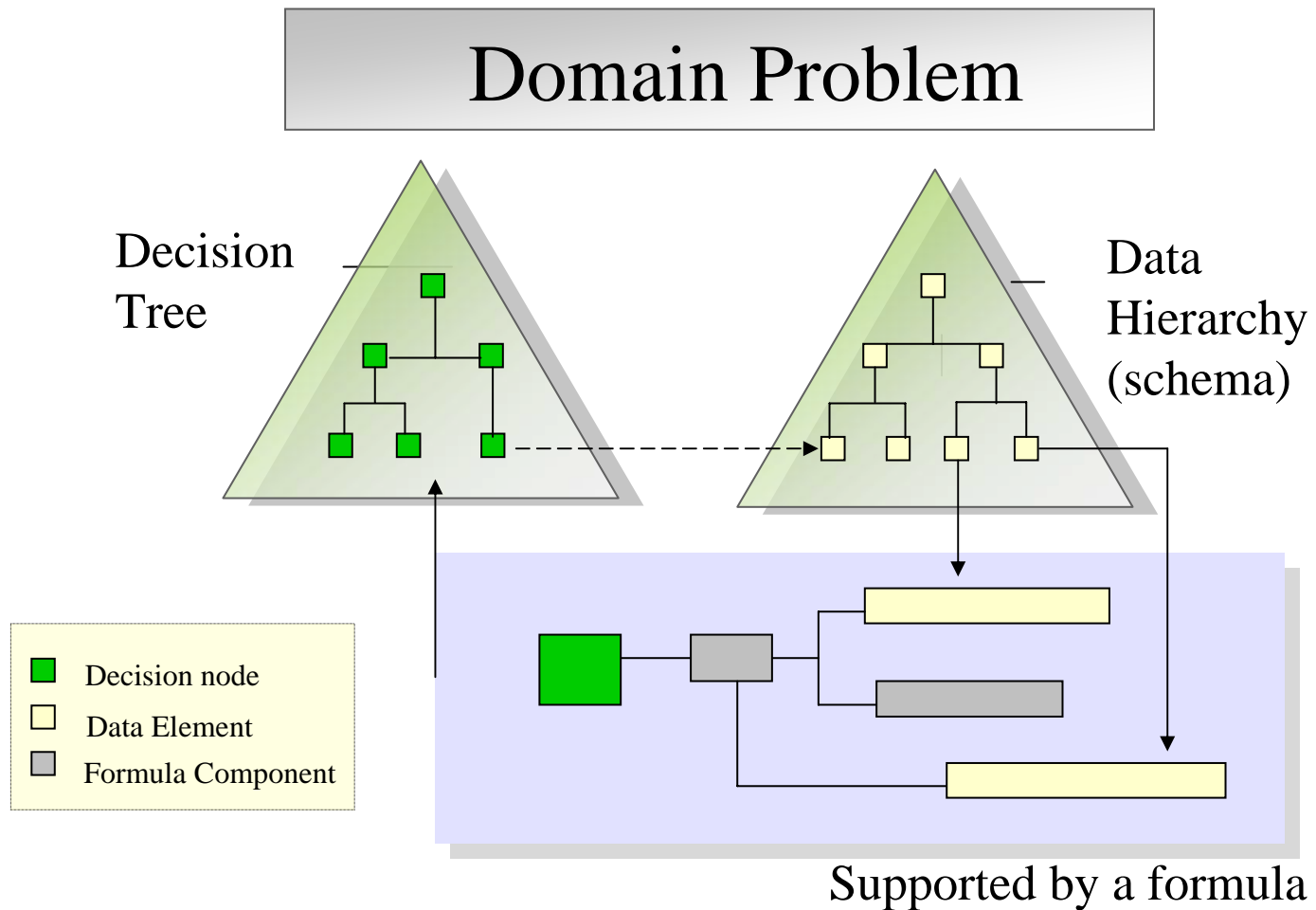
- Major cause of future change and instability is isolated and managed – permanently
 - Gartner estimate 5-40% saving of IT costs before savings of labour
[Gartner estimate from the Symposium ITXpo, Sydney Australia, November 2002]



Decisions are the core artefact of “intelligent process” definition

- From our first **inventory of decisions** we observed that we could reverse engineer:
 - Required data, including structure
 - Implied processing model
 - Implied cultureIn short, an entire business operation!
- This is NOT true for data and/or process models
 - **Decision making cannot be inferred**
- Therefore decision making is a more complete description of the business than either data or process models
- And, codified and automated decision making is a primary reason for the development of modern systems

Designing decision services

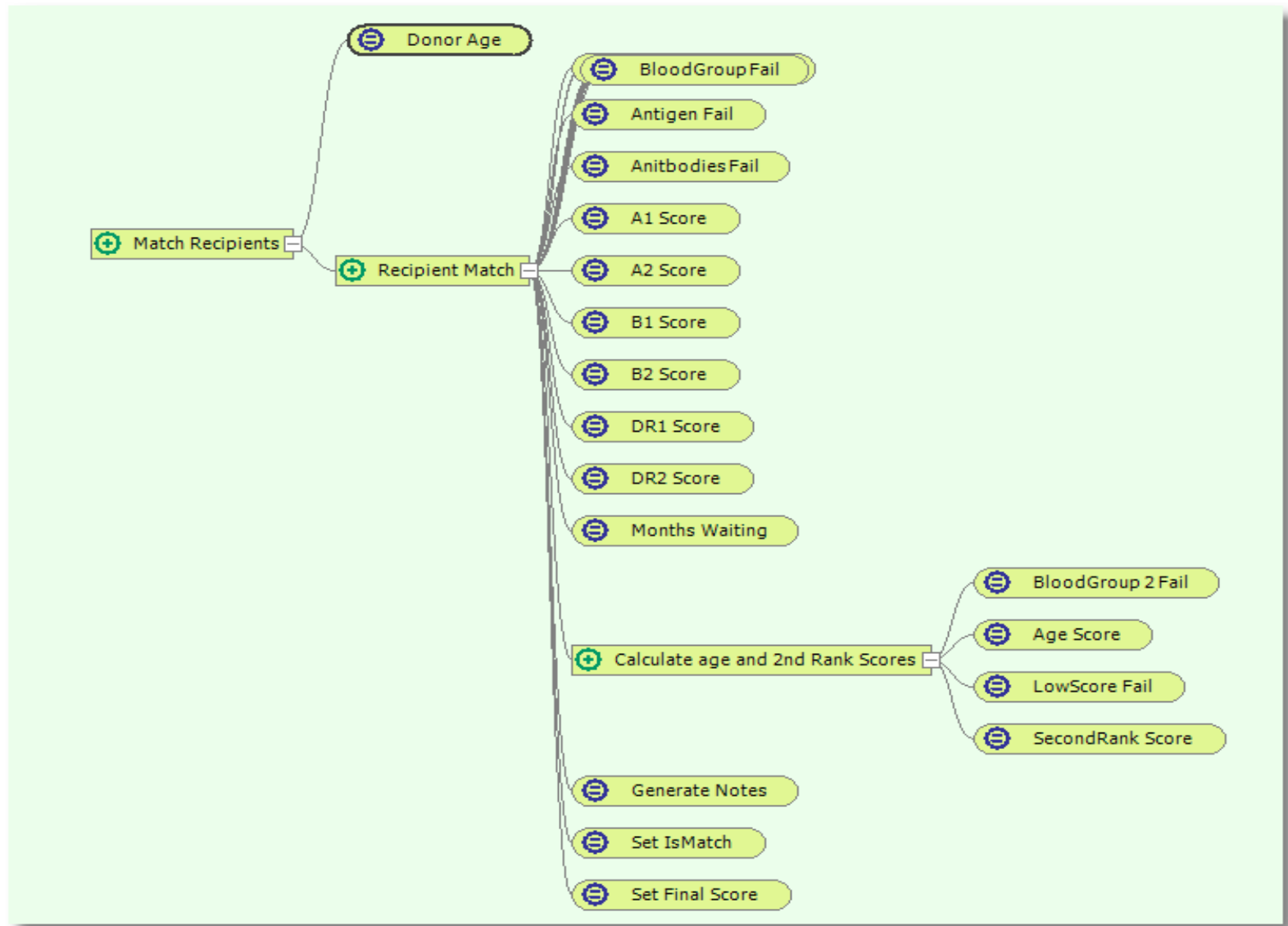




Creating the decision tree

- Ask of the 'source of truth'
 - 'what decisions do you need to make in order to [resolve the domain problem]'
 - **Examples: Underwrite a fleet of vehicles; Admit a patient**
- Use 'mind-mapping' techniques to decompose the high level decisions down to single valued, simple decisions
 - **These are now suitable for translation into formulas**
- Reconcile continuously with the Fact Model - one or more schemas that is supplying the 'decision vocabulary'

Example: Decision Tree

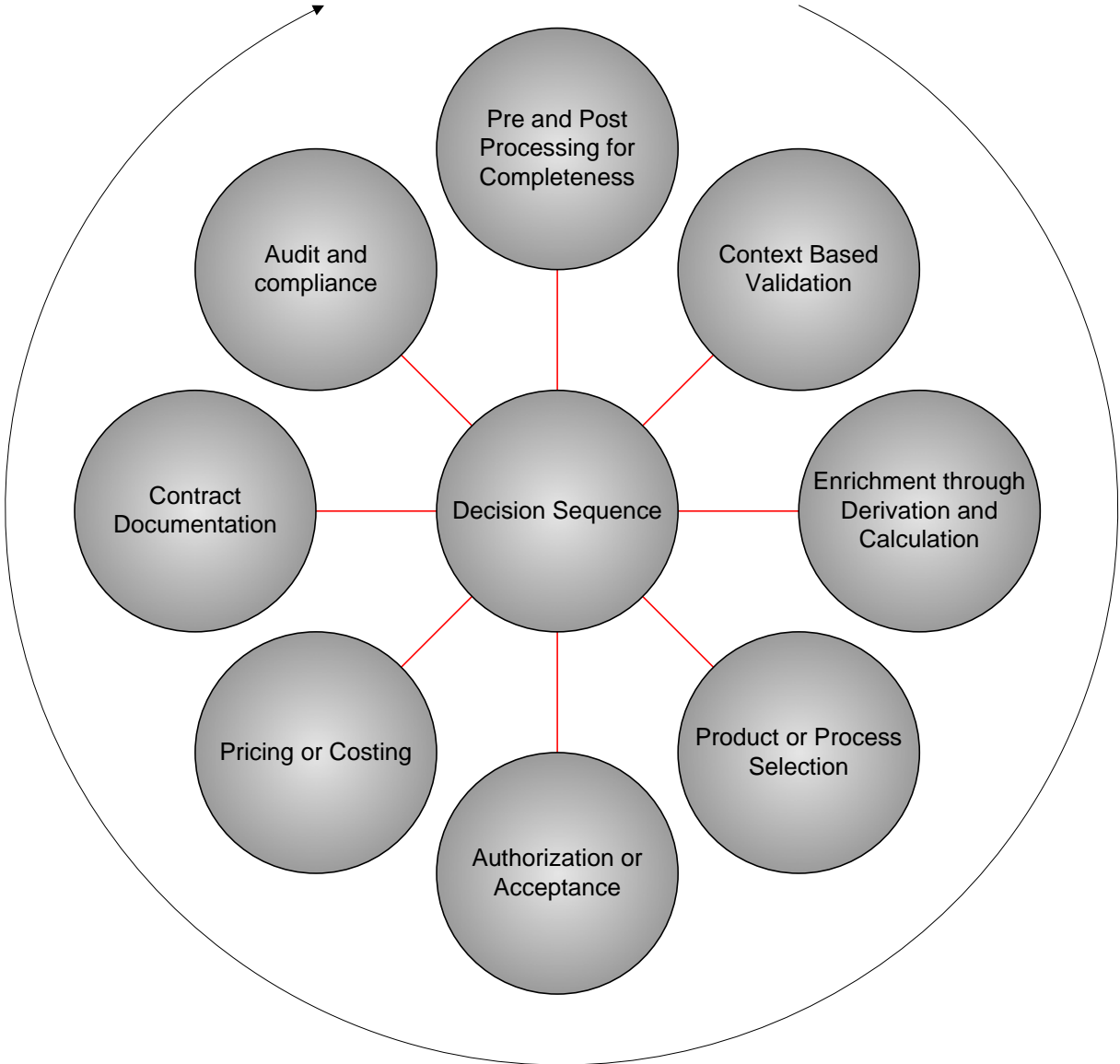




Typical applications for decision services

- Codifying and automating contracts
- Cleaning, consolidating, and transforming data
- Automating the validation, acceptance, valuation, and/or auditing of any pre-defined XML document, for example, an order or request within a B2B transaction.
- Controlling system behavior through dynamic configuration of system parameters
- Managing process flow by providing the decisions that control when and how processes execute
- Replacing human decision makers within business transactions

Common decision categories within a Service





Decisions should not constrain architecture

Many SOA integration options:

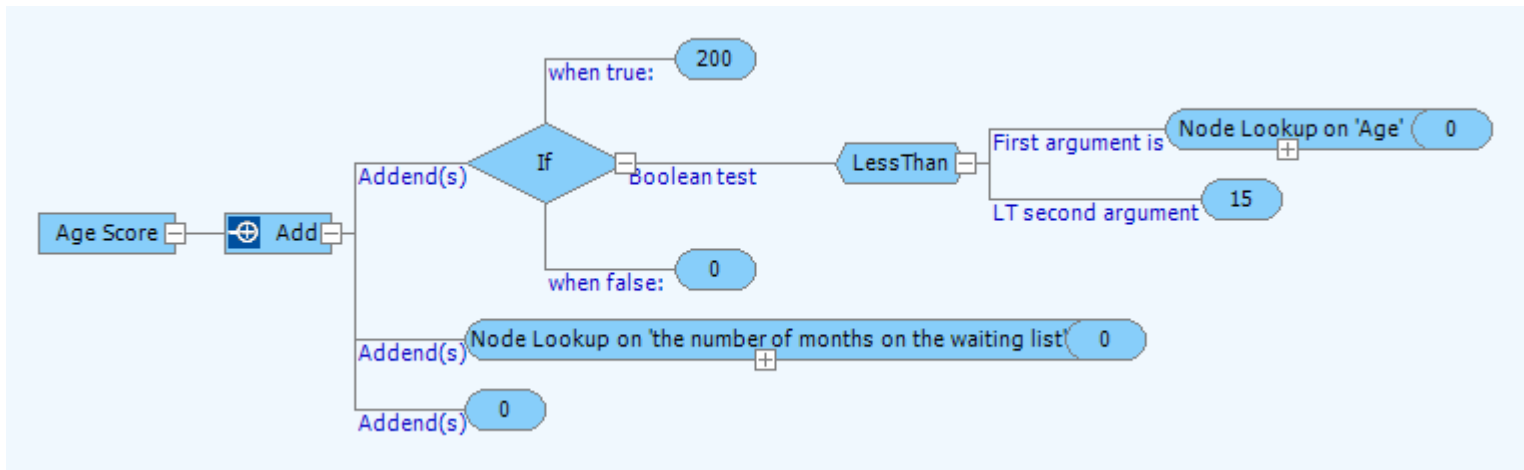
- Local call in the language of choice - Java, .NET, C++
- Web Services
- Messaging Service (MSMQ, MQSeries, Sonic)
- COM Wrapped DLL
- Proprietary service containers such as:
 - IBM's business rule beans; Crossworlds container; MQSI Node
 - Sonic Software ESB containers
 - Microsoft BizTalk Server



Healix demonstration

- **Donor Recipient Matching System** (National Kidney Allocation System)
- A simple system comprised only of rules and data handling adapters
- Specific decision structures are used to select the Recipient including
 - Social policy criteria
 - Clinical criteria
 - Criteria are scored to select the most suitable Recipient
- Rules were built in a few hours
- Rules are maintained personally by the Medical Director, without technical or BA support

Formula for Age Score (original)

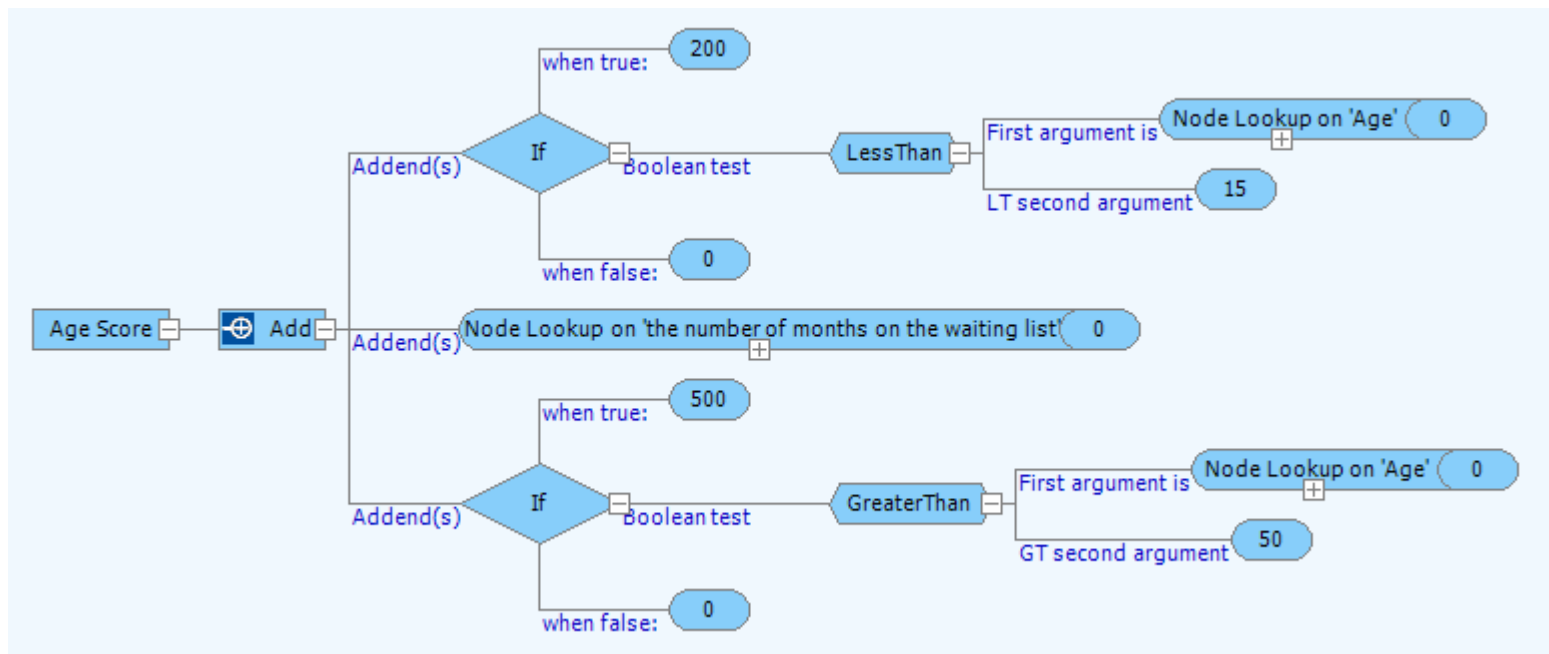


The formula "Age Score" is defined as follows.

Return A + the number of months on the waiting list + 0

- ⊕ A: When Age < 15
 - THEN return 200
 - ELSE return 0

Formula for Age Score (updated)



The formula "Age Score" is defined as follows.

Return A + the number of months on the waiting list + B

Ⓜ A: When Age < 15
 THEN return 200
 ELSE return 0

Ⓜ B: When Age > 50
 THEN return 500
 ELSE return 0



Summary

- What we have seen:
 - Decisions are the core services in any process in which they participate
 - Software developers now need to recognize and respond to this
- Why is it important to you?
 - Aligning software development to this reality improves many outcomes: “Greatly increased business agility - faster, at lower cost, and with less risk”
 - Service discovery and design starts with the decision-making services
- How can you benefit?
 - Start thinking about businesses in terms of the decisions they make and how they make them – it is what makes them, and you, succeed or fail!