CS377A – Introduction to Cybernetics

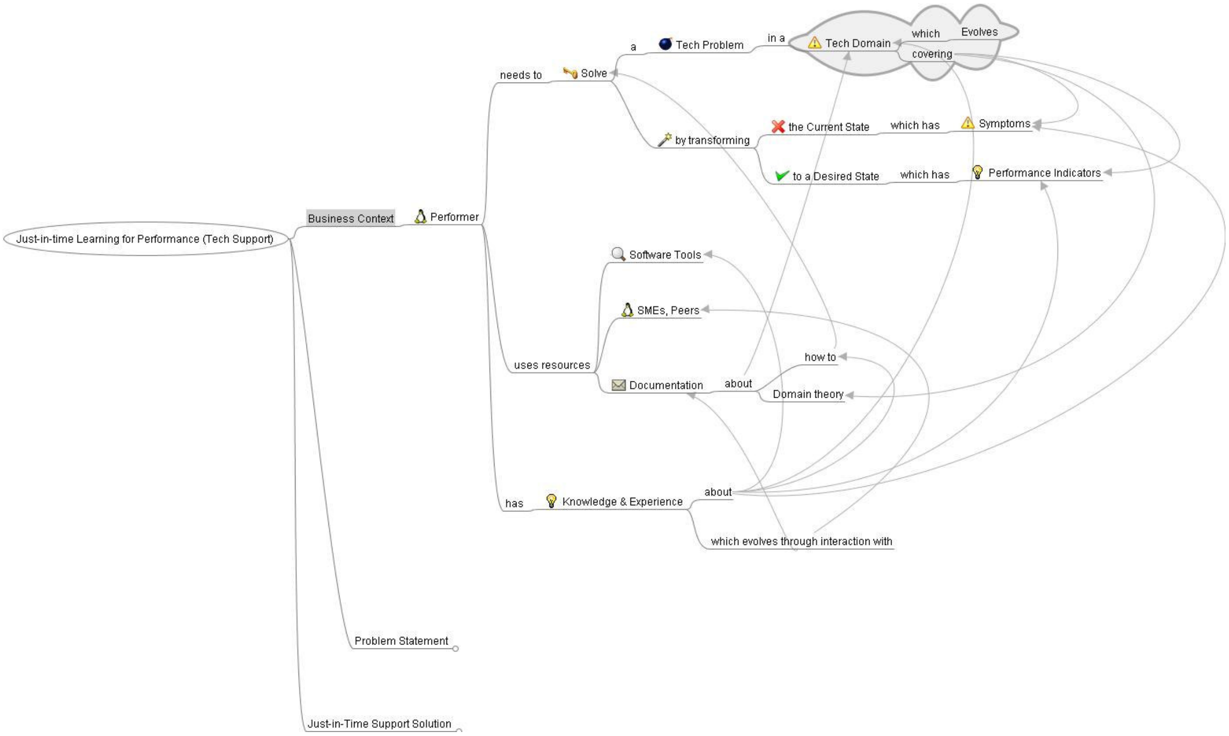
Haggai Mark (hmark@stanford.edu)

12/4/07

Final Project:
Just-in-Time Performance Support System (JIT PSS)

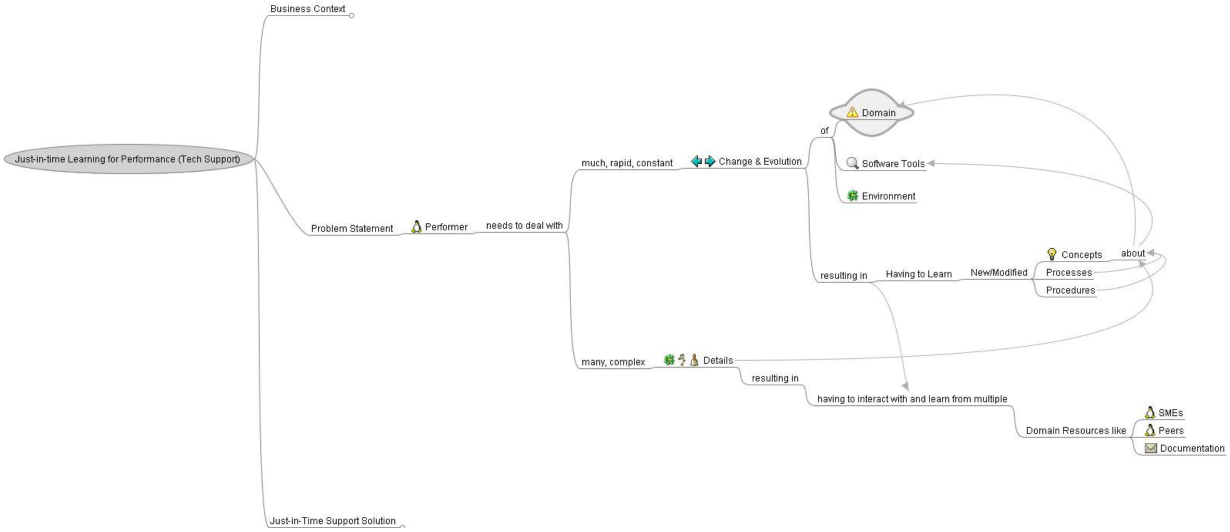
Business context (read top to bottom, left to right)

Briefly, a tech support worker (the performer, as well as learner), needs to be able to fix problems by transforming a given, “bad” state (e.g. malfunctioning computer network) into a desired state (where for example, the network works again).
In order to do that, the performer needs to be knowledgeable in the domain (e.g. computer networking), in related tools (e.g. networking-related tools such as network sniffers, configuration collectors, network management systems), and so on.
The performer also usually interacts with peers, Subject Matter Experts (SMEs), and product and domain documentation.



Business Problem

The current situation in many cases, is that the domain as well as the environment the performer is working in, are very complex and dynamic. The performer has to master both a lot of depth and breadth, many subjects, as well as many details, all of which are evolving fast. In complex domains (e.g. computer networking) this turns out to be a very hard problem, causing stress to the performer, many mistakes/misdiagnoses, lost time, productivity and money to the stakeholders involved.

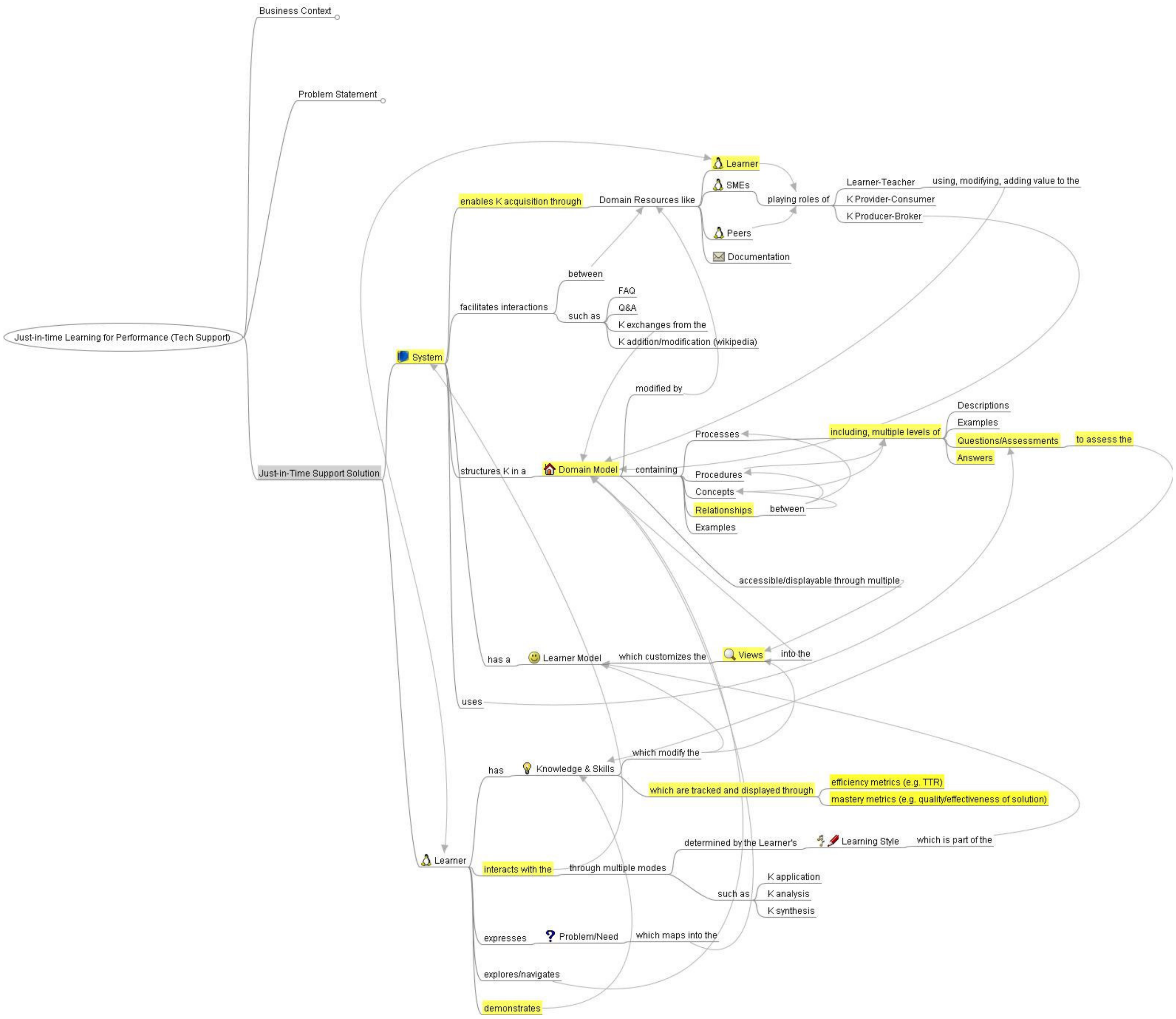


Proposed Solution (JIT PSS = Just-in-Time Performance Support System)

The typical situation, in most cases is that a tech support performer goes through initial training on the domain, tools, products, and available resources. In this process, they collect notes, tips, pointers, etc. and organize them somehow (e.g. in notebooks, binders, on the web, and so on). Sometimes, their employer provides some structure and tools (in many cases, no more than a search engine to an intranet (since “everything is on the web”… ☹) or a case-based database).

Some problems with this de-facto “solution” are: it puts most of the burden for learning, performing, and keeping current, on the performer. This approach doesn’t indicate to the performer what he knows and where his gaps are, so both the incentives and the metrics for improvement are not obvious. In most cases, it also doesn’t leverage “social networks” of peers, mentors, and SMEs to help the performer expand and excel. Also, there is no “continuous learning”, as the performer is usually “hit” periodically with updates, refreshers, etc., which in most cases come to him/her at random times, without any relevance to where s/he is (in terms of performance, skills, learning style), with no connection to his/her problem at hand.

The proposed solution, provides several improvements to the current state. It provides a structured framework for resources (both knowledge and people) a performer can leverage. It is continually updated with new knowledge, new people, new resource types (e.g. documentation, procedures, videos) and so on. It reflects the learner’s knowledge, skills, and interests (e.g. voice networks), as well as similar attributes of other people within the “social network” or “user community” interacting/conversing with the system and with each other. The system provides different ways to explore and learn, from informal browsing of the domain, to formal test taking and assessments.


Just-in-time Learning for Performance (Tech Support)
Business Context
Problem Statement
Just-in-Time Support Solution
System
enables K acquisition through
Domain Resources like
Learner
SMEs
Peers
Documentation
playing roles of
Learner-Teacher
K Provider-Consumer
K Producer-Broker
using, modifying, adding value to the
facilitates interactions
between
such as
FAQ
Q&A
K exchanges from the
K addition/modification (wikipedia)
modified by
structures K in a
Domain Model
containing
Processes
Procedures
Concepts
Relationships
between
Examples
including, multiple levels of
Descriptions
Examples
Questions/Assessments
Answers
to assess the
accessible/displayable through multiple
has a
Learner Model
which customizes the
Views
into the
uses
Learner
has
Knowledge &amp; Skills
which modify the
which are tracked and displayed through
efficiency metrics (e.g. TTR)
mastery metrics (e.g. quality/effectiveness of solution)
interacts with the
through multiple modes
determined by the Learner's
Learning Style
which is part of the
such as
K application
K analysis
K synthesis
expresses
Problem/Need
which maps into the
explores/navigates
demonstrates

Main business processes for the proposed JIT PSS

One of the important activities of the user community is to update the system with new capabilities and knowledge, through the knowledge acquisition process. The performer, peers and SMEs can enter new knowledge into the system, updating and extending its Domain Model. The Domain Model consists of Topic Nodes, associations/relationships, and resources (details to be described below). The system also maintains a User Model for each user within the community. Users can then exchange knowledge (somewhat analogous to "trading" portions of their User Domains, based on interests, needs, etc.).

Any user can express an interest or need (remember, the performer is trying to solve a technical problem) which will map his request to a relevant area(s) within the Domain Model, and based on the performer's "profile" (e.g. mastery level, preferred learning style) as captured in the User Domain.

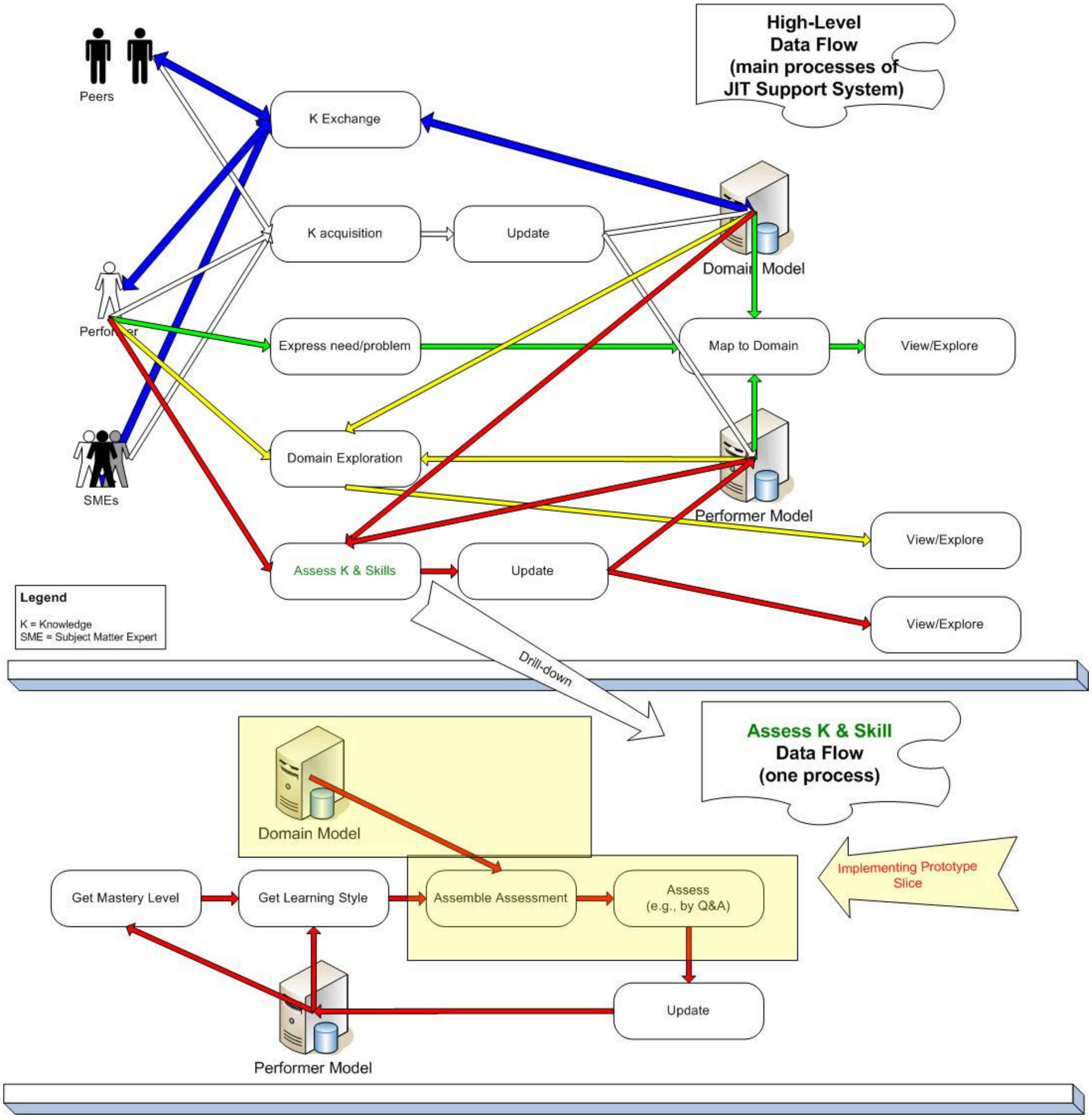
Any user can also explore the Domain Model (its topic nodes, relationships and resources) and thus expand his/her knowledge and skills. This in turn updates the User Model ("profile").

A user can also take formal assessments of knowledge and skills, in order to demonstrate mastery, advance in his/her career (certification), etc.

A few cybernetic concepts at this level:

The performer can enter new knowledge into the system (through the knowledge acquisition process). This knowledge can be discussed (conversation, real or virtual) and/or challenged by peers and SMEs, resulting in (hopefully) increased understanding by the participants, updates to the Domain Model, as well as the relevant/participating User Models.

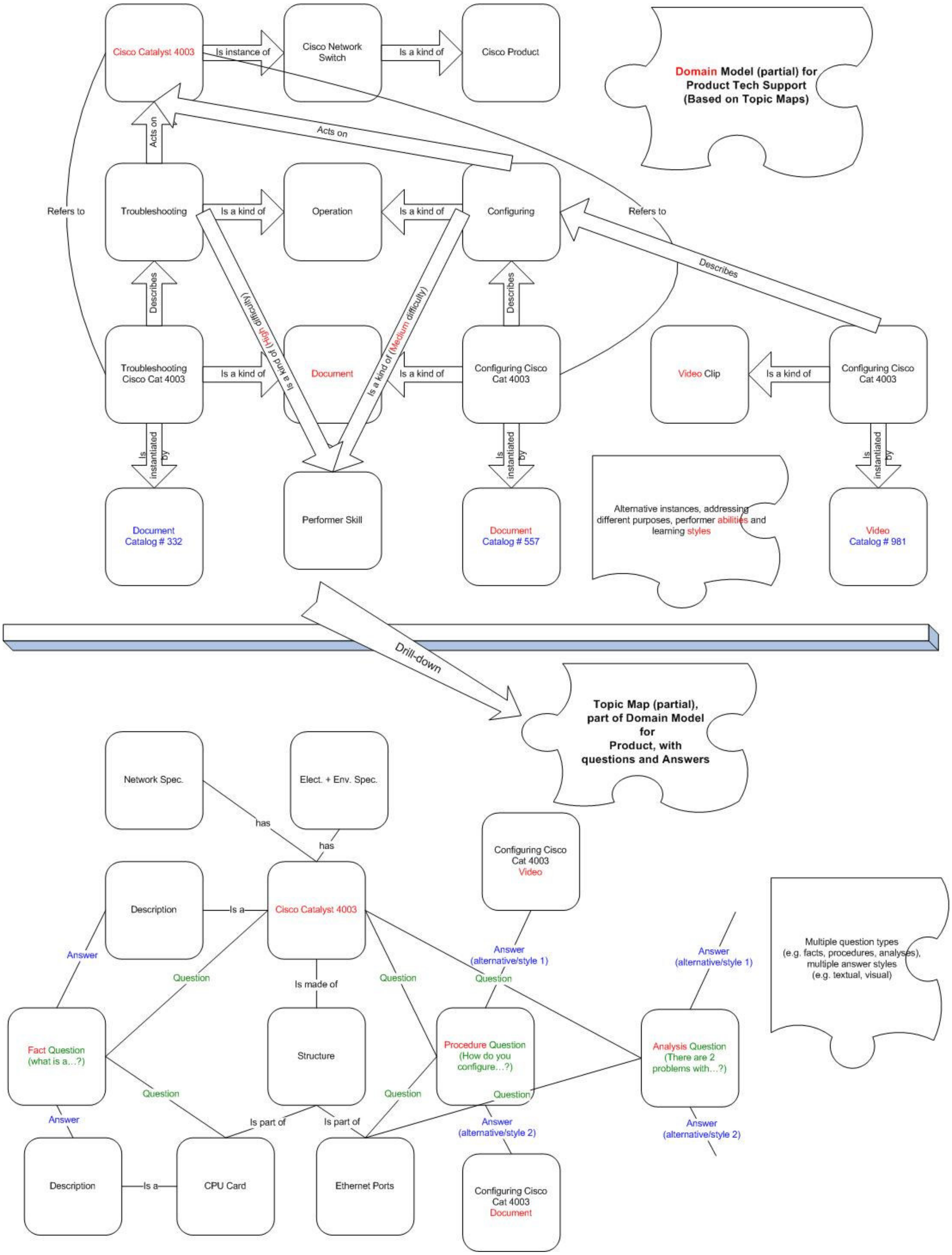
The performer can take some formal or informal training (through the Assess Knowledge and Skills process), which draws from the Domain Model, and the User Model (e.g. current mastery level, preferred learning style). The assessment can involve different levels of cognitive challenges (e.g. memorization, analysis, synthesis), and different kinds of tasks/questions (e.g. configuring a network device, troubleshooting a network segment). This, in turn, results in updating the User Model to reflect performer progress, domain coverage (areas of strength and difficulty), overall mastery level, and so on.


High-Level
Data Flow
(main processes of
JIT Support System)
Peers
Performer
SMEs
K Exchange
K acquisition
Update
Domain Model
Express need/problem
Map to Domain
View/Explore
Domain Exploration
Performer Model
View/Explore
Assess K & Skills
Update
View/Explore
Legend
K = Knowledge
SME = Subject Matter Expert
Drill-down
Assess K & Skill
Data Flow
(one process)
Domain Model
Implementing Prototype
Slice
Get Mastery Level
Get Learning Style
Assemble Assessment
Assess
(e.g., by Q&A)
Update
Performer Model

The Domain Model

The domain model I've selected is based on the Topic Maps (TM) ISO standards (**ISO/IEC 13250:2003**). It enables the capture of topics (e.g. a "Cisco 4003 LAN Switch"), relationships (e.g. "is an instance of a") with other topics (e.g. "Cisco Product"), and resources (e.g., a 360 degree video of the Cisco 4003 switch, or a JPG photo of the switch's front panel). The knowledge captured in a TM "resides" in both the topic structure and the relationships, as well as in the resources associated with those topics; in other words, a TM without any resources (i.e. "empty") is still a valuable knowledge source (or Domain Model). One important feature of TMs is that the relationships/associations are "first rate citizens", in other words, they can also be associated amongst themselves, as well as linked to resources of their own (a much more powerful model compared to HTML links and documents, for example).

The Domain Model captures knowledge about the domain, the tools, the kinds of resources, their type (e.g. media types like text/documents, video, graphics, simulation, and formats like Microsoft Word, mov, jpg), level of difficulty/mastery required, cognitive level (e.g. memorization, analysis, synthesis), and so on. The model also captures questions, hints (help topics), and answers, to be used for knowledge and skills assessment. All this rich information is used by the system to tailor performer interactions like domain exploration or assessments. It enables the assessment process, for example, to pick a topic from the Domain Model, select a question based on the user model (profile data such as current mastery level, type of cognitive level, learning style), and based on the performer/learner answer assess hi skill/mastery level, and select the next question, and so on.


Domain Model (partial) for Product Tech Support (Based on Topic Maps)

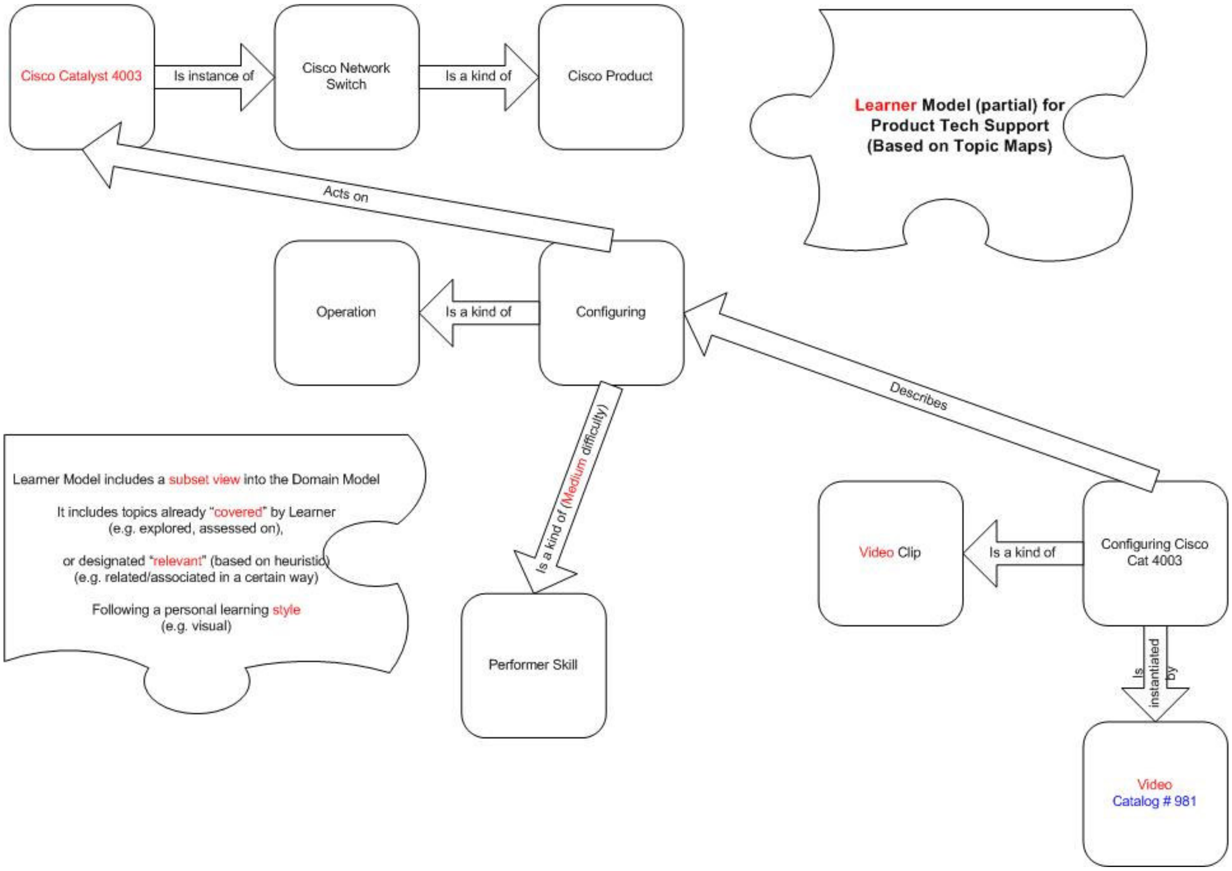
Alternative instances, addressing different purposes, performer abilities and learning styles

Topic Map (partial), part of Domain Model for Product, with questions and Answers

Multiple question types (e.g. facts, procedures, analyses), multiple answer styles (e.g. textual, visual)

User Model

The system keeps a User Model for each participant/user. The User Model is a “view” into the Domain Model, in the sense that it is a subset of the Domain, including topics, associations, and resources. The subset is determined by things like the topics the performer is interested in (or responsible for, in a case of a tech support help-desk), the topics they have already covered and mastered (through assessments/tests, for example), their mastery level (e.g., beginner or advanced topics), their learning style (e.g. audio-visual resources), permissions, and so on. This last “filter” (permissions) can be one mechanism for enabling knowledge exchange between the performer and his/her peers or SMEs. By, for example, a SME removing certain restrictions in the Domain Model, the performer gains access to additional areas in the Domain Model.

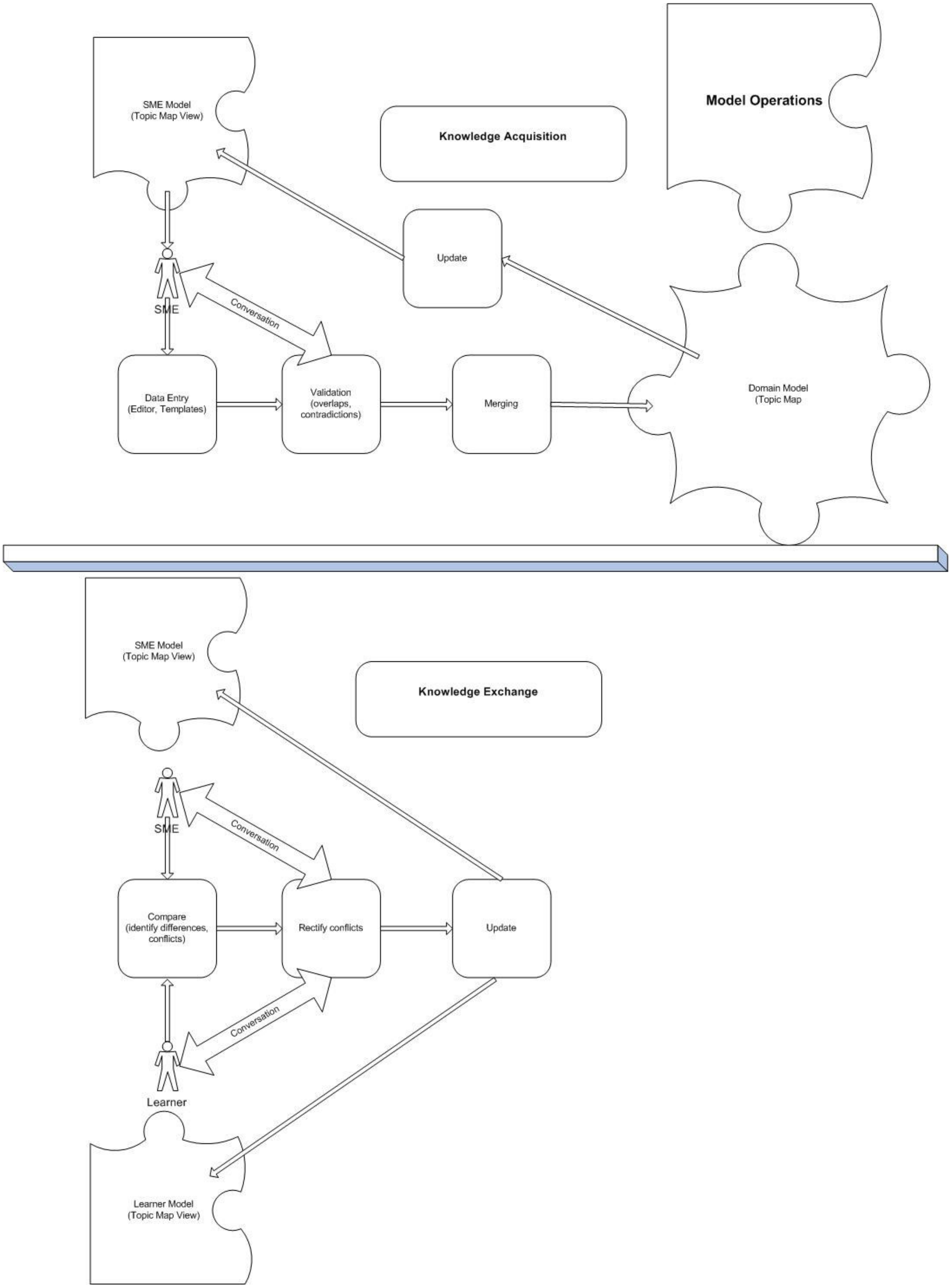

Cisco Catalyst 4003
Is instance of
Cisco Network Switch
Is a kind of
Cisco Product
Learner Model (partial) for Product Tech Support (Based on Topic Maps)
Acts on
Operation
Is a kind of
Configuring
Describes
Learner Model includes a subset view into the Domain Model
It includes topics already “covered” by Learner (e.g. explored, assessed on),
or designated “relevant” (based on heuristic) (e.g. related/associated in a certain way)
Following a personal learning style (e.g. visual)
Is a kind of (Medium difficulty)
Performer Skill
Video Clip
Is a kind of
Configuring Cisco Cat 4003
Is instantiated by
Video Catalog # 981

Some Domain Model Operations

The nature of the Domain Model (implemented using the Topic Map technology), enables a relatively straight forward implementation of certain operations covered above (under “Main business processes”).

For example, the knowledge acquisition process (where a system user enhances or modifies the Domain Model), consists of interactive sessions between user (performer, peer, SME) and system, through templates and editors, to allow the knowledge entry necessary to define topics, relationships, and resources. As part of this process, knowledge validation and verification needs to happen between the knowledge enterer and existing knowledge known to the system (i.e. already in the Domain Model) or to its users (e.g. SMEs, peers). This is a process of conversing, as understanding and agreement needs to be reached before the knowledge is incorporated. This newly entered knowledge will be reflected in the Domain Model, and in all relevant User Models (a cybernetic loop).

A similar conversation needs to happen when two users (e.g. SME and performer) decide to exchange some knowledge (e.g. a SME is mentoring a performer/learner, introducing him to a new area within the domain). There, too, there are loops for conversation, and understanding validation.


SME Model
(Topic Map View)
Knowledge Acquisition
Model Operations
Update
SME
Conversation
Data Entry
(Editor, Templates)
Validation
(overlaps,
contradictions)
Merging
Domain Model
(Topic Map
SME Model
(Topic Map View)
Knowledge Exchange
SME
Conversation
Compare
(identify differences,
conflicts)
Rectify conflicts
Update
Conversation
Learner
Learner Model
(Topic Map View)

The JIT Tutoring sub-system

The knowledge and skills assessment sub-system is described in a separate document. It is implemented as a second order cybernetic system, where the first order loop (FOL) implements a question-answer delivery system, drawing the questions, hints (help topics), answers, alternative questions, different media types, and so on, from the Domain Model, based on the User Model (profile). The goal of the FOL is to get the learner to perform at a given/set Mastery Level, within a given Topic or subject within the Domain.

The second order loop (SOC) is responsible progressing the learner through the various topics within the domain, while raising their mastery level. The goal of the SOC is to have the learner cover all domain topics (or all relevant topics, established by some criteria determined by the environment, such as the employer of the tech support employee), at a desired/highest level of performance/mastery.