Chapter 5

Knowledge Flow

Knowledge is power, but knowledge is not just statically stored. It evolves through being shared and developed by roles, people, and various resources within the cyber-physical-socio environment.

A *knowledge flow* is a passing of knowledge between people or through machinery. It has three crucial attributes: *direction* (sender and receiver), *carrier* (medium) and *content* (shareable). Good knowledge flow enables intelligent participants (people, roles and devices) to cooperate effectively.

5.1 Definition

Although knowledge flow is intangible, any teamwork relies on it, even if team members are unaware of its happening. They share knowledge using various forms of networking, where the links are knowledge flows working like the conveyor belts in a production line. Any team member can put knowledge onto the appropriate belt to have it automatically conveyed to the team member who needs it. Team members can be helped by knowledge from the "conveyor belts" connected to them when working on a task. The linkages of such "knowledge conveyor belts" together with the team members as active nodes make up a knowledge flow network. Designing the network properly, and controlling its operation effectively, will raise the efficiency of knowledge sharing within teams (H. Zhuge, "A Knowledge Flow Model for Peer-to-Peer Team Knowledge Sharing and Management", *Expert Systems with Applications*, 23(1)(2002)23-30).

Effective knowledge flow will avoid redundant knowledge passing between team members, recognizing that different members may be given different kinds of tasks and need different kinds of knowledge. Members then do not need to spend time and energy in searching for knowledge in a traditional centralized repository.

The carrier can be the Internet, local networks, various wireless networks, and even sensor networks. The content being shareable means that the knowledge can be understood by all team members. A connective network ensures that the content can be passed from any team member to any other member.

Knowledge content can be specified as being within a knowledge space where each point places knowledge of a specific type and level at a specific location (H. Zhuge, "A Knowledge Grid Model and Platform for Global Knowledge Sharing", *Expert Systems with Applications*, 2002, 22(4)(2002)313-320). Such a specification meets the following needs:

People working in different roles need knowledge at different levels.

People working at different kinds of tasks need different kinds of knowledge.

Thus the knowledge of a flow KF has a field (a two-dimensional region in a knowledge space) defined by a type field TFd and a level field LFd: $Field(KF) = <TFd, LFd>$, where $TFd = <t \mid t$ is a knowledge type$>$ and $LFd = <level \mid level$ is a knowledge level$>$.

The operation $TFd_1 \cup TFd_2$ is a set union such that the order of the knowledge types of each along the knowledge type axis is maintained. Similarly, the set operations $\cup$, $\cap$, $-$ can be carried out between any two TFds and between any two LFds.

Let $<TFd_1, LFd_1>$ and $<TFd_2, LFd_2>$ be the fields of two knowledge flows KF_1 and KF_2. The following operations hold:

(1) $<TFd_1, LFd_1> \cup <TFd_2, LFd_2> = <TFd_1 \cup TFd_2, LFd_1 \cup LFd_2>$;
(2) $<TFd_1, LFd_1> \cap <TFd_2, LFd_2> = <TFd_1 \cap TFd_2, LFd_1 \cap LFd_2>$;

(3) $<TFd_1, LFd_1> - <TFd_2, LFd_2> = <TFd_1 - TFd_2, LFd_1 - LFd_2>$;

(4) $<TFd_1, LFd_1> \subseteq <TFd_2, LFd_2>$ if and only if $<TFd_1 \subseteq TFd_2, LFd_1 \subseteq LFd_2>$.

A knowledge node, the sender or receiver of a flow, can also generate and request knowledge. What a node can put out depends on what knowledge it has stored and what it can get in. A node can be an automaton that holds its own store of knowledge and uses an agent to help team members use that knowledge.

When a knowledge node is working it is said to be *active*. Otherwise, it is *inactive*. A node switches between these states. Active nodes can self-organize a knowledge organization that can effectively share knowledge.

The following are properties of a good knowledge flow network.

(1) A knowledge flow network is *connective* if there is a flow path between every pair of nodes. A connective knowledge flow network requires a connective actual network, but a connective actual network does not ensure a connective knowledge flow network.
(2) A knowledge flow network is *complete* for a task if it is connective and its nodes correspond to the team members or their roles in the task. A complete network means that no team member is isolated from the knowledge of any other.
(3) A complete knowledge flow network is the *smallest* if it has the fewest possible flows between nodes. A smallest network can not only eliminate isolation but also achieve effective team knowledge sharing.
(4) A smallest complete knowledge flow network has no redundant paths between any two nodes.

5.2 A Knowledge Flow Process Model

Knowledge can flow through any of the following four types of connections:

Sequential connection. Two flows, KF_1 and KF_2 merge into one, KF_1/KF_2, such that a) $Field\,(KF_1/KF_2) = Field\,(KF_1) = Field\,(KF_2)$, or b) $Field\,(KF_1/KF_2) = Field\,(KF_2)$ if $Field\,(KF_1) \subseteq Field\,(KF_2)$.
Join-connection. Two or more flows converge to form one, denoted by $KF_1 \wedge KF_2 \wedge \ldots \wedge KF_n \Rightarrow KF$, such that $Field\,(KF_1 \wedge KF_2 \wedge \ldots \wedge KF_n \Rightarrow KF) = Field\,(KF_1) \cup Field\,(KF_2) \cup \ldots \cup Field\,(KF_n) = <LFd_1 \cup LFd_2 \cup \ldots \cup LFd_n, TFd_1 \cup TFd_2 \cup \ldots \cup TFd_n>$.

Split-connection. A flow KF can be split into two or more flows, denoted by $KF \Rightarrow KF_1 \vee KF_2 \vee \ldots \vee KF_n$, such that $Field\,(KF \Rightarrow KF_1 \vee KF_2 \vee \ldots \vee KF_n) = Field\,(KF_1) \cup Field\,(KF_2) \cup \ldots \cup Field\,(KF_n) = <LFd_1 \cup LFd_2 \cup \ldots \cup LFd_n, TFd_1 \cup TFd_2 \cup \ldots \cup TFd_n>$.

Broadcast. A flow KF can be broadcast to many flows $KF_1, KF_2, \ldots, KF_n$ such that $Field\,(KF = (KF_1, KF_2, \ldots, KF_n)) = Field\,(KF_1) = Field\,(KF_2) = \ldots = Field\,(KF_n)$.

The difference between workflow (http://www.wfmc.org) and knowledge flow has the following aspects:

(1) A knowledge flow can take in the knowledge generated at a node as it flows through it. Workflows do not.
(2) Much knowledge flow content comes from team members' experience carrying out a task and cannot be anticipated. Workflow networks reflect existing business domains and can be designed.
(3) Knowledge flow content comes from team members, while workflow content reflects either data or execution dependence between activities (tasks).

For teamwork, the knowledge flow can be made consistent with the workflow by having the same roles in both networks.

5.3 Peer-to-Peer Knowledge Sharing

Team members are called peers if they do the same work for the same type of tasks at the same level of the organizational hierarchy. Knowledge sharing makes use of the knowledge within a team to solve problems more quickly or effectively. Sharing between peers is more effective than that between non-peers for the following reasons.

(1) Peers work on the same types of tasks so their experiences are more relevant for sharing with each other to solve their problems.
(2) Peers have similar knowledge structures so can understand each other more easily when sharing knowledge.
(3) Peers have more interests in common so they can more effectively share knowledge. For example, two programmers can better share programming knowledge than either can with a manager.

Organizational innovation is one of the key issues of knowledge management. A successful large-scale knowledge organization tends to have fewer middle layers than an unsuccessful one. Organizations in some domains, like orchestras, may even have no middle layers at all (P.F. Drucker (ed.), "Harvard Business Review on Knowledge Management", *Boston, MA: Harvard Business School Press*, 1998). So peer-to-peer knowledge sharing is also a useful aim in structuring a large-scale organization.

Example. Software development by distributed teams focuses on work cooperation and resource sharing between physically dispersed team members during the development. Research on such work focuses only on aspects of technique. Human cognitive characteristics are seldom addressed. The following are reasons for incorporating formal knowledge flow into software development by distributed teams:

(1) *Software development is a knowledge-intensive process*. Team members can improve their work not only by using software tools but also through cognitive cooperation.

(2) *Cognitive cooperation cannot be planned*. Team members' development knowledge is gained and gathered as their work proceeds, so cognitive cooperation among them cannot be planned, though it must be encouraged. Cooperation in the form of knowledge flows is essential.
(3) *A distributed team requires effective and low cost communication*. Planned and disciplined knowledge flow can cut the cost of communication and can better reflect the actual work process of project development.
(4) *A development team should be supported by a formal experience accumulation procedure*. All team members can use the experience of their predecessors accumulated while working on previous projects, so that the team can avoid fruitless work and adapt to any change of participants or of roles.

There are five cognitive levels of knowledge in software development, given here from low to high.

(1) *Coding knowledge* helps members to share programming skills. The skills of this level are in the form of problem-solution pairs.
(2) *Reuse knowledge* helps members to reuse code components.
(3) *Knowledge of methods* enables team members to apply known problem solving techniques. Such knowledge is in the form of problem-method pairs, where a method can be a process, a pattern, or an algorithm.
(4) *Rules for development and cooperation* encourage team members to share knowledge and experience, which flow on to others to improve their software development generally. Rules for cooperation can make sharing more efficient, and are very useful for bringing new members successfully into a team.
(5) *Decision and evaluation knowledge* is meta-knowledge gained from developing the knowledge of the other four levels. It reflects the manner of making decisions during the development process, and provides guidance in making new decisions, even in quite new circumstances.

5.4 Knowledge Intensity

Knowledge flows can be used to transfer capability and expertise in an orderly and effective way. The major obstacle is the absence of criteria for assessing the effectiveness of a knowledge flow network and for ensuring its optimal operation. Effectiveness lies in essence in having a good path for needed knowledge to flow from where it resides to where it is needed—across time and space and within and between organizations as necessary.

Knowledge intensity is a critical parameter in this process, whereby a team member with profound knowledge is qualified to occupy a position of very high intensity in the flow network. Good management will keep knowledge flowing from those who are more knowledgeable to those who are less, and so avoid wasted flows.

The notion of intensity reflecting degree of knowledge leads to principles that provide objective laws for the existence and development of effective knowledge flow.

To set up a reasonable scope for research, the following assumptions specify the nature of equality, autonomy, and generosity in knowledge flow networking.

Assumption. *Nodes in a knowledge flow network are able to acquire, use and create knowledge*. It is reasonable to assume that people in an organization all have some ability to generate, use and spread knowledge.

Assumption. *Knowledge nodes share knowledge autonomously*. This limits research to the passing and sharing of knowledge among nodes independently, without outside influence. Then we can just focus on team members' effectiveness and the needs of the task at hand when designing knowledge flow networks.

Assumption. *Nodes share useful knowledge without reserve*.

Knowledge within a team usually covers several areas, classified according to discipline. Knowledge can be also classified into five levels as outlined above (H. Zhuge, "A Knowledge Grid Model and Platform for Global Knowledge Sharing", *Expert Systems with Applications*, 22(4)(2002)313-320).

Knowledge area and level are two dimensions of knowledge space. An area i and a level j determine a *unit knowledge field* (or unit field for short) denoted by UFd (i, j).

Knowledge intensity is a parameter that expresses a node's degree of knowledge and reflects the corresponding person's cognitive and creative abilities in a unit field. The intensity of a knowledge node and its change determine the node's "rank" in a network. It is in direct proportion to the aggregate knowledge held by the node.

A node with superior knowledge and ability to learn, use and create knowledge will be of high intensity. Thus, we estimate the intensity of a node in a unit field by assessing how much knowledge in the unit field is held by the node.

The knowledge intensity of a node will be different in different unit fields. We use the following four-dimensional orthogonal space *KIS* to represent the knowledge intensity of a knowledge node:

KIS (*knowledge-area*, *knowledge-level*, *knowledge-intensity*, *time*).

Any point in this space represents the knowledge intensity of a node in a certain unit field at certain time. At the given time t, the intensity of node u in unit field UFd (i, j) for a given *task* is KI $(task, u, i, j, t)$.

In every unit field some nodes need to pass knowledge to others. We can define a knowledge flow network for every unit field with flows that avoid unnecessary knowledge passing. Cooperation within a task can involve many networks, one for each unit field.

5.5 Knowledge Flow Principles

It is ineffective if knowledge flow through persons with the same knowledge structure. Therefore, we have the following principle.

Principle. *Knowledge only flows between two nodes when their intensity differs in at least one unit field*.

This can be formally expressed as follows. Let u and v be two knowledge nodes, and $KI(task, u, i, j, t)$ and $KI(task, v, i, j, t)$ be their intensity in $UFd(i, j)$ at time t. If the following formula holds, knowledge will flow between u and v.

$$\exists i \exists j (KI(task, u, i, j, t) - KI(task, v, i, j, t)) \neq 0$$

Principle. *A knowledge flow network is effective if and only if every flow is to a node of lower intensity than its source*.

This can be formally expressed as follows. If u_k is any node in a knowledge flow network, with u_{k-1} its predecessor, then the network is effective if the following formula holds.

$$\forall k (KI(task, u_{k-1}, i, j, t) - KI(task, u_k, i, j, t)) > 0 .$$

Just as for water or electricity, knowledge naturally flows from high intensity nodes to low intensity nodes.

Principle. *The intensity difference between any two nodes in a knowledge flow network always tends to zero. That is, the following formula holds*:

$$\forall i \forall j \lim_{t \to \infty} (KI(task, v, i, j, t) - KI(task, u, i, j, t)) = 0 .$$

Let nodes *u* and *v* be the two ends of a knowledge flow in unit field *UFd* (i, j). If they share their useful knowledge *without reserve*, the one with lower intensity will learn from the other, and the difference in their knowledge intensity in *UFd* (i, j) will become smaller and smaller with the passing of time. This effect will be apparent in a closed environment, one in which there is no flow into the network from outside. In such an environment, all nodes are likely to have similar knowledge in the long term simply from learning together and sharing. All flow could stop. This principle implies that a team will improve its performance more by learning from outside the team than by only exchanging knowledge within the team.

Principle. *If knowledge depreciation is ignored, the intensity in any unit field at any node will never decrease.*

If *KI* $(task, u, i, j, \Delta t)$ is the change in knowledge intensity of node *u* in *UFd* (i, j) in a period of time $\Delta t > 0$, then *KI* $(task, u, i, j, \Delta t)$ should not be negative. So we have:

$$KI(task, u, i, j, \Delta t) = KI(task, u, i, j, t + \Delta t) - KI(task, u, i, j, t) \geq 0$$

Knowledge depreciation can be ignored if the flow duration is rather short or the depreciation rate in the unit field is quite low.

When the intensity at a node changes, the knowledge flow network should be reformed if it will improve the flow.

In a competitive team, each node will attempt to increase its intensity so as to raise its position and rewards. This incentive inspires team members to learn, create and contribute as much as possible.

5.6 Computational Model of Knowledge Intensity

5.6.1 *Computing knowledge intensity in a closed environment*

We first discuss the simple case where node *v* is the only predecessor of node *u*, *u* is the only successor of *v*, and the knowledge intensity of node *v* is a constant. The knowledge intensity of node *u* in unit field *UFd* (i, j) at time *t*, KI_{closed} (*task, u, i, j, t*), has the following features:

(1) KI_{closed} (*task, u, i, j, t*) monotonically increases.
(2) KI_{closed} (*task, u, i, j, t*) tends to that of its predecessor in the long term. So the eventual stable value of KI_{closed} (*task, u, i, j, t*) is that of its predecessor, that is, $KI_{fs} = KI_{closed}$ (*task, v, i, j*, 0).
(3) The rate of increase of KI_{closed} (*task, u, i, j, t*) is in direct proportion to two factors; one is its current intensity and the other is the ratio of its difference from its possible stable value of KI_{fs} to the value of KI_{fs}.

From the above analysis, we obtain the following non-linear differential equation, where λ is the proportionality coefficient and KI_{u0} is the initial intensity of *u*:

$$\begin{cases} \dfrac{dKI_{closed}(task, u, i, j, t)}{dt} = \lambda\left(\dfrac{KI_{fs} - KI_{closed}(task, u, i, j, t)}{KI_{fs}}\right)KI_{closed}(task, u, i, j, t) \\ KI_{u0} = KI_{closed}(task, u, i, j, 0) \end{cases}$$

The following is the solution of above equation:

$$KI_{closed}(task, u, i, j, t) = \frac{KI_{fs}}{1 + \left(\dfrac{KI_{fs}}{KI_{u0}} - 1\right)e^{-\lambda t}}$$

While the knowledge intensity at the source node *v* changes with time, consequent intensity change at *u* comes after that at its predecessor.

Let the intensity of node v in unit field $UFd(i, j)$ at time t be $KI_{closed}(task, v, i, j, t)$. Then we have the following equation:

$$\begin{cases} \dfrac{dKI_{closed}(task,u,i,j,t)}{dt} = \lambda\left(\dfrac{KI_{closed}(task,v,i,j,t) - KI_{closed}(task,u,i,j,t)}{KI_{closed}(task,v,i,j,t)}\right)KI_{closde}(task,u,i,j,t) \\ KI_{u0} = KI_{closed}(task,u,i,j,0) \end{cases}$$

The following is the general solution of the above equation, where C_1 is a constant:

$$\begin{cases} KI_{closed}(task,u,i,j,t) = \dfrac{1}{C_1 e^{-\lambda t} + \lambda e^{-\lambda t}\int_0^t \dfrac{e^{\lambda t}}{KI_{closed}(task,v,i,j,t)}dt} \\ KI_{u0} = KI_{closed}(task,u,i,j,0) \end{cases}$$

We can get the solution of $KI_{closed}(task, u, i, j, t)$ by replacing $KI_{closed}(task, v, i, j, t)$ in the above formula with the appropriate expression.

For example, in a closed team composed of three nodes, let a be the predecessor of b, and b be the predecessor of c. Let $KI_{closed}(task, a, i, j, t)$ be a constant KI_{a0}. Then we get b's intensity function from the above formula as follows:

$$KI_{closed}(task,b,i,j,t) = \frac{KI_{a0}}{1 + \left(\dfrac{KI_{a0}}{KI_{b0}} - 1\right)e^{-\lambda t}}.$$

And c's intensity function is

$$KI_{closed}(task,c,i,j,t) = \frac{KI_{a0}KI_{b0}}{KI_{b0} + (KI_{a0} - KI_{b0})\lambda t e^{-\lambda t} + \dfrac{(KI_{a0} - KI_{c0})KI_{b0}}{KI_{c0}}e^{-\lambda t}}.$$

5.6.2 *Computing knowledge intensity in an open environment*

In an open environment, a knowledge node can learn from the external environment as well as from within its team.

Let $KI_{open}(task, u, i, j, t)$ (in short, $KI(u, t)$) be the knowledge intensity value of node u in $UFd(i, j)$ at time t in an open environment. It is composed of the intensity coming from within the team (denoted by $KI_{in}(u, t)$) and that from the external environment (denoted by $KI_{out}(u, t)$). The overall intensity at node u is thus

$$KI(u,t) = KI_{in}(u,t) + KI_{out}(u,t) \ .$$

In an open environment, the nodes that have higher intensity will absorb knowledge more rapidly than those with lower. The rate of increase of $KI_{out}(u, t)$ is in direct proportion to u's intensity. And when the intensity of predecessor node v is higher than that of u, $KI_{in}(u, t)$ can be computed as described above.

Therefore, we can obtain the following non-linear differential equations, where λ and δ are proportionality coefficients and KI_{u0} is the initial knowledge intensity of u:

$$\begin{cases} \begin{cases} KI(u,t) = KI_{in}(u,t) + KI_{out}(u,t) \\ \dfrac{dKI_{out}(u,t)}{dt} = \delta KI(u,t) \\ \dfrac{dKI_{in}(u,t)}{dt} = \lambda\left(\dfrac{KI(v,t) - KI(u,t)}{KI(v,t)}\right) KP(u,t) \\ KI_{u0} = KI(task, u, i, j, 0) \end{cases}, & if\ KI(v,t) - KI(u,t) > 0 \\ \begin{cases} KI(u,t) = KI_{in}(u,t) + KI_{out}(u,t) \\ \dfrac{dKI_{out}(u,t)}{dt} = \delta KI(u,t) \\ KI_{in}(u,t) = 0 \\ KI_{u0} = KI(task, u, i, j, 0) \end{cases}, & if\ KI(v,t) - KI(u,t) \le 0 \end{cases}$$

Its general solution is as follows, where C_2 is a constant:

$$\begin{cases} \begin{cases} KI(u,t) = \dfrac{1}{C_2 e^{-(\delta+\lambda)t} + \lambda e^{-(\delta+\lambda)t} \displaystyle\int_0^t \frac{e^{(\delta+\lambda)t}}{KI(v,t)} dt}, & if\ KI(v,t) - KI(u,t) > 0 \\ KI_{u0} = KI(u,0) \end{cases} \\ KI(u,t) = KI_{u0} e^{\delta t}, & if\ KI(v,t) - KI(u,t) \le 0 \end{cases}$$

By appropriately replacing *KI* (*v, t*) in the above formula, we can get the solution for *KI* (*u, t*).

Using this approach, we can compute changes in knowledge intensity at nodes of a network in a closed or open environment based on their initial intensities, their learning ability and their predecessor nodes' intensities.

The following will discuss how to estimate the initial knowledge intensity.

5.6.3 Knowledge Intensity Evaluation

Knowledge can be explicit or tacit. Explicit knowledge is expressible, linguistic, and simple to encode. Tacit knowledge comes more from experience and intuition, and is therefore much more difficult to pass on (K.C. Desouza, "Facilitating Tacit Knowledge Exchange", *Communications of the ACM*, 46(6)(2003)85-88; I. Nonaka, "A Dynamic Theory of Organizational Knowledge Creation", *Organization Science*, 5(1)(1994)14-37).

Explicit knowledge is easy to assess by using objective methods such as statistics. Tacit knowledge is difficult to assess because it is often at least partly subconscious (M. Mitri, "Applying Tacit Knowledge Management Techniques for Performance Assessment", *Computers & Education*, 4(2003)173-189).

Combining an objective evaluation approach with a subjective one could be a good way to assess knowledge intensity (H. Zhuge, "A Dynamic Evaluation Approach for Virtual Conflict Decision Training", *IEEE Transactions on Systems, Man and Cybernetics*, 2000, vol.30, no.3, pp.374-380; H. Zhuge and J. Liu, "A Fuzzy On-line Collaborative Assessment Approach for Knowledge Grid", *Future Generation Computer Systems*, 20(1)(2004)101-112).

The objective approach uses the quantity and quality of a node's explicit knowledge. The subjective approach uses questionnaires for completion by the node itself and by others, and assessment of achievement. Although the tacit knowledge and the cognitive and creative abilities of a node are hard to assess, they can be inferred subjectively to some extent. The node with more knowledge always emits better information and one with more ability always gets better evaluations.

5.7 Knowledge Spiral Model

Knowledge spirals are formed when knowledge flows in networks. A node can deliver knowledge to its successors either by forwarding knowledge from a predecessor, or by passing on its own.

Fig. 5.1 depicts a knowledge spiral, which consists of nodes with two types of flow: external—knowledge passed between nodes; and internal—knowledge created at a node, for example through abstraction, analogy, synthesis or reasoning.

The knowledge spiral model is very similar to the hypercycle model (K. Oida, "The Birth and Death Process of Hypercycle Spirals", in: R.K. Standish, M.A. Bedau, H.A. Abbass, edd., *Artificial Life VIII*, *MIT Press*, 2002). The self–replication arc and the catalytic–support arc of the hypercycle correspond to the knowledge passing and the knowledge processing respectively. The differences are twofold: self–replication in a hypercycle is carried out within nodes but knowledge passing is between nodes; and catalytic–support in a hypercycle happens between nodes but knowledge processing happens within nodes.

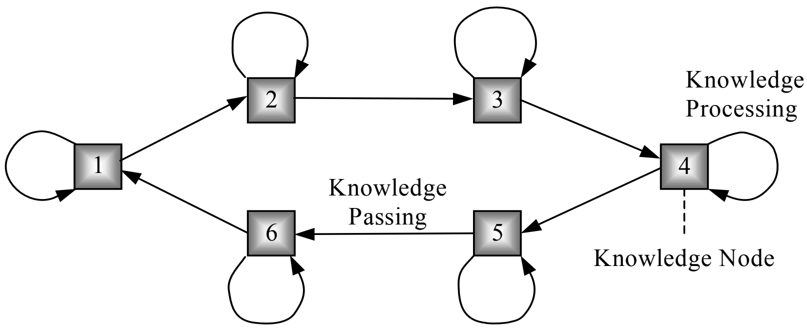

Fig.5.1 Knowledge spiral process model.

An effective knowledge spiral should maintain the intensity differences between nodes and ensure that only needed knowledge is passed between nodes. The processing at a knowledge node can be modeled as an automaton. Knowledge can also be modeled in a conflict environment (H. Zhuge, "Conflict decision training through multi-space cooperation", *Decision Support Systems*, 29(2000)111-123).

In general, work and knowledge both flow within a team (H. Zhuge, "Workflow-based cognitive flow management for distributed team cooperation", *Information and Management*, 40(5)(2003)419-429). A team member can take on one or more roles, and a role can also be part of other roles. Some roles take part in knowledge flow spirals and others carry out the tasks specified in work lists.

A knowledge flow spiral can be in one of the following four states.

(1) *Static*: creating and storing knowledge.
(2) *Active*: fulfilling roles.
(3) *Suspension*: waiting for something.

(4) *Termination*: reaching either the successful or the unsuccessful exit node.

5.8 Knowledge Flow Network Planning

Planning the knowledge flow network for a team means describing and designing a network free of unnecessary flows so that the network is efficient and effective. The success of the planning depends on the experience of the planner. Planning a large network is time consuming and may need a team of planners. Without an agreed abstraction method, planners will find it hard to work together and to bring about a coherent plan. These difficulties are the main obstacles to planning successful large knowledge flow networks.

5.8.1 *Composition operations and principles*

A knowledge flow network can be made from two or more existing networks by using the following composition operations.

(1) *Merge*: overlay common nodes.
(2) *Add flow*: connect nodes between networks.
(3) *Add condition*: add a *join* or *split* to express the relationship between flows related to the same node.
(4) *Embed*: put one network entirely within a node of another.
(5) *Graph operations*: combine networks with union, intersection, or subtraction.

Flows should be added whenever nodes have unit fields in common. Conditions should be added when a node is itself a network.

Composing knowledge flow networks also involves composing their roles. Let Rel_i be the relationship between the roles in $RoleSet_i$, $Roles_1 = <RoleSet_1, Rel_1>$ and $Roles_2 = <RoleSet_2, Rel_2>$ the role models of two networks KFN_1 and KFN_2 of the same team (maybe created by different planners), and KFN the union of KFN_1 and KFN_2 (that is,

$KFN_1 \cup KFN_2$). The role model of *KFN* can be obtained by using the following union operation:

$$Roles = Roles_1 \cup Roles_2 = <RoleSet_1 \cup RoleSet_2, Rel_1 \cup Rel_2>.$$

People, teams and tasks are the three main considerations in building a knowledge flow network. Composition of networks should respect the following principles:

The flow effectiveness principle. Composition of knowledge flow networks should ensure the effectiveness of the composed network. Effectiveness will be achieved if flows in the same chain share the same knowledge space or subspace so that the right knowledge can be delivered to the node in need of it, and so that the content of a flow can be stored at the right node. Where there are intensity differences between nodes, knowledge flow is only effective from the node with higher intensity to that with lower.

The organizational effectiveness principle. Composition of knowledge flow networks will not be effective unless it meets the regulations and targets of the team. If the composition requires that the team expand then the expansion should help meet regulations and targets, for example in respect of profit, security and copyright.

The task relevancy principle. Knowledge gained by the composite team should help the team complete its tasks. If knowledge resulting from the composition does not help task completion, then the composition is ineffective.

The mutual benefit principle. All members of the team should benefit from the composition, for example by gaining helpful knowledge or by increase in some reward. Otherwise, the team may suffer from lessened cooperation in the long run.

The minimum coverage principle. The composite knowledge flow network should be the smallest that includes all the nodes and flows of the original networks. In other words, there must be no redundant flows or nodes. Otherwise effective knowledge sharing cannot be assured in the composite network.

The trust principle. Effective cooperation requires that team members trust each other as much as possible.

5.8.2 *Knowledge flow network components*

A large–scale building block used in the design of knowledge flow networks is the knowledge flow component. It is a knowledge flow network that is *independent*, *encapsulated*, and *complete*.

Independence. Processing within a component should be relatively independent of that in other components. Consequently the density of knowledge flow paths within a component is usually higher than that between components.

Encapsulation. It can itself be used as a knowledge node. A knowledge flow component can be normalized to have just one initial node and one successful final node. Any external knowledge flow can only use the component through those two nodes.

Completeness. The knowledge flow process is complete in both build–time (definition phase) and run–time (execution phase).

A knowledge flow network component is called *definition complete* if: (1) every internal node has at least one input and one output flow, (2) every internal flow except from the final node goes to an internal node, (3) the final node can be reached from the initial node, and, (4) there is no isolated node or subnetwork.

Execution completeness requires that all restrictions and conditions be met during execution, and that the execution of the knowledge flow component can be treated as that of a single knowledge node.

Components can be used to compose a knowledge flow network. Using known and well-understood patterns of flow can help planners compose effective new networks in the same way that using design patterns leads to effective software engineering (E. Gamma, et al., "Design patterns: elements of reusable object-oriented software", *Pearson Education*, 1995). It can also promote understanding between planners.

A knowledge flow network pattern is an abstraction of a mode of teamwork. In the pattern, every node should be reachable from every other node via a path of nodes and flows under certain constraints. The flow characteristic of the pattern is peer–to–peer.

Further work needs to be done on the following aspects:

(1) Mathematical models for adapting a knowledge flow network to new conditions;
(2) Algorithms for matching patterns and components and for selecting usable ones; and,
(3) Approaches that consider intention, trust and belief (B.J. Grosz and S. Kraus, "Collaborative plans for complex group action", *Artificial Intelligence*, 1996, vol.86, pp.269-357).

5.8.3 *The team organization principle*

Trust between team members is an important factor that affects team cooperation. People more trusted by current team members should be preferred when recruiting.

The distribution of page ranks of the Web obeys the "power-law" and "the rich get richer" rules, and so do aspects of many other networks (L.A. Adamic and B.A. Huberman, "Power-Law Distribution of the World Wide Web", *Science*, 287(24) (2000)2115).

The distribution of trust levels is somewhat similar to that of Web page ranks because nodes with high trust levels have more opportunities to cooperate than nodes with low trust levels.

However, knowledge differences tend to level off, a "the poor get richer" rule, because a node with less knowledge can gain from nodes with more (that is, knowledge intensity always tends to equilibrium).

The following principle can now be affirmed.

Principle. *A team prefers the recruit who has more knowledge and is highly trusted by more team members*.

5.9 Resource-Mediated Knowledge Flows

Knowledge flows can be also generated and carried out by asking and answering; knowledge flowing from the person/role who answers a question to the person / role who asks the question.

Some relationships between resources reflect knowledge flows between resources' authors. For example, citation relationships between scientific papers reflect knowledge flows from an author of a paper being cited to an author of the paper that cites it. A citing paper is a confluence of incoming knowledge flows and a source of output knowledge flows conveying the innovation of its author(s). So a resource–mediated knowledge flow management tool would be very useful in managing knowledge and in exploring the nature of innovation in scientific research.

Hyperlinks between resources reflect a kind of weak citation relationship. Semantic relationships between resources can be set up to refine the citation relationship by using text–mining approaches (J. Han and M. Kambr, "Data Mining: Concepts and Techniques", *Morgan Kaufmann Publishers*, 2000).

In resource-mediated mode, knowledge flows through four types of links: *question answering links*, *citation links*, *hyperlinks*, and *semantic links,* as shown by the broken lines in Fig. 5.2.

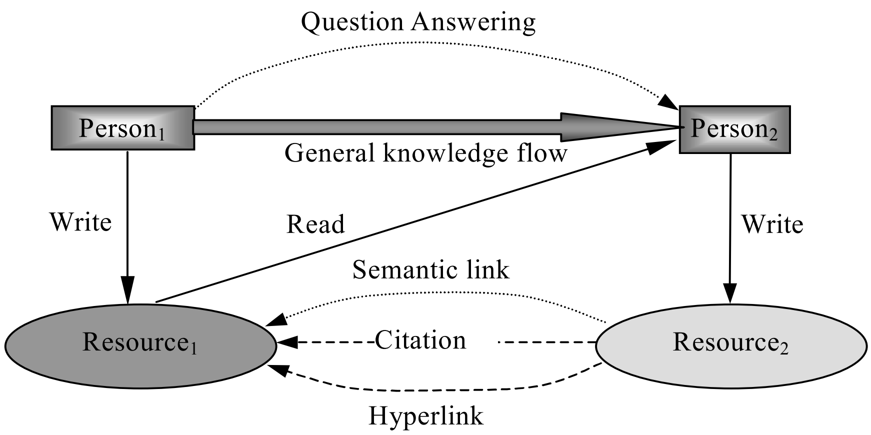

Fig. 5.2 Resource-mediated knowledge flows.

Algorithms for computing the ranks in a knowledge flow network can be designed with reference to the PageRank algorithm (J. Kleinberg and S. Lawrence, "The structure of the Web", *Science*, 294(30)(2001)1849-1850).

A Knowledge Grid environment has three flows: *knowledge*, *information* and *service*. Exploring their common features can lead to the design and implementation of a uniform flow model. In-depth investigation of knowledge flow involves interdisciplinary research into management, cognition, psychology and epistemology.

Chapter 5 in:

H.Zhuge, The Knowledge Grid, World Scientific, 2004.