行政院國家科學委員會專題研究計畫　期中進度報告

以工作階段性為基礎之工作相關知識探勘與知識支援之研究(1/2)

計畫類別：個別型計畫
計畫編號：NSC94-2416-H-009-015-
執行期間：94 年 08 月 01 日至 95 年 07 月 31 日
執行單位：國立交通大學資訊管理研究所

計畫主持人：劉敦仁

計畫參與人員：陳韋孝、吳怡瑾、賴錦慧、柯志坤

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 95 年 5 月 24 日

行政院國家科學委員會專題研究計畫期中進度報告

以工作階段性為基礎之
工作相關知識探勘與知識支援之研究(1/2)

Research on Mining and Supporting Task-relevant Knowledge based on Task-stages (1/2)

計劃編號：NSC 94-2416-H-009-015
執行期間：94 年 8 月 1 日至 95 年 7 月 31 日
主持人： 劉敦仁 交通大學 資訊管理研究所
計劃參與人員：陳韋孝、吳怡瑾、賴錦慧、柯志坤

摘要

組織之運作主要為執行各項知識密集工作，以達成企業組織之營運目標；而在工作之執行，常需逐步執行階段性任務以完成工作。因此本研究主要探討如何根據知識工作者在不同工作階段之需求，提供有效之工作階段相關知識支援。計畫執行進度包括：(1) 分析知識密集工作之特性，設計以工作階段性為基礎之工作相關知識探勘與知識支援系統架構；(2)運用資訊檢索技術與改良階層式凝聚分群演算法，提出一工作階段探勘方法，從歷史工作資料發掘工作階段知識需求，進而建構工作階段需求特徵檔; (3) 實驗評估驗證所提方法有效提供工作階段知識支援。

關鍵詞：知識管理、知識支援、分群、工作階段需求特徵檔

Abstract

Organizations mainly conduct knowledge-intensive tasks to achieve organizational goals. In such task-based environments, knowledge workers usually accomplish their tasks by stages, while their task-needs may be different at various stages of task performance. This project mainly investigates the issues related to providing task-relevant in knowledge-intensive environments based on task-stages. The research progress of this project includes the following. (1) Analyzing the characteristics of knowledge-intensive tasks, and designing a system framework to support task-relevant knowledge based on task-stages; (2) A *task-stage mining method* is proposed for discovering task-stage needs from historical task executions. The method adopts information retrieval techniques and a modified hierarchical agglomerative clustering algorithm to identify task-stage needs by analyzing codified knowledge (documents) accessed or generated during task performance. Furthermore, task-stage profiles are generated to model the task-stage needs; (3) Empirical evaluations were conducted to demonstrate that the proposed method provides a basis for effective knowledge support based on task-stages.

Keywords: knowledge management; knowledge support; clustering; task-stage profile

1. Background and research objective

Reusing knowledge assets extracted from historical task executions is the key to providing effective knowledge support for conducting tasks. Among different type of knowledge management, the repository of structured and explicit knowledge, especially in document form, is a codified strategy for managing knowledge [4][15]. Thus, intellectual content is generally codified into explicit form to facilitate knowledge reuse and sharing [3][10].

As the operations and management activities of enterprises are mainly task-based, organizing and delivering relevant knowledge from the perspective of business task is regarded as a desirable and effective way to fully reuse organizational knowledge assets [1][5]. The KMS developed by the *KnowMore* project is an example of proactive delivery of task-specific knowledge according to the contexts of tasks within a process [1]. Moreover, for knowledge-intensive tasks, such as research projects in academic organizations, project management in firms, product development in R&D departments, it is more difficult to supply task-relevant knowledge during the progress of task execution. Generally, knowledge workers require a longer time to accomplish knowledge-intensive tasks. As observations from the pilot studies in information search process areas [8][14], the information needs of knowledge workers will vary according to different stages of task performance. Vakkari [14] concentrated on the user's information seeking activities during the progress of task performance.

This work focuses on providing knowledge support based on task-stages for knowledge-intensive tasks within organizations. In our previous study, we built a task-based knowledge support system without considering knowledge workers' stages of progress towards task performance [9]. Therefore, this work aimed to support task-relevant knowledge according to workers' task-needs at different task-stages. A task-stage mining technique for discovering task-stage

needs from historical task sets is proposed. The proposed method adopts information retrieval techniques and a modified hierarchical agglomerative clustering (HAC) algorithm to identify task-stage needs by analyzing codified knowledge (documents) accessed or generated during task performance. Hierarchical agglomerative clustering (HAC) is a class of clustering approaches [6][7], which produces clusters by merging, depended upon similarity of two existing clusters. In this work, we modified the traditional HAC algorithm to generate task-stages (clusters) by setting a time window to select candidates for merging. Once the stages of a task has been identified by the modified HAC algorithm, task-stage profiles are generated to model workers' task-stage needs and are used for delivering task-relevant knowledge at various task stages. Finally, we conduct empirical evaluations to confirm the effectiveness of the proposed technique.

2. Research result

This project investigates the issues of deploying knowledge support systems based on task-stages. The task-stage knowledge support method and system framework is proposed to tackle the problem of knowledge support for knowledge-intensive task.

Firstly, the contents of documents and workers' usage data are pre-processed and analyzed. A task-stage mining method is proposed for discovering task-stage needs from historical task executions. That is, a *TSHC* algorithm, task-stage hierarchical clustering algorithm, is proposed to cluster each task document sequence into task stages. The valuable knowledge of each task-stage is extracted to build the task-stage profile to meet the workers initial task-needs at each stage. Experiments have been conducted to evaluate the effectiveness of the proposed technique. Currently, we are investigating task-stage identification models to determine the changes of a worker's task-stage. In our ongoing work, we will integrate the proposed method with a task-stage identification model to provide more effective knowledge support based on task-stages.

3. Related work

3.1. Task-based knowledge management

In task-based business environments, an important issue of deploying KMS is how to support task-relevant knowledge by considering the characteristics of tasks in organizations [1][5][9].

Organizations mainly conduct knowledge-intensive tasks to achieve organizational goals. In such task-based environments, knowledge workers usually accomplish their tasks by stages, while their task-needs may be different at various stages of task performance. Accordingly, another challenge of deploying KMS is how to support task-relevant knowledge according to workers' task-needs at different task-stages. The Vakkari studies [14], which focus on a user's information seeking activities during task performance (e.g., writing a proposal, completing a project, etc.), show that information needs vary according to different task stages.

3.2. Information retrieval and information filtering

Information retrieval (IR) deals with the representation, storage, organization of, and access to information items [2]. Document pre-processing by term transformation and term weighting [13] is widely used to represent documents in the form of vectors. Documents can be retrieved and presented to users according to the degree of similarity calculated by cosine similarity measure. Moreover, information filtering (IF) stresses on maintaining a user profile, in which the system routes the proper information relevant to the user profile [2].

IR and IF technologies applied in document management systems are generally the first pace of knowledge management initiatives. Maintaining and learning user profiles is important in modern Knowledge Management Systems.

3.3 Clustering

Clustering is an unsupervised process of grouping the data (objects) into classes or clusters. The data (objects) within a cluster have similarity in comparison to one another, whereas dissimilar to data (objects) in other clusters. Generally, clustering techniques can be divided into two classes: one is partitioning clustering and the other is hierarchical clustering [6]. Partition method outputs only one partition while hierarchical method produces a nested partition. The hierarchical clustering method creates a hierarchical decomposition of the given data set [7].

Hierarchical agglomerative clustering (HAC) produces clusters by merging, depended upon similarity of two existing clusters [7]. The process repeats combining the two most proximity clusters; likes a binary tree, starts at leaves and grows up to root. The operations of HAC can be summarized into three steps as the following. (1) Each data item is considered as a singleton item initially; (2) Select the nearest pair of clusters and merge them; (3) Repeat step 2 until meeting the stopping condition. In this work, we modified the Hierarchical agglomerative clustering (HAC) algorithm to develop the proposed task-stage mining method.

4. Problem formulation of mining task-stage needs

This work broadly refers to a task as a unit of work in organizations, such as a project, research work, or activity. Moreover, this work uses profiling approach to model workers' task-stage needs, namely information needs (profiles) on task-stages. Task-stage profiles are generated to model worker's initial

task-stage needs at each stage and are used further to provide task-relevant knowledge at various task stages. A task-stage profile specifies the key subjects of a task stage and can be represented as a feature vector of weighted terms. The profile is used to retrieve relevant codified knowledge in the repository. The key contents of codified knowledge, namely document profiles, are also represented as a feature vector of weighted terms. Relevant documents can be retrieved to provide knowledge support for task execution according to the similarity measures between task-stage profiles and document profiles.

A task class is defined to specify a type of tasks with similar properties (similar tasks, for brevity). Task classes are identified based on existing tasks. A task that belongs to a task class is called a task instance of the task class. Task class/instance resembles the concept of object class/instance. Tasks in different task classes generally have different task-stage needs, while tasks belonging to the same task class may have similar task-stage needs. Accordingly, task-stage profiles generated by analyzing existing similar tasks of the same task class are useful to provide needed relevant knowledge in supporting the execution of on-going tasks.

The term *virtual task* is used to represent a task class of similar tasks. A virtual task may have more than one task instances. For example, a virtual task "Research on recommender systems" has several task instances including "Hybrid approaches for product recommendations" and "Comparisons of collaborative filtering for recommendations". Each existing task instance is associated with documents accessed or generated during task performance. Accordingly, the problem of identifying task-stage needs of a virtual task is described as follows.

"*Given a virtual task and a set of task instances with associated documents, find the number of task stages and the corresponding profile of each task-stage.*"

Definition I. Task: A task is a fundamental unit in business. This work broadly refers to a task as a unit of work in organizations, such as a project, research work, or activity.

- **T:** *Task set*; T = {t_1, t_2,…, t_r, …, t_n}, where a task is either an executing-task (on-going task) or an existing task in the task-based working environment.
- ***t*$_r$:** *Existing-task*; An existing-task is a historical task accomplished within the organization.
- ***t*$_v$:** *Virtual task;* A virtual task represents a task class of similar tasks.
- *Task instance***:** A task instance of t_v is an existing-task belongs to the class of t_v.
- **T$_v$:** *Set of task instances;* A set of task instances of the virtual task t_v.

Definition II. Task Document Sequence: A task document sequence is a sequence of documents retrieved (accessed or generated) during task performance. Documents are sorted according to their retrieval time. Each task instance has its own task document sequence.

- **TDS(*t*$_r$):** *Task Document Sequence of* t_r*;* TDS(t_r) = <d_1, d_2,…,d_m>, where a sequence of documents accessed/generated while conducting a task t_r.

Definition III. Task Stage: Task document sequences of task instances are clustered into stages based on similarity measures and retrieval time of documents. The clustering result forms task stages of a task.

- **TS(*t*$_r$):** *Task-stages of a task instance*; A task t_r comprises several task stages. **TS**(t_r) = <ts_r[1], ts_r[2],…, ts_r[*k*]>, where ts_r[*i*] denotes the task-stage *i* of t_r .
- **TS(*t*$_v$):** *Task-stages of a virtual task*; **TS(*t*$_v$)** = <ts_v[1], ts_v[2],…, ts_v[*k*]>, where the task-stages of a virtual task t_v are derived from task stages of task instances in T$_v$.

Definition IV. Task Stage Documents and Profiles: Each cluster of documents represents a task stage with associated documents, named task stage documents. Documents in task stage *k* of task t_r is denoted as ts_r[*k*].docs, while documents in task stage *k* of virtual task t_v is denoted as ts_v[*k*].docs. Each task-stage profile can be derived from the feature vectors of documents in each task-stage. Let ts_r[*k*].profile denote the profile of task stage *k* of task t_r, and ts_v[*k*].profile denote the profile of task stage *k* of virtual task t_v.

5. Mining and supporting task-stage knowledge

This work employs information retrieval techniques [2] to conduct text processing. The key contents of a document can be represented as a feature vector of weighted terms in *n*-dimensional space. The term transforming and term weighting steps are employed to find the most discriminating terms [2]. We use the normalized *tf-idf* ap-proach [13] to derive the term weight.

The document database and the system log record the historical tasks with associated documents and usage data. The documents of a task are preprocessed and organized as a task document sequence (TDS) according to their accessed/generated time during task performance. Clustering techniques are then employed to cluster the task document sequences of similar tasks belonging to the same task class. A task-stage hierarchical clustering (TSHC) algorithm is proposed to cluster the task document sequences based on similarity measures and retrieval time of documents. Each cluster represents a task stage with associated documents of a task. Finally, the feature vector of each task stage is extracted from each cluster of documents

to construct the task-stage profile. The relevant codified knowledge (documents) can then be retrieved to provide knowledge support for task execution according to the similarity measures between task-stage profiles and document profiles.

5.1 Mining of task-stage needs: A modified hierarchical clustering technique

Given a virtual task t_v and a set of task instances with associated documents, the mining process identifies the number of task stages and the corresponding profile of each task-stage for t_v. Notably, a virtual task t_v denotes a task class of similar tasks and has several task instances. A task instance of t_v is an existing-task that belongs to the class of t_v. The documents of a task instance are preprocessed and organized as a task document sequence (TDS) according to their accessed/generated time during task performance. The mining procedure is described step by step, as shown in Fig. 1.

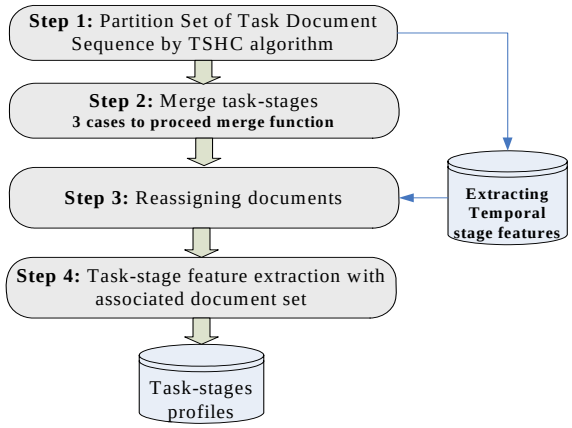

Fig. 1: Process of mining task-stage knowledge

Step1. Identifying task-stages of each task instance.

This step identifies the task-stages of each task instance. A *TSHC* algorithm, task-stage hierarchical clustering algorithm, is proposed to cluster each task document sequence into task stages. The proposed algorithm modifies the hierarchical agglomerative clustering (HAC) algorithm by considering time variant to determine clusters. Table 1 shows the TSHC algorithm. Let TDS(t_r)=<d_1, d_2,…,d_m> be the task document sequence of a task instance t_r. The task document sequence is clustered into several clusters, which represent the stage information of the target task t_r. A task t_r comprises several task stages. **TS**(t_r)=<$ts_r[1]$, $ts_r[2]$,…, $ts_r[k]$>, where $ts_r[i]$ denotes the task-stage i of t_r . Each cluster represents a task stage with associated documents, where $ts_r[i]$.docs denotes documents in task stage i of task t_r.

***TSHC* algorithm:** The traditional Hierarchical Agglomerative Clustering (HAC) algorithms [7] start with each individual item in its own cluster and iteratively merge clusters until all items belong in one cluster. For each level, the HAC algorithms determine new clusters by merging clusters from the previous level. Several techniques can be used to determine if two clusters are to be merged according to the distance threshold and the distance between any two points in the two target clusters. Two clusters are merged, if the minimum (maximum or average) distance between any two points in the two target clusters is less than or equal to the distance threshold. The HAC algorithm examines all clusters in the previous level to determine new clusters, and does not consider the time dimension.

The proposed *TSHC* algorithm adopts the HAC algorithm by considering the time dimension, since the task document sequence (TDS) is generated by ordering documents based on their retrieval time during task performance. A time window size w is set to define the scope of candidate clusters to be merged. Notably, the index position in the sequence rather than the actual time is used to decide if the candidate cluster is within the time window. The *TSHC* algorithm is listed in Appendix 1.

Step2. Merging Task-stages.

Step 1 uses the *TSHC* algorithm to generate the task stages of each task instance. This step generates the task stages of the virtual task t_v by merging the stages of task instances of t_v. $\mathbf{T}_v$ is the set of task instances of t_v. A kernel task instance t_r is selected as the representative of the virtual task t_v. Then, the stages of other task instances are merged to the stages of t_r. The kernel task is the task instance with the best average Q value of clusters among task instances. During the merging process, documents from the stages of other task instances are merged to the stages of the kernel task instance.

Generally, several candidate stages in t_r are possible to merge a document of t_f. For each stage i of t_f, each document d in $ts_f[i]$.docs is merged into the candidate stage k of t_r that has the maximum similarity measure $cos(d.\text{profile}, ts_r[k].\text{profile})$. There are three cases to select the candidate stages of t_r for merging a stage i of t_f.

Case 1: $n_r > n_f$: The number of stages of t_f is less than the number of stages of t_r. The candidate stages of t_r selected are from stages i to $i+(n_r-n_f)$, where $n_r - n_f + 1$ stages of t_f are chosen as the candidate stages.

Case 2: $n_r = n_f$: The number of stages of t_f is equal to the number of stages of t_r. The candidate stages of t_r selected are from stages i-1 to i+1, where three stages of t_f are chosen as the candidate stages.

Case 3: $n_r < n_f$: The number of stages of t_f is greater than the number of stages of t_r. The candidate stages of t_r are from stages $max(1, i-(n_f-n_r))$ to $min(i, n_r)$.

Step3. Reassigning documents.

This step reorganizes the documents in task stages of t_v to enhance the homogeneousness of each task-stage. Documents may be reassigned to other task

stages according to the similarity measures (cosine measures) of document profiles and task-stage profiles. A document *d* is assigned to the stage *k* of t_v that has the maximum similarity measure, i.e., cos(*d*.profile, ts_v[*k*].profile), among all task stages of t_v. Some noisy or irrelevant documents may be removed from t_v, if their maximum similarity measures are below a defined threshold. After the reassignment, task stages with zero or one document will be removed. Consequently, we derive the final document set at each task stage of the virtual task t_v.

Step 4. Extracting Task-stage profiles.

The profile of each task stage can be derived by averaging the feature vectors of documents in the corresponding stage. Notably, the task-stage profile specifies the key subjects of a task stage. For a stage k of virtual task t_v a task-stage profile ts_v[*k*].profile is the vector obtained by averaging the feature vectors of documents in ts_v[*k*].docs.

5.2. Knowledge support

Relevant documents can be retrieved to provide knowledge support for task executions according to the similarity measures (e.g. cosine measures) between task-stage profiles and document profiles. For conducting an on-going task t_e, the system first identifies the task class t_v that t_e belongs to. Then, the task-stage profiles of t_v are used to provide task-relevant knowledge at each stage of executing t_e. A document *d* is provided to knowledge workers at the stage of *k*, if the similarity measure *cos*(*d*.profile, ts_v[*k*].profile) is greater than a defined threshold. Alternatively, top-*N* approach can be used to provide *N* documents with *N*-highest similarity measures at each task stage.

6. Experiment evaluations

Experiments were conducted using a real application domain on conducting research tasks in a laboratory of a research institute. The tasks concerned are writing research papers or conducting research projects. Notably, the real application domain restricts the sample size of the data and participants in the experiments.

The effectiveness of knowledge support is measured in terms of precision, recall and F1-measure, as in the research area of information retrieval [11][12]. Precision is the fraction of retrieved items (tasks or documents) that are relevant, while recall is the fraction of total known relevant items that are retrieved. The F1-metric could be used to balance the trade-off between precision and recall. F1 metric assigned equal weight to precision and recall and was given by, 2×precision×recall/(precision+recall).

A time window size *w* is set to define the scope of candidate clusters to be merged. The smaller the time window size is, the higher the timeliness is. When time window size is equal to *n*, the proposed *TSHC* algorithm is the same as standard hierarchical clustering algorithm (HAC). The result reveals that generally the smaller the time window size, the better the effectiveness (the higher precision, recall, and F-measure) of knowledge support.

The *task stage knowledge support method* is compared with the baseline method, *non-stage knowledge support method*. In general, the average precision, recall and F1-measure of task stage knowledge support method are greater than that of non-stage method. The result reveals that task stage knowledge support method can provide more effective knowledge support when supporting fewer number of task-stage relevant documents.

7. Discussions

Codifying knowledge into explicit form is a widely adopted strategy in contemporary knowledge management systems (KMS) to facilitate knowledge reuse. However, workers still encounter difficulty in accessing information from vast amount of codified knowledge. This work proposed a task-stage mining technique to tackle the problem. The valuable knowledge of each task-stage is extracted to build the task-stage profile to meet the workers initial task-needs at each stage. Experiments have been conducted to evaluate the effectiveness of the proposed technique. In our ongoing work, we are investigating task-stage identification model to determine the changes of a worker's task-stage. We will also integrate the proposed task-stage mining technique with the task-stage identification model to investigate the contribution of the task-stage knowledge support model empirically.

8. Project evaluation

Providing task-based knowledge supports is crucial for enterprises in effectively managing business knowledge and achieving competitive advantages. Our work not only contributes to the practice of knowledge management but also contributes to further research on task-based knowledge support. The progress of this project includes proposing novel idea, developing new technology and conducting experiment evaluations.

References

[1] Abecker, A., Bernardi, A., Maus, H., Sintek, M., Wenzel, C.: Information Supply for Business Processes: Coupling Workflow with Document Analysis and Information Retrieval. Knowledge Based Systems. 13(1) (2000) 271-284.

[2] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. The ACM Press. New York (1999)

[3] Bolloju, N., Khalifa, M., Turban, E.: Integrating Knowledge Management into Enterprise Environments for the Next Generation Decision Support. Decision Support Systems 33(22) (2002) 163-176

[4] Davenport, T. H. Working knowledge: How Organizations Manages What They Know. Harvard Business School Press, Boston MA (1998)

[5] Fenstermacher, K. D.: Process Aware Knowledge Retrieval. Proc. of the 35th Hawaii Intl. Conf. on System Sciences, Hawaii, USA. (2002) 209-217

[6] Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. *ACM* Computing Surveys 31(3) (1999) 264-323

[7] Johnson, S. C.: Hierarchical Clustering Schemes. Psychometrika 2 (1967) 241-254

[8] Kuhlthau, C.: Seeking Meaning: A Process Approach to Library and Information Services. Ablex Publishing Corp., Norwood, NJ (1993)

[9] Liu, D.-R., Wu, I.-C., Yang, K.-S.: Task-based K-Support System: Disseminating and Sharing Task-relevant Knowledge. Expert Systems with Applications 29(2) (2005) 408-423

[10] Markus, M.L.: Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situation and Factors in Reuse Success. Journal of Management Information Systems 18(1) (2001) 57-94

[11] van Rijsbergen, C.J.: Information Retrieval, Second ed. Butterworths, London (1979)

[12] Riloff, E., Lehnert. W.: Information Extraction as a Basis for High Precision Text Classification. ACM Transaction on Information System 12(3) (1994) 296-333

[13] Salton, G. and Buckley C.: Term Weighting Approaches in Automatic Text Retrieval. Information Processing & Management, (1988) 24(5), pp. 513–523.

[14] Vakkari, P.: Cognition and Changes of Search Terms and Tactics during Task Performance: A Longitudinal Case Study. Proceedings of the RIAO'2000 Conference. Paris: C.I.D (2000) 894-907

[15] Zack, M.H.: Managing Codified Knowledge. Sloan Management Review 40(4) (1999) 45- 88

Appendix 1. Task Stage Hierarchy-based Clustering (TSHC) Algorithm

Input: $TDS(t_x)$: Task Document Sequence of t_x
Output: Task stages of target task

```
function TSHC(TDS) {
1    Set a constant offset be 0.01;
2   do {
3       foreach cluster c_i in TDS {
4         c_temp={c_i-w, c_i-w+1, ..., c_i-1, c_i , c_i+1, ..., c_i+w-1, c_i+w};
          //According to time window size to determine clusters
5         foreach cluster c_j in c_temp where c_i <> c_j {
6           Calculate similarity of c_i and c_j as similarity;
7           if (similarity is greater than Threshold) then
8              add {c_i, c_j, similarity} to the merge_list;
          }
        }
9       do while (merge_list is not empty) {
10          Select and remove the pair (c_i, c_j) with maximum similarity from
              merge_list;
11          if (c_i and c_j had not been merged with another cluster)
12       then Merge c_i and c_j in TDS
        }
13      if (there is no cluster merged) then
14         Decrease the Threshold by offset;
15      if (number of clusters in TDS ≥ MaxNumStages) then {
16          Let Q value be the quality of clustering value of TDS;
17         Add {TDS, Q} to the list result;
        }
      }
18    while (number of clusters in TDS > MinNumStages)
19    return result; }
```