**Preliminary**

~~~~This section defines some key terms ~~of formal research~~ of the service composition that I will continue to use in this paper.

**Service**

~~~~A service S is a reusable system that provides functionalities that are documented in a service description. This description defines a 5-tuple {¥em IOPEQ = (S.I, S.O, S.P, S.E, S.Q)}, where {¥em S.I, S.O} are abbreviated from the required inputs and outputs of the S, {¥em S.P, S.E} are the preconditions and effects which denote the necessary condition for utilizing S and the consequence of running S, and {¥em S.Q} is the set of the QoS attributes of the S.

~~~~In this paper, {¥em S.P, S.E} are ignored and therefore a service S is modeled only with a 3-tuple {¥em IOQ = (S.I, S.O, S.Q)}.

**QoS**

~~~~QoS is abbreviated from quality-of-service, i.e. quality apart from the functionality the service ~~can~~ provides, including the price, the execution time and the reliability of the service.

The S.Q of a service S consists of a number of QoS attributes which is normalized in a form that 0 is the worst result and 1 is the best result possible.

**Service Compliance**

%Connectivity between two services

~~~~If there exists an output {¥em o} $¥in$ {¥em S.O} of a service S that is compatible with an input of {¥em i}  $¥in$ {¥em S'.I} of another service S', it is defined that there is a service link between S and S', written as S $¥to$ S'. It means that / Thus, the type of {¥em o} is the same with or a subtype to i. ¥¥

~~~~In this paper, a service is constrained to have only one input and one output, and therefore S $¥to$ S' iff:

¥begin{center}

{¥em o} = {¥em S.O} ,  {¥em i} = {¥em S'.I} ,  {¥em o} $¥in$ {¥em i}

¥end{center}

**Functional requirements**

**Workflow**

~~~~A workflow is a sequence of two or more linked services. The functions of a workflow depend on function combinations that consist of the input parameter of the workflow's first service and the output parameter of the workflow's last service. A workflow template contains service tasks instead of actual services and they are characterized as abstract functionalities that can be replaced by actual services. There are generally two ways to assign services to a task. One is to compare the functions of the services with the functional requirements of the task. The other is to collect services depend on documents such as service descriptions, which are usually distributed with the service by the provider. Finally, each task is assigned with a set of services that meet the functional requirements of the task.

Selection algorithms receive a workflow template with fixed functional requirements and a dozen~~s~~ of services, and select one or more services for each task that guarantee the obtained workflows QoS are optimized and then return the obtained workflow as an output.

Figure 1 shows an example of a workflow template and a possible service selection.

**QoS optimization**

**Service composition**