Master Thesis

Market-oriented Resource Allocation for Volunteer-based Free Internet Services

Supervisor Professor Toru Ishida

Department of Social Informatics
Graduate School of Informatics
Kyoto University

Naoki MIYATA

February 7, 2008

ボランティアによる無料インターネットサービスのための市場指向資源割り当て

宮田 直輝

内容梗概

近年，ネットワーク上で提供された資源を利用可能にするボランティアによる無料インターネットサービスの開発が進められている．有限なシステム資源が利用者間で共有され，提供者はシステム資源を用いて有用なサービスを提供する資源を提供する．利用者は提供されたサービスを非営利の目的でのみ無料で利用することができる．

こうしたシステムにおける課題の一つは，多くの資源が無料で利用できるために，利用者間で資源の取り合いになる可能性がある点である．資源の利用にコストがかからないため，利用者が実際には必要のない需要である擬似需要を記述する可能性がある．擬似要求が記述されると適切な資源割り当てが妨げられ，システムが有効に活用されなくなる．しかし，一般に利用者の需要が真であるかを判断することは困難である．

本研究では，ボランティアによる無料インターネットサービスにおいて適切な資源割り当てを実現する方法を検討した．適切な資源割り当てを実現する際の課題は下記の 2 点である．

1. 無料インターネットサービスにおける資源割り当ての問題設定
   既存のシステムでは資源の利用は有料であり，利用者は価格の制約を満たしながら自らの効用を最大化する．提供者の目的は資源提供により得られる利潤を最大化することである．無料インターネットサービスでは無料で資源を利用できるため，利用者に価格の制約がなく，提供者の目的はより好ましい利用者に多くの資源が利用されることである．こうした資源が無料で利用可能なシステムの目的と制約や，資源割り当てを決定する手法に必要な特徴を明らかにする必要がある．
2. 多数の利用者と提供者が存在するシステムにおける資源割り当て
   定めた資源割り当て問題において，実用的な時間で資源割り当てを決定する手法が必要である．無料インターネットサービスにおける利用者と提供者の規模は，システムの規模が拡大するにつれて増加する．利用者と提供者が増加すると，資源割り当てを決定するために必要な時間は増加する．そ

こで，多数の利用者と提供者が存在するシステムにおいても，実用的な時間で資源割り当てを決定する手法が必要である．

本研究では無料インターネットサービスにおける適切な資源割り当てを実現するために，問題のモデル化を行い，市場モデルを用いた資源割り当て手法を提案した．市場が持つ調整機構によって利用者と提供者の選好を反映した資源割り当てを行い，かつ限られたシステム資源を活用する手法を提案する．上記で述べた課題に対する本研究の主な貢献は以下の2点である．

1. 無料インターネットサービスにおける資源割り当て問題の定式化
   無料インターネットサービスの実システムに基づいてシステムの要求を明確にし，その要求を扱うモデルを提案した．無料インターネットサービスにおいては，利用者は各サービスに関する選好を持ち，代替可能な複数のサービスを含む欲求を持つ．提供者は利用者に関する選好を持ち，効用を最大にするように利用者への割り当てを決定する．こうしたモデル化を行い，資源の割り当てを決定する際の目的と制約を定めた．また，資源割り当てを行う手法に必要な特徴について述べた．
2. ヒューリスティクスを用いた資源割り当て手法の提案
   本研究では資源割り当て問題を現在・未来モデルと消費者・生産者モデルを用いた市場モデルに対応付けた．消費者・生産者モデルにより，利用者は資源を取得するための財をシステムから割り当てられ，現在と未来に各資源から得られる効用を基に，提供者に対する財の割り当てを決定する．提供者は割り当てられた財を基に資源を生産し，利用者に対する評価値と需要に基づいて利用者への割り当てを決定する．そして，実用的な時間で資源割り当てを決定するために，資源から得られる効用を基に利用者が需要を修正するヒューリスティクスを用いた資源割り当て手法を提案した．本研究では，提案手法が無料インターネットサービスの資源割り当てを行う際に満たすべき特徴を持ち，利用者と提供者の選好に基づいた資源割り当てを実現できることを示した．

以上の貢献により，ボランティアによる無料インターネットサービスにおいて本手法を用いることで，利用者と提供者，管理者の選好を反映した資源の割り当てを実現することができる．また，利用者に真の需要を記述する動機付けを行い，擬似需要が記述された場合も他の利用者に与える影響を小さくするといった，資源割り当てを行う際に必要な性質を満たすことができる．

Market-oriented Resource Allocation for Volunteer-based Free Internet Services

Naoki MIYATA

Abstract

Recently, volunteer-based free Internet services, which have many useful resources available on the Internet, have been developed. Volunteer-based free Internet services have finite system resources. There are providers who volunteer to offer useful services and users who utilize the provided services without charge only for non-profit or research purposes.

One of the problems in the system is that users may scramble to obtain as much resource as possible because they can use the resources for free. Users may also input 'sudo-demand', which is not actually true demand. Sudo-demands prevent the system to appropriately allocate resources, so that the system may not be utilized well. Unfortunately, it is difficult to determine whether the demands of users are true or not, either beforehand or posteriori.

In this research, I propose an approach to realize an appropriate resource allocation for volunteer-based free Internet services. Issues in the resource allocation are listed as follows.

1. Problem establishment of resource allocation in volunteer-based free Internet services
   In existing general systems, users intend to maximize their utility with satisfying cost constraints since resources are not free. The purpose of resource providers is to maximize the profit of providing users with their resource. On the other hand, users in volunteer-based free Internet services have no cost constraints since resources are charge-free. The purpose of providers is to utilize their resources well. To realize an appropriate resource allocation, it is necessary to clearly define the purposes and constraints of the resource allocation and the characteristics which allocation methods should have.
2. Appropriate resource allocation in large-scale systems
   The more large-scale and open the system is, the more users and providers exist in the system. It often spends much time to allocate resources in such large-scale systems. It is necessary to propose a method which can

appropriately allocate resource in applicative time in such large-scale open systems and has the necessary characteristics for the resource allocation.

In this research, I model the resource allocation problem and propose an approach which allocates resource based on a market-based model. The adjustment mechanism of the market model is able to reflect preferences of users and providers on the resource allocation and utilize the finite system resource. The contributions of this research are as follows.

1. Formulate the resource allocation problem in volunteer-based free Internet services

   At first, I clarify and formulate the requirements of the resource allocation based on actual systems and propose a model to deal with the requirements. In the systems, users and providers have preferences for each other. Users have desires including multiple interchangeable services. Providers decide the allocation of their resources for users. Additionally, I describe the characteristics which allocation methods should have.

2. Propose a resource allocation method using heuristics

   In this research, I introduce a market-based model using current-future model and consumer-producer model. In the market model, users are assigned system resource to obtain service resources from providers. Users decide the assignment of the system resource based on the utility obtained in current and future state. Providers produce service resource using the assigned system resource and decide the assignment of the service resource for users based on their preference and demands of users. I propose a resource allocation method using heuristics which can adjust user's system resource assignment based on the obtained utility. The experimental results show that my approach can allocate resources based on preferences of users and providers and has the characteristics necessary for the resource allocation in the systems.

The above contributions realize the appropriate resource allocation for volunteer-based free Internet services, based on preferences of users and providers. It has the characteristics which should be satisfied in the systems.

Market-oriented Resource Allocation
for Volunteer-based Free Internet Services

Contents

Chapter 1 Introduction

Recently, the systems to utilize resources in the Internet have been developed. Language Grid project [1] aims to develop the system where language resources are implemented as Web service and users can compose new language resource of multiple language resources. In Grid computing area, the technologies to provide high capacity services by connecting computational resource on the Internet have been studied. PRAGMA, which is one of the actual Grid systems, provides large-scale Computational Grid and Data Grid by connecting many computers in the Internet.

In these systems, users can use the resource for free as long as they obey the policy of the project or system. Users in Language Grid can use the system for free when their purposes is non-profit use. Users in PRAGMA can use the resource for free only for research purposes. In this paper, I term the Internet service where providers volunteer to provide their resources and users can use the resources for free, volunteer-based free Internet services.

In volunteer-based free Internet services, not only users have preferences for resources, but also resource providers have also preferences for users. The purpose of providers is not the profit obtained from providing their resources, but that their resources are used by users who motivate the provider to provide their resource. For example, a motivation of a university which provides a useful resource is that when its students need to use the resource, they can preferentially use it. Otherwise, other users can use it. In volunteer-based free Internet services where providers cannot obtain profit, it is very important to reflecting preferences of providers on the system.

Volunteer-based free Internet services allocate its resource on users based on preferences of users and providers. Since the resource of the system is finite, the crush of demand leads to overloading of the system. In the case where resources are allocated on users based on demands which users input, users may input sudo-demands, which is actually not true-demand since the resources are charge-free. Another approach which charges a fee for the resource is not appropriate since the appropriate user may not have appropriate amount of money and the purpose of the system is not to obtain profit.

In order to motivate users to input true demands and allocate resources based on preferences of users and providers, resources are manually allocated on users in existing actual systems. For example, the administrator receives the request of users and allocated resources on users based on the requests. The manual allocation motivates users to send true demands. But, the larger and more open the system is, the more difficult it is to allocate resources based on preferences changing overtime of users and providers.

In this research, I consider the dynamical resource allocation for volunteer-based free Internet services. The problems in allocating resources in the systems is the follows.

- Problem establishment of resource allocation in volunteer-based free Internet services

  The requirements of resource allocation for volunteer-based free Internet services should be clarified and formulated. Users in volunteer-based free Internet services have no cost constraints since resources are charge-free. The purpose of providers is to utilize their resources. To realize the appropriate resource allocation, it is necessary to clearly define the purposes and constraints of volunteer-based free Internet services and the characteristics which allocation methods should have.

- Allocate resources in large-scale systems appropriately

  The more large-scale and open the systems is, the more users and providers exist in the system. It increases the computational time to allocate resource. I need a method which can appropriately allocate resource in applicative time in such large-scale open systems and has the necessary characteristics for resource allocation.

In this research, I model the resource allocation problem in volunteer-based free Internet services to realize an appropriate resource allocation. Then, I propose a resource allocation method which uses market-based model. The adjustment mechanism of the market model is able to reflect preferences of users and providers on the resource allocation and utilize the limited system resource. . Additionally, it can motivate users to input true demands since sudo-demands waste goods to obtain the resource satisfying their true demands. I propose a resource allocation method using a heuristic for the proposed market-model. The method can allocate resource in applicative time. The results of my simulations show that my approach can allocate resources based on preferences of users and providers and has the characteristics necessary for the resource allocation in volunteer-based free Internet services.

Chapter 2 Related Works

At present, researches on service selection for Web service systems based on QoS include works [2, 3, 4, 5, 6, 7, 8]. Zeng, L. et al. [5] formulates the problem of web service composition in terms of QoS and proposes AgFlow which selects appropriate services using integer programming. AgFlow has a service quality model to evaluate the overall quality of composite web services. AgFlow also have two service selection approaches for composite service execution. One selects services based on each task. The other selects services based on the global allocation of tasks using integer programming. Menasce, D. A. et al. [9] proposes an architecture which allocates QoS based on user utilities in Service Oriented Architecture. In the proposed approach, users provide QoS broker with their utility functions of the consumer and cost constraints about required services. Service providers register with the broker by providing service demands for each of the resources used by the services provided and cost function for each of the services. Consumers request services from the QoS broker, which selects a service provider that maximizes the consumer's utility function subject to its cost constraint. The QoS broker uses analytic queuing models to predict the QoS values of the various services that could be selected under varying workload conditions.

These related works focus on the problem that users select appropriate services which satisfy their cost constraints, execution time and so on when they use resources based on their preferences. This research assumes the system where users can use resources for free. Since users do not have cost constraints, users may input sudo-demands. Consequently, I provide a method which motivate users to input true-demand. Additionally, in volunteer-based free Internet services, provider's utility is based on not the profit obtained from providing users with services but their preference for users and the use amount of resources. I provide an approach which allocates resources based on preferences of users and providers.

In Web service area, an approach which controls user access based on policies written by resource providers in order to reflect preferences of resource providers on the system is proposed. Yuan, E. et al. [10] propose Attributed Based Access Control (ABAC) which dynamically controls user access based on the policy and attribute of users. Service providers describe their access policy about users, resources and environments. The system realizes their policy by dynamically controlling user access based on the described policy. Access control based on user's attributes and its rights is more flexible and more powerful than existing role-based access control.

They propose an approach which controls user access based on the attributes of users. In volunteer-based free Internet services, determining whether a user can access the resource or not is not enough to reflect the preferences of providers on the system. In order to motivate providers to provide resource without charge, it is necessary to reflect preferences of providers on resource allocation. I model the preferences of users and providers in volunteer-based free Internet services and propose a method which allocates resources based on the preferences.

In Grid service area, researches on resource allocation of grid systems include works [11, 12, 13]. Buyya, R. et al. [13] describe an approach of introducing market model to general grid systems. There are various users and providers. There are also various purposes, strategies and patterns of demand and supply. They introduce a competitive market model in order to realize the system where user and provider can maximize their utility. As the result, resources are allocated on users based on various utilities of users and providers.

These researches assume that the utility of users includes requirements for service QoS and cost constraints. the utility of providers is based on the profit obtained from providing resources. In the system which I assume, there are no cost constraints since most of resources are charge-free. Additionally, the utility of providers is based on not the profit but the preference for users and the amount of resource which users use. I model the preferences of users and providers in volunteer-based free Internet services and propose a method which allocates resources based on the preferences in applicable time.

Yamaki, H. et al. [14] proposes an distributed approach which allocates QoS based on a dynamically market model in distributed multi-media systems. They focus on the problem of allocating the finite bandwidth in the system based on the preferences of users and providers. Since the sum of bandwidth is constant over time, the bandwidth is divided to users each time. The resource allocation is based on the equilibrium of the market using preferences of consumers and producers.

The problem on which their resource and my research focus is the same in that they allocate free and finite resources based on preferences of users and providers. They assume that the utility of providers is the profit obtained from providing their resources. On the other hand, I assume that providers have preferences for users and their utility is based on the preference. Additionally, they uses equilibrium to decide the resource allocation. Since there are many users and providers in volunteer-based free Internet services, calculating equilibrium spends too much time to apply the method to actual systems. I propose a heuristics method which allocates resources in applicative time even if the system has many users and providers.

In volunteer-based free Internet services I assume, resources have to be allocated based on preferences of not only users but also providers. Then, I model the users and providers in the system. Users have preferences and desires for resources. Providers have preferences for users. Resources have to be allocated based on these preferences. In large-scale and open systems, there are many users and providers. I provide an approach which allocates resource using heuristics in applicable time even in such large-scale systems.

Chapter 3 Resource Allocation for Volunteer-based Free Internet Services

In this chapter, I describe the resource allocation problem in volunteer-based free Internet services.

3.1 Architecture of Volunteer-based Free Internet Services

The architecture of volunteer-based free Internet services is shown in Figure 1. Volunteer-based free Internet services provide shared computational resource. Service holders provide the system with their services. The provided services serve uses with running on the shared computation resource. The motivation of service providers is to utilize their resource well. Users select the necessary services out of available services. The administrator observes and adjusts the users and providers in the system so that the system is utilized well. For example, there are shared computational resource, contents creators and users in open course wares. Contents provided by creators are served by shared computational resource. In Language Grid, the provided language resources run on the shared computational resource to provide services.

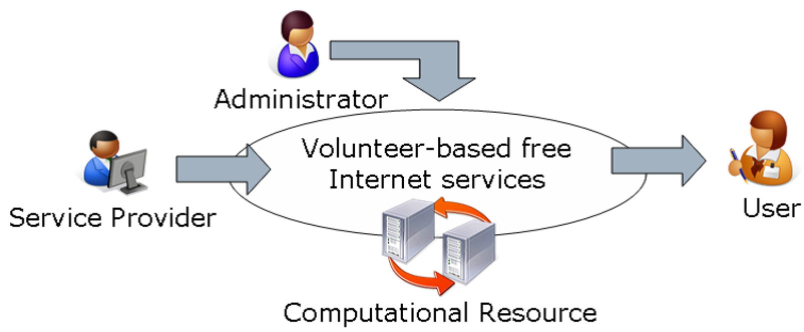

Figure 1: Architecture of volunteer-based free Internet services

Volunteer-based free Internet services become overload by flock access since the shared computational resource is finite. To keep the system from overloading, the finite resource is allocated on users. The resource allocation of volunteer-based free Internet services should be based on the preferences of users and providers. The preference of providers is to utilize their resource well. That of users is to satisfy their desires by using services. The appropriate resource allocation keeps the system from overloading and let users to know the availability of services in advance. Users can adopt their behavior to the allocated resource.

Providers provide their services using the shared system resource. Namely, services have the amount of service resource derived from the amount of allocated system resource. The sum of the allocated service resource on users is less than the amount. The purpose of providers is that their resource is utilized by preferred users. Providers have the evaluated values of users. The utility of the provider is determined by the amount of service resource used by users and the evaluated value of users. The provider limits the amount of service resource allocated by user to protect rights of resources.

Users have desires to use the system for tasks. The desire have the set of services to satisfy the desire. There are multiple interchangeable services for a task in the system. The desire also has the ceiling amount of allocated service resource. User assigns weights to their desires, which means how important the desire is in the set of his desires.

In order to appropriately allocate resources in this model, the system resource is allocated on providers based on the demands of users at first. It determines the amount of service resource which the provider can provide. Then, the allocation of service resources on users is determined based on the preferences of users and providers. The appropriate resource allocation can be realized when most of the demands described by users are true.

3.2 Resource Allocation Problem of Volunteer-based Free Internet Services

The purpose of resource allocation for volunteer-based free Internet services is to realize the appropriate resource allocation based on preferences of users and providers. There are two restrictions in allocating resource in the systems.

- Unable to charge a fee for system resource

  Pricing resources is unable to realize the appropriate resource allocation in volunteer-based free Internet services since the appropriate user may not have appropriate amount of money and the purpose of the system is not to obtain profit.

- Unable to differentiate true demands and sudo-demand

  It is difficult to determine whether the demands of users are true or not, either beforehand or posteriori in Internet services. If the system determines whether the demand is true or not based on whether they actually use the resource, it motivate users to waste the resource in order to avoid the penalty. So the approach is not effective.

When the resource in the system is free, users are not penalize even if they input larger demand than true demand or unnecessary demands as true demands. I term such demand 'sudo-demand'. Sudo-demands interrupt the appropriate resource allocation since the resource allocated on sudo-demands is not actually used.

The function to allocate resources should have the following three characteristics due to the restrictions.

- Motivate users to input true demands

  Since the system is unable to differentiate true demands and sudo-demands, the function should motivate users to input true demands. That users input true demands is required to realize the appropriate resource allocation and utilize the system well.

- Suppress the effects of sudo-demands

  It is impossible to completely eliminate sudo-demands in large-scale systems. The function should suppress the effects of sudo-demands even if there are some sudo-demands.

- Allocate resource in applicable time even in large-scale system

  The function should be able to allocate resources in applicable time even when there are many users and providers in the system.

3.3 Formulation of Resource Allocation Problem

In this section, the formulation of above-mentioned resource allocation problem is described. Service provider's model is shown in Figure 2. The provided service has a product function which determines the amount of utilizable service resource based on the amount of allocated system resource. Let x denote the sum of the system resource allocated on s at t. Let $Q_s(t)$ denote the finite amount of the resource at time t. The sum of service resources which the provider can provide is determined as follows.

$$f_s(x) = Q_s^0 - \frac{Q_s^0}{1 + \gamma_s x} \tag{1}$$

Q_s^0 is the upper limit of resources which provider s can produce. γ_s is the variable indicating how the production is increased by adding the system resource. Providers have evaluated valued of users. Namely, Provider s assigns $\alpha_s^u \quad (0 \leq \alpha_s^u \leq 1)$ on each user u, which is evaluated values of users. Provider also assigns an upper limit of the amount of the resource allocated on u at t. Let $l_s^u(t)$ denote the upper limit for user s. Provider decides on the allocation of the resources on users. Let $q_s^u(t)$ denote the amount of the resource allocated on $u \in U$ by s at time t. $q_s^u(t)$ is less than the upper limit $l_s^u(t)$. The sum of the resource allocated on users is less than $Q_s(t)$. Users on which the resource is allocated decide the amount of resources to use. Let $y_u^s(t)$ denote the amount of resource s which user u use. $y_u^s(t)$ is less than $q_s^u(t)$, which is the amount of allocated resource.

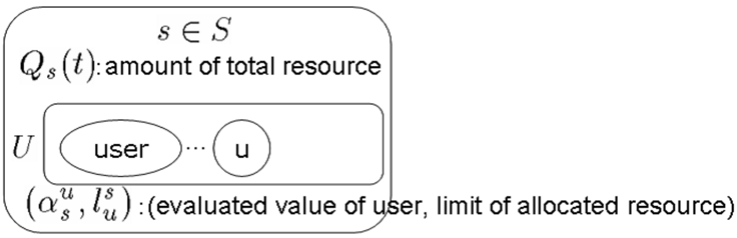

Figure 2: Provider model

- $Q_s(t)$: total amount of the resource at t
- α_s^u: s's evaluated value of u
- $l_s^u(t)$: upper limit of the resource allocated on u at t
- $q_s^u(t)$: amount of the resource allocated on u at t
- $y_u^s(t)$: amount of the resource s which u use at t

The user model in the system is shown in Figure 3. User u has evaluated values of each service available in the system. Let $\alpha_u^s (0 \le \alpha_u^s \le 1)$ denote the evaluated value of user u for service s. User u assigns a large value on the service for which the user has a liking. User u also has desires for resources. Let $D_u(t)$ denote a set of desires at time t. A desire $d \in D_u(t)$ has the following attributes. User u assigns a weight on each desire. Let w_d denote the importance of d in $D_u(t)$. User u assigns a ceiling amount of resources for desires. Let r_d denote the ceiling amount. User u assigns a set of services which desire d uses. Let S_d denote the set. In this problem, it is assumed that users know their future demands. Providers do not know these values. To make the following explanation simply, I assume that sets of services that a user uses at a time is independent. Namely, the equation $S_{d_1} \cap S_{d_2} = \phi(d_1, d_2 \in D_u(t), d_1 \neq d_2)$ holds.

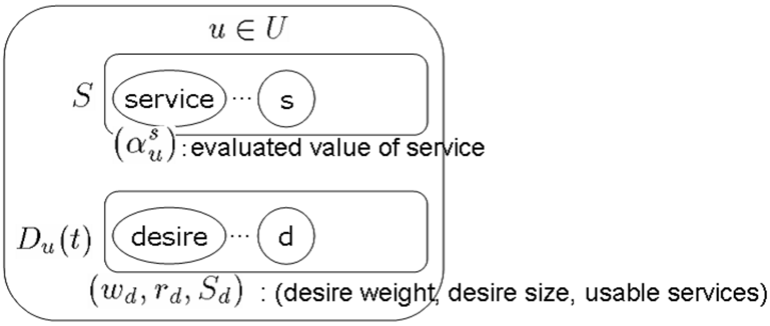

Figure 3: User model

- $D_u(t)$: set of desires of u at t
- w_d: weight of d
- r_d: ceiling amount of resource which d uses per unit of time
- $S_d \subseteq S$: set of services which d uses
- α_u^s $(s \in S_d)$: evaluated value of s

The allocation of system resource on providers and service resource on users should be determined based on the demands of users and preferences of users and providers in order to realize the appropriate resource allocation. The utility of a user equals the sum of the utility of his desires times the evaluated value of services. The utility of a desire equals the sume of the amount of used service resource times the evaluated value of the service. Let $\mathbf{y}_u(t) = (y_u^{s_1}(t), y_u^{s_2}(t), ...)$ denote the amount of resources s which user u uses at t. Let $\mathbf{y}_d(t) = (y_d^{s_1}(t), y_d^{s_2}(t), ...)$ denote the amount of resource which user u uses at t for d. Let $\alpha_u = (\alpha_u^{s_1}, \alpha_u^{s_2}, ...)$ denote u's evaluated values of services. The utility of desire d at t is defined as follows.

$$utility_d(\mathbf{y}_d(t)) = \frac{\alpha_u \cdot \mathbf{y}_d(t)}{r_d} \quad \left(y_d^s(t) \leq q_s^u(t), \sum_{s \in S_d} y_d^s(t) \leq r_d \right) \tag{2}$$

The utility of user u at t is given as follows.

$$utility_u(\mathbf{y}_u(t)) = \sum_{d \in D_u(t)} w_d utility_d(\mathbf{y}_d(t)) \quad \text{subject to} \quad y_u^s(t) \leq q_s^u(t) \tag{3}$$

The utility of providers equals the sum of the evaluated value of users times the amount of used service resource. Let $\mathbf{y}_s(t) = (y_{u_1}^s(t), y_{u_2}^s, ...)$ denote the amount of s resource used by uses. Let $\alpha_s = (\alpha_s^{u_1}, \alpha_s^{u_2}, ...)$ denote the evaluated values of users. The utility of provider s at t is defined as follows.

$$utility_s(\mathbf{y}_s(t)) = \alpha_s \cdot \mathbf{y}_s(t) \quad \text{subject to} \quad \sum_{u \in U} y_u^s(t) \leq Q_s(t) \tag{4}$$

The purpose of users is to maximize the sum of the utilities at each time and defined as follows.

$$\max \sum_t utility_u(\mathbf{y}_u(t)) \quad \text{subject to} \quad y_u^s(t) \leq q_s^u(t) \tag{5}$$

By the same token, the purpose of providers is to maximize the sum of the utilities at each time and given as follows.

$$\max \sum_t utility_s(\mathbf{y}_s(t)) \quad \text{subject to} \quad \sum_{u \in U} y_u^s(t) \leq Q_s(t) \tag{6}$$

The unit of a resource allocated on users varies by the kind of the service. For example, the cost of dictionary service is almost constant for each invocation. The number of invoking the service is the unit of allocated resource. Since the cost of translation service and morphological analysis service varies by passed argument, the unit of allocated resource is the length of translated sentence or the size of analysis result. Otherwise, the length of the request sentence is limited and the unit of the resource is the number of invoking the service.

3.4 Approaches of Existing Systems

In some existing volunteer-based free Internet services, resources are manually allocated. For example, PRAGMA, which provides computational grid and data grid, calls on members to do the following procedures to use resources.

1. A user send a request to an administrator of the system
2. The administrator forwards the request to corresponding request providers
3. The provider decides on approve or reject of the request
4. The user uses the resource, following the reply

Using the above procedures, the administrator and providers appropriately allocate resources on users. Providers allocate resources so that they could provide users with enough good-quality services. Providers can protect rights which their resources possess since they manually decides on approve or reject. Additionally, users can know the availability of resources in advance.

There are also various users in Language Grid. Student and NPO users use the system resource for their experiments and events. Trial Sites which provide anonymous users with services use the system resource for advertisement of Language Grid. The former has high priority and uses specified resources in a certain period. On the other hand, the latter aims to use the services which are not used by other users. In Language Grid, when a user uses the system for his experiment or demo, the administrator tells the other users not to use the system at the same time. The limitation on the users who provide many anonymous users with services is very strict lest the highly-preferred users are not disturbed.

The more large-scale and more open the system is, the more difficult it is to manually allocate resources. For example, when the student users increase, it is difficult to decide an appropriate resource allocation on them based on the history of the user's use. They have to consider whether a user overuses resources, or whether the demand of a user is true or not, and so on.

Since there are many users and providers in this problem, it is difficult for one-way auction to solve it. Even if there is only one provider in the system, it is difficult to solve it in applicable time when the number of users and services is large. Since I assume that it is impossible to expect true demands of users from their behavior and attributes, there are no way to differentiate true demand and sudo-demand. If the system determines whether the demand is true or not based on whether they actually use the resource, it motivates users to waste the resource in order to avoid the penalty. So it is not effective. Although hidden problem can be applied to the system where the purpose of users is unobservable, the assumption that there are some probabilistic relation between actions and results is not acceptable in this system.

In this research, I uses a market model where users uses goods to obtain resources. In the model, agents corresponding users and providers try to maximize their utility. It leads to an appropriate resource allocation based on preferences of users and providers. It motivates users to input true-demands since users have finite goods to obtain resources in the model.

Chapter 4 Market-oriented Resource Allocation

In this research, I provide an approach which uses a market-model and realizes an appropriate resource allocation based on preferences of users and providers in volunteer-based free Internet services. The approach allocates the shared system resource on users in advance. Users allocate the system resource on providers based on their demands. Providers allocate their service resource on users based on the demands of users and their preferences. The method determine the resource allocation using a market-model. There are three reasons to introduce market-model.

- Appropriately allocate resource using market mechanism
  The market mechanism can realize the appropriate resource allocation based on the preferences of users and providers and utilize the finite system resource well. I determine the behaviors of users and providers in the market model in order to realize the appropriate resource allocation.
- Motivate users to input true demand
  In the market model, the finite system resource to obtain system resources is allocated on users in advance. Sudo-demands waste the system resource to obtain service resources for true demands. It motivates users to input true demands.
- Suppress effects of sudo-demand
  Proposed method allocates system resource on users without considering demands. In the market model, the system resource is evenly allocated on users. Even if a user inputs longer and larger sudo-demands than other true demands, the amount of system resource which the sudo-demands can use is less than the other true demands. It can suppress the effects of sudo-demands.

I introduce consumer-producer model and current-future model proposed by Yamaki H. et al. [14]. I extend these model so that it can be applied to volunteer-based free Internet services since the model assume that the purpose of providers is to maximize profit.

In consumer-producer model, the finite system resource is allocated on users corresponding to consumers. Users decides the system resource allocation on providers corresponding to producers based on their demands. Providers produce service resource from the allocated resource and allocate service resource on users. Users evaluate the allocated service resource. This model allocates system resource on users evenly in order to equalize the opportunity of users to obtain service resources. The detail of this model is described in 4.1

In current-future model, the goods exchanged in the market is classified into current and future goods. Users can exchange their goods with each other according to their demands. For example, the user whose current demand is larger than future demand can obtain more current system resource by releasing future system resource. This model allows users to exchange system resource based on their demands. The detail of this model is described in Section 4.2

Users in this model decide the system resource allocation based on current utility, expected future utility and the exchange ratio between current system resource and future system resource. They try to obtain service resource which efficiently increases their utility at the time when their utility is effectively increased. It is more efficient to obtain the service resource which has small demand of few users than that which has large demand of many users.

Providers in this model decide the allocation of produced service resource based on the demand of users and evaluated value of providers. They need to obtain much system resource in order to utilize their resource well. At the same time, they need to allocate more resource on the users to whom they assign high evaluated value. If the provider allocate all of service resource on only highly-evaluated users, the demands of other users decrease and then the amount of system resource allocated on the provider decrease. Providers have to decide the appropriate allocation in which the amount of allocated system resource is enough and the utility of them is large.

The market model has the optimal resource allocation which reflects the preferences of users and providers. It spends too much time to calculate the optimal allocation in large-scale systems. I propose the resource allocation method for the market model using sensitive factor proposed by Kuwabara,

K. et al. [15]. In the proposed method, the demand and supply is repeated until the resource allocation converges. The demand and supply is adjusted based on those at the previous iteration to improve the efficiency of the resource allocation. I describe the resource allocation using sensitive factor in the market model.

4.1 Consumer-Producer Model

The consumer-producer model is shown in Figure 4. In the model, the finite system resource is allocated on users. The system resource is allocated on providers by users. Providers produce service resource from the allocated resource and allocate service resource on users. Users evaluate the allocated service resource. In this model, each user has equal opportunities to use services.

The consumer agent is one-on-one with the user of the system. Users evaluate not system resources but service resources. In other words, the important thing for users is not how much system resource he has, but how much his demand is satisfied with service resources. The preference of user u for service resources is implied in the utility function. The function is based on the evaluated values of services and the amount of resources allocated on the user.

The producer agent in the market model is one-on-one with the provider of the system. The agent converts the system resource allocated by users into the service resource and allocates the produced service resource on users so that their utility would be maximized. In this model, the consumer agent initially has no service resources. All service resources has to be produced by producer agents. Providers allocate its service resource based on the amount of produced service resource and demands and evaluated values of users.

The rectangles at the center of the figure denote goods exchanged in the market model. One is the system resource. The other is the service resource. These goods are classified into current and future goods respectively. The system resource indicates the resources allocated on users by the system. There are current system resource and future one. Similarly, the service resource indicates the resource allocated on users by providers. There are also current service resource and future one. In the figure, q_s^u is the resource allocated on

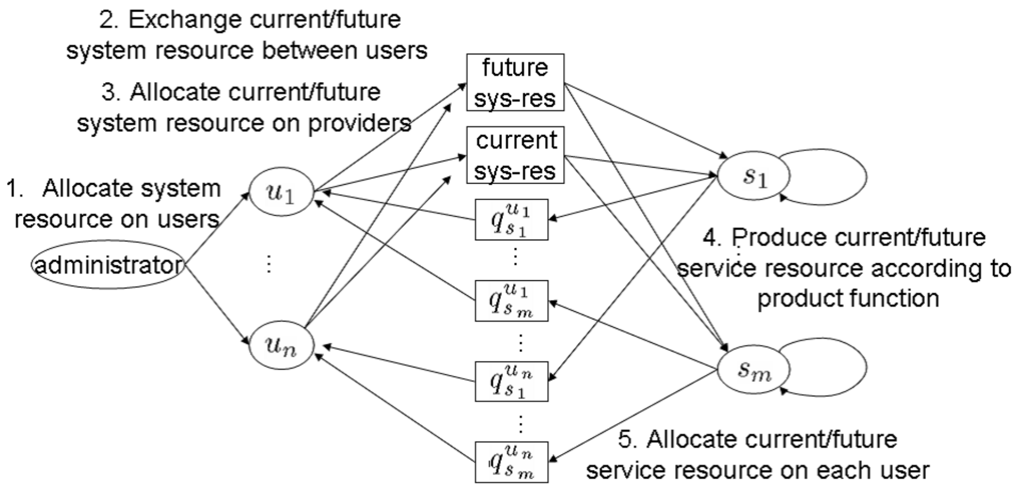

Figure 4: Consumer-provider model

user u by provider s.

The circles in the figure denote agents. There are three kinds of agents in the figure. First is the consumer agent corresponding to users in the system. Second is the producer agent corresponding to providers in the system. Third is the administrator agent corresponding to the administrator in the system.

The arrows in the figure denote the flow of goods between agents. At first, the administrator allocates the system resource on users. Then users exchange their system resource with each other based on their demands and other agent's behavior and then allocate the system resource on providers. Providers produce service resource according to their product function and then allocate them on users.

4.2 Current-Future Model

In current-future model, the goods exchanged in the market is classified into current and future goods. Users can exchange their goods with each other according to their demands in the model.

In current-future model, time is divided into equal interval. A unit of time of the present time is defined as current. A certain period $(T - 1)$ is defined as future. When the total amount of current system resource is β, the total amount of current system resource which users posses equals β. The total amount of future system resource equals $(T-1)\beta$.

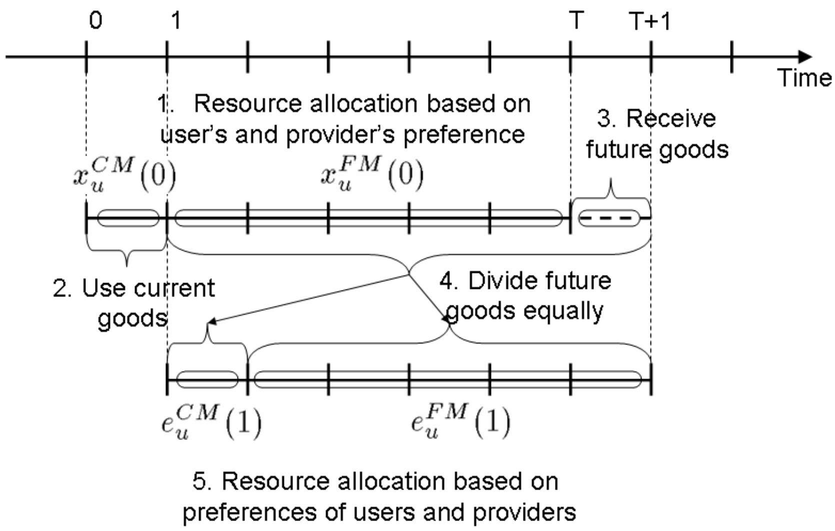

Figure 5: Current-future model [14]

The procedure of dealing with goods in current-future model is shown in Figure 5. At first, resources are allocated based on preferences of users and providers to decide the resource allocation (1 in the figure). Users use the allocated resource (2 in the figure). Next, when time passes for a unit of time, the future goods is reflected on the initial goods in the next time. In concrete terms, the future goods is not enough since there are resources for $(T-1)$ periods although T period is taken into account. $\beta/|U|$ is allocated on each users (3 in the figure). After this procedure, the resource which users have has been updated for new time slice. Users divide their future resource $1:(T-1)$ into the initial resource at next time(4 in the figure). Namely, the current and future system resource which user u initially possesses at time t are determined as follows. Let $e_u^c(t), e_u^f(t)$ denote the initial goods which user possess at time t and let $x_u^c(t-1), x_u^f(t-1)$ denote the system resource which user u has at time $t-1$.

$$e_u^c(t) = \frac{1}{T}\left(x_u^f(t-1) + \frac{\beta}{|U|}\right) \tag{7}$$

$$e_u^f(t) = \frac{T-1}{T}\left(x_u^f(t-1) + \frac{\beta}{|U|}\right) \tag{8}$$

After the above procedure for initial resource allocation, the utility function and demand of users and providers is updated and then the resource is allocated (4 in the figure). Then, the resource allocation is decided on for the new period.

In current-future model, the goods in certain period future is treated as future goods. Providers have to expect the future service resource allocation based on the allocated future system resource. The amount of the future service resource which provider can provide is determined through the following procedures. At first the allocated future system resource divided by considered period is substituted to the product function to calculate the future service resource per a unit of time. The amount of the future service resource is the service resource per a unit of time times the product of the considered period. Namely, when $(T-1)$ future is taken into account and x_f future system resource is allocated, the amount of future service resource is given as follows.

$$(T-1)f_s\left(\frac{x_f}{T-1}\right) = (T-1)\left(Q_s - \frac{Q_s}{1+\gamma_s\frac{x_f}{T-1}}\right) \tag{9}$$

4.3 Resource Allocation in Market Model

This section describes how the appropriate resource allocation is realized by mapping the model of volunteer-based free Internet services to the market model.

At first, I describe the behavior of users. Users allocate the system resource on providers and obtain service resources to satisfy their demand. User have to decide the amount of released system resource and the allocation of system resource on providers in order to increase their utility.

The user who has large demand in current tries to increase his current utility more by releasing future system resource. When the future demand in the system is larger than the current one, the amount of released current system resource is larger than that of released future one. Then, the amount of the current system resource which a user can obtain by releasing their future one is large. The user who has current demands obtain more current system resource by releasing his future one. And vice versa.

Users intend to increase his utility by allocating more system resource on the service which provides large utility per a unit of allocated system resource than others. The services which assigns high-priority to the user or the service which other users do not use allocate much service resource on the users. When the user increases the amount of resource allocated on a service, the utility per allocated system resource is decreased due to the characteristics of the product function. Therefore, it is effective to allocate system resource on multiple services than to allocate that on one service.

The behavior of providers is the follows. Providers allocate service resources based on the evaluated value of users and the amount of system resource allocated by users. I assume that the evaluated value of users and the evaluated value of services do not vary due to the history of resource allocation. There are four kinds of users according to the the amount of allocated system resource and evaluated values of them.

1. User who allocates much system resource and has high evaluated value
   When the user allocates much system resource, the demand of the user for the service is large. Therefore, the provider get much utility from this user by allocating much resources on this user. At the same time, allocating much resource decreases the amount of resource allocated on other users. As the result, the demand of other users and the amount of system resource allocated by other users decrease.
2. User who allocates much system resource and has low evaluated value
   Since the demand of the user for the service is large, the provider can obtain much system resource from this user. Even if the amount of the resource allocated on this user is relatively small, the user allocates much system resource on this service due to his large demand. When the amount of resource allocated on this user is too small, the amount of system resource gained from this user .
3. User who allocates little system resource and has high evaluated value
   Since the demand of the user for the service is small, the provider can obtain little system resource from this user, But since the evaluated value of the user is large, the utility gained from this user is larger than that from users of case 4. Since the demand of user is smaller than user in case 1, the

amount of resource allocated on this user is smaller than that allocated on the user in case 1.

4. User who allocates little system resource and has low evaluated value
   Since the provider can obtain little system resource and little utility gained from this user, the amount of resource allocated on this user is the smallest in the 4 cases.

When a user describes sudo-demands, the sudo-demands waste the finite system resource of the user. Users should input the desires which specify the resource and period in order to increase their utility. It means that this mechanism motivates users to describe true demands.

The effect of sudo-demands on other users in market-based system is smaller than that in demand-based system. Even in using the mechanism which motivates users to input true demands, a part of users describes sudo-demands in large-scale systems. In the proposed approach, the system resource allocated on users is constant over time. The amount of system resource which sudo-demands can use per time is smaller than that of system resource which true demands can use.

4.4 Resource Allocation Using Sensitive Factor

I propose a heuristics approach which decides on the resource allocation for the above-mentioned market model, since it takes much time to calculate an optimal or Pareto-optimal resource allocation. I extend the technique proposed by Kuwabara, K. et al. [15] and apply it to the market model which I use. In this section, the procedure which the proposed approach follows to allocate resources and the behaviors of users and providers is described.

4.4.1 Resource Allocation Protocol

The resource allocation protocol of my heuristics approach is described in Figure 6. In the protocol, users decide on the amount of current and future system resource to release based on the current and future utility and the exchange ratio between current and future system resource. The released system resource is reallocated on users following a rule of the market. Then, users decide the amount of current and future system resource allocated on providers based on the utility gained from the provider. Providers produce service resource based on the amount of allocated system resource and his product function and then decide the allocation of the produced service resource on users. Finally, users decide on the amount of service resource to use for his desires. The utility of users and providers is calculated based on the amount of used service resources. The above-mentioned procedure is repeated until the utilities of users and providers converge.

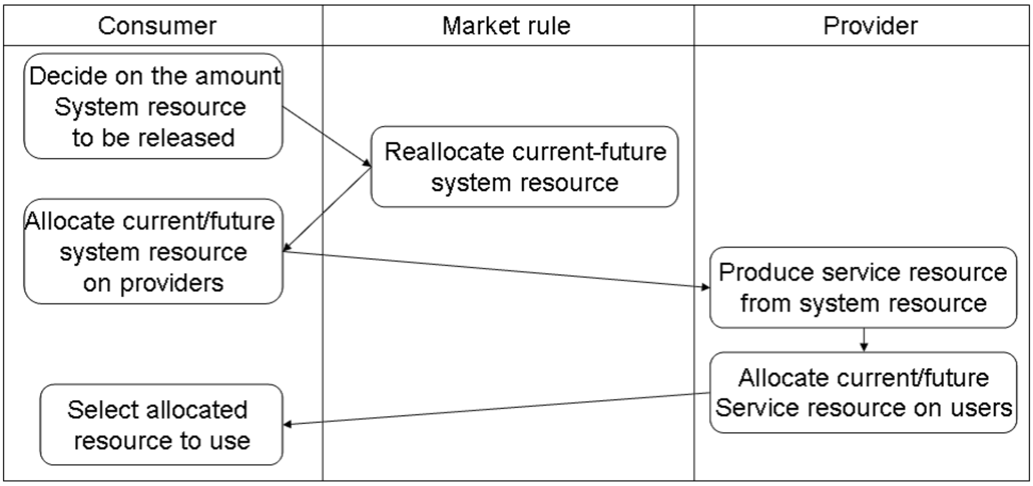

Figure 6: Resource allocation protocol

1. Users decide the amount of current and future system resource to release
   Users decide the amount of current and future system resource to release based on the current and future utility and the exchange ratio between current and future system resource. For example, the user who has no desire in current time, releases all of his current system resource to obtain the future one. When the demand of the target service resource is large and the user estimates that the amount of the service resource allocated on the user decreases in the future, the user releases his future system resource to obtain the current system resource and increase the current utility which is relatively easy to increase.
2. Released system resources are reallocated on users according to a rule of the market

The system resource released by users is reallocated on users according to a rule of the market. In the market model, The amount of future system resource reallocated on a user is derived from the ratio of current system resource released by the user to the sum of released current system resource and vice versa. Let g_u^c denote the current system resource released by user u, and g_u^f denote the future system resource released by user u. Let b_u^c, b_u^f denote the current or future system resource reallocated on user u. b_u^c, b_u^f are given as follows.

$$(b_u^c, b_u^f) = (\frac{g_u^f}{\sum_{u' \in U} g_{u'}^f} \sum_{u' \in U} g_{u'}^c, \frac{g_u^c}{\sum_{u' \in U} g_{u'}^c} \sum_{u' \in U} g_{u'}^f) \tag{10}$$

The detail of the behavior to determine the exchange is described in Section 4.4.2.

3. Users decide the allocation of current and future system resource on providers
   Users decide the allocation of current and future system resource based on the utility gained from providers. The detail of the behavior to determine the allocation is described in Section 4.4.3.
4. Providers produce service resource from system resource allocated by users and allocate them on users
   Providers produce their service resource based on the amount of allocated system resource and his product function. Then, providers decide the allocation of the produced service resource on users. The detail of the behavior to determine the allocation is described in Section 4.4.4. Providers decide the allocation based on the amount of system resource allocated by user and evaluated values of users.
5. Users decide the amount of service resource to use
   Users decide the amount of allocated service resource to use. Users select the service resources which maximize their utility out of the allocated service resources.

4.4.2 Behavior to Exchange System Resources

Users evaluate their current and future utility and determine the amount of current and future system resource to be released. At first, users decide the amount of released system resource based on their current and future demands.

In the repetition of demand and supply, the amount of released system resource is adjusted according to the resource allocation at previous iteration. For example, when the current utility obtained by a unit of current system resource is smaller than the future utility obtained by releasing a unit of current system resource, the user intends to increase his future system resource to increase his utility more efficiently. The initial amount of released system resource is decided based on demands of each user.

At first, users decides the initial amount of released system resources. Let $g_u^c(i)$, $g_u^f(i)$ denote the amount of current and future system resource released by u at i-th iteration. Let $D_u(t)$ denote the set of desires which user u has at time t. To explain simply, $v_u^c(t) = \sum_{d \in D_u(t)} w_d$, $v_u^f(t) = \sum_{t<t'<t+T} v_u^c(t')$ denote the sum of the current or future desire weights.

When the ratio of weights of the current desires to that of the future desires equals to the ratio of the current system resource to the future system resource, the user initially releases no current and future system resource. Namely, $g_u^c(0) = g_u^f(0) = 0 \quad (v_u^c/v_u^f = e_u^c/e_u^f)$.

When the ratio of current desire weights to future desire weights is smaller than that of his current system resource to his future system resource, the user releases his current system resource to obtain future system resource. Let $p(i)$ denote the amount of future system resource gained by releasing a unit of current system resource. $g_u^c(0)$, which is the amount of future system resource which user initially grants, holds the following equation.

$$\frac{v_u^c}{v_u^f} = \frac{e_u^c - g_u^c(0)}{e_u^f + g_u^c(0)p(0)} \quad (v_u^c/v_u^f < e_u^c/e_u^f) \tag{11}$$

Namely, $g_u^c(0)$ is given as follows.

$$g_u^c(0) = \frac{v_u^c e_u^f - v_u^f e^c}{v_u^c + v_u^f p(0)} \tag{12}$$

When the ratio of current desire weights to future desire weights is larger than that of his current system resource to his future system resource, the user releases his future system resource to obtain current system resource. In a similar way of the above case, $g_u^f(0)$, which is the amount of future system

Algorithm 1 Release current and future system resource

α: sensitive factor
i: iteration of resource allocation
$p(i)$: exchange rate of current system resource to future one at i ($p(0) = 1$)
e_u^c, e_u^f: u's initial current/future system resource
$x_u^c(i), x_u^f(i)$: u's current/future system resource after i-th iteration
$g_u^c(i), g_u^f(i)$: u's current/future system resource released at i
$U_u^c(i), U_u^f(i)$: current/future utility per unit system resource at i
θ: threshold of released system resource
if $p(i-1)U_u^f < U_u^c(i-1)$ **then**
$$(g_u^c(i), g_u^f(i)) = \begin{cases} (0, g_u^f(i-1) + \alpha(e_u^f - g_u^f(i-1)) & (g_u^c(i-1) < \theta) \\ ((1-\alpha)g_u^c(i-1), 0) & (\text{otherwise}) \end{cases}$$
else if $p(i-1)U_u^f(i-1) > U_u^c(i-1)$ **then**
$$(g_u^c(i), g_u^f(i)) = \begin{cases} (g_u^c(i-1) + \alpha(e_u^c - g_u^c(i-1)), 0) & (g_u^f(i-1) < \theta) \\ (0, (1-\alpha)g_u^f(i-1)) & (\text{otherwise}) \end{cases}$$
else
$$(g_u^c(i), g_u^f(i)) = (g_u^c(i-1), g_u^f(i-1))$$
end if

resource which user initially releases is given as follows.

$$g_u^f(0) = \frac{v_u^c e_u^f - v_u^f e^c}{v_u^c + v_u^f \frac{1}{p(0)}} \quad (v_u^c/v_u^f > e_u^c/e_u^f) \tag{13}$$

The second iteration of the resource allocation or later, users adjust the amount of released current and future system resource based on the exchange ratio between current and future system resource and their utilities. The algorithm for users to adjust the amount of released system resources is shown in Algorithm 1. Let $p(i)$ denote the exchange ratio of current system resource to future system resource at i-th iteration. The exchange ratio of future system resource to current system resource equals $1/p(i)$ according to the rule of the market. Let $U_u^c(i)$ and $U_u^f(i)$ denote the current or future utility per unit used system resource of user u at i-th iteration respectively.

When the utility per a unit of future system resource is larger than that by the current system resource which user can obtain by releasing a unit of future system resource at the exchange ratio of the previous iteration, the user intends to increase his current utility by releasing his future system resource. In other words, when $p(i-1)U_u^f < U_u^c(i-1)$ holds, the user intends to increase his current system resource. The user who released current system resource at $(i-1)$-th iteration decreases the amount of released current system resource. The amount of released current system resource at i-th iteration is given as follows.

The user who released future system resource at $(i-1)$-th iteration increases the amount of released future system resource. The amount of released future system resource at i-th iteration is given as follows.

$$(g_u^c(i), g_u^f(i)) = (0, g_u^f(i-1) + \alpha(e_u^f - g_u^f(i-1)) \tag{14}$$

When $p(i-1)U_u^f < U_u^c(i-1)$ holds, the user tries to increases his future system resource. The amount of released system resource is adjusted according to the above-mentioned behavior.

$$(g_u^c(i), g_u^f(i)) = ((1-\alpha)g_u^c(i-1), 0) \tag{15}$$

4.4.3 Behavior to Allocate System Resource

After the system resources are reallocated, users allocate their system resources on providers. The behavior that allocates system resources on providers is shown in Algorithm 2. At first, user decides on the allocation of system resources on desires. Although I describe the allocation of current system resource here, that of future system resource is determined in a similar way. The amount of system resource allocated on desires is adjusted based on the resource allocation at previous iteration. The desire which increases the utility the most efficiently in the current desires has more system resource allocated on at this iteration than that at previous iteration. Other desires has less system resource. The initial amount of system resource allocated on desires is determined based on the weights of desires.

Let D_u^c denote the set of desires of user u in current time. Let $rate_d(i)$ denote the rate of system resource allocated on desire d out of his system resource. $rate_d(0)$ equals $w_d / \sum_{d' \in D_u^c} w_{d'}$. User increases the rate of d_{best}, which is the

Algorithm 2 System resource allocation from user to provider

α: sensitive factor

i: iteration of resource allocation

$S_d \subseteq S$: set of services which d uses

$D_u^c \subseteq D_u$: set of demands active in current

$x_u^c(i)$: u's current system resource allocated at i

$d_{best}(i) \in D_u^c$: demand having best utility per unit resource in D_u^c at i

$s_{best}(i) \in S_d$: service giving d the best utility per unit resource in S_d at i

$rate_d(i)$: rate of resource allocated to d at i $\left(rate_d(0) = \frac{w_d}{\sum_{d' \in D_u^c} w_{d'}}\right)$

$rate_d^s(i)$: rate of resource allocated to s by d at i $\left(rate_d^s(0) = \frac{\alpha_d^s}{\sum_{s' \in S_d} \alpha_d^{s'}}\right)$

for all $d \in D_u^c$ **do**

$$rate_d(i) = \begin{cases} rate_d(i-1) + \alpha(1 - rate_d(i-1)) & (d \text{ is } d_{best}) \\ (1-\alpha)rate_d(i-1) & (\text{otherwise}) \end{cases}$$

  $m_d(i) = rate_d(i)x_u^c(i)$: resource allocated on d at i

  for all $s \in S_d$ **do**

$$rate_d^s(i) = \begin{cases} rate_d^s(i-1) + \alpha(1 - rate_d^s(i-1)) & (s \text{ is } s_{best}) \\ (1-\alpha)rate_d^s(i-1) & \text{otherwise} \end{cases}$$

    $m_d^s(i) = rate_d^s(i)m_d(i)$: allocate on s by d at i

  end for

end for

desire providing the best utility per a unit of used system resource at $(i-1)$-th iteration. The ratio of desire at i-th iteration is given as follows.

$$rate_d(i) = \begin{cases} rate_d(i-1) + \alpha(1 - rate_d(i-1)) & (d \text{ is } d_{best}) \\ (1-\alpha)rate_d(i-1) & (\text{otherwise}) \end{cases} \tag{16}$$

Let α denote the sensitive factor indicating how sensitively users adjust the rate of allocated resources based on the previous allocation. Users allocate $m_d(i) = rate_d(i)x_u^c(i)$ on desire d.

After that, desires decide the allocation of the given system resource. The amount of system resource allocated on services is adjusted in the same way as the allocation on . The initial amount of system resource allocated on services is determined based on the evaluated values of services.

Let $rate_d^s(i)$ denote the rate of system resource allocated on provider s by desire d. $rate_d^s(0)$ is based on the evaluated value of service and equals $\alpha_d^s / \sum_{s' \in S_d} \alpha_d^{s'}$. Desires increase the rate of s_{best}, which is the provider providing the best utility per a unit of used system resource in S_d at $(i-1)$-th iteration. The rate is given as follows.

$$rate_d^s(i) = \begin{cases} rate_d^s(i-1) + \alpha(1 - rate_d^s(i-1)) & (s \text{ is } s_{best}) \\ (1-\alpha) rate_d^s(i-1) & (\text{otherwise}) \end{cases} \quad (17)$$

In a similar way of the above-mentioned procedure, the allocation of future system resource is coordinated based on the utility gained by providers at previous iteration. The weight of the future desire equals the weight of the desire times the period that the desire is active in the considered future. That is, w_d^f, which is the weight of the future desire, equals $w_d(\min(t_d^{end}, t+T-1) - \max(t+1, t_{start}))$. $rate_d(0)$ equals $w_d^f / \sum_{d' \in D_u^f} w_{d'}^f$

4.4.4 Behavior to Allocate Service Resource

The behavior of providers who allocate service resource on users is shown in Algorithm 3. Providers decide the allocation of service resource based on the amount of allocated system resource and the evaluated value of user. Providers treat the amount of allocated system resource times the evaluated values of users as the ratio of service resource allocated on users. When the calculated amount is less than the ceiling of the user, the calculated amount of service resource is allocated on the user. When the calculated amount is more than the ceiling of the user, the ceiling amount of service resource is allocated on the user. This procedure is repeated until there are no users who need more service resource or no service resource.

Let $U_{unsatisfied}$ denote the set of users the allocated service resource of which doesn't reach the ceiling. Let q_s^u denote the amount of the service resource s allocated on u. Let c_s^u denote the ceiling amount of service resource s allocated on u. the amount of service resource s allocated on u is given as follows.

$$q = \min\left(Q_s \frac{v_u}{\sum_{u \in U_{unsatisfied}} v_u}, c_s^u - q_s^u\right) \quad (18)$$

Provider allocates service resource on users in $U_{unsatisfied}$ and update $U_{unsatisfied}$.

Algorithm 3 Service resource allocation on users

U_s : set of users which can use s

$q_s^u = 0(u \in U_s)$: resource allocated to u by s

$U_{unsatisfied} = U_s$: set of users unsatisfied with resource

$c_s^u = \min(r_d, l_s^u)$: ceiling amount of resource allocated to u

$q_{left} = Q_s$: left resources which s has

$v_u = m_u^s \alpha_s^u$: evaluated value of u for the allocation

while $U_{unsatisfied} \neq \phi$ and $left > 0$ **do**

    $q_{given} = 0$

    for all $u \in u_{unsatisfied}$ **do**

        $q = \min(q_{left} v_u / \sum_{u \in U_{unsatisfied}} v_u, c_s^u - q_s^u)$

        $q_{given} = q_{given} + q$

        $q_s^u = q_s^u + q$

        if $q_s^u == c_s^u$ **then**

            $U_{unsatisfied} = U_{unsatisfied} \setminus \{u\}$

        end if

    end for

    $q_{left} = q_{left} - q_{given}$

end while

Provider repeats the procedure until $U_{unsatisfied}$ or service resource which the provider can provide becomes empty.

Chapter 5 Simulation of Resource Allocation

In this chapter, I describe the settings and results of the simulations conducted to verify the market model and the behaviors of users and providers.

5.1 Simulation Settings

We conduct simulations to verify the resource allocation based on preferences of users and providers using the above-mentioned market-model and behaviors of users and providers. In this simulation, random number is identically-distributed. the number of users is 100 ($|U| = 100$), and the number of services is 100 ($|S| = 100$). Simulated period is 200. The number of the desires which a user has in the period is a random number from 6 to 10. The period of a desire is a random number from 10 to 30. r_d, which is the size of a desire is a random number from 10 to 20. w_d, which is the weight of a desire is a random number from 0 to 1. $|S_d|$, which is the number of services a desire uses is a random number from 3 to 7. The service has a quality value, which is a random number from 0 to 1. The evaluated value of service is normalized based on the quality value. Q_s, which is the largest amount of service resource provider can provide is a random number from 10 to 20. The period considered as future is 20 ($T = 20$). α, which is a sensitive factor used in user's adjusting system resource allocation is 0.01. The sum of the system resource is 1000 ($\beta = 1000$). It means that each user receive 10 system resource every time.

5.2 Utilities of Users and Providers

In this section, the allocation of goods in the market is described to show how goods is allocated on users and providers. The weights of desires which a user has, the amount of current system resource which the user uses and the utility which the user gains are shown in Figure 7, 8 and 9. The horizontal axis of the Figures denotes time in the simulation. The vertical axis in Figure 7 denotes the sum of weights w_d of the current desires which the user has. The vertical axis in Figure 8 denotes the amount of current system resource which the user uses to obtain service resources. The vertical axis in Figure 9 denotes the utility which the user gains.

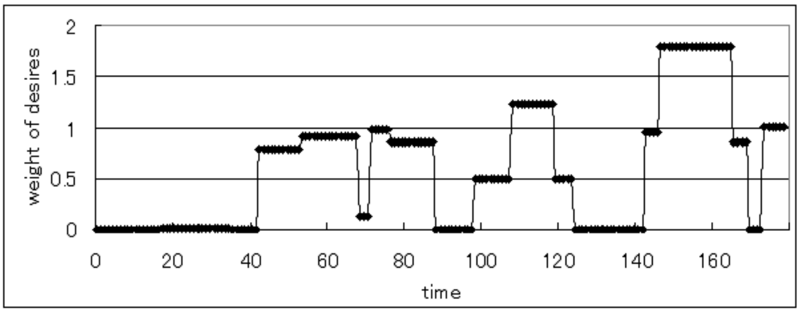

Figure 7: Desire weight of a user

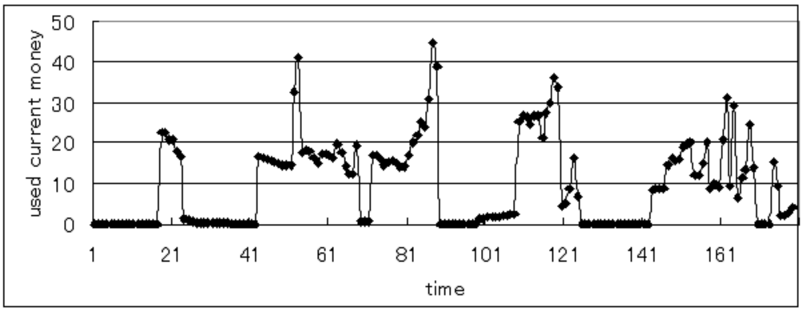

Figure 8: Amount of current system resource which the user uses

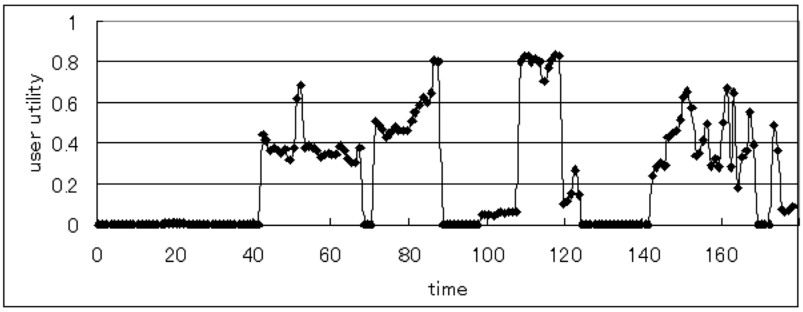

Figure 9: Utility which the user gain

This user has a desire the weight of which is small from 17 to 34. Then, this user has desires from 42 to 87, from 98 to 123, from 142 to 168 and after 173. Until time 16, this user releases all his current system resource to obtain future system resource since he has no current desire. The stored future system resource is used from time 17. Until time 22, he uses relatively large amount of system resource for the light-weighted desire since the period considered as future is 20 and there are no desires weighted more. From 23 to 42, this user releases current system resource for heavily-weighted desires in the future. From 43, the user can obtain much utility by using much current resource stored from 22. In a similar way, he releases future system resource to obtain current system resource, and gain much current utility since the user have little future desire around 122 time. Users manage their system resources based on their desires and behaviors of other users. As the result, they can increase utility according to their desires.

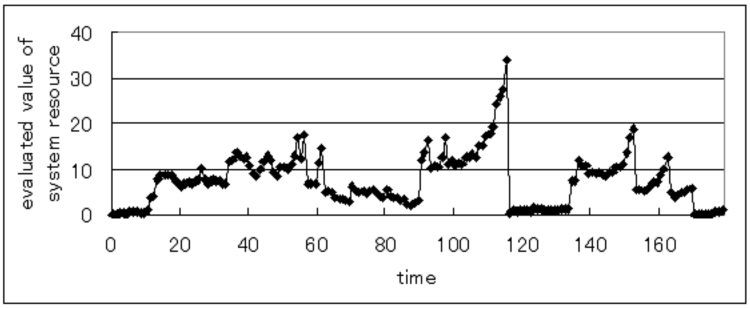

Figure 10: Demand for the provider

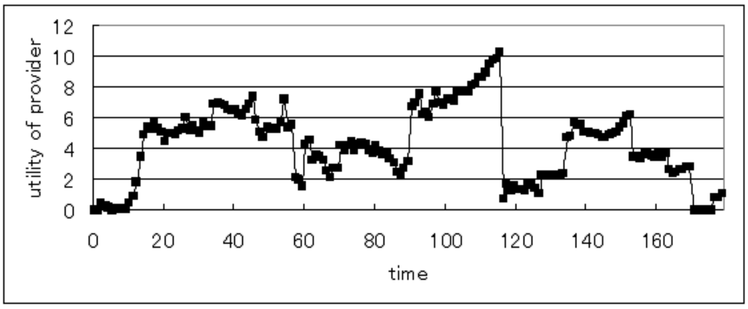

Figure 11: Utility of the provider

The amount of current system resource allocated on a provider times evaluated value of users and the utility of the provider are shown in Figure 10 and 11 respectively. The evaluated value is large when the user who is assigned an large evaluated value allocates a large amount of system resource on the provider. The value is small when the user who is assigned an small evaluated value allocates a small amount of system resource. The vertical axis in Figure 11 denotes the utility of the provider. These graph shows that the utility which the provider gains is large when the evaluated value of demands for the provider is large.

5.3 Competition with a Demand-based Approach

In this section, I compare my approach with a demand-based approach. The demand-based approach allocates the system resource on providers based on demands of users. A lot of system resource is allocated on the provider for which many users have heavily-weighted demands. Providers produce the service resource according to the amount of allocated system resource and his product function and then allocate it on users based on their preferences for users and demands of users.

At first, I compare the preferences of users and providers in changing the rate of the users who input sudo-demands. In volunteer-based free Internet services, users may not specify the period in which they use the target resources although they actually use them in a certain period. The service resource allocated on sudo-resource is not used and wasted by the user. In this simulation, I record the sum of the utilities of users and providers in changing the rate of the users who input sudo-demand.

The average of the sum of the utilities is shown in Figure 12 and 13 respectively. The horizontal axis shows the rate of the users who input sudo-demand. The vertical axis shows the average sum of the utilities. When the rate of users using sudo-demand is 0, the sum of the utility of users in demand-based approach is a little larger than that in my approach. It is because the system resource is allocated on providers based on the demands of users and the resource allocation in demand-based approach is more appropriate for users than that in market-based approach. As described at the above section, the user who has no or little demands in future allocates relatively large amount of current system resource on their light-weighted desires.

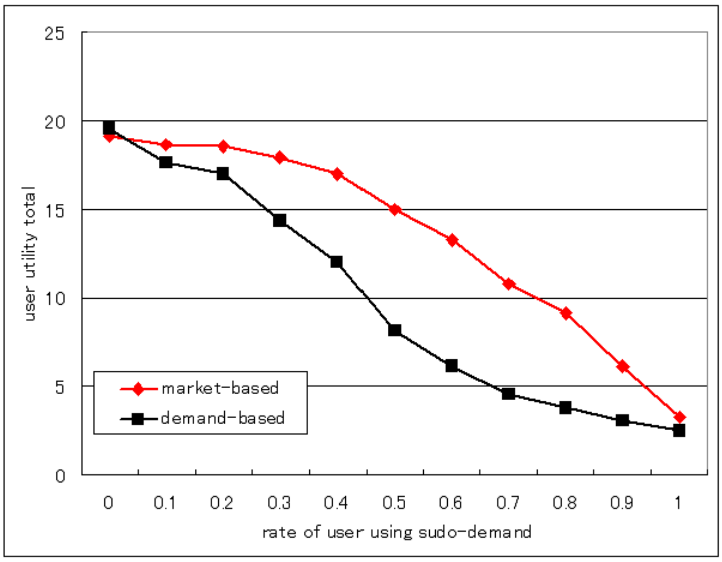

Figure 12: Utility of users in changing the rate of users using sudo-demands

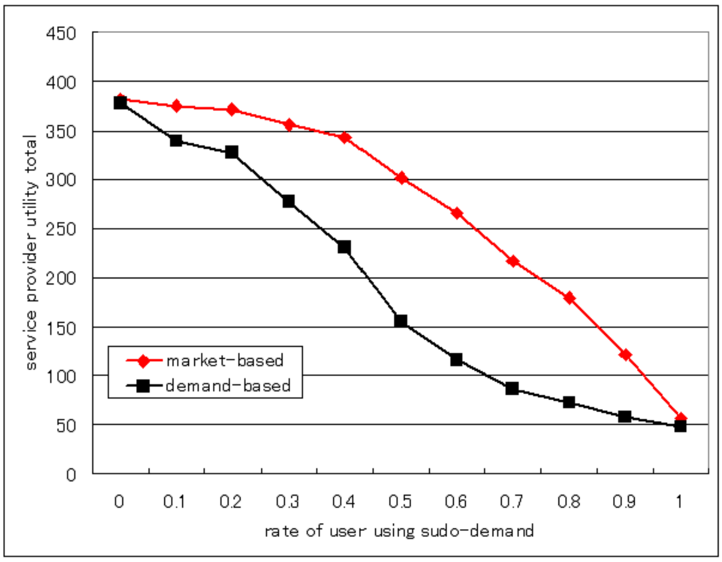

Figure 13: Utility of providers in changing the rate of users using sudo-demands

When the rate of users using sudo-demands increases, the utility of users and providers decreases in either market-based approach or demand-based approach. The service resources allocated on sudo-demands is not used. Since the amount of service resource allocated on true demand decreases, the utility of users and providers decreases.

The decrease of the utility of users and providers in my approach is smaller than that in demand-based approach. In market-based approach, the sudo-demands consume the system resource of the user every time. Then, since the amount of system resource which sudo-demands can use is relatively smaller than that which true demand can use in a certain time, the amount of service resource allocated on sudo-demands is smaller than that allocated on true demands. On the other hand, sudo-demands in demand-based approach can obtain as much system resource as other true-demands can. As the result, the effect of sudo-demand on other users in market-based approach is smaller than that in demand-based approach.

The system needs to motivate users to input their true demands since the increase of sudo-demands leads to the decrease of the social surplus in the system. I compare the utility of a user using true demands with that of the user using sudo-demands. In this simulation, other users input true demands.

In using demand-based approach, the utility of user when the user uses sudo-demands is the almost same as that when the user uses only true demands since the system resource allocation is based on the demands of users. Even if the user wastes much service resource previously, the system resource is used for the user in a similar manner with other users. It is difficult for the system to motivate users to input true-demands.

In using my approach the competition of the utility of the user using true demands with that of the same user using sudo-demands is shown in Figure 14. When the user inputs sudo-demands, the user uses his current system resource to obtain service resource for the sudo-demands. As the result, the utility of the user using sudo-demands is much smaller than that which the user gains by using true demands since the amount of system resource which the user can use for his true demands is small. In order to increase his utility, the user should input his true demands. It means that the system using market-based approach can motivate users to use true-demands.

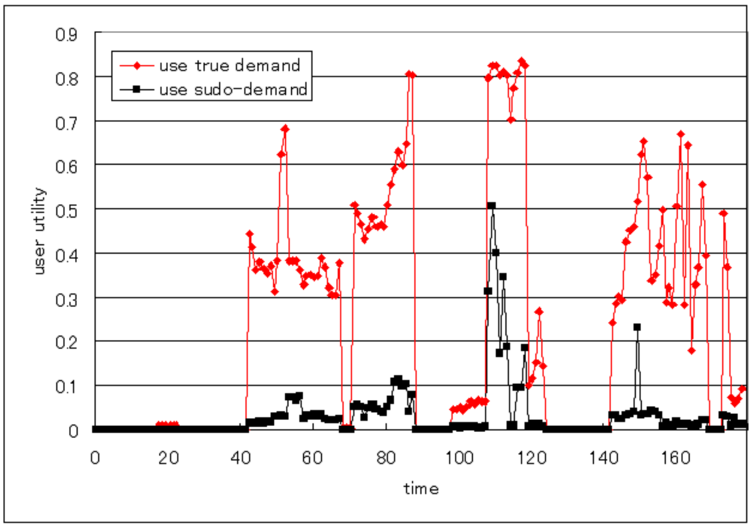

Figure 14: Competiion the utility which a user gains with using sudo-demands with that which the user gains with using true demands

5.4 Sample Scenario in Application

I conduct the simulation assuming that my approach is applied to Language Grid, which is one of the actual volunteer-based free Internet services. Language Grid provides shared computational resource in order to allow language resource to provide services. Language resource providers include NPO's, universities and so on. They provide their language resources for effective utilization. Some providers have preferences for users. For example, a university prefers students to other general users. Users can use the provided language resources only for non-profit purposes without charge. The example of user's preferences is that they prefer high-quality translation engines or specialized dictionaries. The instance of user's desires is that they need English-Japanese translation in a certain time. There are multiple language resources which can translate sentences from English to Japanese in the system.

One of issues of Language Grid is that the resource have to be appropriately allocated on various users. Students and NPO use Language Grid for important experiments and events. They use the certain resource in a certain time. On the other hand, there are some service site such as bilingual corpus sites, trial sites and so on. Trial sites provide anonymous users with services of Language Grid for trial. They use Language Grid on a steady basis. The finite resource in Language Grid should be appropriately allocated on these various users.

The resource of Language Grid is manually allocated on users. For example, when a student use Language Grid for important demos or experiments, it is announces in order for other users not to use the system at the same time. The resource allocated on trial sites is small so that the trial sites do not interrupt student or NPO users. When the resource allocation is dynamically allocated based on the demands, it has the same issues as volunteer-based free Internet services. Namely, the resource should be appropriately allocated based on preferences of various users and providers. And it is possible for users to input sudo-demands. Consequently, I verify the resource allocation which the proposed approach is applied to Language Grid, by simulations.

The following users and services are added to the simulation settings.

- service 1
  - provide high-quality service
  - prefer students to other users
- service 2, 3, 4
  - provider low-quality service
  - have almost same preferences for users
- user 1 (student)
  - use service 1, 2, 3, 4 in a short period
  - prefer service 1 to other services
- user 2 (student)
  - use service 1, 2, 3, 4 in a longer period than user 1
  - prefer service 1 to other services
- user 3, 4, 5 (trial site)
  - always use service 1, 2, 3, 4
  - prefer service 1 to other services

The desires of user 1 and 2 is shown in Figure 15 and 16 respectively. Both user 1 and user 2 have a desire for services from 90 to 110. Then user 1 have a short desire from 140 to 160. user 2 has a short desire from 130 to 170, which is longer than the second desire of user 1.

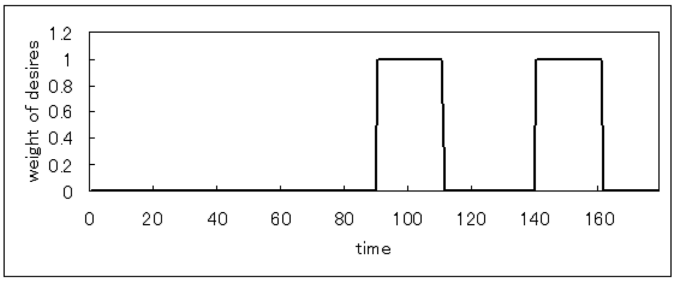

Figure 15: Desires of user 1

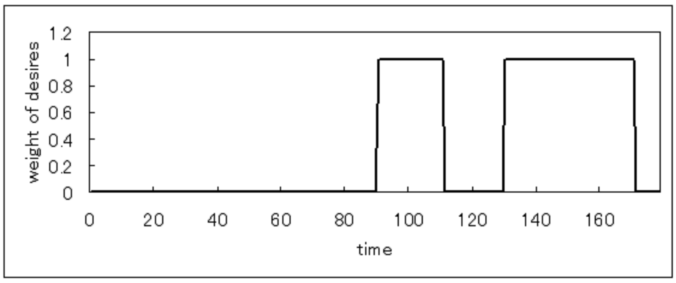

Figure 16: Desires of user 2

The comprising services allocated on trial site 3 is shown in Figure 17. While student users have no desire, user 3 can obtain much service resource of service 1, which is a high-quality service. For shortfall of service resource, he obtains service 2, 3 and 4, which is low quality service. From time 71, the expected future utility is decreased due to the desire of user 1 and 2 beginning from time 90. Then, user 1 releases future system resource to increase his current utility. From time 90, desires of user 1 and 2 start. Since they are assigned high-priority by service 1 and they have released their current system resource until this time, they have more system resources than other trial sites. As expected, the amount of service resources which trial sites can obtain and their utility are decreased. After the desires of student users finished, trial sites can obtain the service resource of service 1 again. Then, although desires of student users start from 130 and 140, the amount of system resources which student users have is relatively smaller than before. Trial sites can obtain service resources of low quality services.

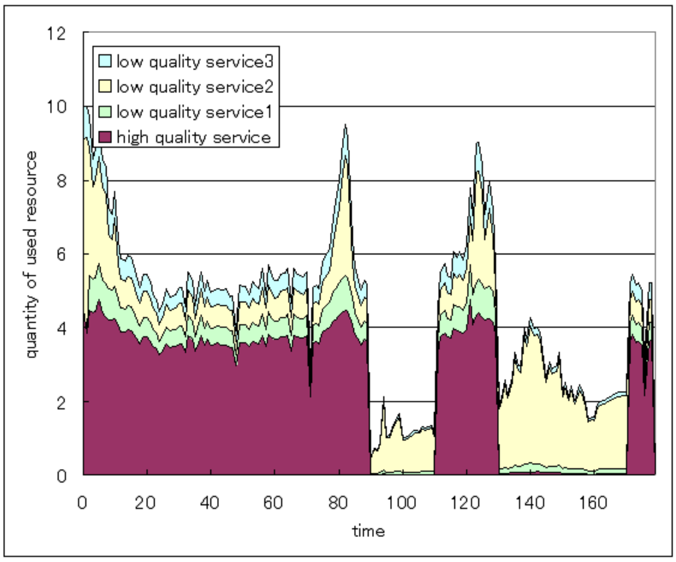

Figure 17: Comprising services allocated on a trial site

The comprising users who use high quality service is shown in Figure 18. While highly-preferred users have no desire, the service resource is allocated on trial sites. When student users have desires for the service, the service resource is preferentially allocated on student users. The amount of system resource allocated on service 1 is large since users try to obtain as much service resource of high-quality service 1 as possible. The amount of produced service resource and the utility of the provider is also large. When multiple users have a desire for the same service, the user who uses the service in a focused way in terms of time and variety of services can procedure much service resource.

The comprising users who use a low quality service is shown in Figure 19. The low quality service is used by the user who cannot obtain enough service resource of the high-quality service. The service is used by trial sites who are not satisfied with the service resource of high-quality service. From time 90 to 110, the service is used by trial sites and student users, since they are not satisfied with the amount of high-quality service resource. From time 140, the

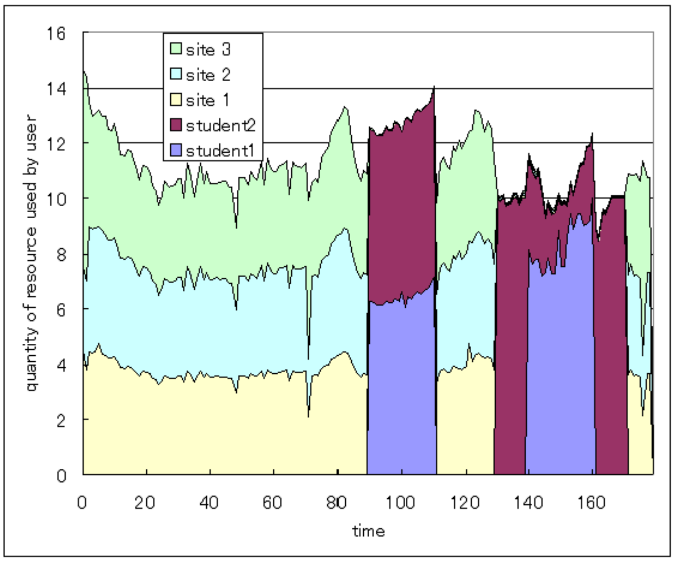

Figure 18: Comprising users who uses a high quality service

service resource is mainly used by trial sites who obtain little service resource of service 1. When the demand of the system is large, the amount of used low-preferred service resource is used. When the demand of the system is small, the amount of that is small.

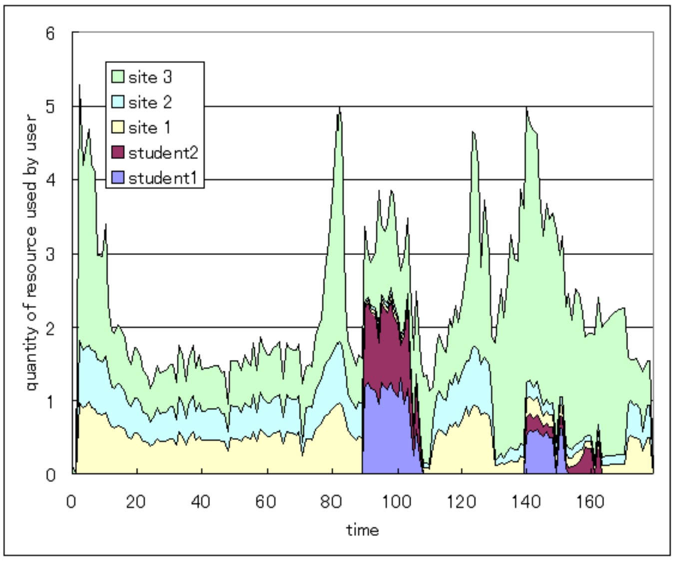

Figure 19: Comprising users who use a low quality service

These results show that the highly-preferred users can obtain the service resource in preference to other users. Other trial sites can obtain the service resource which is not used by highly-preferred users. Trial sites obtain service resources when students have no demands or the service resources which students do not need. Namely, they can use the Language Grid without interrupting student users. High-priority students can obtain high-quality service resources by specifying the time of desires. It means that proposed method can motivate even high-priority users to input describe true demands. These results show that proposed method can realize the appropriate resource allocation in Language Grid which has various users.

5.5 Discussion

The results of the simulations show that users can manage the allocated system resources and increase their utility along with their desires. Providers can increase their utility along with the demand for their resource and their preferences for users. The result of the simulation assuring Language Grid shows that the low-preferred user can obtain the low quality service which is not preferred by other users and the high-quality service only when highly-preferred user have no or little desires. The user who specifies services and periods in a focused way can obtain more service resource than others. Additionally, my approach motivates users to input true demands and decreases the effects of users using sudo-demands on other users.

There is often an administrator who manages the whole system in volunteer-based free Internet services. For example, users have to send request to provider via the administrator in PRAGMA. There is the administrator governing the system in Language Grid, too. The role of the administrators is to adjust the system to work well along with the policy of the system. The administrator in Language Grid may have preferences for users. Student and NPO users who use the system for important experiments and events is more preferred to other anonymous users. The administrator can reflect their preference on the system by adjusting the system resource allocation. The user on which more system resource is allocated can obtain more service resource than other users.

One of the problems of my approach is that inactive user may have much future system resource. In volunteer-based free Internet services, there are some inactive users, who do not use the system for long time. The inactive users release all their current system resource for each time since they have no current desire. After they continue this behavior for long time, they have much more system resources than the users who use the system regularly. The inactive users may obtain much service resource by the large amount of system resources. It interrupts other users who use the system regularly. One of the countermeasures to the problem is that the system adjusts the system resource allocation based on how frequently the user uses the system. The user who frequently uses the system is allocated more system resource than that who rarely uses the system. This can prevent inactive users to have much system resource. But, the mechanism which evaluate users based on the use frequency, may motivate users to input sudo-demands. It is necessary to consider an appropriate countermeasure.

Chapter 6 Application of Proposed Approach

In this chapter, I describe the system architecture of Language Grid, XACML, which is a well-understood access control language, and the way to apply my approach to Language Grid.

6.1 Architecture of Language Grid

The actors of Language Grid are computational resource providers, language resource providers, users and an administrator. Computational resource providers provide computational resource on which language resources run. Language resource providers provide language resource which provides users with service. Users can use the language resources provided in Language Grid. The administrator manages the whole system so that the system works well along with the policy of the Language Grid. For example, he decides the allocation of language resources on computational resources. He also observes whether or not there are any illegal users who use too much language resource. The administrator allows users to use language resources without considering computational resources.

Computational resource providers provide computational resource on which language resources run. The information of the provided computational resource is registered to the system. When the administrator allocates new language resource on a computational resource, he need an acceptance of the computational provider. Therefore, the computational resource provider can limit the language resource deployed on his resource.

Language resource providers provide language resources which provide services. The information of the provided language resource is registered to the system, too. The registered information includes language paths which the language resource deals with, the interface of the resource, the information of the provider, and so on. Language resource providers can limit the use of their resource. On the other hand, the amount of the service resource which the language resource provides is determined by the allocation of computational resource. The allocation of computational resource is determined by the administrator along with the policy of the system and demands of users.

Users can use the provided language resource for free. There are multiple services which provide the same kind of service. They can create composite services by combining multiple services. For example, there are multiple translation engines which provide translation service from Japanese to English. User can create a new translation service from Japanese to Chinese by combining Japanese-English translation service and English-Chinese one.

One of the problems in Language Grid is that users have no way to know the availability of objective services in a certain time. The limit of language resources protects only the rights which the language resources possess. It doesn't assure that users can use the service within the limits. When the demand of the system is high, users may use less service resource than the limitation. For example, a limitation of a service is that a user can use the service 1000 times a day and a user want to use it 1000 times from 13 to 14 o'clock in a day. If the demand of the system in the time is large, the system has so many loads that it may not be able to handle the whole requests from the user. If the user comprehends the demand of the whole system, he may estimate the availability of the service. But I cannot assume that in large-scale and open systems. Applying my approach to Language Grid allows users to know the availability of services in advance. They can adjust their behavior along with the availability based on the resource allocation determined by my approach.

In Language Grid, users have preferences for interchangeable services based on the quality of the services. For example, users assign high preference for the service which provide good quality translation or is specialized for the objective domain. Resource providers also have preferences for users. a university wants to allocate the service resource on students preferentially when students want to use its service. When students do not want to use its service, the university is interested in providing other trial sites with its service. An NPO which provides a dictionary specialized for the domain of their activity provides the same domain groups with the service resource preferentially. The resource allocation has to be determined along with these preferences. Additionally, the system has to motivate users to input true-demands since the users may input sudo-demands for charge-free resources and it is difficult to determine whether they are true or not. Applying my approach allows the system to allocate the resources based on the preferences of users and providers and motivates users to input their true demands.

6.2 XACML

As the use of the resource on network becomes popular, the technologies to protect the rights of resources have been studied. MPEG-21 REL [16] and ODRL are rights expression languages which describe the limitation of resource. XACML [17] is a well-understood access control language to describe policies for the resource. Resource providers describe the access policies with the well-understood rights expression languages to protect the rights of the resource. The system controls the access based on the described policies. The provider who provides a dictionary limits the frequency of use lest illegal users extract all data from the dictionary. The provider who provides a nonfree translation engine for the advertisement strictly limits the frequency of use lest the number of user who uses the nonfree service decreases.

XACML is a well-understood access control language proposed by OASIS. The policy for the resource is described in XML format. The system controls the access based on the described policies. The data structure used in XACML includes the following items.

- Target
  - Subject
  - Resource
  - Action
- Rule
  - Target
  - Condition

Subject is an entity to which the policy is applied. Resource is an entity for which the policy is described. Action is an operation which Subject is allowed to perform on Resource. The policy is applied to Target who corresponds to Subject, Resource and Action. Rule denotes whether the policy accept or reject the access. Rule includes Target and Condition elements. Condition includes the elements which do not belong to Target, such as time and environment.

The data flow in XACML architecture is shown in Figure 20. The XACML policy described by providers is managed in Policy Administration Point (PAP). PAP returns the policy corresponding to Target. The request of users for resource is passed to Policy Enforcement Point (PEP). PEP creates XACML requests based on the request and sends it to context handler. The XACML request includes the identifier of the objective resource and the user who accesses the resource, the way of use and so on. Context handler forwards the request to Policy Decision Point (PDP). PDP inquires context handler about the information which is necessary for check and is not included in the request. Context handler forwards the inquiry to Policy Information Point (PIP). PIP obtains the information from target system and returns it to PDP via context handler. PDP determines whether the access is acceptable or rejectable based on the information and returns the result to PEP. PEP accepts the access only when PDP returns the acceptable result. The above procedure realizes the access policy described by providers.

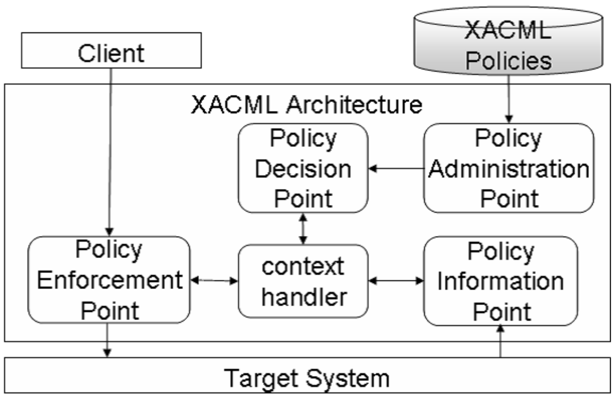

Figure 20: Data flow of XACML [18]

Although XACML is a well-understood access policy language, it is hard for user to describe his policy without excess and deficiency. Kolovski et al. [19] formulates XACML using description logic and proposes a service to analyze the described policy. This approach allows users to describe appropriate policies using XACML.

6.3 Architecture to Apply my Approach

In this section, I describe architecture to apply my approach to Language Grid using XACML. The following three items are used for market-based resource allocation approach.

- Desires and preferences of users for resources
- Preferences of providers for users
- Policies described by providers

Users describe their preferences and desires for services. Providers describe their preference for users. This information is described in original format and managed by the system. Users and Providers can reflect their preferences on the resource allocation by modifying their inputs. Providers describe XACML policies to protect rights which their resources possess. The market-based resource broker allocates service resources on users based on the inputs and outputs XACML policies for the system based on the result of resource allocation. The XACML architecture, which is a gateway of Language Grid, controls the access along with the XACML policies which the resource broker outputs.

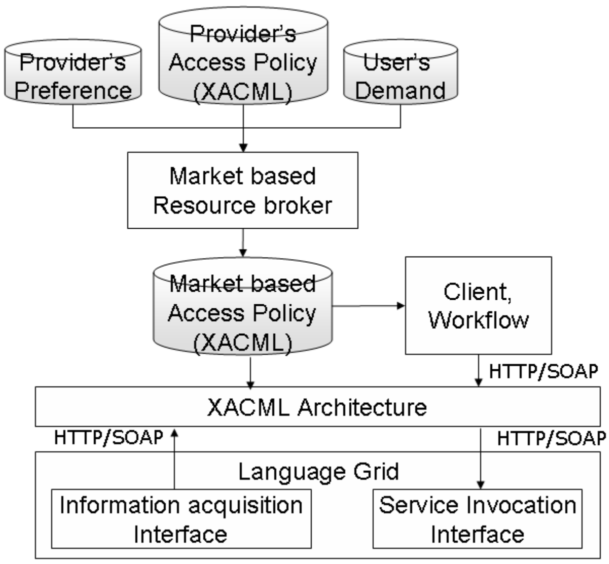

Figure 21: Architecture to apply my approach

For example, a provider of a dictionary service whose identifier is service 1, limits the access for the resource as follows.

- client of the user must display the license of the service
- Number of using the service is less than 100
- Size of response is less than 5KB
- Available from 1st November, 2007 to 31th October, 2008

This policy is described in XACML in the following way.

- Target
  - Subject: user uses the client which is able to display licenses
  - Resource: serviceId is service1
  - Action: Invoke
- Rule
  - Subject: Number of invoking the service in the day is less than 100
  - Action: Size of the response to the request is less than 5KB
  - Condition: Date is from 1st November, 2007 to 31th October, 2008

The resource broker determines whether the user can use the objective resource or not based on the policies and assign the limitation of the resource based on the policy. In the above policy samples, the limitation of the number to invoke the service a day equals the ceiling amount of allocated service resource.

The resource broker outputs XACML policies based on the market-based resource allocation by adding user identifier and a period to the original policy. The following policy is output for the above policy sample.

- Target
  - Subject: userId is user1 and the client of the user can display licenses
  - Resource: serviceId is service1
  - Action: Invoke
- Rule
  - Subject: Number of invoking the service in the day is less than 100 and that in the period of the market is less than 15
  - Action: Size of the response to the request is less than 5KB
  - Condition: Date is from 1st November, 2007 to 31th October, 2008 and during the period which the market defines

The output policy is passed to the XACML gateway in front of Language Grid. The gateway controls the access based on the policies. Users refer to the output policy to know the available of resources. Users can use the services which the policies tell available for them and invoke the workflow which is composed of available services.

The above architecture realizes an appropriate resource allocation based on preferences of users and providers and policies of providers. Providers describe access policies to protect rights of resources with a well-understood access policy language. The proposed architecture conducts a market-based resource allocation using the policies and preferences of users and providers in original format and output XACML policies which assure the availability of services. Users can know the availability of services in advance and select services out of available services. The system controls the access according to the output policies.

Chapter 7 Summary

In this research, I study the resource allocation for volunteer-based free Internet services. In the system, users and providers have preferences for each other. The systems have to motivate users to input true demands since there are no cost constraint for users in the systems and it is hard to determine whether the described demand is true or not. To realize an appropriate resource allocation in the systems, I propose a market-based approach. There are two problems in allocating resources for volunteer-based free Internet services.

- Problem establishment of the resource allocation

  Users in volunteer-based free Internet services have no cost constraints since most resources are charge-free. The objective of providers is that their resources are used by favorable users. To realize an appropriate resource allocation, it is necessary to clearly define the objectives and constraints in volunteer-based free Internet services and the characteristics which allocation methods should have.
- Appropriate resource allocation in large-scale systems

  The system has to allocate resources in applicable time even if there are many users and providers in the system. The technique of the system also has the necessary characteristics for the resource allocation.

In this research, I model the resource allocation problem of the volunteer-based free Internet services to realize appropriate resource allocation and propose a resource allocation method which uses market-based model. This research has the following two contributions.

- Formulate the resource allocation problem

  I clarify and formulate the requirements of the resource allocation in volunteer-based free Internet services based on actual systems and propose a model to deal with the requirements. In volunteer-based free Internet services, users and providers have preferences for each other. Users have desires including multiple interchangeable services. Providers decide the allocation of their resources on users. Additionally, I describe the characteristics which allocation methods should have.

- Propose a resource allocation method using heuristics

  I propose the market model composed of current-future model and consumer-producer model to realize the appropriate resource allocation for volunteer-based free Internet services. I also propose an approach to allocate resources for the market-based model. This approach is able to allocate resource in applicative time even in large-scale systems. I describe that my approach can allocate resources based on preferences of users and providers and has the characteristics necessary for the resource allocation in volunteer-based free Internet services.

Finally, I propose the architecture to apply my approach to the actual volunteer-based free Internet services. The architecture outputs policies which assure the availability of resources based on the preferences of users and providers and policies which providers describe in order to protect rights of resources. The gateway of the objective system controls the access corresponding to the output policies to realize the resource allocation. Providers can describe their policies without the availability of resources.

The above contributions realize the appropriate resource allocation for volunteer-based free Internet services, considering users' and providers' preferences. It can motivate the users to input true demands and decrease the effects of sudo-demands on other users.

Acknowledgments

The author would like to express his gratitude to the supervisor, Professor Toru Ishida at Kyoto University, for his consecutive instruction, pertinent advice and earnest argument.

The author would like to tender his acknowledgments to Associate Professor Shigeo Matsubara and Assistant Professor Hiromitsu Hattori at Kyoto University, for his technical and constructive advice.

The author would like to express his appreciations to the advisers, Professor Hajime Kita at Kyoto University, Researcher Yohei Murakami at National Institute of Information and Communications Technology, and Professor Masatoshi Yoshikawa at Kyoto University for his valuable advice.

Finally, the author would like to thank all members of Ishida and Matsubara laboratory for their various supports and discussions.

References

[1] Ishida, T.: Language Grid: An Infrastructure for Intercultural Collaboration, *SAINT '06: Proceedings of the International Symposium on Applications on Internet*, Washington, DC, USA, IEEE Computer Society, pp. 96–100 (2006).

[2] Jaeger, M. C., Muhl, G. and Golze, S.: QoS-Aware Composition of Web Services: A Look at Selection Algorithms, *ICWS '05: Proceedings of the IEEE International Conference on Web Services (ICWS'05)*, Washington, DC, USA, IEEE Computer Society, pp. 807–808 (2005).

[3] Berbner, R., Spahn, M., Repp, N., Heckmann, O. and Steinmetz, R.: Heuristics for QoS-aware Web Service Composition, *ICWS '06: Proceedings of the IEEE International Conference on Web Services (ICWS'06)*, Washington, DC, USA, IEEE Computer Society, pp. 72–82 (2006).

[4] Maximilien, E. M. and Singh, M. P.: Toward autonomic web services trust and selection, *ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing*, New York, NY, USA, ACM, pp. 212–221 (2004).

[5] Zeng, L., Benatallah, B., Ngu, A. H., Dumas, M., Kalagnanam, J. and Chang, H.: QoS-Aware Middleware for Web Services Composition, *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, pp. 311–327 (2004).

[6] Canfora, G., Penta, M. D., Esposito, R. and Villani, M. L.: An approach for QoS-aware service composition based on genetic algorithms, *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, New York, NY, USA, ACM, pp. 1069–1075 (2005).

[7] Aggarwal, R., Verma, K., Miller, J. and Milnor, W.: Constraint Driven Web Service Composition in METEOR-S, *SCC '04: Proceedings of the 2004 IEEE International Conference on Services Computing*, Washington, DC, USA, IEEE Computer Society, pp. 23–30 (2004).

[8] Guan, Y., Ghose, A. K. and Lu, Z.: Using constraint hierarchies to support QoS-guided service composition, *ICWS '06: Proceedings of the IEEE International Conference on Web Services (ICWS'06)*, Washington, DC,

USA, IEEE Computer Society, pp. 743–752 (2006).

[9] Menascé, D. A. and Dubey, V.: Utility-based QoS Brokering in Service Oriented Architectures, *ICWS '07: Proceedings of the IEEE International Conference on Web Services (ICWS'07)*, IEEE Computer Society, pp. 422–430 (2007).

[10] Yuan, E. and Tong, J.: Attributed Based Access Control (ABAC) for Web Services, *ICWS '05: Proceedings of the IEEE International Conference on Web Services (ICWS'05)*, Washington, DC, USA, IEEE Computer Society, pp. 561–569 (2005).

[11] Weng, C., Li, M., Lu, X. and Deng, Q.: An economic-based resource management framework in the grid context, *CCGRID '05: Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05) - Volume 1*, Washington, DC, USA, IEEE Computer Society, pp. 542–549 (2005).

[12] Galstyan, A., Czajkowski, K. and Lerman, K.: Resource Allocation in the Grid Using Reinforcement Learning, *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, Washington, DC, USA, IEEE Computer Society, pp. 1314–1315 (2004).

[13] Buyya, R., Abramson, D. and Venugopal, S.: The Grid Economy, *Proceedings of the IEEE*, Vol. 93, No. 3, pp. 698–714 (2005).

[14] Yamaki, H., P.Wellman, M. and Ishida, T.: Controlling Application QoS Based on a Market Model, *The transactions of the Institute of Electronics, Information and Communication Engineers*, Vol. 81, No. 5, pp. 540–547 (1998).

[15] Kuwabara, K., Ishida, T., Nishibe, Y. and Suda, T.: *An equilibratory market-based approach for distributed resource allocation and its applications to communication network control*, World Scientific Publishing Co., Inc., pp. 53–73 (1996).

[16] Wang, X.: MPEG-21 Rights Expression Language: Enabling Interoperable Digital Rights Management, *IEEE MultiMedia*, Vol. 11, No. 4, pp. 84–87 (2004).

[17] Lorch, M., Proctor, S., Lepro, R., Kafura, D. and Shah, S.: First experiences using XACML for access control in distributed systems, *XMLSEC '03: Proceedings of the 2003 ACM workshop on XML security*, New York, NY, USA, ACM, pp. 25–37 (2003).

[18] Moses, T.: eXtensible Access Control Markup Language TC v2.0 (XACML) (2005).

[19] Kolovski, V., Hendler, J. and Parsia, B.: Analyzing web access control policies, *WWW '07: Proceedings of the 16th international conference on World Wide Web*, New York, NY, USA, ACM, pp. 677–686 (2007).

Appendix

A.1 Instances of XACML Document

Listing 1: Instance of XACML document written by a provider

```
<Policy PolicyId="GeneratedDictionary1Policy" RuleCombiningAlgId="
    urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:ordered-permit-
    overrides">
  <Description>description</Description>
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:boolean-
            equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#boolean">
              true</AttributeValue>
          <SubjectAttributeDesignator AttributeId="use-license-show-client"
              DataType="http://www.w3.org/2001/XMLSchema#boolean"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
            equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              service1</AttributeValue>
          <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1
              .0:resource:resource-id" DataType="http://www.w3.org/2001/
              XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
            equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              invoke</AttributeValue>
          <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0
              :action:action-id" DataType="http://www.w3.org/2001/XMLSchema#
              string"/>
        </ActionMatch>
      </Action>
    </Actions>
  </Target>
  <Rule RuleId="InvokeRule" Effect="Permit">
    <Target>
      <Subjects>
        <AnySubject/>
      </Subjects>
      <Resources>
```

```
        <AnyResource/>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-
              greater-than-or-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer"
                >5120</AttributeValue>
            <ActionAttributeDesignator AttributeId="response-byte-size"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-less-
          than">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-
            and-only">
          <SubjectAttributeDesignator AttributeId="invoke-num-per-day-
              Dictionary1" DataType="http://www.w3.org/2001/XMLSchema#integer"/
              >
        </Apply>
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">100
            </AttributeValue>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-greater
          -than">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-one-
            and-only">
          <ActionAttributeDesignator AttributeId="invoke-dateTime" DataType="
              http://www.w3.org/2001/XMLSchema#dateTime"/>
        </Apply>
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">
            2006-11-20T19:59:20.531000000+12:00</AttributeValue>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-less-
          than">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-one-
            and-only">
          <ActionAttributeDesignator AttributeId="invoke-dateTime" DataType="
              http://www.w3.org/2001/XMLSchema#dateTime"/>
        </Apply>
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">
            2008-11-20T19:59:20.531000000+12:00</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
  <Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```

Listing 2: Instance of XACML document created by the resource broker

```
<Policy PolicyId="GeneratedDictionary1Policy" RuleCombiningAlgId="
    urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:ordered-permit-
    overrides">
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:boolean-
            equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#boolean">
              true</AttributeValue>
          <SubjectAttributeDesignator SubjectCategory="
              urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              AttributeId="use-license-show-client" DataType="http://www.w3.org
              /2001/XMLSchema#boolean"/>
        </SubjectMatch>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
            equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              user1</AttributeValue>
          <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0
              :subject:subject-id" DataType="http://www.w3.org/2001/XMLSchema#
              string"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
            equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              service1</AttributeValue>
          <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1
              .0:resource:resource-id" DataType="http://www.w3.org/2001/
              XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
            equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              invoke</AttributeValue>
          <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0
              :action:action-id" DataType="http://www.w3.org/2001/XMLSchema#
              string"/>
        </ActionMatch>
      </Action>
    </Actions>
  </Target>
  <Rule RuleId="InvokeRule" Effect="Permit">
```

```
<Target>
  <Subjects>
    <Subject>
      <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-
          greater-than">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer"
            >10</AttributeValue>
        <SubjectAttributeDesignator AttributeId="invoke-num-on-interval"
            DataType="http://www.w3.org/2001/XMLSchema#integer"/>
      </SubjectMatch>
    </Subject>
  </Subjects>
  <Resources>
    <AnyResource/>
  </Resources>
  <Actions>
    <Action>
      <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-
          greater-than-or-equal">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer"
            >5120</AttributeValue>
        <ActionAttributeDesignator AttributeId="response-byte-size"
            DataType="http://www.w3.org/2001/XMLSchema#integer"/>
      </ActionMatch>
    </Action>
  </Actions>
</Target>
<Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-less-
      than">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-
        and-only">
      <SubjectAttributeDesignator SubjectCategory="
          urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
          AttributeId="invoke-num-per-day-Dictionary1" DataType="http://www
          .w3.org/2001/XMLSchema#integer"/>
    </Apply>
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">100
        </AttributeValue>
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-greater
      -than">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-one-
        and-only">
      <ActionAttributeDesignator AttributeId="invoke-dateTime" DataType="
          http://www.w3.org/2001/XMLSchema#dateTime"/>
    </Apply>
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">
        2006-11-20T19:59:20.531000000+12:00</AttributeValue>
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-less-
      than">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-one-
```

```
              and-only">
            <ActionAttributeDesignator AttributeId="invoke-dateTime" DataType="
                http://www.w3.org/2001/XMLSchema#dateTime"/>
          </Apply>
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">
              2008-11-20T19:59:20.531000000+12:00</AttributeValue>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-less-
            than">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">
              2007-11-22T20:59:02.046000000+09:00</AttributeValue>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-one-
              and-only">
            <EnvironmentAttributeDesignator AttributeId="
                urn:oasis:names:tc:xacml:1.0:environment:current-dateTime"
                DataType="http://www.w3.org/2001/XMLSchema#dateTime"/>
          </Apply>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-less-
            than">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-one-
              and-only">
            <EnvironmentAttributeDesignator AttributeId="
                urn:oasis:names:tc:xacml:1.0:environment:current-dateTime"
                DataType="http://www.w3.org/2001/XMLSchema#dateTime"/>
          </Apply>
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">
              2007-11-29T21:59:02.046000000+09:00</AttributeValue>
        </Apply>
      </Condition>
    </Rule>
    <Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```