Adriano César Machado Pereira

Caracterização da Reatividade de Usuários em Sistemas Distribuídos

(*Characterizing User Reactivity in Distributed Systems*)

Belo Horizonte
Julho de 2007

Adriano César Machado Pereira
Orientador: Wagner Meira Jr.

Caracterização da Reatividade de Usuários em Sistemas Distribuídos

(*Characterizing User Reactivity in Distributed Systems*)

Tese apresentada ao Curso de Pós-graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Belo Horizonte
Julho de 2007

À minha querida esposa Maira Pereira,

Aos meus futuros filhos,

Aos meus queridos avós, pais, irmãs e familiares.

Resumo

Nas últimas décadas testemunhamos um crescimento contínuo do uso da tecnologia da informação em todas as áreas do conhecimento humano. Em paralelo, também se observa um aumento do grau de interatividade provida pelos sistemas computacionais como conseqüência tanto de uma melhor conectividade quanto melhores interfaces, incluindo recursos multimídia, como vídeo, áudio, ou realidade virtual. Assim, pode-se dizer que sistemas computacionais interativos, isto é, sistemas com os quais usuários interagem continuamente, requisitando e provendo informações, têm se tornado populares em nossa sociedade. Desde transações bancárias até telefones celulares, estamos quase sempre interagindo com sistemas e até mesmo com outros usuários através desses sistemas. Deve-se notar que os sistemas atuais possibilitam não somente as interações homem-computador, mas também uma variada gama de interações usuário-usuário, que são mais ricas e complexas.

Uma parte significativa das interações intermediadas por sistemas é síncrona, ou seja, o usuário submete informação ou uma requisição ao sistema e espera pela resposta, posteriormente solicita outra parte de informação e esse processo interativo continua. A Web, em particular o protocolo HTTP, é um exemplo de interação síncrona. De fato, padrões de interação similares existem desde o primeiro sistema computacional nos anos sessenta.

Essas interações costumam ser complexas e intrigantes. É bastante difícil determinar exatamente os fatores que levam o usuário a comportar-se conforme observado. Primeiramente porque as informações que temos sobre os usuários são esparsas e variáveis, em termos de condições instantâneas que abarcam o comportamento observado e do seu histórico de ações. É notável observar que as interações não são isoladas, mas que as interações sucessivas constituem um ciclo de realimentação, onde o comportamento do usuário afeta o sistema e vice-versa.

Existem fortes evidências de que boa parte do comportamento de um usuário seja reativo, ou seja, o usuário reage a condições instantâneas em tempo real. A hipótese principal desta dissertação é que o mundo real é reativo. Esse conceito significa que o comportamento do usuário varia de acordo com alguns fatores relacionados ao servidor e

a aplicação que está sendo provida.

A principal hipótese deste trabalho de tese é que sistemas computacionais interativos possuem um forte componente reativo. A tese desta dissertação é definir, caracterizar e modelar a reatividade em sistemas interativos, em particular em sistemas Internet. Entender a reatividade do usuário tem aplicabilidade em muitos cenários, desde avaliação de sistemas até melhorias dos mesmos, como através de personalização. O maior desafio deste trabalho é identificar qual o limite da caracterização da reatividade.

A principal motivação deste trabalho é entender como os usuários interagem com o sistema, possibilitando prover serviços com melhor desempenho e entender melhor as aplicações que envolvem negociação.

As contribuições deste trabalho são a formalização do conceito de reatividade, a especificação de um modelo de reatividade multi-nível, a elaboração de metodologias de caracterização para modelar a reatividade do usuário e a validação do modelo e metodologia através de sua aplicação a cenários relevantes, como serviços Internet e *e-business*. Essas contribuições estão publicadas em alguns artigos internacionais.

Vislumbramos como benefícios deste trabalho melhorias na caracterização de cargas de trabalho, novas técnicas de geração de carga de trabalho que considerem a reatividade, novas técnicas de controle de qualidade de serviço (QoS) e melhorias na modelagem e caracterização de serviços de *e-business*.

Abstract

We have been witnessing a continuous growth in the use of information technology in all areas of human knowledge in the last decades. In parallel, we may also observe an increase in the degree of interactiveness provided by computer-based systems as a consequence of either better connectivity or improved interfaces, including multimedia resources such as video, audio, or virtual reality. We may say that interactive computer systems, that is, systems with which users interact continuously, getting and providing information, become a common place in our society. From bank transactions to cell phones, we are almost always interacting with systems and even with other users through these systems. We should note that current systems allow not only human-computer interactions, but also a wide variety of user-user interactions, which are obviously richer and more complex.

A significant portion of system-mediated interactions is synchronous, that is, users submit information or a request to the system and wait for a response, then submit another piece of information and the process continues. The Web, in particular the HTTP protocol, is an example of synchronous interaction. In fact, similar interaction patterns exist since the first computer systems in the 1960's.

These interactions are usually complex and intriguing. It is quite hard to determine exactly the factors that led a user to behave as observed. First because the information we have about users is sparse and variable, in terms of both the instantaneous conditions surrounding the observed behavior and his or her background. It is remarkable to note that the interactions are not isolated, but successive interactions become a loop feedback mechanism, where the user behavior affects the system behavior and vice-versa. Further, we should note that the number of users in Internet services is usually large, and they interact in an indirect fashion through the server, where their simultaneous demands may generate contention.

There is strong evidence that much of the user behavior is reactive, that is, the user reacts to the instantaneous conditions at the action time. The main hypothesis of this dissertation is that the real world is reactive. This concept means that the user behavior

varies according to some factors related to the server and the application being provided.

The main hypothesis of this dissertation is that interactive computer-based systems has a strong reactive component. The thesis of this dissertation is to research, define, characterize, and model reactivity in interactive systems, in particular Internet-based systems. Understanding users' reactivity has applicability to several scenarios, from system evaluation and simulation to system improvement, such as personalization. The main challenge is to identify what is the limit of reactivity characterization.

The main motivation of this work is to understand how users interact with the system, allowing to provide services with better performance and to understand better the business applications.

The contributions of this work are the formalization of reactivity concept, the specification of a multi-level reactivity model, elaboration of characterization methodologies for modeling user reactivity, and the validation of the model and methodology applying them to relevant scenarios, such as Web services and e-business. These contributions are presented in some research papers published in international conferences and journals.

We foresee as benefits from this work improvements on workload characterization, new workload generation techniques that consider reactivity, novel quality of service(QoS) control techniques, and improvements on modeling and characterizing e-business services.

Resumo Estendido do Trabalho

Nas últimas décadas tem se testemunhado um crescimento contínuo do uso da tecnologia da informação em todas as áreas do conhecimento humano. Em paralelo, também se observa um aumento do grau de interatividade provida pelos sistemas computacionais como consequência tanto de uma melhor conectividade quanto melhores interfaces, incluindo recursos multimídia, como vídeo, áudio, ou realidade virtual. Assim, pode-se dizer que sistemas computacionais interativos, isto é, sistemas com os quais usuários interagem continuamente, requisitando e provendo informações, têm se tornado popular em nossa sociedade. Desde transações bancárias até telefones celulares, estamos quase sempre interagindo com sistemas e até mesmo com outros usuários através desses sistemas. Deve-se notar que os sistemas atuais possibilitam não somente as interações humano-computador, mas também uma variedade de interações usuário-usuário através de sistemas computacionais, que são mais ricas e complexas.

Uma parte significativa das interações intermediadas por sistemas é síncrona, ou seja, o usuário submete informação ou uma requisição ao sistema e espera pela resposta, posteriormente solicita outra parte de informação e esse processo interativo continua. A Web, em particular o protocolo HTTP, é um exemplo de interação síncrona. De fato, padrões de interação similares existem desde o primeiro sistema computacional nos anos sessenta.

O entendimento destas interações síncronas é necessário para melhorar tanto os sistemas quanto a forma de interação dos usuários. Entretanto, essas interações costumam ser complexas e intrigantes. É tarefa árdua determinar exatamente quais fatores afetam o

comportamento dos usuários. Primeiro porque a informação que temos dos usuários é esparsa e variável, em termos de informações do instante de suas atuações bem como informações passadas. Segundo porque o comportamento do usuário é algo muito complexo.

É importante destacar que as interações não podem ser consideradas de forma isolada, mas podemos considerá-las como uma sequência, onde interações sucessivas definem um mecanismo cíclico (que denominamos "loop feedback"), onde o comportamento do usuário afeta o sistema e vice-versa. Sistemas Web são um típico cenário onde se observa esse ciclo. Por exemplo, quando um usuário requisita dados de um servidor Web sobrecarregado, ele deve esperar mais do que o usual, devido à contenção observada nesse servidor. Neste caso, ele pode reagir de algumas formas distintas, como: esperar pacientemente pela resposta, realizar novamente a requisição causando carga ainda maior no servidor, ou simplesmente desistir de sua solicitação. Note que esse exemplo serve de base para observar como o comportamento do sistema afeta o usuário e este afeta o sistema.

Podemos notar que o número de usuários de serviços Internet é tipicamente grande, e que estes interagem de modo indireto com o servidor, sendo que suas demandas sucessivas podem gerar contenção. Outro exemplo é um serviço de leilões eletrônicos, onde usuários interagem com o servidor através de seus lances, optando por fornecer um lance de maior valor que o atual vencedor da negociação, esperar para ver a evolução da negociação, ou até mesmo desistir de participar do leilão. Novamente, vemos que o comportamento desses usuários possui correlação uns com os outros, criando um ciclo.

Uma observação importante que podemos fazer ao observar essas interações é que sua análise não pode se realizar somente de uma perspectiva, uma vez que todos os participantes podem realizar ações que afetam os demais, assim temos que avaliar, da melhor forma, quais razões levam cada um deles a atuar de determinada forma.

Uma ação pode ser definida como o que o agente, uma entidade, um participante, pode fazer. Ações se relacionam a atividade, que está presente em tudo na natureza e é base da vida. A relação entre quaisquer entidades representa uma interação. A maior questão que motiva este trabalho de tese é como entender as interações entre usuário e sistema. E o maior desafio desta é como entender e modelar a interação entre usuário e sistema (ou serviço que está sendo provido) em um cenário de aplicações de ciência da computação.

Considerando a tradicional arquitetura cliente-servidor que foi herdada para os serviços Web, distinguimos três perspectivas com as quais podemos realizar nosso trabalho: o usuário, o sistema, e a interface usuário-sistema. Olhando sob a perspectiva do usuário, distinguimos três componentes que estão relacionados à forma como os usuários se comportam ao interagir com um sistema computacional:

Habilidade: O componente habilidade representa a capacidade do usuário para interagir com o sistema e usar de forma adequada seus melhores recursos. Existe uma linha de pesquisa ativa da área de Interação Humano-Computador (IHC) relacionada a usabilidade e efetividade, bem como o desafio de como melhorar as interfaces dos sistemas.

Interesse: O interesse contempla todo o conhecimento do usuário acerca do que pretende realizar com o serviço provido. Isso inclui a parte de informações e bens pode pretender adquirir com o serviço. Em geral, não é possível qualificar e quantificar completamente o interesse de um usuário apenas observando-o, uma vez que é algo muito subjetivo e difícil de ser diretamente observado.

Reatividade: A reatividade é o componente que contempla as condições e variáveis do ambiente que podem afetar o comportamento do usuário e também como podem fazê-lo. Definimos reatividade como o conjunto de ações que o usuário realiza como consequência de um determinado cenário vivenciado. Ou seja, a reatividade está ligada às variáveis que o usuário percebe no ambiente e sua ação frente a isso. Dois exemplos de variáveis de um cenário típico de serviços Web são a latência percebida pelo usuário ao solicitar algo e receber sua resposta, e a sequência de lances que caracteriza um leilão.

Em todas as interações usuário-sistema (ou usuário-serviço), esses componentes afetam diretamente a natureza da sequência de interações. Embora habilidade (e suas dimensões) tenha sido estudada com certa profundidade, os conceitos de reatividade e interesse não estão claros nos dias atuais. Isso porque existe dificuldade em diferenciá-los e também porque estes são diretamente dependentes da aplicação onde a interação ocorre. Esta diversidade de aplicação aumenta o desafio para caracterizar e modelar a reatividade.

Para modelar a reatividade, precisamos modelar tanto as potenciais causas da reação quanto suas consequências no ambiente. Iniciamos por estudar a aplicabilidade da *Lei de Causa e Efeito.*

O filósofo grego Socrates, em 425 A.C., indicou simplesmente que nós vivemos em um mundo de leis governadas por um sistema, onde nós compreendemos os princípios por trás disso ou não. [40]. Ele usou suas grandes e respeitadas habilidades para forçar povos a pensar sobre as conseqüências lógicas de seus pensamentos e comportamentos. Em seu tempo, esta idéia ficou conhecida como "Lei Socrática de Causalidade". Hoje, esta é denominada "Lei de Causa e Efeito".

A *Lei de Causa e Efeito* expressa que tudo ocorre por uma determinada razão. Para todo efeito observado em sua vida, existe uma causa, ou uma série de específicas, mensuráveis, definidas, identificáveis causas. Esta lei expressa que, se existe algo que você quer em sua vida, algum efeito que você deseje que ocorra, então você pode encontrar alguém que já alcançou esse efeito ou resultado, e realizando as mesmas coisas que esta pessoa tenha feito você pode desfrutar os mesmos resultados.

A Lei de Causa e Efeito determina que absolutamente tudo ocorre por uma razão. Todas as ações têm consequências e produzem resultados específicos. As escolhas que nós fazemos são causas, sejam estas conscientes ou não, e produzirão resultados ou efeitos correspondentes. A lei trabalha da mesma forma para todos em todo momento.

Uma versão da *Lei de Causa e Efeito* é a lei de ação e reação [40]. Proposta inicialmente e explicada pelo Senhor Isaac Newton, esta lei estabelece que para cada ação existe uma reação igual e oposta[70], isto é, as ações têm conseqüências, e o relacionamento entre ações e conseqüências é importante. Inicialmente, você pode decidir-se pela ação, pode controlá-la. Entretanto, uma vez que você tenha determinado uma ação particular, as conseqüências já estarão frequentemente fora de suas mãos. Isso explica porque as pessoas bem sucedidas tendem a ser muito pensativas sobre o que dizem e fazem, enquanto as mal sucedidas tendem a ser inconsequentes, até mesmo descuidadas acerca de suas recomendações ou comportamentos [70].

González Pecotche também tratou sobre esta lei, como pode ser visto no parágrafo a seguir de seu livro "O Mecanismo da Vida Consciente": "Dentro da grande estrutura cósmica, e como uma expressão cabal e absoluta do Pensamento Supremo, aparecem configuradas em suas respectivas jurisdições as Leis Universais, regulando e regendo a vida cósmica tanto quanto a humana. Entre as mais direta e estreitamente vinculadas ao homem, citaremos as de Evolução, Causa e Efeito, Movimento, Câmbios, Herança, Tempo, Correspondência, Caridade, Lógica e Adaptação" [91].

A *Lei de Causa e Efeito*, ação e reação, é atemporal, baseia-se em princípios universais que existiram desde o começo da humanidade.

Há uma forte evidência de que muito da ação do usuário (a maneira com que um usuário se comporta) é reativa, isto é, o usuário reage às circunstâncias instantâneas no momento de tomar uma determinada ação. Conseqüentemente, tornam-se necessárias ferramentas de análise e modelagem para compreender eficazmente a reatividade, em consequência da complexidade do comportamento e de sua natureza multiperspectiva. A seguir, apresentamos como pretendemos trabalhar essas questões nesta tese de doutorado.

Hipótese da Tese (*Statement of the Thesis*)

A hipótese principal desta dissertação é que o uso de sistemas computacionais interativos possui um forte componente reativo. A tese desta dissertação é pesquisar, definir, caracterizar e modelar a reatividade em sistemas interativos, em particular sistemas Web. O entendimento da reatividade tem aplicabilidade em diversos cenários, desde avaliação de desempenho de sistemas e simulação, até caracterização e modelagem de modelos de *e-business*.

Contribuições Esperadas

As principais contribuições deste trabalhado são:

- Formalização do conceito de reatividade: isto é apresentado no Capítulo 3.
- Especificação de um modelo de reatividade multinível: isto é previamente explicado em alto nível no Capítulo 3, e depois nos contextos específicos das aplicações de um serviço Web e *e-business* nos Capítulos 4 e 5, respectivamente.
- Elaboração de metodologias de caracterização para modelagem da reatividade: isto é descrito nos Capítulos 4 e 5, para cada estudo de caso realizado.
- Validação dos modelos de reatividade e respectivas metodologias, aplicando-os a cenários práticos reais de um site Web e um leilão eletrônico: isto é apresentado e detalhados nos Capítulos 4 e 5.

Além disso, essas contribuições estão também presentes em alguns artigos publicados em conferências internacionais, que estão citados e sumarizados na Seção 6.2.

Como consequência deste trabalho, obtemos os seguintes benefícios:

- Melhorias para a caracterização de cargas de trabalho.
- Novas técnicas de geração de carga de trabalho que consideram a reatividade.

- Novas propostas para técnicas de controle de QoS.

- Melhorias na caracterização e modelagem de serviços de *e-business*, como leilão eletrônico.

Organização do texto

Este texto de dissertação de doutorado está organizado em cinco capítulos, além da introdução. O Capítulo 2 apresenta uma visão geral dos trabalhos correlatos. O Capítulo 3 explica o conceito de reatividade, apresentando um detalhamento de como este pode ser modelado. O Capítulo 4 apresenta um estudo de caso completo para um *Web Site*, descrevendo cada passo e as contribuições alcançadas. O Capítulo 5 descreve outro estudo de caso real, uma aplicação de *e-business*. Por fim, apresentamos nossas contribuições e principais conclusões no Capítulo 6. Além disso, há apêndices que apresentam outros resultados interessantes, nossa metodologia de trabalho adotada e um outro estudo de caso com outros dados reais de leilão eletrônico.

Agradecimentos

Esta é uma parte especial de minha tese, pois é aqui onde traduzo em singelas palavras o meu grande sentimento de gratidão a todos que colaboraram nesta empreitada acadêmica. De certa forma, o que registro aqui é uma continuidade do que já manifestei na ocasião de minha formatura na graduação (2000) e da culminação de meu mestrado (2002).

A gratidão é um dos mais nobres sentimentos do ser humano. Penso que ela engrandece a quem a testemunha e estimula e faz feliz a quem a recebe. Porém a gratidão é um sentimento que vai ficando mais raro à medida que o ser humano, invertendo a hierarquia de valores, tem se envaidecido, olvidando a Fonte Suprema que o criou, as suas raízes espirituais e os seus deveres para com o seu semelhante.

Gratidão remete a recordação, e me recordo de um valioso conhecimento aprendido que expressa que "recordar o bem recebido é fazer-se merecedor de tudo quanto amanhã possa nos ser brindado".

A conclusão desta etapa, deste trabalho, é um marco importante para mim. Retrata um esforço não só meu, mas de muitos, com quem tenho a alegria e o dever moral de compartilhar esta conquista!

Inicialmente, gostaria de expressar minha gratidão a Deus, o Criador de tudo quanto existe, pela maior de todas as oportunidades, que é a de viver, e por presidir sempre todos os momentos de minha vida.

Outro conhecimento aprendido diz respeito à gratidão a Deus e à verdadeira oração que penso ser eficaz, que se resume na conduta, não em palavras calcadas em crenças absurdas.

Dessa forma, vale dizer também que "é bom recordar o que mais de uma vez afirmamos: os seres invocam a Deus em seus momentos de desventura, pretendendo um amparo imediato, sem perceberem, por outro lado, que poucos o fazem como homenagem de gratidão por seus momentos de felicidade e, menos ainda, para mostrarem-lhe o fruto de seus esforços por vincular-se à sua maravilhosa Vontade, plasmada na Criação. É preciso, pois, recordá-lo também nos momentos de alegria; a recordação não só perde assim o caráter especulativo, como também brota da gratidão pela felicidade que se vive. Então sim, o espírito individual pode elevar a alma e vinculá-la a vibrações superiores."

Tenho procurado aplicar com mais frequência o seguinte conhecimento em minha vida: "Que sempre seja Deus quem presida suas horas de alegria, oferecendo-Lhe, do mais íntimo do coração, sua gratidão por tudo o que Lhe deve e possui em felicidade, em conhecimento, em conforto, em triunfos."

Sou eternamente grato as Américas, avós amáveis e companheiras, ao avô Zizico (sempre presente, apesar da distância física), brincalhão e paciente, avô Wilson, sinônimo de luta e responsabilidade.

A meus queridos pais, Elza e Lourival, pelo incondicional e verdadeiro amor, que se traduz em afeto em nossa convivência. Agradeço o apoio, o carinho e os ensinamentos de inestimável valor. A Lê e Déia, por sempre me oferecer tanto apoio, carinho e amor. Ao Eduardo, tios e tias, primos e primas, que completam esta minha maravilhosa família. Merecem minha recordação também a família que herdei de minha querida esposa: Marcinha, Maísa e Urutan, Marly e Marcos, sem falar nos demais familiares formiguenses que a complementam. Ao ainda pequenino Lucas, que tanta felicidade tem proporcionado à nossa família.

A Maira, minha querida Lindona, com quem compartilho os mais valiosos ideais, por seu amor especial e ascendente, afeto grandioso, respeito, companherismo, tolerância e paciência. Agradeço também por saber respeitar minhas lutas diárias, minhas imperfeições, minha individualidade, meu trabalho, por me auxiliar em importantes decisões e compreender minhas ausências e incontáveis horas de estudo. Obrigado por estar tornando minha vida ainda mais feliz e significativa gradativamente!

Ao professor Wagner Meira Jr., orientador deste trabalho e de tudo quanto realizei nos últimos 10 anos de pesquisa acadêmica e estudos na UFMG, agradeço pelos valiosos ensinamentos, pelas oportunidades, desafios, orientações e incentivos. Você costuma dizer que "o que vale no curso de mestrado ou doutorado é o sofrimento". No início, não entendia isso, hoje compreendo que na verdade esse sofrimento a que você se refere é a luta que, com muito esforço e perseverança, culmina em capacitação e conhecimento. Sou muito grato por sua colaboração. Você foi um grande apoio para que eu conseguisse chegar até aqui, sou grato por todas as oportunidades, por tudo que aprendi! Saiba que pode contar comigo e espero mantermos o contato sempre!

Agradeço também ao professor Paulo Góes, co-orientador desta tese, por todo o apoio recebido, pelo auxílio nas atividades da tese, pelas semanas vividas nos Estados Unidos, quando me recebeu com tanta generosidade, sou muito grato por tudo! Espero podermos manter o contato e nos encontrar sempre que for possível.

Aos demais membros da banca, os professores Virgílio Almeida, Mohammed Zaki, e Raquel Prates, ilustres pesquisadores de reconhecimento internacional, agradeço pelas importantes orientações e conselhos.

Agradeço aos companheiros de pesquisa Gustavo Gorgulho, Walter Santos e Leonardo Araújo, pelas inúmeras contribuições a este trabalho, onde juntos formamos uma pequena grande equipe e alcançamos méritos com muito esforço, dedicação e perseverança. Para mim foi grande a importância da colaboração de vocês, sei que não teria chegado até aqui sem esse grande apoio.

Da mesma forma, tenho que expressar como foi importante a equipe de trabalho formada com o Fernando Mourão e Leonardo Rocha, pessoas que estiveram presentes praticamente todo o tempo na minha segunda parte do trabalho desta tese. Aprendemos juntos tantas coisas, vivemos tantos desafios e dificuldades, e também colhemos bons frutos, e sei que outros frutos ainda virão.

Aos amigos, que felizmente são muitos, tantos que não caberiam nesta seção do trabalho. Em especial, aos grandes amigos de infância e do Colégio Logosófico, entre os quais destaco Alexandre Silveira, Ana Carolina Chaves, Ana Flávia Nagem, Anita Lima, Cynthia Porto, Daniel Rocha, Danilo Nagem, Guilherme Cavalieri, e Gustavo França. Aos queridos amigos da Fundação Logosófica, que são muitos, mas que especialmente quero citar Maria Lúcia da Silveira, Cláudio Melo, Paula Nadabe, Luciana Caixeta, Elmo Ballesteros, Michelle Ballesteros, Cassiana Ranieri, Carla Nagem, Rodrigo Chagas, Renata Salles, Maurício Chagas, Silvia Salles, Liara Salles, Paulo Salles, Lilian Nagem, Hudson Abreu, Pauline Moysés, Virgínia Guedes, Maria Natália, Carlos Aude, Fernando França, Aparecida França, Márcio Patrus, Ilacir França, José Geraldo, Marcos Breder, Renato Ribeiro, e Raphael Gideoni. Cabe ressaltar que tantas outras amizades que possuo na Fundação são de inestimável valor para minha vida.

Aos amigos do grupo *OGDGA*, onde destaco Luciana Avelar e Rodrigo, o Foca. Aos integrantes das minhas famílias colaterais, das quais destaco a família Neves Cavalieri, França, Silva Gontijo, e Alves Pereira. Aos companheiros de *Smart Price*, incluindo os que lá estão e os que por lá já passaram. A todos os colegas e amigos da *2000 e-mail*. A todos os amigos da UFMG. Aos amigos José Henrique e Leonardo Santos. Prestar um agradecimento nominal não me parece a forma mais generosa de manifestar meu agradecimento, pois o verdadeiro reconhecimento é visível na amizade e no verdadeiro afeto que caracteriza a nossa convivência.

Aos companheiros da UNA, pela confiança e apoio recebido, em especial ao Anderson Ceolin e Maria do Carmo Caixeta. Aos colegas e amigos professores, em especial, Ildevan Medeiros, Ronaldo Faria, Paulo Gondim, Róbson Abreu, Bruno Teixeira, Maria Aparecida Araujo, Alexandra Maia, e Liliene Pereira.

A todos do Laboratório *e-SPEED*, do Departamento de Ciência da Computação da UFMG, por tornarem alegre, rico e divertido nosso convívio diário. Aos companheiros de sala 3005 (Tassni Cançado, Bernardo Annoni, Max Machado, Daniel Guidoni, Renan Cunha, Rainer Couto, e outros), pelo bom convívio e conversas agradáveis desfrutadas.

Não poderia deixar de agradecer ao Departamento de Ciência da Computação (DCC) da Universidade Federal de Minas Gerais (UFMG), onde estou a exatos 11 anos, pelo ambiente propício ao desenvolvimento do trabalho, muitos professores capacitados e competentes funcionários de que dispõe. Agradeço especialmente aos funcionários do DCC, que com grande eficiência deram suporte ao meu curso de pós-graduação. Também agradeço aos colegas de pós-graduação - mestrado e doutorado - pelo apoio e exemplos úteis com os quais pude aprender algo.

Finalmente, dedico este trabalho a todos que não estão diretamente mencionados, mas que de alguma forma colaboraram para a conclusão do mesmo. Meu sincero obrigado! Espero que este trabalho possa ser fonte de estudo e pesquisa para muitos seres e base para desenvolvimento de novas pesquisas.

Agora que estou a finalizar estes dizeres, noto que palavras não são suficientes para expressar meu afeto e gratidão, tamanha é a intensidade do que tudo isso manifestado representa para mim!

Para concluir, a exemplo do que fiz em outras ocasiões especiais, recorro então a um pensamento de um grande Mestre, o maior sábio que já conheci até os dias atuais, a quem tanto sou grato por seus ensinamentos de verdadeira transcendência, pois são estes os que tem verdadeiramente ampliado a minha vida e a tornado mais feliz: "A vida não deve terminar como terminam as horas do dia, agonizando em um entardecer. A vida tem que ampliar seus horizontes; fazer longas as horas da existência para que o espírito, incorporado na matéria, experimente a grandiosidade de sua criação. Para isso tem que renovar-se no passado e no futuro. No passado, reproduzindo constantemente na tela mental todas as passagens vividas com maior intensidade; no futuro, pensando no que ainda resta por fazer, naquilo que se pensou fazer, e, sobretudo, no que se quer ser nesse futuro. E quanto mais gratidão o homem experimente pelo passado, quanto mais gratidão guarde pelas horas felizes vividas nele, assim como pelas de luta ou de dor, que sempre são instrutivas, tanto mais abrirá sua vida a novas e maiores perspectivas de realização."

“Eu guardo, para todos aqueles que de uma ou outra forma contribuíram para fazer-me mais grata a vida, uma eterna gratidão, e estampo nessa gratidão a lealdade com que conservo essa recordação, a qual jamais pôde empalidecer ali onde se encerra tudo quanto constitui a história de minha vida.” (González Pecotche, Humanista e Criador da Ciência Logosofia - RAUMSOL)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We have been witnessing a continuous growth on the use of information technology in all areas of human knowledge in the last decades. In parallel, we can also observe an increase in the degree of interactiveness provided by the computer-based systems as a consequence of either better connectivity or improved interfaces, including multimedia resources such as video, audio, or virtual reality. We may say that interactive computer systems, that is, systems with which users interact continuously, getting and providing information, have become a common place in our society. From bank transactions to cell phones, we are almost always interacting with systems and even with other users through these systems. We should note that current systems allow not only human-computer interactions, but also a wide variety of user-user interactions, which are obviously richer and more complex.

A significant portion of system-mediated interactions is synchronous, that is, users submit information or a request to the system and wait for a response, then submit another piece of information and the process continues. The Web, in particular the HTTP protocol, is an example of synchronous interaction. In fact, similar interaction patterns exist since the first computer systems in the 1960's.

Understanding these synchronous interactions is necessary for improving both the systems and the user experience. However, these interactions are usually complex and intriguing. It is quite hard to determine exactly the factors that led a user to behave as observed. First because the information we have about users is sparse and variable, in terms of both the instantaneous conditions surrounding the observed behavior and his or her background. Second because user behavior is very complex.

It is remarkable to note that the interactions cannot be considered in isolation, and we should analyze them as a sequence, where successive interactions become a “loop feedback” mechanism, where the user behavior affects the system behavior and vice-versa. Internet systems are a typical scenario where such cyclic sequences of interactions arise. For instance, when users request data from overloaded Web servers, they may wait longer than the usual response time, as a consequence of contention on the server. In this case, they react in one of three ways: they may wait patiently for the response, they may retry the request causing even more load on the server, or they may simply give up on obtaining the information desired from the server. Notice that both the user and server behaviors are correlated, and affect each other since the user reaction will relieve the server or overload it even more, affecting user actions that follow.

Further, we should note that the number of users in Internet services is usually large, and they often interact in an indirect fashion through the server, where their simultaneous demands may generate contention. An example is an Internet-based auction service, where users interact through their bids, that is, while competing for an item, they may place higher bids towards winning the auction, wait to see how the auction evolves, or give up. Again, the behavior of the bidders is correlated, and affect each other. These two examples, although significantly different at first glance, are quite similar regarding their interaction-related dynamics, since in both cases we can identify the “loop feedback” that characterizes the user-system or user-user interactions.

One interesting observation when we try to understand these interactions is that the analysis cannot be performed from just one perspective, since all participants in the interaction may affect each other and we should understand, as much as possible, the reasons that lead them to behave as expected. In order to understand the interaction is important to define what is an action.

An action may be defined as what an agent, an entity, a participant, may do. Actions relate to activity, which is present in everything in nature and is the basis of life. The relation between any entities represents an interaction. The major question that motivates this dissertation is how to understand the user-system interactions. The major challenge in which this dissertation is inserted is to understand and model the interaction between user and system (or service being provided) in a Computer Science application scenario.

Considering the traditional client-server architecture that is inherent to common Web services, we distinguish three perspectives from which we may perform the understanding task: user, system, and user-system interaction. We are interested in the user-system interaction in this work, however is important to emphasize that we are going to characterize both user and system in order to model this interaction. As the user is the entity that causes a change in this interaction, we have to begin studying it. From the user perspective, we distinguish three components that may be used to explain the user behavior when interacting with a system:

Ability: The ability component represents the user skills for interacting with the system and use properly their resources. There is an active research in the Computer-Human Interaction area related to interface usability and effectiveness, as well as on how to improve the system interfaces.

Interest: The interest component comprises all user background and willingness towards getting services, which include information and goods. It is not usually possible to qualify and quantify completely the user's interest by just observing him or her, since there may be interests that manifest just once or are never expressed.

Reactivity: The reactivity component concerns about the service environmental conditions and variables and how they affect user behavior. We define reactivity as the set of actions a user takes as a consequence of the scenario, that is, any distribution system environment, the user can perceive. Two examples of environment variables from this scenario are the user-perceived latency and the sequence of bids of an auction.

In all user-system (or user-service) interactions these components affect directly the nature of the sequence of interactions. Although ability (and its dimensions) has been studied in some depth, the concepts of reactivity and interest are not really clear, first because of the difficulty in differentiating them, second because they are heavily dependent on the application where the interaction takes place. This application diversity raises a challenge towards having a unified view of reactivity in interactive systems, being a challenge to find common grounds for understanding reactivity.

In order to model the reactivity, we need to model both the causes of the reaction and its consequences. We start by analyzing the applicability of the Law of Cause and Effect to this modeling demand.

The Greek philosopher Socrates, in 425 BCE, stated simply that we live in a world governed by a system of laws whether we understand the principles behind it or not. [40]. He used his considerable, rhetorical skills to force people to think through the logical consequences of their thoughts and behaviors. In his time this idea was known as the "Socratic Law of Causality". Today we call it the "Law of Cause and Effect".

The Law of Cause and Effect says that everything happens for a reason. For every effect in your life, there is a cause, or series of specific, measurable, definable, identifiable causes. This law says that, if there is anything you want in life, an effect that you desire, you can find someone else that has achieved the same result or effect, and that by doing the same things that they have done over and over you can eventually enjoy the same results and rewards.

The Law of Cause & Effect states that absolutely everything happens for a reason. All actions have consequences and produce specific results, as do all inactions. The choices we make are causes, whether they are conscious or unconscious, and will produce corresponding outcomes or effects. The Law works the same for everyone at all times.

An instance of the law of cause and effect is the law of action and reaction [40]. First proposed and explained by Sir Isaac Newton this law states that for every action there is an equal and opposite reaction [70], that is, actions have consequences, and the relationship between actions and consequences is important. At the beginning, you can decide upon the action. You can control it. But once you have launched a particular action the consequences are often out of your hands. Once you have done or said a particular thing the consequences take on a power and a force of their own.

There are other people who has also talked about this law. González Pecotche, in this paragraph of his book "The Mechanism of Conscious Life", says that "Within the great cosmic structure, and as a precise and absolute expression of the Supreme Thought, there are, operating in their respective jurisdiction, the Universal Laws which rule and regulate both the cosmic and the human life. Among the laws that have a more direct and close relation to man, we shall mention the Laws of Evolution, Cause and Effect, Movement, Change, Inheritance, Time, Reciprocity, Charity, Logic, and Adaptation." [91].

The law of cause and effect, action and reaction, sowing and reaping, are timeless truths, universal principles that have existed since the beginning of the mankind.

There is strong evidence that much of the user action (the way a user behaves) is reactive, that is, the user reacts to the instantaneous conditions at the time he or she takes an action. Therefore, it seems that we do need better analysis and modeling tools to effectively understand reactivity, as a consequence of both the behavior complexity and its multi-perspective nature. Next we start discussing how we intend to tackle these issues.

1.1 Statement of the Thesis

The main hypothesis of this dissertation is that the use of interactive computer-based systems has a strong reactive component, as discussed in the last section. The thesis of this dissertation is to research, define, characterize and model reactivity in interactive systems, in particular Internet-based systems. Understanding users' reactivity has applicability to several scenarios, from system evaluation and simulation to system improvement, such as personalization.

1.2 Contributions

We believe that the contributions of this work are:

- Formalization of the reactivity concept: it is presented in Chapter 3.
- Specification of a multi-level reactivity model: first it is previously explained in high-level in Chapter 3, and then in specific contexts of a Web site and e-business service in Chapters 4 and 5, respectively.
- Elaboration of characterization methodologies for modeling user reactivity: it is described in Chapters 4 and 5, according to each case study.
- Validation of the model and the methodology applying them to relevant scenarios such as Web services and e-business: it is presented and detailed in Chapters 4 and 5, for each of these case studies.

Moreover, these contributions are also presented in some research papers published in international conferences. These publications are cited and summarized in Section 6.2.

Further, this work achieves the following results:

- Improvements on workload characterization: we provide new characterization methodologies that consider reactivity, as proposed and validated in the case studies presented in Chapters 4 and 5.
- New workload generation techniques that considers reactivity: we develop a new workload generator, that considers reactivity, based on a known workload generator tool, described in 4.
- Novel QoS control techniques: we propose novel QoS control techniques, based on admission control and scheduling techniques and considering user reactivity, to guarantee QoS in Internet services for workloads with different characteristics. 4 describes these techniques and validate them.
- Improvements on modeling and characterizing e-business services.

1.3 Thesis Proposal Outline

This doctoral dissertation is organized in five more chapters. Chapter 2 presents an overview of related work. Chapter 3 explains the concept of reactivity, discussing how it may be modeled. Chapter 4 presents the work we have done for a Web site scenario, describing each step and the main contributions achieved. Chapter 5 describe another case study, an e-business application. Finally, we present our main contributions and conclusions in Chapter 6.

Chapter 2

Related Work

This section reviews the work in the following subjects that are related to this doctoral dissertation:

- User Behavior: focusing on how it has been studied by several different research areas, and also including an analysis of the research in computer-human interaction (CHI).
- Performance of Web Applications: studying the main aspects related to performance of computer systems, more specifically the ones that we adopted in this work: workload characterization and generation, performance evaluation, and quality of service (QoS).
- e-Business: investigating what is the business rules, an important aspect directly related to one of the main parts of our research, and online auctions, the main application we have chosen to work with in this research.

The next sections present the related works according to these three main subjects.

2.1 User Behavior

The user behavior can be analyzed using several variables observed in a Web log, for example. One can use the list of requests submitted to the server, navigational patterns, the types of services (functions [76]) accessed, the think-times, among other information. But, in these cases, aspects related to the quality of service provided are not considered. Several works characterize these behavioral aspects, but the understanding of the reactions of users as function of the performance of the server has not been completely studied as discussed next.

[56] uses the latency to study the user behavior in the specific context of a game application, detecting that network delay has effects on players' behavior, causing users to quit the service. [32] tried to model the click-stream in the context of Web advertising and [38] analyzes the correlation between requests in streaming media applications, trying to find trends in user interaction process.

[19] characterizes the user behavior in a public wireless network, considering distribution of the users, session duration, data rates, application popularity and mobility. The author tries to optimize the quality of access provided by wireless services.

[59] proposed a user behavior model framework, built in a top-down manner, consisting of various layers and based on mathematical models. This work was used to produce a user-oriented workload generator [58].

None of these studies, however, models the behavior of users to the performance provided by the service. They do not capture the aspects related to the reactivity of users to the quality of service. Despite the importance of this subject, it has not been completely studied, demanding works to understand and use it to develop techniques to deploy better services, with more quality of service.

Finally, [94, 49, 92] presents a methodology to characterize, understand and model the user behavior, based on their interaction with servers according to variations of performance. The methodology is based on the analysis and modeling of the correlation between the response time and the time between two requests of the same user. They present a methodology to characterize and simulate the user behavior named *USAR*.

2.1.1 Computer Human Interaction

Human-Computer Interaction(HCI) [103, 102] is about people, computers and how they impact on each other. It deals with the study of how humans interact with computers, and how to design computer systems that are easy, quick and productive for humans to use.

A basic goal of HCI is to improve interaction between user and computers, by making computers more user-friendly and easier to use. HCI is also concerned with:

- Methodologies and processes for designing interfaces (i.e., given a task and a class of users, design the best possible interface within given constraints, optimizing for a desired property such as learnability or efficiency of use).
- Methods for implementing interfaces (e.g., software toolkits and libraries; efficient algorithms).
- Techniques for evaluating and comparing interfaces.
- Developing new interfaces and interaction techniques.
- Developing descriptive and predictive models and theories of interaction.

Recently, the field of human-centered computing has emerged as an even more pronounced focus on understanding human beings as actors within socio-technical systems.

We studied some basic concepts about HCI in order to identify how reactivity relates to the research areas of the field, however we conclude later that reactivity is not a subject directly investigated by HCI topics.

2.2 Performance of Web Applications

2.2.1 Workload Characterization

Several studies tried to understand workloads of web applications and have identified several common characteristics such as file size and popularity distributions, self-similarity in web traffic, reference locality and user request patterns, as can be seen in [9, 13, 39, 77]. Menascé et al. [81] and Veloso et al. [115] proposed hierarchical methodologies to characterize e-business and streaming media workloads, respectively, considering three levels of characterization: request, function, and session. Nevertheless, these works did not focus on the user behavior analysis.

Streaming media workload characterizations were studied in [7, 8, 115], however the user behavior understanding is superficial. [37] studied the correlation between requests, trying to determine trends in the user interaction process.

There are other works that study user interaction with Internet services. [32] tried to model the click-stream in the context of web advertising, and [57] considered the latency to study the user behavior, but in the specific context of a game application, and detected that network delay has some effect on players' behavior. [19] characterizes the user behavior in a public wireless network, considering distribution of the users, session duration, data rates, application popularity and mobility. [59] proposed a user behavior model framework, built in a top-down manner, consisting of various layers and based on mathematical models.This work was used to produce a user-oriented workload generator [58].

2.2.2 Workload Generation

Workload generators are tools designed to generate synthetic logs, composed of requests that simulate real user requests. They are capable of exercising a server simulating the behavior of a set of user sessions (or just requests), submitting to the server a sequence of generated HTTP requests. Workload generation is fundamental to evaluate Web systems, since they permit us to test different server load scenarios. They have been studied in several works.

SPECweb99 [111], WebBench [117] and TPC-W [1] are benchmarks for evaluating the performance of World Wide Web Servers. They provide representative benchmarks for measuring a system's ability to act as a Web server. SURGE [23] and *httperf* [83] are workload generators, developed to exercise Web servers through the submission of a set of requests with different characteristics of load. Beyond them, other workload generators were designed to be applied to more specific scenarios. Gismo [63] and MediSyn [113] were designed to evaluate streaming media applications. ProWGen [28] is suited to evaluate proxy-cache servers.

These workload generators are powerful tools but are not capable of simulating user behavior patterns related to the reactions of the users according to the performance provided by the service. They adopt an arrival process independent of the performance provided, generating the same workload, despite the variations observed in the quality of service provided.

Therefore, the study of techniques to fill this gap is important and may provide more realistic workloads, improving existing workload generators and providing the possibility to conduct more precise evaluation of Web servers. This will help to improve not only the performance of the servers, but also some other important aspects such as the scalability of the service.

2.2.3 Performance Evaluation

Performance evaluation is a very important subject in order to verify the quality of service provided by a Web site when different workloads are applied to it. Several works address this problem, studying many aspects related, as will be presented in the next paragraphs.

Some works evaluate the impact of response time on users behavior, such as [107]. [24] presents an experiment where the users were able to classify the quality of service provided by an e-commerce application with variable response time answers. The authors perceived how the users change their behavior, as for example, ending (interrupting) the session in some cases. These aspects suggest that the reaction of users to the delay in getting the response should be considered in Web server performance evaluation. [89] presents a mechanism for Web servers to measure the response time perceived by the clients, using a model of TCP that analyzes the broken connections. Other works such as [67] analyzes the connectivity of the clients in order to take actions to improve the response time provided.

[47] shows that performance of a computer system depends on the characteristics of the workload it must serve. Therefore, performance evaluations require the use of representative workloads in order to produce dependable results. [14] presents a workload characterization study of six different data sets with different characteristics, discussing performance issues and suggesting enhancements for Web servers, focusing on caching strategies.

[86] studies how the environment where a Web server is located affects how it performs. The authors point out that the existing benchmarks at that time did not consider this component. In order to do this, they emulate the WAN environment by introducing elements such as delay and loss of packets, and perform tests to analyze the impacts of the new scenario on the throughput and response time of the server.

[16] presents an approach to test performance applying it in a case study. First of all, data is collected and analyzed in the point of view of its characteristics. Also, the performance impact of each type of request is evaluated in terms of CPU cost. Then, the bottlenecks are identified and solutions are proposed to solve them.

[33] proposes a back-end monitor to measure Web site performance by passively collecting packet traces. The paper presents the initial implementation of the monitor and analyzes the performance across three different Web sites.

2.2.4 Quality of Service - QoS

There are several efforts in QoS that focus on the computer networks level, but the same does not apply to Internet servers. Iyengar [60] analyzes a web server and finds out that there is a need to use some admission control in order to achieve good performance.

Bhatti and Friedrich [25] discuss QoS issues for web servers and develop a queueing model that provides varying priorities to different classes of customers, using it to investigate a simple admission control mechanism. Cherkasova and Phaal [34, 35] develop a session-based admission control that rejects new customers during overload period. Carter and Cherkasova [29] develop a method to detect inactive sessions. By removing requests belonging to such sessions, the server performance is improved. Abdelzaher and Bhatti [2, 3] propose a mechanism to adapt the content provided instead of rejecting requests to decrease the server load. Nevertheless, none of these efforts take into consideration the user behavior while designing the admission control strategy.

[45] presents a method for admission control and request scheduling for multiple-tiered e-commerce Web sites, achieving both stable behavior during overload and improved response times. The proposed method externally observes execution costs of requests online, distinguishing different request types, and performs overload protection and preferential scheduling using relatively simple measurements and a straightforward control mechanism. The authors present an implementation, called Gatekeeper, using it with standard software components on the Linux operating system. They evaluate the proxy using the industry standard TPC-W workload generator in a typical three-tiered e-commerce environment. The results show consistent performance during overload with a 15 percent penalty to large jobs.

[114] presents the Cataclysm server platform for handling extreme overloads in hosted Internet applications. The primary contribution of this work is to develop a low overhead, highly scalable admission control technique for Internet applications. Cataclysm provides several desirable features, such as guarantees on response time by conducting accurate size-based admission control, revenue maximization at multiple time-scales via preferential admission of important requests and dynamic capacity provisioning, and the ability to be operational even under extreme overloads. The authors implement a prototype Cataclysm hosting platform on a Linux cluster and demonstrate the benefits of their integrated approach using a variety of workloads.

2.3 e-Business

2.3.1 Business Rules

Business rules describe the operations, definitions and constraints that apply to an organization in achieving its goals [54, 10]. For example a business rule might state that no credit check is to be performed on return customers. Others could define a tenant in terms of solvency or list preferred suppliers and supply schedules.

In the web services literature there are several approaches dealing with the monitoring of the assertions over service-enabled business processes. The WS-Policy framework [84] provides a general purpose model for describing a broad range of service requirements, preferences, and capabilities. Typically, it is used when the provider describes the set of conditions (business rules) the requester should satisfy before invoking the service. RuleML [53, 54] is a powerful technique for expressing business rules over semantically annotated service. On the negative side is the lack of any support for run-time monitoring of the business rules.

In the context of this work, we adopt the business rules concept to identify the application functioning, that is, the application's logical rules. This concept is useful to describe one of the reactivity dimensions that we model.

2.3.2 Online Auctions

An auction is the process of buying and selling goods by offering them up for bid, taking bids, and selling the item to the highest bidder. In economic theory, an auction is a method for determining the value of a commodity that has an undetermined or variable price. In some cases, there is a minimum or reserve price; if the bidding does not reach the minimum, there is no sale (but the person who puts the item up for auction still owes a fee to the auctioneer). In the context of auctions, a bid is an offered price.

Online auctions present several aspects that violate the common assumptions made by the traditional economic auction theory. The auction duration is typically much longer than in traditional auctions; bidders may come and exit at any time; bidders are geographically dispersed all over the world; they have very distinct backgrounds and it is hard to predict how many bidders will end up participating in the auction. Instantaneous reactivity in such environments plays an important role, which we plan to address in our research.

Online auctions have been studied extensively lately. Many studies focus on validating concepts from the classic economic theory of auctions in the online environment. For example, Lucking-Reiley [71] checks the validity of the well-known results of revenue equivalence. Bajari and Hortacsu [18] address how the starting bid, set by the seller, affects the winner's course. Gilkeson and Reynolds [52] show the importance of a proper starting bid price to attract more bidders and make an auction successful. These works provide some interesting analysis, but do not focus on explaining the dynamics of the auction negotiation patterns.

There are some studies on bidding behavior analysis. Bapna et al. [21] develop a simulation model emulating bidders' behavior to analyze their impact in the outcome of the auctions. Using data from ubid.com, Bapna et al. [22] develop a cluster analysis approach to classify online bidders into five categories: early evaluators, middle evaluators, opportunists, sip-and-dippers, and participators. Moreover, they argue that bidders pursue different bidding strategies that realize different chances of winning and different levels of consumer surplus.

The widespread use of reputation and feedback systems and their impact on the outcome of online auctions has also received considerable attention. Resnick and Zeckhauser [104] and Ba and Pavlou [17] examine the effects of bidder and seller reputations on auction outcomes, concluding that seller reputations are correlated to auction success on eBay.

Auction design has been the focus of significant theoretical [106, 73, 82, 85] and experimental attention [64]. It has also been the subject of some limited empirical work [69]. However, most research focuses on the bid-taker's perspective and assumes a certain bidder behavior, none of them considering our concept of reactivity.

Menasce and Akula have several works in characterization of online auctions. [74] provides a workload characterization of auction sites including a multi-scale analysis of auction traffic and bid activity within auctions, a closing time analysis in terms of number of bids and price variation within auctions, some analysis of the auction winner and unique bidder. In this work they use data from Yahoo! Auctions site [15] and present some interesting overall conclusions about online auctions. In [6] they present a two-level (site and a user level) workload characterization of a real online auction site using data collected by automated agents. The main contributions of this paper are as follows: (i) a detailed workload characterization of a real auction site; (ii) an analysis of the presence of heavy tailed distributions in this workload; (iii) an analysis of the bidding activity during closing minutes of auctions; and (iv) an analysis of the arrival rate process of bidders and bids within clusters based on different attributes. One application of their work can be devise dynamic pricing and promotion models to improve revenue throughput of online auction sites.

For addressing reactivity in online auctions, it is important to take into account the work that has been done on analyzing bidders' and sellers' behavior in online environments [43]. Roth and Ockenfels [105] study the timing of bids, and the impact of different auction deadline rules. By comparing eBay and Amazon auctions, they found that auctions held with a "soft" ending time discourage late bidding (known as "sniping"), a common behavior observed on eBay.

For instance, while an auction is in progress, participants in the auction will be influenced by various types of value signals (e.g., minimum bid, seller reputation, other participants' bids, number of bids submitted up to that point etc.) which can, in turn, impact their decision dynamics for the auctioned item [12]. Recently, the standard assumption of bidder rationality in online bidding behavior has been questioned in a variety of empirical settings. In particular, Dholakia and Soltysinski [41] reported evidence of herd behavior bias, and Kamins, Dreze and Folkes [65] found an effect of minimum bid on the final auction price.

However, these studies are still limited regarding how they explain bidding behavior over the entire sequence of bids, as opposed to simply outcome summaries (e.g., final auction prices, and number of bids) in an auction [12, 31].

Reactivity has been widely studied in the database context [119, 90], and more recently has been applied also in Web semantic research [27]. Also, event-condition action (ECA) paradigm [46, 30] is an interesting topic in this context.

Our reactivity concept can be considered in any Web-based systems characterized by user-system interactions. The reactivity concept we are presenting in this work has a novel semantic, since its objective is to model the dynamics of e-business applications, like online auctions. In this context, there are no specific related works.

The most related work we have found in online auctions has been developed in Robert Smith School of Business (*www.smith.umd.edu*) by Galit Shmueli and Wolfgang Jank. In the work “Modeling the Dynamics of Online Auctions: A Modern Statistical Approach” [108], they investigate the bidding dynamics of auctions. Rather than focusing solely on the outcomes, they look at speed, acceleration and other dynamic characteristics [61].

Understanding auction dynamics is important for understanding and evaluating the information gained by other methods of analysis. For example, Jank and Shmueli’s research reveals that auctions are not homogeneous, even when the exact same product is being offered over an identical time frame. “Even if the end price is the same, they may have gotten to that end price very differently”, says Shmueli. The work we have done in online auctions shows similar conclusions in terms of dynamic auction analysis.

In other work [62], they propose a dynamic model for forecasting price in online auctions. This model operates during the live-auction, which makes it different from previous approaches that only consider static models. In their work, while one part of the model is based on the more traditional notion of an auction’s price-level, another part incorporates its dynamics in the form of a price’s velocity and acceleration. In that sense, it incorporates key features of a dynamic environment such as an online auction. They use novel functional data methodology that allow to measure, and subsequently include, dynamic price characteristics.

2.4 Summary

In this chapter we have presented the related work to this doctoral dissertation. Initially we focus on user behavior studies, workload characterization and generation. Performance and QoS are topics related to the first case study research, presented in this doctoral dissertation in Chapter 4. Afterward we identify a new topic to study - business rules. This concept is important once we adopt it as a second reactivity dimension, as will be presented in Chapter 3. Human-Computer Interaction (HCI) is another related area, even though it will not be directly applied in this work. Interaction between users and computers occurs at the user interface (or simply interface), which includes both hardware (i.e. peripherals and other hardware) and software (for example determining which, and how, information is presented to the user on a screen). A number of diverse methodologies outlining techniques for HCI design have emerged since the rise of the field in the 1980s.

The topics presented in this chapter are useful to understand the reactivity modeling (Chapter 3), the Web case study (Chapter 4) that concerns the performance aspects, and the e-business case study (Chapter 5), where we deal with online auctions. At the best of our knowledge, this work is innovative because reactivity has not been studied yet in related works.

Chapter 3

Reactivity

This chapter presents the concept of reactivity (Section 3.1), which motivates our research, and describes how we model it with some examples (Section 3.2), which we use later in the complete case studies that will be modelled, characterized and described in Chapters 4 and 5 of this doctoral dissertation.

3.1 Concept

In this section we formalize the concept of reactivity. This formalization is based on the Law of Cause and Effect, as previously introduced in Chapter 1, and also is directly related to bounded-rationality [112].

Developed by American behaviorist Herbert Simon (1916-2001), bounded rationality [112] is an analysis of decision-making which accepts that there are cognitive limits to an individual's knowledge and capacity to act rationally. Herbert Simon, in the book Models of My Life [110], points out that most people are only partially rational, and are, in fact, emotional/irrational with respect to their actions. We believe that reactivity fits in this rational part of user behavior.

We believe the reactivity is bounded (limited) because:

- It is not possible to identify all perception aspects that a user may consider.

- The way users act is not completely understandable, considering the rationality is bounded [109].

As we explained, considering reactivity as bounded, in order to understand the reactivity, it is necessary to observe two aspects:

- Action: what the user does during the interaction with the system.

- Perception: what the user perceives during the interaction with the system.

Reactivity is the correlation between one user's perception and consequent actions (and consequences of them) while interacting with a system or other user. One representation of reactivity is by qualifying (and quantifying whenever necessary) the pairs perception-action that characterize the user behavior. The perception criteria are a function of the application. Therefore, we can say reactivity can be modeled by a set of perception-action pairs according to specific criteria. In our work we consider two types of perception criteria:

1. <u>Performance</u>: in this case, the user action relates to performance metrics, such as response time.

2. <u>Business Rules</u>: in this case, the user action relates to business rules, such as the negotiation parameters of an e-business application.

Next we are going to describe some examples of these perception criteria. In a Web application scenario, there is a performance criterion, where the user perceives the response time to the submitted requests and performs actions after some time. The amount of time between the requests is the inter-arrival time (IAT). In this case the perception is the response time, that is, the amount of time spent until the user receives the request's answer, and the action may be the user submits for a new request before, during, or after receiving the answer, or goes away.

We may also find other scenarios where we may identify reactivity. In the context of online games, performance is an important factor, that determines a user's perception.

And based on this, the user acts doing new game plays before, during or after the result of the last play. In a voice over IP service, there exists some perception aspects: the initial connection delay, communication transfer rate (conversation quality), and the inter-communication time (ICT). These aspects can affect the user actions, that will determine to continue the conversation or not. In a streaming media service, the perception aspects are the transfer rate of the video and the transfer progress (which percent of the object has been already downloads). According to this, the user can do actions in order to pause, rewind, forward, re-initiate or cancel. There are other types of Internet services, such as P2P and chat, that consists of interesting scenarios to analyze reactivity in terms of performance.

There are services where the business rules dimension may be very important, such as e-business applications. One example is an online auction, where the negotiation parameters may affect directly the bidder's behavior. Examples of these parameters are number of bids, current value of the auction, number of participants, and target object. These aspects can affect directly the bidder in the decision to place one or more bids, which value to offer etc.

Our goal in this work is to qualify, quantify, and understand the reactivity in Web systems, such as the examples of systems presented. Obviously, reactivity may result from very complex scenarios, but we may be able to identify some key issues (directly related to a criterion) that are behind it. By defining common behaviors, cause-effect relationships, and trends associated with reactivity, we acquire a fundamental resource for characterizing user-service (user-system) interaction that may be used to improve the systems.

3.2 Reactivity Model

In this section we present and discuss our proposed model for reactivity. The main objective of it is to model the service provided by the application, detailing the interaction between the entities, that may be a typical client-server application or applications such as peer-to-peer (P2P), among others. Using the model it is possible to characterize the interaction lifecyle, including the reactivity dimension.

We distinguish several goals in defining such model:

- Determine the application details that are relevant for sake of characterizing and modeling reactivity.
- Define formally where reactivity takes place in a given application.
- Create a representation that support the understanding and modeling of user reactivity.

One strategy for modeling reactivity is to define the user behavior as a set of rules, where each rule has conditions as cause and reactions as a consequence.

The reactivity model may be divided into 3 parts, as follows:

1. **Conceptual Level**: presents a reactivity-oriented abstraction. It describes the application purpose and services provided, as well as the entities that interact around the application, the actions that each of them may perform and the application protocol, that is, common sequences of actions by specific actors.
2. **Definition Level**: makes explicit the reactivity by enumerating the perception criteria, that is, the criteria that a user may consider when reacting. Again, it is not only application specific, but allows to focus on specific aspects of the analysis.
3. **Modeling Level**: comprises the behavior model, where perception criteria are associated with actions, establishing the reactions causally.

There are some requirements in order to apply the reactivity model:

- It is necessary to identify the interaction between entities in a distributed system must take place, characterizing a pair action/reaction.
- Determine some perception criteria that qualify and/or quantify the reactivity.

It is important to emphasize the importance of well understanding the service's environment, once it is essential to specify the criteria that affect the user behavior.

The overall strategy to model reactivity is to characterize each user and then group users that are modeled by similar rules. As any model, it tries to abstract the relevant factors that affect reactivity, and evaluating its success is performed through comparison against other modeling strategies or even through verifying empirically whether the behavior is about the same across time.

In some cases, the behavior model is static, but in other cases it is completely dynamic, and subjective, requiring data mining techniques to identify reactivity patterns.

Formally, we define the first two parts of the reactivity model for any application as a tuple $< S, E, Ac, P, PC >$, where S is the set of states that the application may assume, E represents the set of agents, or entities, that exists, Ac is the set of actions that may be executed by the entities, P is the application protocol (that defines the valid actions according to the application's state), and PC is the set of perception criteria that may characterize the reactivity. P represents the protocol, that is, the application functioning - the rules that define which actions may be executed by the entities according to the application states. Once we have a reactivity model, we can try to identify some correlation between the perception criteria (PC) and the set of actions that an entity may execute. This function will be used to add the reactivity concept to any traditional interactivity model.

Next we present two applications where we model reactivity. The first two parts of modeling reactivity (Conceptual and Definition) are presented, and the third one (Modeling - behavior model) is going to be presented later in each detailed case study (Chapters 4 and 5).

3.2.1 Web Site Example

In this section we model a simple Web site, where there are clients requesting data and one or more servers answering them.

In a Web site there are servers answering the requests submitted by users. Each user is associated with a user session, that has the set of requests submitted by him/her to the Web site over a period of time. Each user request is answered by the Web site server after a period of time, denoted response time.

The reactivity model applied in the context of a Web site is presented in Table 3.1.

Element	**Value**
S	Active Inactive
E	User
Ac	User: submit requests
P	Active - User - submit requests
PC	Response Time (latency)

Table 3.1: Reactivity Model - Web Site

As can be seen, the reactivity model enables the conceptual definition of the application and also provides the perception criteria to consider in order to characterize the reactivity. These aspects contemplate the first two parts of the model, which are the basis to do the third part, which provides the rules to explain reactivity. The function that correlates the pairs perception-action to provide the final part of reactivity modeling will be described later in each case study (in Chapters 4 and 5), since it demands a deep investigation of the application's scenario and a complete characterization methodology.

3.2.2 E-business: Auction Case Study

An auction is the process of buying and selling things by offering them up for bid, taking bids, and then selling the item to the bidder who gave the highest bid.

The English auction is the most well known model. Participants bid openly against one another, with each bid being higher than the previous bid. The auction ends when no participant is willing to bid further, or when a pre-determined "buy-out" price is reached, at which point the highest bidder pays the price. The seller may set a "reserved" price and if the auctioneers fail to raise a bid higher than this reserved price the sale may not happen.

In an auction there are two entities, the buyer and the seller. The English auction is an ascending-price auction. Each auction instance represents the session of the auction engine. There are many states that an auction session may be assigned to: *Active*, *Active with Bids*, *Active with Buy-it-now Option*, *Active with Bids and Buy-it-now Option*, *Cancelled*, *Ended with Buy-it-now Option*, *Ended with Sale*, and *Ended without Sale*. Buy-It-Now is the simplest way to buy on eBay. It allows you to buy an item when you want it, at a known set price. When this option is set, the seller gives you the opportunity to purchase the item right away without waiting for an online auction to end for the defined price. The seller may create an auction session, cancel it, and set the "Buy it now" option. The buyer may place a bid, the most common action, or perform a "Buy it now" offer. There are several attributes that may affect the buyer's action, such as: the number of bids, the current price, the seller's feedback, and the payment method. Despite this, we have analyzed thoroughly several attributes in order to identify which of them are more semantically relevant to capture the user's perception in terms of e-business. We identify some attributes that are more related to the e-business negotiation, such as the *time of the negotiation*, *winning price*, *who is winning the auction*, and *type of competition*. Therefore we have decided to adopted these attributes as perception criteria for our research in online auctions. These attributes will be described in the case study of online auctions, presented in Chapter 5.

Applying the reactivity model in this scenario we get the result presented in Table 3.2.

Element	**Value**
S	Created Active Active with Bids Active with Buy-it-now (BIN) Option Active with Bids and Buy-it-now (BIN) Option Cancelled Ended with Buy-it-now (BIN) Option Ended with Sale Ended without Sale
E	Buyer Seller
Ac	Buyer: MakeBid, MakeBuyItNowOffer Seller: Cancel, SetBuyItNow
P	Created - Seller - Cancel Active - Buyer - MakeBid Active - Seller - SetBuyItNow Active with Bids - Buyer - MakeBid Active with Bids - Seller SetBuyItNow Active with BIN Option - Buyer - MakeBid Active with BIN Option - Buyer - MakeBuyItNowOffer Active with BIN Option - Seller - Cancel Active with Bids and BIN Option - Buyer - MakeBid Active with Bids and BIN Option - Buyer - MakeBuyItNowOffer Active with Bids and BIN Option - Seller - Cancel
PC (related to Buyer)	Negotiation Time Winning Price Winning Bidder (who is winning?) Competition Type

Table 3.2: Reactivity Model - Auction

3.3 Summary

In this chapter we have presented the reactivity concept, that is the main subject of this work. Reactivity defines that the user action varies according to some perception criterion. We present two dimensions of reactivity: performance and business rules. We presented a way to model reactivity and an explanation of how to apply it to many applications, such as Web sites, e-business, and even other applications, such as streaming media and online games. The next chapter deals with the performance dimension, presenting a complete case study, involving characterization, modeling, workload generation and QoS. The Chapter 5 presents a case study of an online auction, which comprises the business rules dimension. We believe that these two case studies are representative enough to illustrate how to analyze and model reactivity.

As can be seen, the reactivity model enables the conceptual definition of the application and also provides the perception criteria to consider in order to characterize the reactivity. These aspects contemplate the Conceptual and Definition levels of the model, which are the basis to do the third one (called Modeling level), which provides the rules to explain reactivity. The Modeling level will be described in Chapters 4 and 5, since it demands a deep investigation of the application's scenario and a complete characterization methodology.

Chapter 4

Case Study: Web Site Application

One motivation of this work is to understand how users interact with the system, allowing to provide services with better performance and to understand better the business applications.

The design of systems with better performance and scalability is a real need to fulfill the user demands and generate profitable Web services. Achieving such performance, however, has been a constant challenge, motivating the development of several techniques and mechanisms for both guaranteeing those requirements and evaluating how well they are addressed. The latter has been tackled by several workload characterization studies and workload generators that mimic the behavior of actual users and allow the experimental evaluation of servers performance.

Workload generators so far have addressed several aspects of Web workloads, such as arrival process and resource popularity. However, there is one aspect that has not been modeled: how users react according to the server-side aspects, such as the performance of the system, that is, how the behavior of the user changes as a function of the response of a server. It is not clear how the changes in user behavior affect the workload submitted to the server, and how the process iterates, when the server, answering to a reactive behavior, changes again the user behavior and so on. Another issue is whether and how the cross-dependences among server and users reach an equilibrium. Considering these aspects, reactivity in Web server performance analysis raises new challenges, in terms of both realistic workload generation and scalability mechanisms. Moreover there are new characteristics of the service provided that can affect the user reaction, such as the parameters of an e-business service (an auction, for example), that can determine whether the user will participate in it, and how he/she behaves over the negotiation session.

Figure 4.1 represents the interaction between a user and a Web server. The time a user takes to receive the answer to a request, after sending it, is named *latency*. The latency may vary according to several aspects, such as the server load. After or even before receiving the response, the user may submit another request, in a process that repeats over and over until the user decides to get out. Therefore, the quality of service perceived is related to the latency and affects the way users interact with the service.

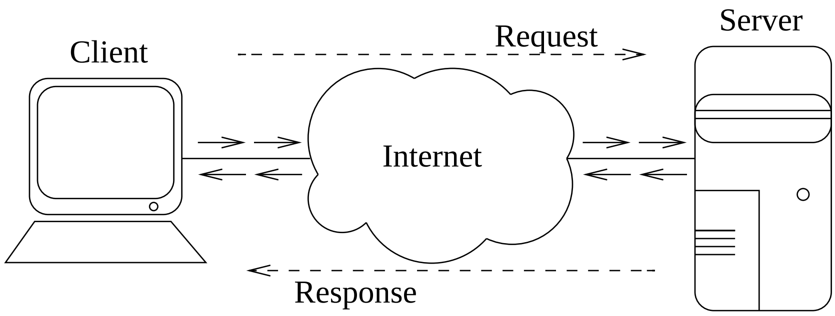


Figure 4.1: Client/Server - Web Interaction Process

This chapter presents a case study where we focus on the performance dimension of the reactivity. The next sections describe how we characterize and model the reactivity, the use of it in workload generation, our reactive simulator, and the use of reactivity in QoS control mechanisms.

4.1 Characterizing and Modeling Reactivity

Understanding the nature and characteristics of Internet services workloads is very important to design systems with better performance and scalability, which are features that affect directly the quality of service experienced by the users. Workload characterization techniques, such as [81], are popular strategies for this task.

Workload characterization may also be the starting point to build synthetic workload generators. These generators are an effective way for exercising a system capabilities through the request of operations that are similar to actual workloads. Nevertheless, most of the Web-related workload generators do not take into consideration system variations in terms of quality of service, generating the same workload despite the server response time, for instance.

Accounting for user behavior is expected to aggregate valuable information to the modeling and generation of workloads through the analysis of criteria such as navigational patterns, actions requested, and the inter-arrival time (IAT) between requests. From the server or proxy perspective, the user behavior is materialized through the sequence of actions performed by a user (that is, click in a link). Although desirable, it is not usually possible to observe other criteria such as the user activities between actions. Characterizing and replicating these behavior-related criteria is a challenge that we address in this work. In particular, we focus on the relation between IAT and server response latency (the time to process and answer a request) for each action.

We propose *USAR*, a hierarchical workload characterization model. The model also comprises a validation methodology through simulation of user reactions. As mentioned, we consider behavior as how the interaction of users with web applications is affected by variable latencies. We define six user profiles and describe how they are identified while analyzing the stream of user actions and the respective server response times. In order to demonstrate and validate the *USAR* model, we present its application and validation using actual data from a proxy-cache server.

To the best of our knowledge, this work has three innovative contributions: the hierarchical workload characterization model that adds the user level; the modeling of the user actions based on the analysis of the relation between IAT and latency; and the accurate simulation of these actions.

4.1.1 The *USAR* Characterization Model

In this section we present both the *USAR* characterization model and its validation strategy.

4.1.1.1 Workload Characterization Model

Previous efforts employed hierarchical methodologies to characterize workloads, considering user-side and server-side metrics, but ignoring the correlation between them.

In order to fill this gap we propose a new characterization model, named *USAR*, which extends [77, 81, 93]. The *USAR* model consists of four levels: User, Session, Action, and Request. Each level is described as follows:

User (U): This level models the user behavior considering the offered quality of service, abstracting from the operations performed by him/her.

Session (S): This level models the sets of user actions performed within a time interval, that is, actions that are apart less than a pre-defined threshold τ. The characterization considers the following criteria: session length, session duration, session composition in terms of actions, bytes transferred per session, and IAT between sessions.

Action (A): This level models the actions performed by the user, which are usually clicks that activate a link. Besides characterizing the probability distribution of the action types, we also characterize the correlation between response time and IAT.

Request (R): This level models the objects associated with a user action. We study the request arrival process and the probability distribution of its features.

The idea of the hierarchy of levels (see Figure 4.2) is to guide the analysis of the workload into different perspectives. This eases the characterization process, once it may be done according to different views associated to each level. Therefore this process becomes clear and produces a more detailed characterization.

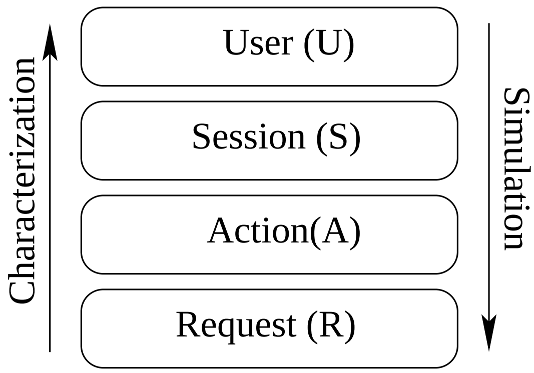


Figure 4.2: The *USAR* Model

In the context of this work, our main objective is to describe the characterization at the *user level*. As mentioned, in this work this level intends to model the user behavior, more specifically the reaction of the user to variable latencies, that is, how the response time (quality of service) affects the user actions. Next we describe the 8-step methodology we propose to analyze and characterize this level.

1. Prepare log, generating a temporary log Lu by putting together the sessions of each unique user;

2. Analyze users from the following perspectives: IATs between requests of the same user, latency associated with requests of the same user, IAT and latency ratio, and IAT and latency difference;

3. Discretize IAT and latency measures using a function that correlates them. In our case, the function takes into consideration the ratio and the difference between them, clustering user actions that are similar in terms of the two measures;

4. Transform user sessions into sequences of user action classes using the aforementioned discretization criterion;

5. Evaluate the sequences in order to group them according to a similarity criteria;

6. Process the log Lu applying a function $f(Lu)$, which maps sequence of classes to the groups defined in the last step;

7. Apply a clustering technique such as K-Means [51] to determine clusters of similar user sessions;

8. Analyze the clusters and classify them according to the probability associated with user action, concluding the user behavior analysis.

4.1.1.2 Validation

In this section we present our validation methodology, which is based on generating a synthetic log that mimics the behavior of a user. The characteristics of the generated log is matched with the characteristics of the actual log so that we are able to verify the precision of the modeling process. The generation of reactivity logs enables the enhancement of workload generators such as SURGE [23] or httperf [83], so that they generate Web workloads considering the interaction between the users and the system.

Current workload generators [23, 83] do not allow the simulation of user reactions and we believe that this innovation may provide significant improvement in the quality of the workload generated. Therefore, a major goal of the *USAR* validation strategy is to be the starting point for the construction of a simulator that mimics actual user reactions, based on the relation between IAT and latency.

To capture and replay the properties of the user behavior, we model and implement a user action simulator, parametrically defined, as shown in Figure 4.3, that deals with three types of data:

1. Input data: specify the characteristics of the users that interact with the system. The input data defines the types of users, their behaviors, the user profiles that compose each behavior, and the following distributions for each *USAR* level:

   - User: we define the probability distribution of user types, the probability distribution of behaviors that compose each type and the probability distribution of user profiles (groups obtained in step 6 of the characterization methodology) that compose each behavior.

- Session: the probability distribution of session length is defined. This distribution will be used to determine the number of user actions in each user session.
- Action: we determine the distribution of user action classes in sequences that are directly related to user profiles. As a result, we specify the popularity of the actions in each profile.
- Request: the last level involves the generation of the requests that compose the Web workload. For a precise workload generation, we must determine the popularity of the objects and their size probability distribution. These distributions are not used in the generation of user action classes, but are important to transform these classes into real requests.

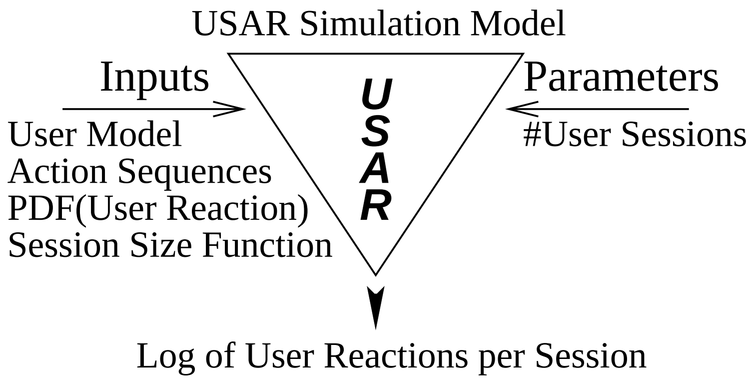


Figure 4.3: The *USAR* Simulator

2. Execution parameter: parameter that are set in a per execution basis, such as number of sessions that must be simulated.

3. Output data: a log file composed of user sessions. Each session is composed of classes of user actions that can be applied to a workload generator to produce Web workloads that are representative for evaluating the performance of the server and the network and simulates accurately the user behavior.

4.1.2 Experiments and Results

This section presents a case study that demonstrates the effectiveness and applicability of the characterization model and its validation strategy. We focus on the innovative features of the model, more specifically the user level characterization. Further, we will just evaluate basic aspects and explain the main results related to the other levels.

Our characterization is based on one month of log files from the Squid proxy-cache server of the Federal University of Minas Gerais (UFMG). The log contains an entry to each request performed by a user and consists of the following information: request timestamp, latency to obtain the resource, requester IP, object status in cache and HTTP status code, object size (bytes transferred), method (GET, POST), URL of the resource requested, peer status and host (for caches that participate in a hierarchy), and mime type.

We start by performing a detailed analysis of this workload in order to determine the diversity of the user population, the variety of Web sites being accessed, and the latency variation that the users experience while accessing these sites, among other criteria.

4.1.2.1 Workload Characterization Model

In this section we present the resulting characterization of our case study, from the request to the user level.

Request level characterization

In the request level characterization we focus on the requests regardless the action, session, and user associated. The log L records the accesses of users from one of the largest federal universities in Brazil and it has a considerable number of requests per day.

We apply the methodology at the request level, analyzing about 9 million requests issued from about 500 unique IP addresses statically assigned, generating a traffic of almost 90 Gbytes. As expected, 98% of the requests are for HTTP objects and the remaining requests are either for FTP or HTTPS objects. Table 4.1 presents general information about the log, which contains more than two million requests per week, a significant amount of unique

Attribute	Week			
	1	2	3	4
# Requests ($x10^6$)	2,44	2,10	2,15	2,08
MegaBytes ($x10^3$)	34,4	19,9	16,8	20,7
# Unique IPs	488	485	497	507
# Sessions	6952	6979	7369	7756
# Unique Obj. ($x10^5$)	4,82	4,13	4,29	3,97

Table 4.1: General information about the workload

objects, unique IPs, and user's sessions.

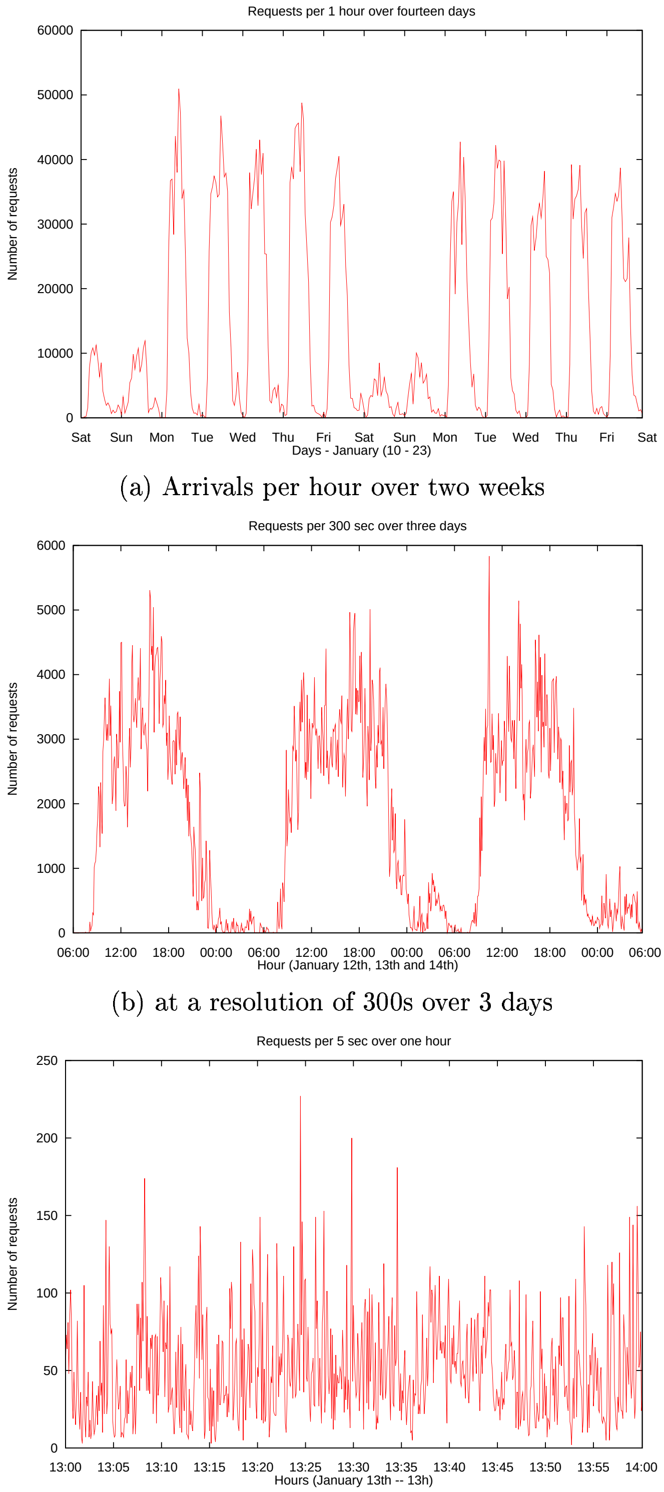
Its analysis demonstrates that traditional characterization dimensions, such as object size, object popularity, and request arrival distribution, match previous characterizations of the same type of traffic [23, 81].

Figure 4.4 shows three graphs that plot the request arrival process at three different time scales, nominally, one hour, five minutes and five seconds. The analysis of the graphs show clearly that the arrival process of requests presents a self-similar characteristic.

Figure 4.5 presents two graphs that show the probability distributions of objects and object sizes. In both cases, we can clearly see that both distributions are quite skewed, as observed in other characterizations of Web-related traffic.

Action level characterization

The action level characterization demands the generation of a temporary log with the relevant requests, that is, requests that are directly associated with actions. The characterization consists of the identification of the action types, which are directly related to the functions provided by the application, and the multi-scale analysis of the actions, among other relevant analysis.

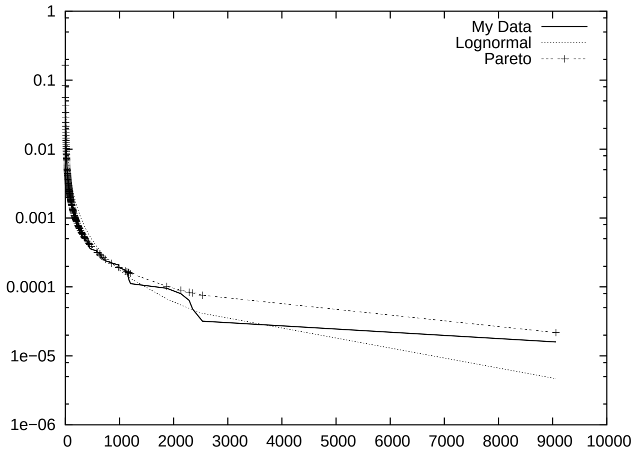


(a) Arrivals per hour over two weeks

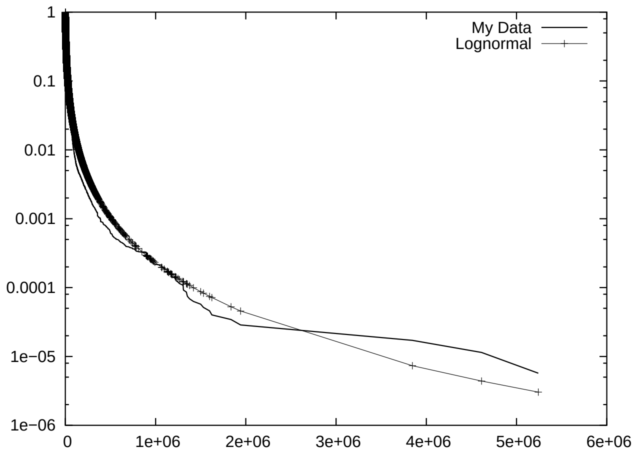
(b) at a resolution of 300s over 3 days

(c) at a resolution of 5s over one hour

Figure 4.4: Total number of requests arriving at proxy-cache at various time scales



(a) Popularity of Objects



(b) Object Size

Figure 4.5: Request Level Characterization

Since our analysis is based on a proxy log, it is not possible to determine the type of the user action by simply analyzing its URL and other data that composes the log, since we do not have contextual information about the service being provided. Therefore, in the context of our case study, the action types are not as relevant as for other application domains, such as an E-business service.

Considering only the requests to HTML objects, the total number of user actions is 160585. We modeled the probability distribution of the action latencies and found that it fits a lognormal distribution, as shown by graph in Figure 4.6. This fitting demonstrates the high variability of the observed latencies, which gives room to investigate the correlation between the server response time and the user reaction.

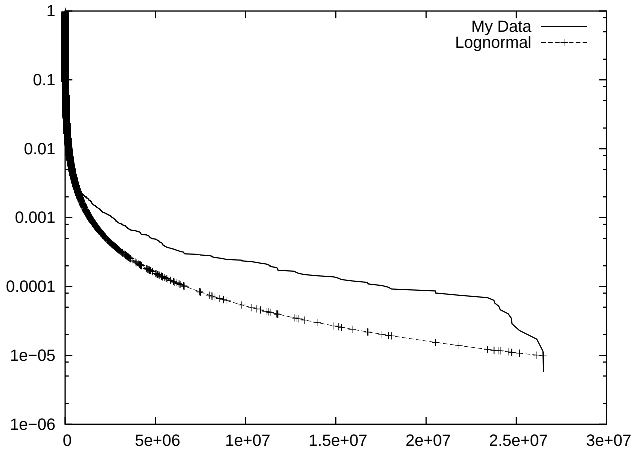


Figure 4.6: Action Latency

Session level characterization

Considering just the requests to 62770 unique HTML objects and using a threshold of 1800 seconds for session duration, we identified 14352 user sessions associated with 518 unique IP addresses. One initial step of the session level characterization is to generate a log Ls for each session s.

We then analyze the session length (Figure 4.7), in terms of number of number of requests, and notice that more than 90% of the sessions are composed by at most 26 requests, the average and the maximum session lengths are 12.2 and 1105 requests, respectively. The analysis of the session duration shows that most of the sessions (almost 80%) last for less than 1800 seconds and the average session duration is 1172 seconds. We also observe a significant variation in the distribution of requests among sessions. Note that the observed average session length shows the suitability of this case study log (workload) to our work, since short sessions do not provide enough information to model user behavior.

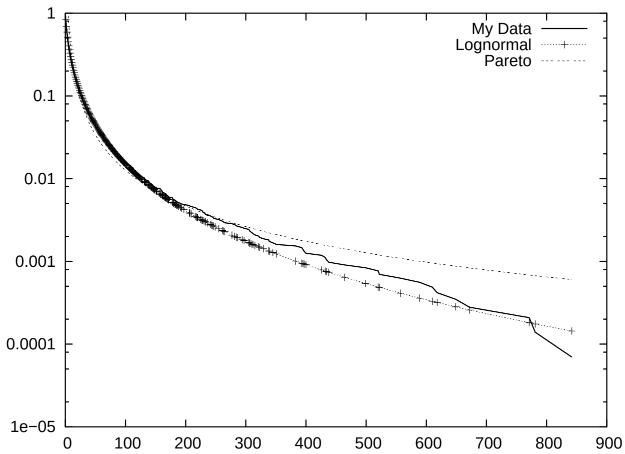


Figure 4.7: Session Length

User level characterization

This subsection describes the user level characterization performed for the case study according to the proposed methodology. The first step is to generate a temporary log Lu by putting together the sessions of each user u, although the session identifiers are kept.

We then analyzed the user data from the following perspectives: IATs between consecutive requests, latency associated with user requests, IAT-latency ratio, and IAT-latency difference.

We discretized IAT and latency measures using functions that correlates them, more specifically the ratio (RAT) and the difference (DIF) metrics. These metrics are defined as:

$$DIF(k) = I(k, k+1) - L(k), \ \forall \ k \in Lu;$$

$$RAT(k) = \begin{cases} \text{I(k,k+1)/L(k)} & \text{, DIF(k)} > 0 \\ \text{L(k)/I(k,k+1)} & \text{, DIF(k)} < 0 \\ 1 & \text{, DIF(k)} = 0 \end{cases};$$

where k is a user request, $I(k, k+1)$ is IAT between request k and $k+1$, and $L(k)$ is the latency associated to the request k.

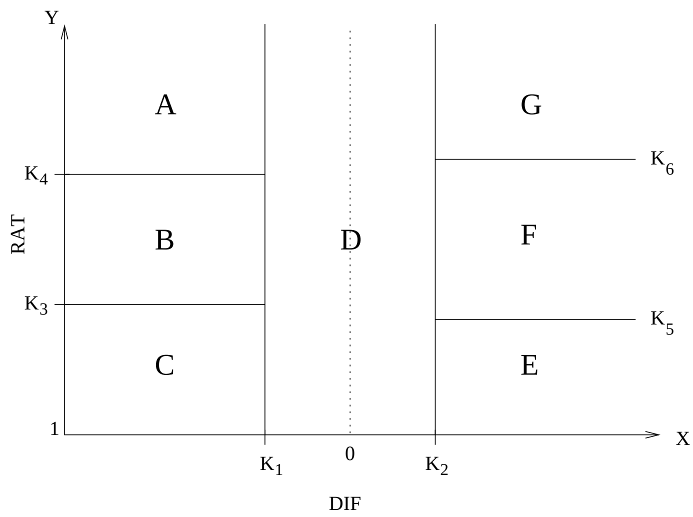


Figure 4.8: Discretization Model

Figure 4.8 depicts the discretization model based on the two functions. The x axis is associated with the DIF function and the y axis with the RAT function. The model defines seven user action classes (A to G), using two limit values for each axis. Values k_1 and k_2 divide the positive and negative sides of DIF function, defining a zone close to zero, where we can not say much about the user behavior. This zone represents values of IAT and latency very close to each other, which can represent situations such as: users who request objects and ask another one few seconds before the first request answer arrives; and users who request objects and do not process the answer, since they request another object immediately after the request answer arrives. As shown in Figure 4.8, we define an interval D between k_1 and k_2, which comprises all values of RAT, motivated by the fact that the value of the RAT function does not affect it. Values k_3 and k_4 break the vertical scale into three different zones, according to RAT function that quantify the correlation between IAT and latency. We then define three classes (A, B and C) for DIF values that are less than k_1. These classes represent behaviors where users do not wait for the answer to their requests before asking another object. The same strategy is applied to the actions that have a DIF greater than k_2, that is, we define three classes (E, F and G) that represent behaviors where users wait for the answer to their requests before asking another one. The boundaries of these classes is defined by two other constants: k_5 and k_6. For instance, Class

E represents actions where the user requests a new object a short time after receiving the previous object. On the other hand, class C represents users who do not receive the object within the expected delay, but wait a significant amount of time, considering the object's latency. We plot the points' histogram and decide the values that divide the classes. (k_1 and k_2 are assigned to the values -0.1 and 1, respectively. The values of k_3 and k_4 are 2 and 4, respectively, for the discretization of classes A, B and C. For the delimitation of classes E, F, and G we choose the values 4 and 8 for k_5 and k_6).

We then transform user sessions into sequences of user action classes using this discretization strategy. This transformation is a direct map one-to-one from application of functions RAT and DIF to each request of user session, defining a user action class. As a result, we define, for each action, a pair $u(DIF((k), RAT(k))$ of user request, where k is the current request in the user session. This pair corresponds to a location in the discretization model, defining the user action class for each of his or her actions. The total number of user actions is 160585. Table 4.2 presents the frequency of user actions.

User Action Classes(%)						
A	**B**	**C**	**D**	**E**	**F**	**G**
1.56	1.47	2.45	3.80	8.40	8.97	73.35

Table 4.2: Distribution of User Action Classes

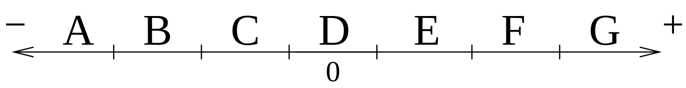


Figure 4.9: User Profile Tendency - Patience Scale

Considering the scale presented in Figure 4.9, we define six user profile trends, each of them representing a user behavior trend in this scale, with the following characteristics.

Impatient: The sequence of actions are in the negative side of the scale, including zero (e.g., $C \to A \to B$, or $A \to C \to B$, or $C \to D \to B$, or $C \to C \to C$, or $A \to A \to A$). Represents a variation in the user behavior, keeping the impatient tendency.

Patient: The sequence of actions are in the positive side of the scale, including zero (e.g., $E \rightarrow G \rightarrow F$, or $G \rightarrow E \rightarrow F$, or $F \rightarrow D \rightarrow G$, or $E \rightarrow E \rightarrow E$, or $G \rightarrow G \rightarrow G$). Represents a variation in the user behavior, keeping the patient tendency.

Continuous: The sequence of actions shows low variability, staying in fixed class at zero (e.g., $D \rightarrow D \rightarrow D$). It represents a fixed tendency, situations where the latency of the requested object and the IAT are very close. This is a typical web robot behavior or users whose tendency is not well-defined.

Impatient tendency: The sequence of actions represents a impatience tendency (e.g., $G \rightarrow D \rightarrow A$, or $C \rightarrow B \rightarrow A$, or $G \rightarrow F \rightarrow E$, or $G \rightarrow A \rightarrow C$, or $E \rightarrow B \rightarrow C$, $F \rightarrow G \rightarrow A$, or $E \rightarrow G \rightarrow B$).

Patient tendency: The sequence of actions represents a patience tendencies that usually move right in the scale (e.g., $A \rightarrow D \rightarrow G$, or $A \rightarrow B \rightarrow C$, or $E \rightarrow F \rightarrow G$, or $B \rightarrow G \rightarrow F$, or $C \rightarrow F \rightarrow E$, or $B \rightarrow A \rightarrow E$, or $C \rightarrow B \rightarrow F$).

Inconstant: The sequence of actions that move from the negative (positive) side to the positive (negative) side and return to the negative (positive) side or zero - and sequences that move from the negative side or zero to the positive side and return to the negative side. (e.g., $C \rightarrow E \rightarrow B$, or $A \rightarrow G \rightarrow D$).

We then translate the user session log into a sequence of user profiles. The new user session representation consists of sequences of actions, where each tendency fits to a patience scale variation.

We use a Java-based tool for machine learning and data mining [51], which implements regression, association rules and clustering techniques. More specifically, we use the algorithms *K-Means* (KM) and *Expectation Maximization* (EM) [66] to perform clustering.

We finally got the results with *K-means*, which show interesting conclusions related to the user behavior. Adopting "within cluster sum of squared errors" as a metric for cluster analysis, we identify 7 as the best configuration for the number of clusters. The distribution of user profiles for each cluster is presented in Table 4.3. It shows the identification of clusters, the percentage of user sessions and the distribution of user actions according to user profiles, in each cluster.

	Sess	User Profile (%)					
Id	**(%)**	**1**	**2**	**3**	**4**	**5**	**6**
1	50	0.01	99.34	0.01	0.21	0.37	0.06
2	17	0.57	67.69	0.86	14.92	10.14	5.81
3	3	31.41	0.48	33.77	0.88	6.17	27.28
4	4	0.2	0.6	0.0	97.63	1.48	0.09
5	11	1.28	36.28	0.56	41.65	13.86	6.37
6	11	0.27	50.04	0.14	6.96	40.16	3.44
7	4	0.0	0.03	0.13	1.22	98.11	0.24

Table 4.3: Clusters - Distribution of Sessions and User Profiles

Analyzing the clusters, we can describe them as:

Cluster 1: Almost all users clustered in this group presents a *Patient* profile during their sessions.

Cluster 2: Presents a significant occurrence of two profiles (*Patient Tendency* and *Impatient Tendency*), a little of *Inconstant*, and the majority of *Patient* profile.

Cluster 3: Represents a balanced occurrence of three profiles (*Impatient*, *Continuous*, and *Inconstant*).

Cluster 4: Almost all users clustered in this group presents a *Impatient Tendency* profile during their sessions.

Cluster 5: The profiles *Patient* and *Impatient Tendency* have been identified in this cluster as the most significant. Also there is a considerable amount of *Patient Tendency* in this group.

Cluster 6: The profiles *Patient* and *Patient Tendency* have been identified in this cluster. It is a group of typical patient users, that maintain a strong patient tendency under some variation.

Cluster 7: Almost all users clustered in this group presents a *Patient Tendency* profile.

The most popular cluster is number 1, which corresponds to half the amount of user sessions. Cluster number 2 has 17%, followed by clusters 5 and 6, both with 11%. The remaining 11% is divided by clusters 4 (4%), 7(4%), and 3(3%). Analyzing them we can observe the predominance of *Patient* profile, but the occurrence of *Patient Tendency* and *Impatient Tendency* is also very significant. The profiles *Impatient*, *Continuous*, and *Inconstant* are significant in the cluster with the small popularity.

The characterization is complete and now we are going to validate it, as described in the next section.

4.1.2.2 Workload Characterization Validation

In this section we present the generation of user action classes, as a strategy for validating the characterization model and simulating the behavioral characteristics of the users of the proxy-cache server of the Federal University of Minas Gerais (UFMG).

1. One user type, seven user behaviors represented by the clusters (Table 4.3) and six user profiles.

2. The probability distribution applied to the user behaviors use the values obtained in the clusters (second column of Table 4.3) to determine the percentage of occurrence of behaviors for the user.

3. The probability distribution associated with the user profiles (impatient, patient, continuous, impatient tendency, patient tendency, and inconstant) uses the values listed in Table 4.3 to determine the percentage of occurrence of profiles in each user behavior.

4. The session length distribution, according to the number of user action classes per session, follows *log-normal* distribution (see Figure 4.7) with $\sigma = 1.463065$ and $\zeta =$

1.430258. The least square computed for this function is 0.022657, a very good value. The log-normal distribution has the following probability density function:

$f(x;\zeta,\sigma) = \frac{e^{-(\ln x-\zeta)^2/(2\sigma^2)}}{x\sigma\sqrt{2\pi}}$,

for $x > 0$, where ζ and σ are the mean and standard deviation of the variable's logarithm, respectively.

5. The probability distribution of user action classes in sequences uses the percentage of occurrence of each sequence in the real log - considering sequences of size 1, 2 and 3 - to determine weights for the further simulation. We explain how the sequences are composed later in this section.

6. The characterization shows that the popularity of objects follows *Pareto* distribution with $\alpha = 0.980308$ and $k = 0.158868$. To certify the quality of the distribution fit, we calculate the least square measure and obtained a value of 0.000653 for *Pareto*. The size of the objects are strictly correlated to the latency observed in the server and does not follow any known distribution (see Figure 4.5).

The characterization process defines seven user action classes that correlate IAT and latency. To map these classes to the six user profiles, we decide to group the action classes into sequences. Combining the seven classes we achieve an amount of 343 possible sequences of size three, 49 of size two, and 7 of size one - sequences of size 2 and 1 are used as the last sequence in case of the session length not being multiple of 3. Each sequence has a weight, calculated according to the absolute value of its occurrence in the real log and the percentage of sessions where it appears. At last, the sequences are mapped to a user profile according to a tendency, as presented in USAR.

This generation process is being used to validate the workload. Using these data and considering the number of sessions informed as a parameter, the generation process, for each session, follows these steps:

- Choose a user behavior according to the distribution given as input data (for this case study a probabilistic distribution).

- Calculate the session length, number of user action classes, and the number of sequences that compose the session.
- Generate the sequences of user actions following the distribution of the user profiles that compose the chosen user behavior (see Table 4.3). Some sequences are more frequent than others considering the weight calculated for each sequence.

At the end of the generation process, the output data is a synthetic log containing sessions of user action classes that considers the quality of service of the system to simulate the real interaction of users with the proxy-cache server. We should emphasize that the generation of an accurate synthetic log is a challenge because we created quite an abstract model of behavior that is often highly variable. As we see next, the precision achieved by the model is one of the contributions of this work.

The objective of the simulation is to show the applicability of the characterization model in order to generate a synthetic log that gives the user action in response to the quality of service provided. Using an efficient method for generating discrete random variables with general distributions [116], we simulate the distribution of the sessions among user behaviors accurately. Further, the simulation produces precise results in terms of the distribution of sequences over the six user profiles observed in each cluster (see Table 4.3). It is hard reproduce exactly the same sequences because the session length distribution was calculated for the whole log.

Nevertheless, a careful analysis shows a good result regarding the frequency of occurrence of each user action class in the synthetic log, when compared to the real one. Table 4.4 presents the results for our proxy log.

The simulation values for all user action classes are very close to the real log characterization and demonstrate the feasibility of the adoption of well-defined models, based on the user behavior, to characterize web services workloads, considering the server-side view of the user interaction under variable latencies.

Finally, during the validation, we identify some directions to improve this work, such as the analysis of the correlation of session length and user profile, which can minimize the

Table 4.4: Distribution of User Action Classes

Log	User Action Classes (%)						
	A	**B**	**C**	**D**	**E**	**F**	**G**
Real	1.56	1.47	2.45	3.8	8.4	8.97	73.35
Synthetic	1.54	1.44	2.4	3.76	8.38	8.95	73.51

difficulty of simulating the real distribution of the sequences among user profiles.

4.1.3 Discussion

Several previous studies [9, 37, 39, 77, 81, 115] focused on characterization and generation of workloads from Internet service providers. However, none of them models the user's reactions to the service response time. We believe that this information is very useful to improve the understanding of Web workloads and to construct more realistic workload generators.

We presented the *USAR* characterization model, which guides the characterization of the user actions based on the latency and the IAT of the requests. The model also comprises a validation strategy, that is a starting point towards generating more realistic workloads. We validate the model through a case study, using the log of an actual proxy-cache server, where we demonstrate the key features of the *USAR* model.

This work is the basis of future efforts in workload generation, considering our user behavior approach, which reduces the gap between the existing models of characterization, which are non reactive, and the actual workloads.

4.2 Using the Reactivity Model to Generate Workload

This section discusses the problem of generating workloads considering reactivity, showing the main changes we have made to the *httperf* workload generator [83] in order to provide reactive workloads.

The ability to generate a stream of HTTP requests that mimics a population of real users is important for performance evaluation and capacity planning of Web servers. However, generating representative Web reference traces is a hard challenge because of the complexity of the aspects involved.

There are several workload generators, but they do not consider the impacts of response time in the way users react. They generate similar workloads despite of the response time perceived. The main objective of this work is to show that there are significant differences between the workload applied to a Web server that considers the impact of response time in user reaction and the other one that does not.

The *USAR* Model [94] brings new concepts that make possible to model and replicate how the user reacts to variations in the response time. To apply these concepts to generate workloads, some important characteristics are needed by a workload generator:

- It must be able to initiate a new request, even if the last one has not finished yet, i.e., has not received its answer yet. This aspect is very important mainly when the response time grows, since some users could have a kind of impatient behavior, since he does not wait completely for the last answer.

- The inter-arrival time must not be static, but may vary dynamically according to the response time perceived and the user action class of the *USAR* model, that associates to each burst a value that correlates the response time and the inter-arrival time.

  Moreover, it is important to introduce some important concepts related to our work:

- Burst: consists of a sequence of requests for fetching a web page with embedded objects (like pictures). A burst of requests is submitted to the server when a user clicks on a link or requests a Web page. Bursts mimic the typical browser behavior where a click causes the browser to first request the selected Web object and then its embedded objects.

- Session: a sequence of bursts in which the time between any two consecutive burst is below a specific threshold.

4.2.1 Workload Generation using *httperf*

In this work we use *httperf* as the tool for workload generation. We chose it because it provides an effective way of generating HTTP workloads and measuring performance. For programming, *httperf* is a framework based on events and highly customizable. We have created a new version of *httperf* in order to allow it to generate new workloads and make it compatible with the *USAR* Model. The next paragraphs describe the new features we have modeled and implemented.

httperf has a module called *wsesslog*, which submits requests based on a user session file. It contains many aspects of user sessions, such as the number and sequence of requests, HTTP method (POST/GET), think-time and *burst* length. In order to aggregate the reactivity model created by *USAR*, we have added information about the user action class to the user session structure.

In order to determine the user action class, according to *USAR* model, we need the value of the response time observed by the client (in this case, the *httperf* itself) and the client think-time. A typically *wsesslog* file contains the think-time, so we only have to get the response time.

The value of response time can be easily obtained in *httperf*, since it is built around the concept of events. For example, every time that a session is created or destroyed, or a request is started or finished, a new event is triggered. These events (see Figure 4.10) can be captured through call-back handles, defined using *httperf* API functions. There is a response time associated to each request and another one that belongs to the burst. Basically, the values of response time could be obtained using two events, one triggered by request start and the other one triggered when the response completely arrives. To obtain real values of response time, we submitted a workload file based on TPC-W [50, 78] using the original version of *httperf* to the test environment.

Figure 4.10 shows the traditional mechanism of executing a user session. It represents the client and server sides and some events associated to the execution. The vertical space represents the time. The figure illustrates the session duration and the concepts of response time, think-time and IAT. The main request of the burst is represented by a bold line and the embedded requests are single lines.

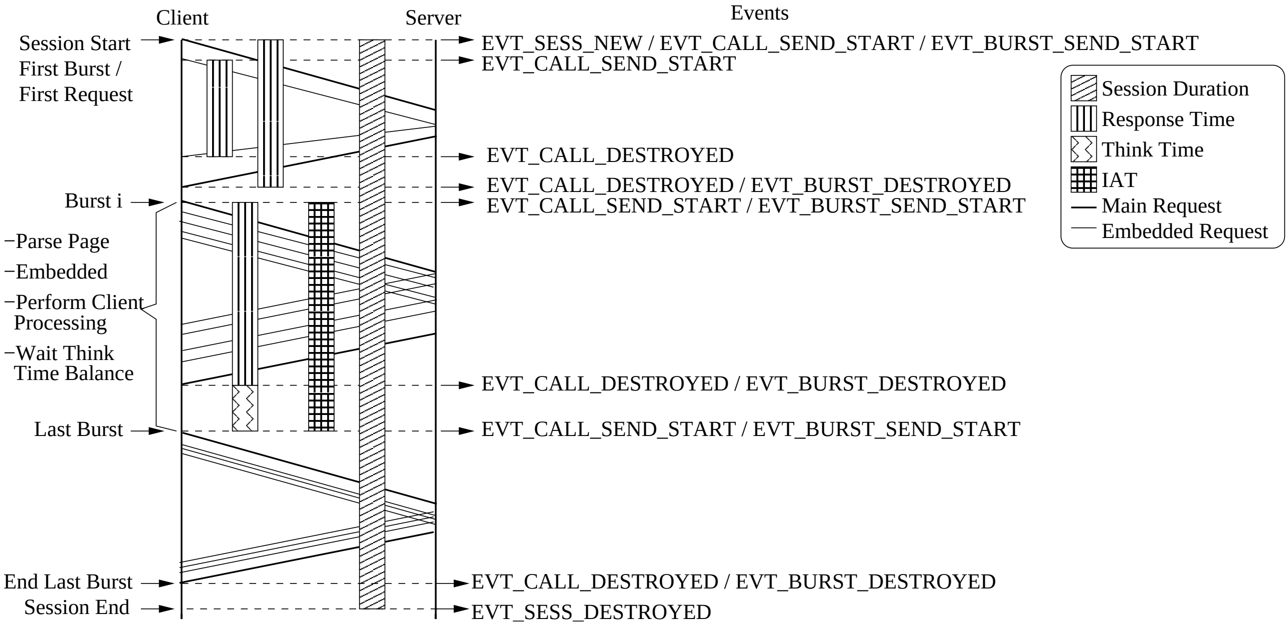


Figure 4.10: *httperf* - Traditional Mechanism

4.2.2 Reactive Workload Generation using *httperf*

This section explains how to generate the reactive workload. First, it is important to define the concept of “user impatience”. Traditional workload generators model that a new request of the user must wait the last one to be completed before dispatching a new one. This approach does not allow to represent the situation where the user wants to send a new request although the last one has not finished yet. The user can do this because the response time for the last request is unacceptable for him/her, for example. We name here “user impatience” this new situation that can be modeled, demanding the ability of the workload generator to allow non-blocking sessions, i.e., a burst of requests can begin before the last burst has completed.

The original *httperf* (see Figure 4.10) does not consider "user impatience", i.e., if a request takes a long time to complete, the program keeps waiting for it forever. The only parameter close to this concept is the time-out value, but it is a radical solution: when a request times out, the *httperf* finishes the entire session. The best way to do it would be to continue the execution of the session, even if some requests had timed out (in this case, reporting it).

In order to reproduce "user impatience", we changed the way how *httperf* schedules the burst that is submitted for each session. The original implementation waits until the last submitted burst finishes to start a timer event that triggers the next burst. We adapted the *wsesslog* to start a timer event as soon as the first request of the burst was submitted. But how long should *httperf* wait before triggering a new burst? The value can be calculated using the user class and the response time of the former request.

We instrumented *httperf* to record some important events. We recorded the following values for each request and burst: session identifier (SESSID), time when it was sent (SNDREQ, SNDBUR) and received (RCVREQ, RCVBUR), response time (client perspective), bytes received, if the request has timed out and if it could not be completed (error); and for each session created or destroyed, how many sessions are active (SESCNT).

At the end we have a new version of *httperf*: non-blocking and reactive. This version supports submitting requests that time-out after a period specified by the user think-time that is obtained during execution time, through these steps:

- Each session records its last response time observed, i.e., how long the last successful burst submitted spends up to the end, and the burst size (in bytes). These two metrics are useful in order to estimate the next expected response time (ER) for the new burst, i.e., the amount of time that the user expects that his/her burst will be served.

- Each burst has associated to it its user class and the exact moment of time when the first request was issued.

- The Expected Inter-arrival Time (EIAT) can be computed considering the think-time (Z).

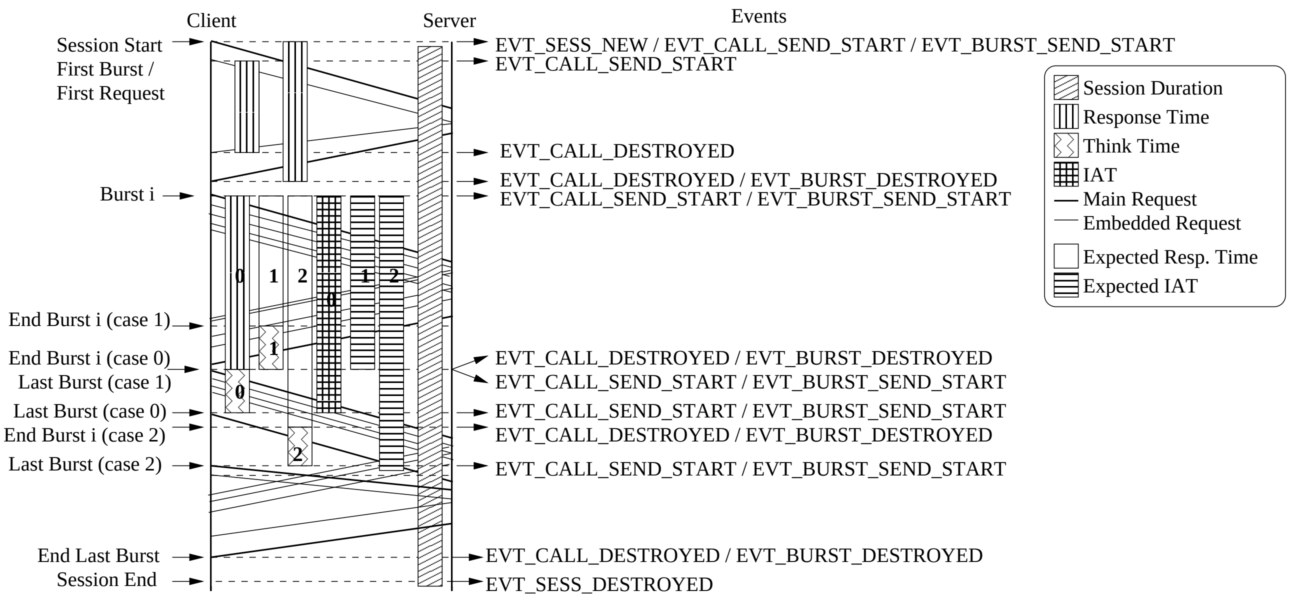


Figure 4.11: New Version of *httperf* - Non-Blocking Reactive

Figure 4.11 shows the reactive version of the workload generator. In the figure, the elements with the flag 0 models the traditional mechanism. There are two new elements in the figure, the expected response time and the expected IAT. The shapes with the numbers 1 and 2 represent the new kind of behavior for the client considering the reactivity. In this two examples the IAT is computed dynamically according to the server response time.

4.2.3 Experiments and Results

In this section, we present the results of our experimental study to analyze the impact of workloads that simulates the reaction of users to the quality of service provided.

The simulation environment of our experiments is composed of a HTTP Server (Apache), an application server (Apache Tomcat), a relational database server (MySQL) and a client (*httperf*), each running on different machines. Each machine runs Linux with kernel version 2.4.25, having a Intel Pentium 4 1.80GHz CPU, and 1GB of RAM memory.

For best performance, we have turned off all unnecessary services and configured the operating system to support a number of file descriptors that was enough for our experiments (65000 file descriptors).

We have used a Java Servlet implementation of TPC-W benchmark. We adapted the workload generated based on TPC-W to add the information related to the user behavior, following some steps:

1. To obtain a synthetical workload that could follow the same navigation rules as would a real user, we create a base workload following TPC-W recommendations and its CBMG. The workload generated, *wl-tpcw*, is composed of 5000 user sessions with medium session length of 124 bursts.

2. We convert the *wl-tpcw* workload on a new one, *wl-httperf*, which is compatible to the format used by the *httperf*'s module *wsesslog* [83].

3. We submit the workload *wl-httperf* to our simulation environment using the original version of *httperf* and record the real response times under the client's perspective.

4. With the recorded response times and the workload *wl-httperf*, we apply the *USAR* characterization model, obtaining the distribution of user actions for each burst of requests.

5. We add to the workload *wl-httperf* the information obtained in the last step, obtaining the workload *wl-httperf-react* that can be used by the new version of *httperf* to generate workloads with reactivity.

Our main objective with these experiments is to analyze the impact of reactive and non-reactive workload in the performance of servers. The reactive workload models users who act dynamically according to variations in the response time. And the other workload consists of users who act in a static way according to the inter-arrival times predefined for their sessions.

It is important to emphasize that the number of simultaneous users are defined by the number of active sessions over the experiment time. We execute experiments with many different workloads setup and here we present three of them:

- A: a workload with 100 sessions with a rate of 100 sessions initiated per second.
- B: a workload with 1000 sessions with a rate of 100 sessions initiated per second.
- C: a workload with 5000 sessions with a rate of 100 sessions initiated per second.

We choose these workloads, since we want to assess the impact of reactive workloads in a light, medium and heavy conditions. For each workload setup, we have the reactive an non-reactive approaches. We focus our analysis in the most overloaded period - the first ten minutes. The experiments evaluate a set of metrics for each scenario:

- Throughput: in our experiments we show both the output and input throughput. The latter corresponds to the rate of requests submitted per unit of time to the server. The former represents the rate of responses received by the client per unit of time.
- Cumulative throughput: in the same way as the throughput measure, we analyze both the cumulative throughput of input and output.
- Response Time: here we refer to the user perceived response time, consisting of the time between the submission of the request and the time when the client finishes to receive the response. It is important to explain the concept of response time, that is directly related to the way user reacts. Response time is a critical factor to users of interactive systems [75]. It is evident that user satisfaction increases as response time shortens. Modest variations around the average response time are acceptable, but large variations may affect user behavior. Regarding the response time of a system:

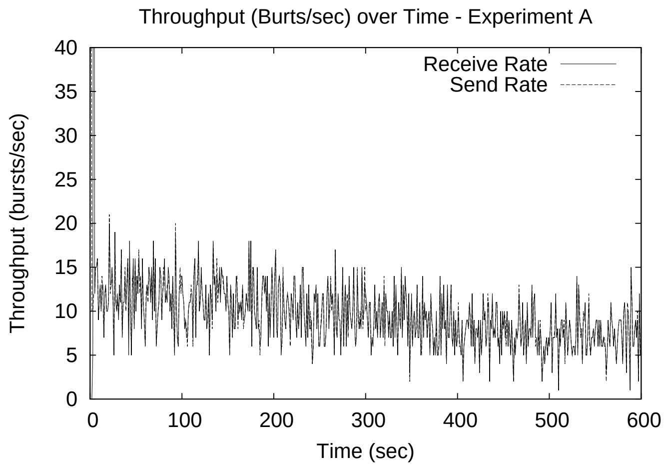
  - 0.1 sec: is about the limit when a user perceives that the system is reacting instantaneously.
  - 1.0 sec: is about the limit when the flow of thought of a user is not interrupted, although the user may notice the delay.
  - 10.0 sec: is about the limit when a user loses attention and the interaction with the system is disrupted.

- Active bursts: this information capture the number of bursts requested to the server but not yet responded, for each period of time.
- Active sessions: this information represents the number of sessions initiated but not yet finished, i.e., that have bursts that are active or have not been submitted yet.

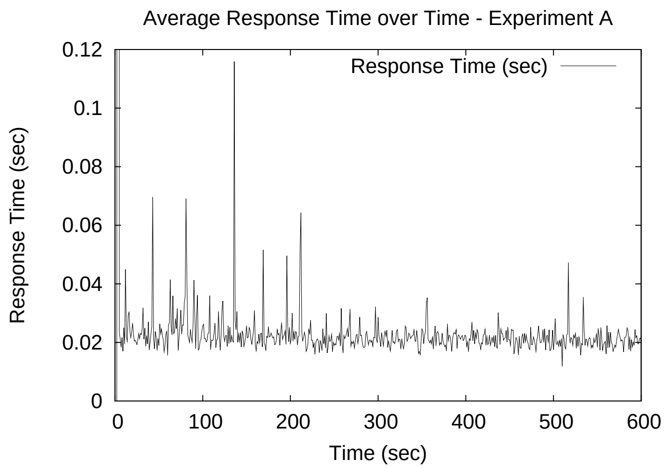
Figures 4.12 and 4.13 present the throughput(a) and response time(b) considering the bursts of requests for the workload A and the non-reactive and reactive experiments, respectively.

The non-reactive experiment A achieves 6000 bursts, with an average throughput of 9.2 bursts/second. The average response time is around 0.027 seconds. Considering the requests, the non-reactive experiment A achieves 50000 requests, with an average throughput of 100 requests/second. The average response time is very small, near zero (instantaneously). This confirms the non-overloaded state.

All the experiments have a rate bigger than the normal condition, once the workload generator dispatch all of them almost in the same time. A few time after this, the load becomes ready to analyze. The number of active bursts during the non-reactive experiment A is very low, once there are not problems with performance. During this experiment, 45% of sessions ended.



(a) Throughput



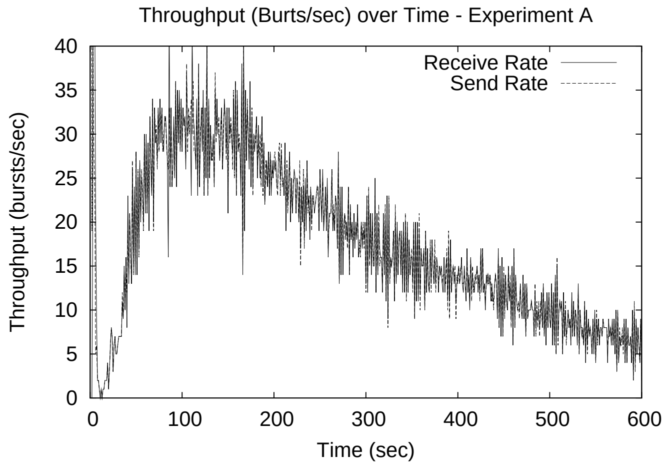
(b) Response Time

Figure 4.12: Experiment A - Non-reactive - Bursts

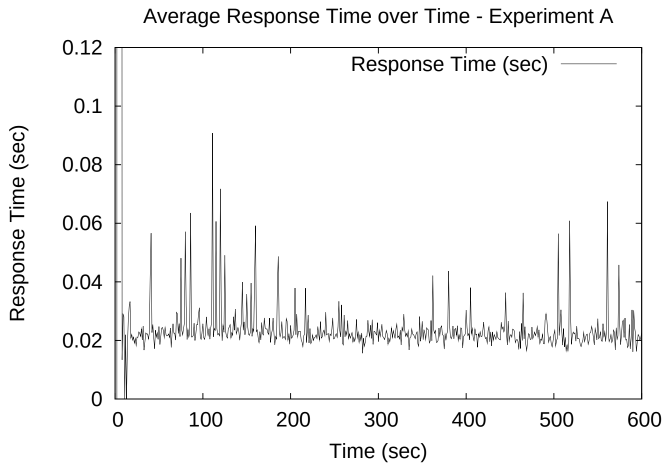
The reactive experiment A achieves more than 10000 bursts, with an average throughput of 16.1 bursts/second. The average response time is around 0.039 seconds. We can conclude from this analysis that throughput raises without changing the response time, achieving a better performance.

Considering the requests, the reactive experiment A achieves 90000 requests, with an average throughput of 190 requests/second. The average response time is very small, as observed by the non-reactive approach.

The number of active bursts during the non-reactive experiment A is very low, once



(a) Throughput



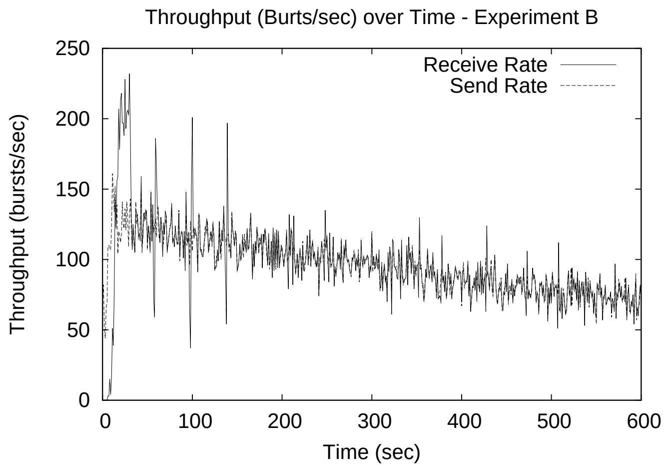
(b) Response Time

Figure 4.13: Experiment A - Reactive - Bursts

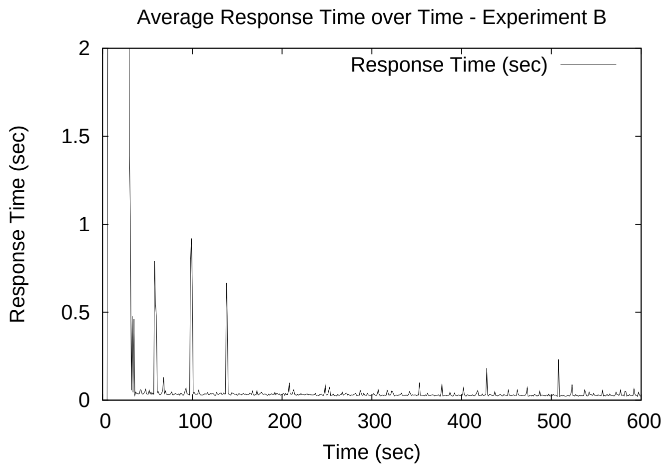
there are not problems with performance. During this experiment, 85% of sessions ended, showing that reactivity allows users to reduce the estimated session time once the response time to their bursts of requests are very small.

These two experiments (workload A) show the server achieves a very good performance, guaranteeing that users perceives an instantaneous answer to their bursts of requests. A good response time rate allows users from the reactive experiment to request faster their new bursts. The increase in the throughput rate without changing the response time rate shows the server was not overload. The decrease in the bursts execution time causes the reactive experiment to end more sessions than in the non-reactive one.

Figures 4.14 and 4.15 present the throughput(a) and response time(b) considering the bursts of requests of the workload B for the non-reactive and reactive experiments, respectively.



(a) Throughput



(b) Response Time

Figure 4.14: Experiment B - Non-reactive - Bursts

The non-reactive experiment B achieves 57000 bursts, with an average throughput of 92.2 bursts/second. The average response time is around 0.24 seconds. Considering the requests, the non-reactive experiment B achieves 500000 requests, with an average throughput of 800 requests/second. The average response time is very small, near zero (instantaneously) with peaks under 1 second. This confirms the non-overload state.

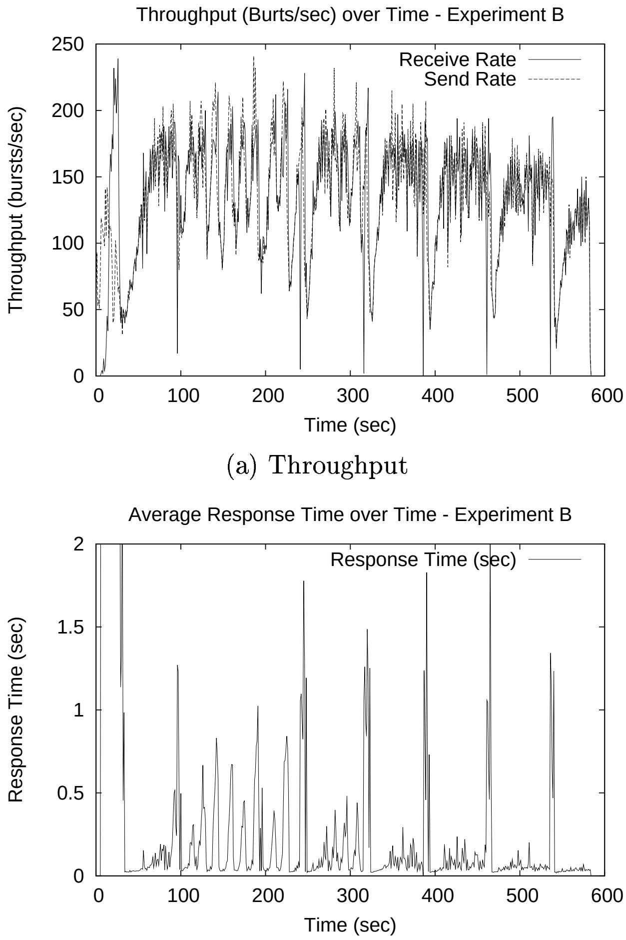
The number of active bursts during the non-reactive experiment B presents a stable behavior, once there are not problems with performance. There are few peaks, that can be explained comparing it with the behavior of the response time. These peaks occur exactly when the response time presents some delay. During this experiment 45% of the sessions has ended, the same percentage of the number of sessions that has ended in the non-reactive experiment A.

The reactive experiment B achieves 78000 bursts, with an average throughput of 114.4 bursts/second. The average response time is around 0.35 seconds. Considering the requests, the reactive experiment B achieves 650000 requests, with an average throughput of 1200 requests/second. The average response time is still small, but not instantaneously as the first experiment(A). There are response time peaks under 2 seconds, but isolated situations that not endanger the performance of the server. In this few situations, the users can observe a small delay in the server response.

The number of active bursts during the reactive experiment B presents variations, that can be explained comparing them with the response time behavior of this experiment. These peaks occur exactly when the response time presents some delay, as expected. During this experiment, 90% of sessions finished.

It is interesting to note that the throughput rate of reactive experiment B decreases exactly when response time rates raise, but in this case the change in the users reaction causes the throughput rate to raise again after some time. From the graphics, we can observe the application server keep a very good response rate to the requests (around 1200/sec) without overload.

Figures 4.16 and 4.18 present the throughput(a) and response time(b) considering the



(a) Throughput

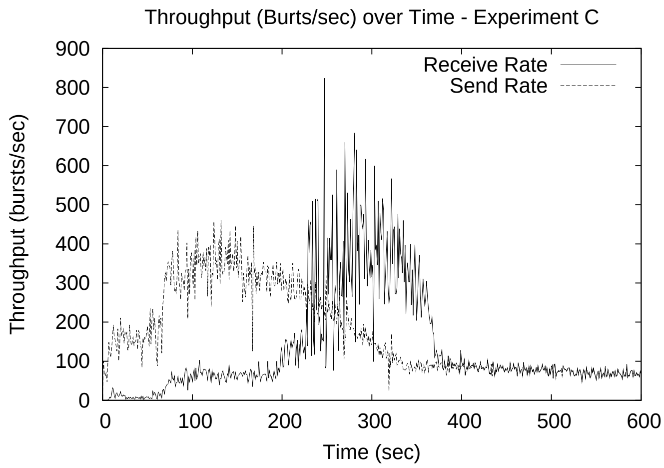
(b) Response Time

Figure 4.15: Experiment B - Reactive - Bursts

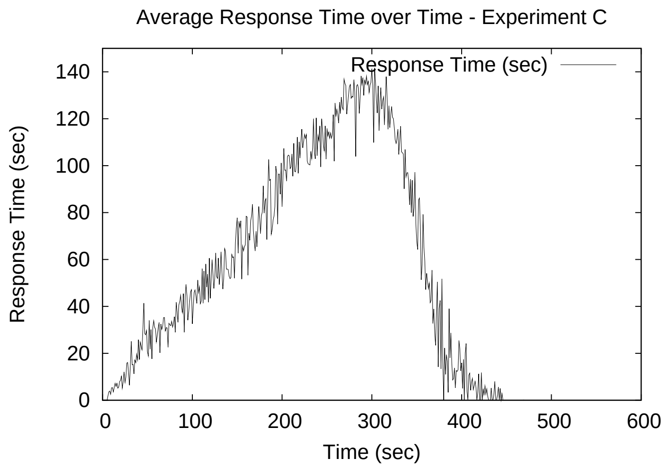
bursts of requests for the workload C and the non-reactive and reactive experiments, respectively.

The non-reactive experiment C executes 80000 bursts, with an average throughput of 123 bursts/second, varying from 100 to 400 bursts/second after the initial seconds. The response time raises from few seconds to more than 120 seconds, with an average time of 40.7 seconds.

Considering the requests, the non-reactive experiment C achieves 580000 requests, with a throughput varying from 200 to 1600 requests/second. This amount of request is only 20% higher than the amount observed in experiment B. It is easy to observe that the server became overloaded. After 30 seconds from the beginning, the response time has already achieved the 10-second limit.



(a) Throughput



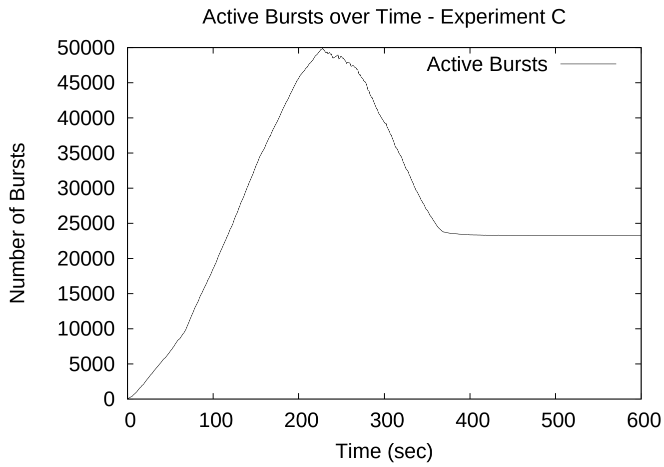
(b) Response Time

Figure 4.16: Experiment C - Non-reactive - Bursts

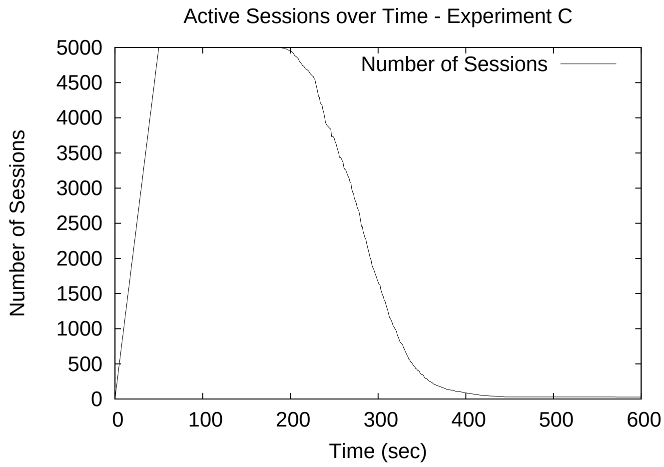
It is important to analyze what happened near 360 seconds. The following aspects are registered: the response time begins to decrease, the throughput decreases, the number of active bursts has balanced, and the number of active sessions has decreased fast. A detailed investigation shows the cause of this anomaly: the TCP/IP connection has timed-out, represented by the system error number 110 in Linux operating system. This problem has caused the following behaviors in the experiment metrics:

- Cumulative throughput: the difference between send and receive rates is more than 20000 bursts. This occurs because these amount of bursts do not finished as connections have closed after the TCP/IP time-out.
- Throughput - Figure 4.16-(a): the throughput rate decreases and keeps near 100 bursts/second, consequence of bursts of a small number of sessions that stay active after the problem.
- Response time - Figure 4.16-(b): the average response time decreases fast when the problem with TCP/IP occurs. After 450 seconds, it achieves acceptable values by users.
- Active bursts - Figure 4.17-(a): the number of active bursts continues high, as many bursts have not finished in consequence of the time-out of TCP/IP. After 400 seconds the value becomes balanced, with vary small variation.
- Active sessions - Figure 4.17-(b): the number of active session decreases fast, which demonstrates that a lot of sessions begin to fail in consequence of the error identified. When the workload generator tries to open or to send requests and the TCP returns the error, the current session fails and close after there are no more connections available for it. Only an amount of 100 sessions become active after 400 seconds, representing the users who generate load to server from this point to the end of the experiment.

In this non-reactive experiment C we identify a big overload in the server, which causes a very worse performance. The response time values observed are unacceptable. Moreover the unavailability of the server represents a big problem, one of the most serious problems that overload can cause - around 80% of the users stay waiting for server's answer without success.



(a) Active Bursts



(b) Active Sessions

Figure 4.17: Experiment C - Non-reactive

The reactive experiment C executes 80000 bursts, with an average throughput of 133 bursts/second, varying from 25 to 250 bursts/second after the initial seconds. The response time raises from few seconds to more than 60 seconds, with an average time of 13.7 seconds.

Considering the requests, the reactive experiment C achieves 690000 requests, with a throughput varying from 50 to 1800 requests/second. This amount of request is only 6% higher than the amount observed in experiment B. It is clear that the server became overloaded. After 20 seconds, from the beginning the response time has already achieved the 10-second limit.

In Figure 4.16-(a), we can see the receive rate increasing and send rate decreasing from the period between 100 and 200 seconds. It is possible to observe a correlation between average response time and active bursts, as expected. The change in the way users react when the server is overload causes a delay in the session duration of the users - 75% of sessions are still active after the experimental time.

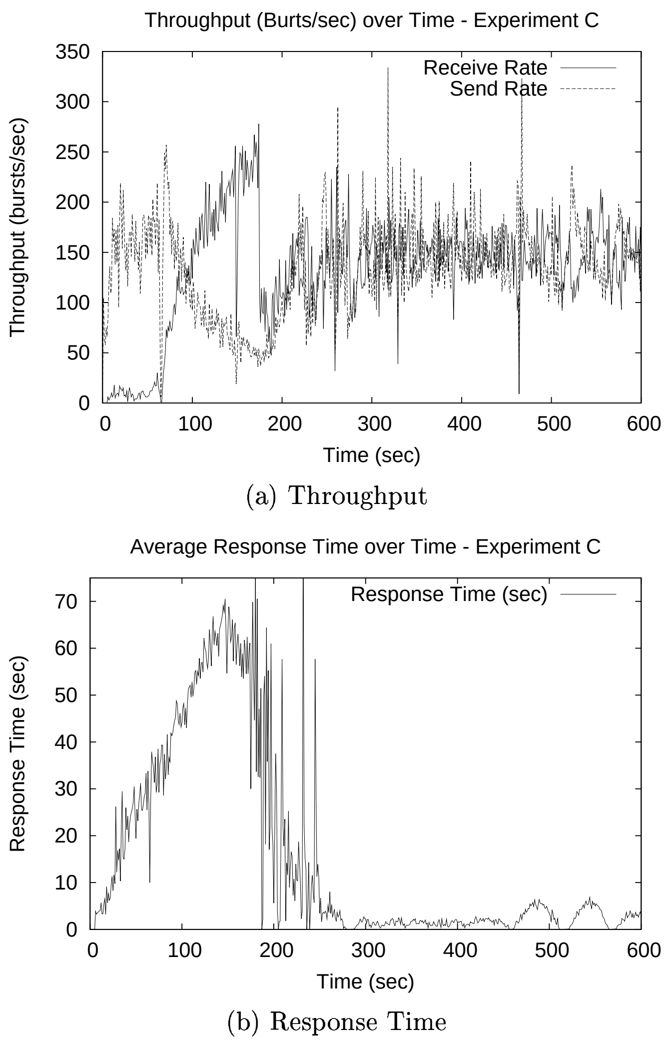
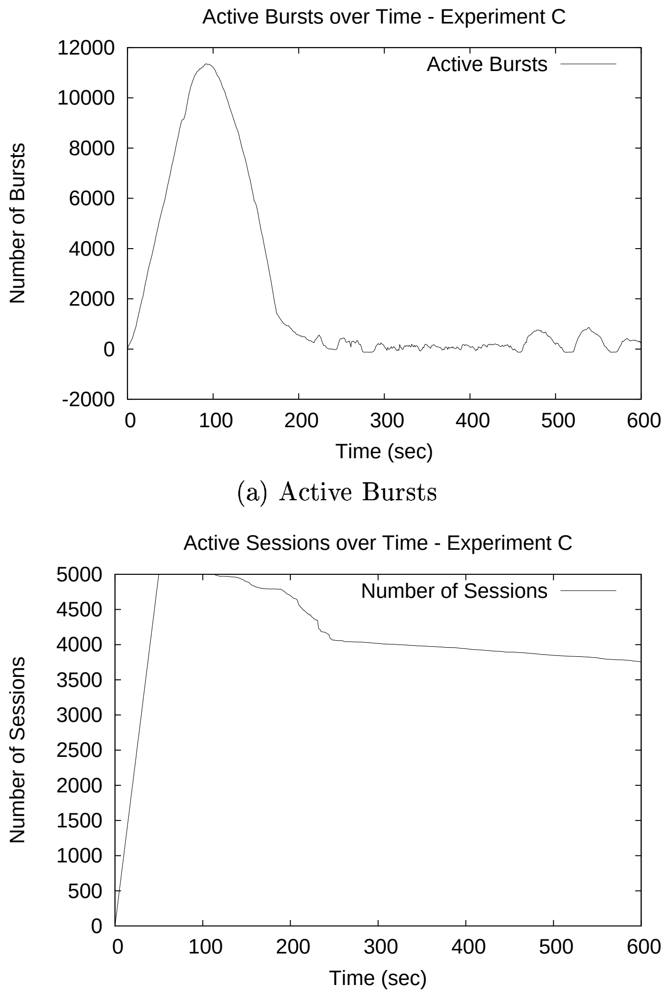


Figure 4.18: Experiment C - Reactive - Bursts



(a) Active Bursts

(b) Active Sessions

Figure 4.19: Experiment C - Reactive

The non-reactive and reactive experiments present a very different scenarios, the first one has caused a heavy overload in the server, that keeps it unavailable for most part of the users. The reactive one has overloaded the server, but the reaction of users to unacceptable response time values changes their global behavior, allowing the server to save resources and returns to acceptable response times after this.

Analyzing the two versions of workloads, the non-reactive and reactive ones, we observe that they cause different situations in all scenarios. This result is interesting, once it can be the base for research in QoS techniques that can consider the influence of user reaction in the performance of servers.

4.2.4 Discussion

We have analyzed the impact of applying workloads with reactive aspects on the performance of Web applications. In order to generate these workloads we have designed new versions of *httperf* workload generator to create user sessions with dynamic inter-arrival time. We have developed two *httperf* implementations: the non-blocking non-reactive and the non-blocking reactive versions. This last one is based on the information obtained by the application of the *USAR* model [94]. The new versions of *httperf* are used to generate workloads based on the TPC-W benchmark in a experimental study. We evaluate the performance of a Web application using three distinct scenarios of load, focusing on the analysis of the following metrics: throughput, response time, active bursts and active sessions for each scenario.

The results of our experiments show the changes in the performance of the Web server when the workload with reactivity is applied to it. The graphics show changes both in the values of each variable in time as in the form of each curve. Both the throughput and response time values have significant variations.

Analyzing the results we can conclude that the actual models of workload generation are limited since they do not consider the user reactivity as an element of the workload model. This indicates that models should be improved. In the same way, the actual models of Web server performance have to be improved since they do not consider the user reactivity in its models also.

Moreover, the results demonstrate how important is to consider the correlation between user and server sides, once it can decrease the gap between the real and model scenarios. In the experiment A, users demanded a higher throughput from the Web server, that answers them with a very good response time. Experiment B presents another case, where the reactivity raises the throughput and the response time of the system, but it has achieved this in a balanced way. Experiment C has shown an interesting situation for the non-reactivity scenario, where a heavy workload has broken down the system. Adopting traditional workload generation mechanisms, the unavailability of the system was an expected situation, once the changing in the way users react is not considered. Our new model has presented a completely different result, demonstrating the importance of understanding better the user-server interactivity process. These results can suggest new improvements in the overload control strategies.

Therefore we are going to study and propose new strategies to improve the performance of Internet services considering the results of our experiments. We are going to develop new strategies of admission control and request scheduling considering reactivity, that is, based on the user characteristics of reaction to the quality of service provided, that will be described in the next section.

4.3 Applying Reactivity in QoS Control Mechanisms

Several new applications have very different requirements from those that the Internet was originally designed for. One issue is performance assurance. When users try to reach a Web site, for example, the server may be so busy that they experience poor performance. Quality of service (QoS) techniques may be used to address this issue.

Internet applications, although contend for system resources, must fulfill QoS requirements, such as accurate timing, reliability, security constraints, as well as other application-specific requirements. In order to satisfy application demands, many QoS mechanisms have been proposed, becoming a topic of great interest to several research areas. Despite this, aspects related to the user behavior are not completely understood, mainly the user response to the quality of service provided by the service.

In this part of the work we are interested in understanding how the reaction of users to QoS measures such as response time may be exploited for sake of designing novel and more effective QoS strategies. Previous work has identified and quantified the impact of user behavior on the performance of Internet services [99], showing that the performance provided by the server affects reciprocally the user-side behavior, in an iterative fashion.

These evidences suggest that effective QoS strategies and mechanisms should take into consideration the user behavior.

We propose and analyze the use of admission control and scheduling techniques based on user reactivity to guarantee QoS in Internet services for workloads with different characteristics. We use the *USAR-QoS* simulator that allows the evaluation of the proposed QoS strategies considering the dynamic interaction between client and server sides in an Internet service scenario. The experiments show the gains obtained by the reactive strategies, also presenting a hybrid strategy that considers reactivity, combining both admission control and scheduling.

4.3.1 Quality of Service and Reactivity

As individuals and organizations increasingly rely on Internet services for their daily activities, it becomes crucial to guarantee that these services are reliable, scalable, robust, trustworthy, and secure, that is, they provide QoS guarantees.

Quality of service (QoS) has been receiving wide attention in the recent years in many research communities including networking, multimedia systems, real-time systems and distributed systems. These systems have a requirement that applications contending for system resources must satisfy timing, reliability and security constraints as well as application-specific quality requirements. These requirements demand QoS guarantees.

Because the Internet treats all packets the same way, it can only offer a single level of service. The applications, however, have diverse requirements. Interactive applications such as Internet telephony are sensitive to latency and packet losses. When the latency or the loss rate exceeds certain levels, these applications become literately unusable. In contrast, a file transfer can tolerate a fair amount of delay and losses without much degradation of perceived performance. Customer requirements also vary depending on what the Internet is used for. For example, organizations that use the Internet for bank transactions or for control of industrial equipment are probably willing to pay more to receive preferential treatment for their traffic. For many service providers, providing multiple levels of services to meet different customer requirements is vital for the success of their business. The capability to provide resource assurance and service differentiation in a network is often referred to as quality of service (QoS). Resource assurance is critical for many new Internet applications to flourish and prosper. The Internet will become a truly multi-service network only when service differentiation can be supported. Implementing these QoS capabilities in the Internet has been one of the toughest challenges in its evolution, touching on almost all aspects of Internet technologies and requiring changes to the basic architecture of the Internet. For more than a decade the Internet community has made continuous efforts to address this issue and has developed a number of new technologies for enhancing the Internet with QoS capabilities.

There are essentially two ways to provide QoS guarantees. The first is simply to provide resources enough to meet the expected peak demand plus a substantial safety margin. This is simple but some people believe it is expensive in practice, and is not able to handle peak demands that increase faster than predicted, since deploying the extra resources takes time.

The second one is to employ QoS control techniques, such as admission control, scheduling, and load balancing. These techniques can guarantee that QoS measures are below or within specified bounds, according to the QoS policy adopted. One example of such technique is admission control.

An effective way to provide QoS is to employ control techniques, such as admission control, scheduling, and load balancing. These techniques can guarantee that QoS measures are below within specified bounds, according to the policy adopted.

The impact of reactive workloads on the performance of Internet services is discussed in [98], where we adopt the *USAR* model to evaluate the performance of a Web application and present a new version of the workload generator *httperf* [83], capable of reproducing the user reactivity. We performed experiments using an actual Web server using three distinct scenarios of load, and analyzing the following metrics: server throughput, response time, rate of submitted requests and number of active sessions. The results show that reactive workloads behave differently when compared to non-reactive workloads submitted to the Web server. The experiments demonstrate that different response times affect the workload, changing the rate of requests submitted and, consequently, changing the load of the server. Therefore, it is shown that server-side affects the client-side behavior, and vice-versa, proving the importance of the reactivity.

The main consequence of demonstrating the fact that reactivity impacts the server performance is that the actual models of Web server performance may be improved since they do not consider it. This motivates the investigation of new QoS strategies that may even mitigate negative effects of the reactivity and reinforce the positive ones.

We present new QoS strategies that consider reactivity. First it is important to introduce the concept of burst, since our policies are based on it. Bursts consist of sequences of requests for fetching a web page and its embedded objects (like pictures). A burst is submitted to the server when a user clicks on a link or requests a Web page during its session. Bursts mimic the typical browser behavior where a click causes the browser to first request the selected Web object and then its embedded objects. Burst is a synonym for the term action we referred on the *USAR* model. A session consists of a sequence of bursts in which the time between any two consecutive bursts is below a certain threshold.

The proposed strategies are based on admission control and scheduling strategies. The basic idea is that bursts must be classified into user reaction classes, based on the reactivity model we described that establishes seven user classes from A to G. The main novelty of this strategy is the use of the reactivity to improve QoS and the idea to combine the proposed admission control and scheduling techniques. The next sections present our workload generator and Web environment simulator and each QoS control strategy with experimental results.

4.3.2 *USAR-QoS*: Workload Generator and Web Environment Simulator

We design and implement a simulator in order to have a easier way to validate our experiments and the QoS control strategies. The main reason is that with simulation we avoid many environmental conditions that can impact negatively the results and obtain results easier and faster.

We implemented a simulator named *USAR-QoS* using the Simpack Toolkit [48], a complete simulation environment, based on C++, that provides simulation routines, random number generators, queuing algorithm and other resources.

The *USAR-QoS* simulates the same scenario presented in [99, 100] to evaluate the impact of the reactivity. The architecture of it USAR-QoS is event-driven and mimics a complete Web system, consisting of the workload generator that supports reactivity and the web application environment. We use a synthetic workload as input for the simulation model. Although synthetic, the workload can be considered representative, because it was generated following the TPC-W benchmark[1, 78]. The server is composed of one or more queues and a processing unit with a certain limited processing capacity. The clients behavior is based on the reactive version of the *httperf* workload generator presented in [101]. The Figure 4.20 presents a state diagram describing how the simulation works.

The *USAR-QoS*' architecture is based on events and respects modularity. The simulation events can be briefly described as:

- *EvntSource()*: starts the execution of *USAR-QoS*, loading the workload parameters necessary for the execution and the QoS policies classes to be used by the execution. After its execution it schedules the event *Sess_Create* for each session to be created.
- *Sess_Create()*: represents the creation of a new session. It schedules *Call_Send_Start*.
- *Call_Send_Start()*: represents the submission of a burst to the server. Schedules the events *Call_TimeOut* and *Sess_Admission_Control*.
- *Sess_Admission_Control()*: verifies if the current session must be accepted to be served by the server according to the session admission control policy being executed. If the session is accepted, the event *Burst_Admission_Control* is scheduled. Else, the session is finalized through the execution of the event *Sess_Destroy*.
- *Burst_Admission_Control()*: verifies if the current session burst must be accepted to be served by the server according to the burst admission control policy executed. If

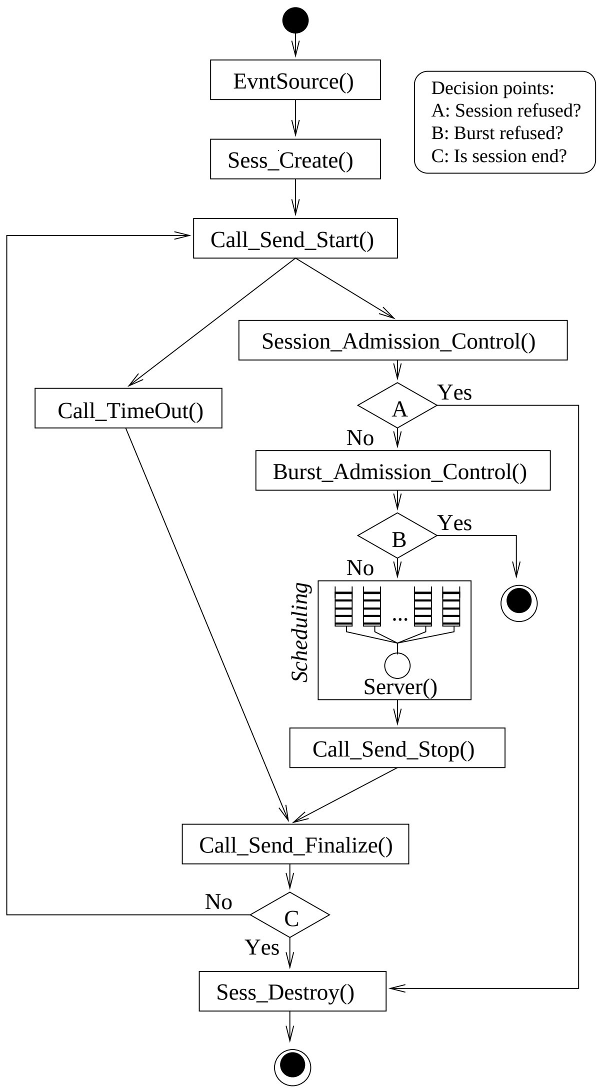


Figure 4.20: State diagram representing the events of the *USAR-QoS* simulator

accepted, the event *Server* is scheduled. Else, nothing should be done since the event *Call_TimeOut* will be executed later for that burst.

- *Server()*: represents the server and its queues. It is the component where the scheduling policy to be evaluated is executed. After execution it schedules *Call_Send_Stop*.
- *Call_Send_Stop()*: finishes the execution of a burst. Schedules *Call_Send_Finalize* to process the next burst.
- *Call_Send_Finalize()*: this event is executed once for each burst. It is scheduled by

the event *Call_Send_Stop* or by *Call_Timeout*, according to the client behavior. This mechanism guarantees that each burst is correctly finalized. For the last burst of a session, if all the other bursts have already been executed, this event is responsible for scheduling the event *Sess_Destroy*. If there are bursts to be executed, *Call_Send_Start* is scheduled.

- *Call_Timeout()*: used to control the time to wait for the answer of a requested burst. If the reply of the corresponding burst has not arrived when this event is executed it schedules the event *Call_Send_Finalize* to process the next burst. During periods of server performance degradation, this event simulates the user impatience, being triggered instead of the *Call_Send_Stop*.

- *Sess_Destroy()*: executed when a session must be destroyed. Finishes the execution of the simulator when all sessions have already been executed.

The *USAR-QoS* was prepared to record the following information about each simulation:

- Bursts Replied Throughput: the rate of bursts answered by the server at each period of time.

- Bursts Requested Throughput: the rate of bursts the users request to the server at each period of time.

- Bursts Expired Throughput: the rate of bursts that the users request the next one before receiving its response, due to impatience and high response times.

- Response time: user perceived response time, comprising the time interval between the request of a burst and the time the client finishes to receive the response.

- Server Queue Size: number of bursts waiting to be served in the server queues.

- Server Utilization: proportion of time the server is busy.

- Number of Active Sessions: number of sessions active, i.e., sessions that have not finished.
- Cumulative number of sessions: cumulative number of completed sessions.

Information of throughput and response time are recorded also applying a smoothing function named *smooth bezier* that provides a better observation of the overall behavior of data.

It is important to mention that the simulator is built in a modular fashion, allowing its extension with new burst or session admission control policies and/or other features, such as scheduling policies.

4.3.3 Admission Control

Admission control mechanisms reject some requests whenever the arrival rate is too high and it is necessary to maintain an acceptable load in the system. Without admission control, throughput may decrease when the system saturates, since several requests will experience long response times and then be dropped.

Traditionally, server utilization or queue length are criteria used in admission control schemes. For sake of service control, the main objective of the control scheme is to protect the system from overload. As long as the system utilization is below a certain threshold, the response times are also low.

Traditional QoS policies do not consider user reaction. Using the reactive approach, we are able to create new admission control policies more suitable for real scenarios.

4.3.3.1 Our Proposed Approaches

This section presents three approaches for QoS admission control: the burst-based, the session-based and the two-level strategies.

Burst-Based Approach

The burst-based admission control rejects bursts of requests when the policy identifies a fulfilled rejection rule. Traditional policies adopt just one limit to response time and start to reject all bursts of requests once this limit is achieved. These policies do not consider any information associated with the user.

Using the *USAR* model, we propose a new policy that considers the way users tend to react according to Internet service's server performance. We define three possible policies as a function of the response time of a service (R):

- $\alpha \leq R < \beta$: reject bursts of user action classes A, B and C.
- $\beta \leq R < \theta$: reject bursts of user action classes A, B, C and D.
- $R \geq \theta$: reject bursts of all user action classes.

In these rules, α, β and θ are response time values determined based on empirical results and literature criteria [75].

This policy is based on the idea that users who have more impatient profile will react faster (that is, reload or submit a different request) than other users when the server presents high response times, degrading server's performance. Considering this, the policy has a multiple criteria rule, minimizing the rejection impact, once less users may have bursts of requests refused from the QoS admission control policy.

In summary, this policy has the premise that, under overload scenarios, it may be better to give priority to users who have more chance to wait for the response to their requests, minimizing the number of unhappy users.

Session-Based Approach

The session-based admission control rejects user sessions when the policy identifies a rejection rule. Traditional policies employ a single response time threshold and start to reject all user sessions once this limit is reached. Similar to traditional burst-based policies, the session-based ones do not check any information associated with the user session.

The burst-based admission control policy may affect all users, but the session-based policy is different, since the trend is to affect fewer users. Considering this, we expect that reactivity will be very important to identify which user sessions have to be dropped.

Using the *USAR* model we propose a new admission control policy that considers how users tend to react according to Internet service's server performance. We monitor the average response time (R) and the user session profile for each session (USP), i.e., the average user class of each burst of his/her session that have already been served. We define the following three criteria:

- $\alpha \leq R < \beta$ and $USP < \sigma_1$: reject user sessions with more impatient profile ($\sigma_1 < 4$ - average user action classes of A, B and C.

- $\beta \leq R < \theta$ and $USP < \sigma_2$: reject user sessions with a balanced profile (nor non-impatient neither patient) - $\sigma_2 < 5$ - average user action classes of A, B C, and D.

- $R \geq \theta$: reject all user sessions.

This policy is based on the same idea presented in the reactive burst-based policy, however applied to session-based control mechanism. For both approaches the idea consists of improving the QoS control by choosing the sessions that will produce the minimum impact on the user satisfaction.

Two-Level Approach

We propose a new approach - a two-level one - which presents a hybrid mechanism that adopt both burst and session-based strategies. The idea of this new admission control policy is to put together the advantages of each strategy in a balanced approach. As occurs in the session-based approach, the rejection of sessions is drastically started as response time grows. In the two-level approach, first of all, burst rejection is started, before the rejection of sessions. This strategy smooths the session rejection through a previous step.

Once the burst rejection is not effective to slow down the response time session rejection is activated. We define the following criteria:

- $\alpha_1 \leq R < \beta_1$: reject bursts of user action classes A, B and C.
- $\beta_1 \leq R < \theta_1$: reject bursts of user action classes A, B, C and D.
- $R \geq \theta_1$: reject bursts of all user action classes.
- $\alpha_2 \leq R < \beta_2$ and $USP < \sigma_1$: reject user sessions with more impatient profile ($\sigma_1 < 4$ - average user action classes of A, B and C.
- $\beta_2 \leq R < \theta_2$ and $USP < \sigma_2$: reject user sessions with a balanced profile (nor non-impatient neither patient) - $\sigma_2 < 5$ - average user action classes of A, B C, and D.
- $R \geq \theta_2$: reject all user sessions.

This policy rejects both bursts and sessions, but according to different limit values. This makes possible to balance the rejection of bursts and sessions, avoiding the rejection of sessions when the response time is not up to the defined limits but trying already to slow down the response time through the rejection of bursts.

4.3.3.2 Experimental Methodology

This section explains how we evaluate the new policies proposed.

The main objective of the experimental evaluation is to analyze the behavior of the reactive QoS policies in the performance of Internet servers under reactive workloads. In order to do this, we have implemented the admission control policies in the *USAR-QoS*, evaluating the following important metrics:

- Throughput: we measure both throughput of requested and replied bursts (requests) in the point of view of the server. The requested bursts throughput represents the rate in which requests arrive at the server. The replied throughput represents the rate in which requests are answered by the server.

- Response time: we refer to the user perceived response time, comprising the time interval between the submission of the request and the time when the client finishes to receive the response.

- Number of active bursts and sessions: this information captures the number of bursts requested to the server but not yet responded, for each period of time, or the number of sessions initiated but not yet finished, respectively.

- Server queue size: represents the number of bursts of requests waiting to be served.

- Server utilization: represents the proportion of time the server is busy.

- Cumulative number of sessions: number of completed or rejected sessions (admission control)

- Bursts rejected and expired-bursts: we identify the number of bursts rejected by admission control and the number of bursts from which their response time are greater than the time one user waits for it at that moment.

As explained in Section 4.3.1, each reactive QoS policy has a set of values (α, β, and θ) that define the admission control functioning. These values must be carefully chosen since the effectiveness of each policy depends on this factor. We adopt 10-second as the time limit when a user loses attention and the interaction with the system is disrupted [75]. We use the following values: $\alpha = 6.0$, $\beta = 7.5$, and $\theta = 9.0$ for burst and session-based policies; and $\alpha_1 = \alpha_2 = 6.0$, $\beta_1 = \beta_2 = 7.5$, and $\theta_1 = \theta_2 = 9.0$, for the two-level approach.

We simulate many scenarios using *USAR-QoS* to observe how the application server behaves over different workloads. We present three scenarios in this case study that represent load situations for different types of workload, where QoS strategies are necessary. For each scenario we obtained a TCP-W-based synthetic workload with 5000 sessions with an average rate of 1 session initiated per second, varying uniformly from 0.5 to 1.5. It is important to emphasize that the number of simultaneous users are defined by the number of active sessions over the experimental time. Each workload has a different set up of

Workload	A	B	C	D	E	F	G
1	5%	10%	10%	15%	15%	15%	30%
2	22%	15%	10%	6%	10%	15%	22%
3	30%	15%	15%	15%	10%	10%	5%

Table 4.5: User reaction classes distribution for each workload

user reaction classes (presented in Section 4.1.1) in order to verify the effectiveness of each policy under different conditions. Table 4.5 shows the distribution of user reaction classes for each scenario.

Workload 1 has predominance of patient classes, once 60% of it consists of classes E, F and G. The second one has a balanced distribution over patient and impatient scale, presenting a similar distribution of classes A and G, B and F, C and E. And workload 3 represents an impatient profile with 60% of classes A, B and C.

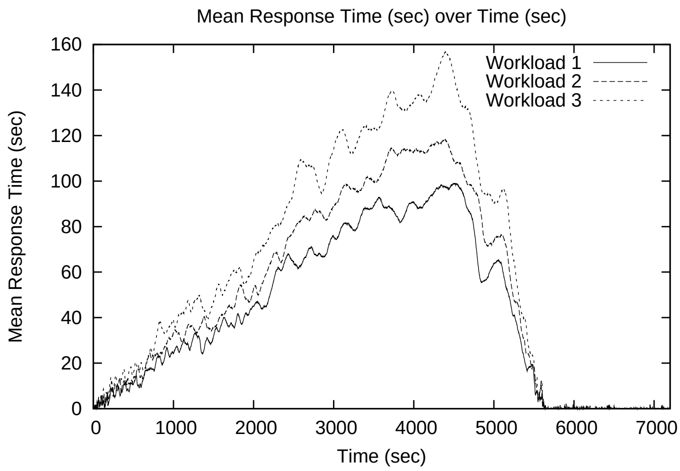


Figure 4.21: No Admission Control - Average Response Time

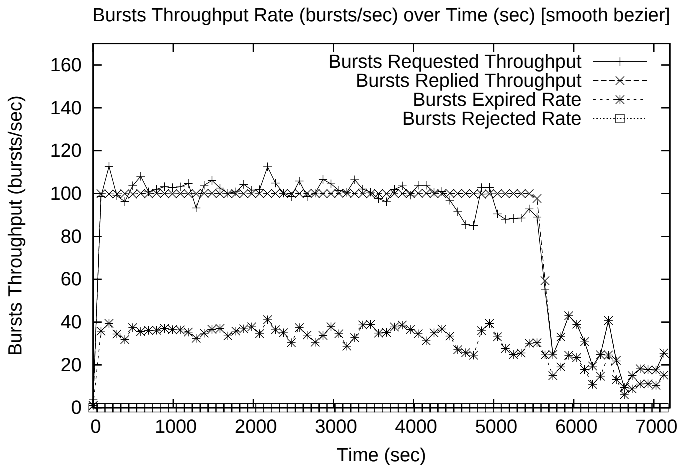


Figure 4.22: No Admission Control - Average Throughput - Workload 1

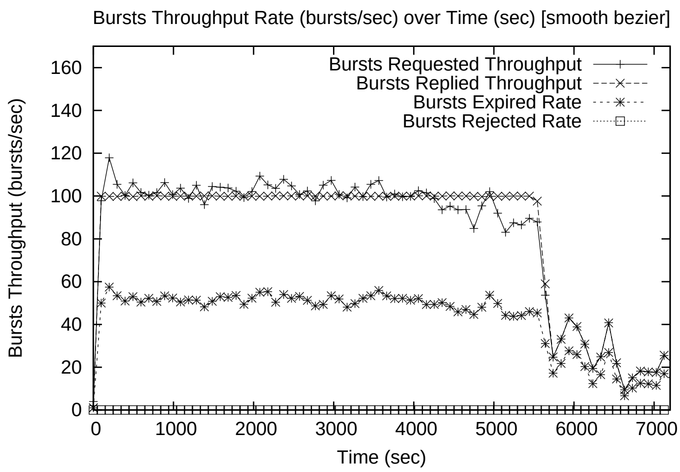


Figure 4.23: No Admission Control - Average Throughput - Workload 2

4.3.3.3 Experimental Results

We notice the differences between each workload setup when we simulate their behavior. Figures 4.21, 4.22, 4.23 and 4.24 show experimental simulations where no admission control policy is active in the simulator. Figure 4.21 shows the average response time and Figures 4.22, 4.23 and 4.24), the average throughputs. As we can observe, workload 3 applies the heaviest load to the server, since its mean response time is greater most of the time than the other two workloads and its average throughput of expired bursts is the biggest one. Workload 1, for instance, presents an intensive load to the server but not

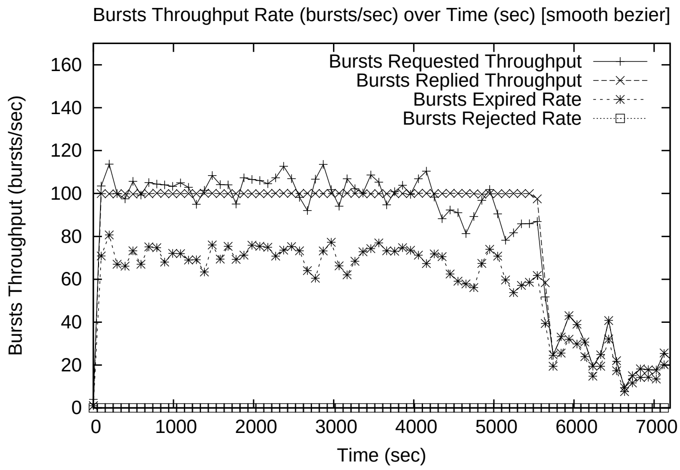


Figure 4.24: No Admission Control - Average Throughput - Workload 3

so heavy as the other two workloads. It is important to notice that the average response throughput is the same for each workload since it is dependent of the server capacity that has a limit value, achieved most of the time.

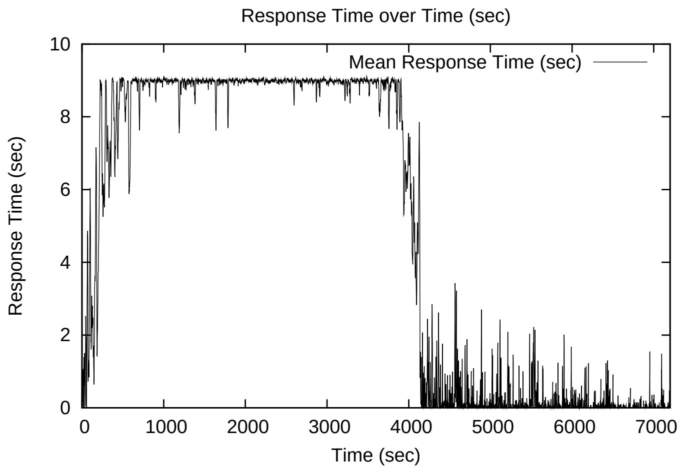
We study the behavior of each policies for each workload configuration as we show in the next subsections.

Burst-Based Admission Control

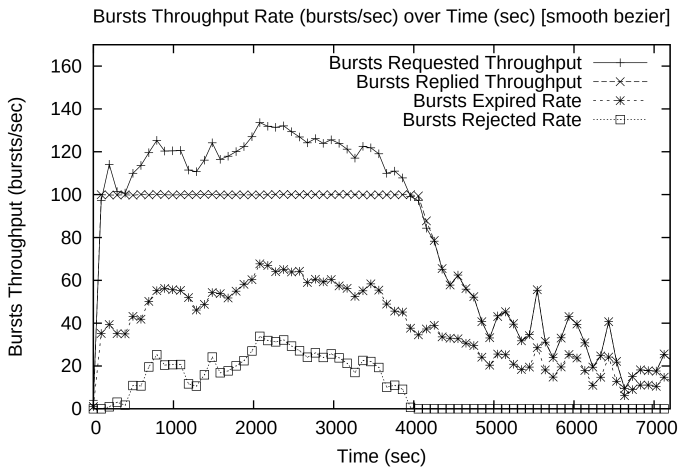
Using the workload scenarios described on the previous session we evaluate the effectiveness of the burst-based admission control approaches.

Figure 4.25 shows a scenario where the non-reactive admission control was activated. In (a) we see the average response time and we observe an upper-bound of 9 seconds, once the admission control is set to be activated at this limit. This upper-bound could be set to a larger number, however we decide to use this because the literature adopts 10-second as the limit of user tolerance to response time [75]. In (b) we have the average throughput.

In Figure 4.26 we observe the system behavior in a reactive burst admission control scenario. In (a) we see that the average response time grows and then established after the activation of the reactive control admission policy ranging from 4 to 7.5 seconds. In (b) we see clearly that the rate of bursts rejected by the admission control elevates as the average response time raises.



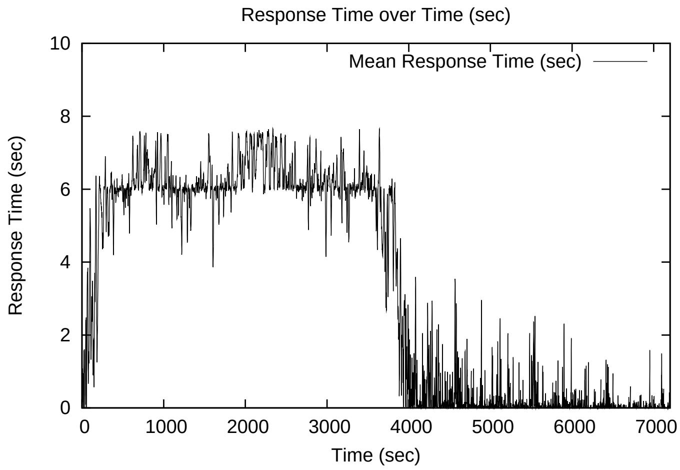
(a) Average Response Time



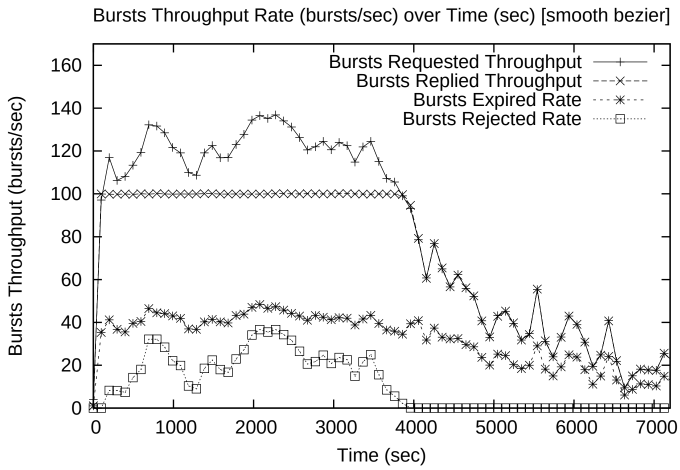
(b) Average Throughput

Figure 4.25: Non-Reactive Burst-Based Admission Control (Workload 1)

Compared to the approach without admission control, the experiment with reactive admission control policy shows a decrease of 40% to 90% in the average response time.



(a) Average Response Time

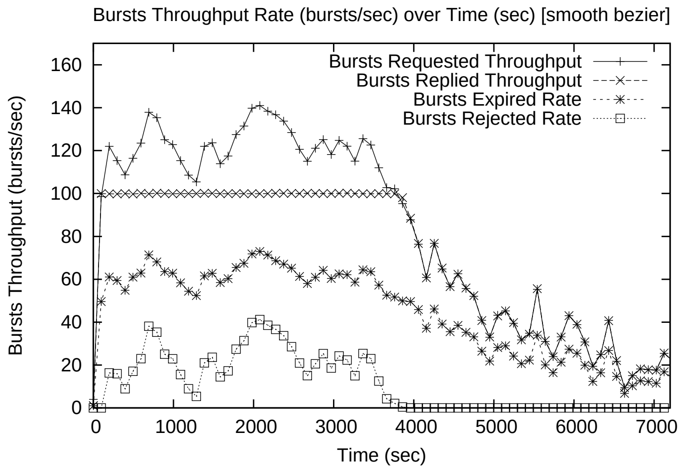


(b) Average Throughput

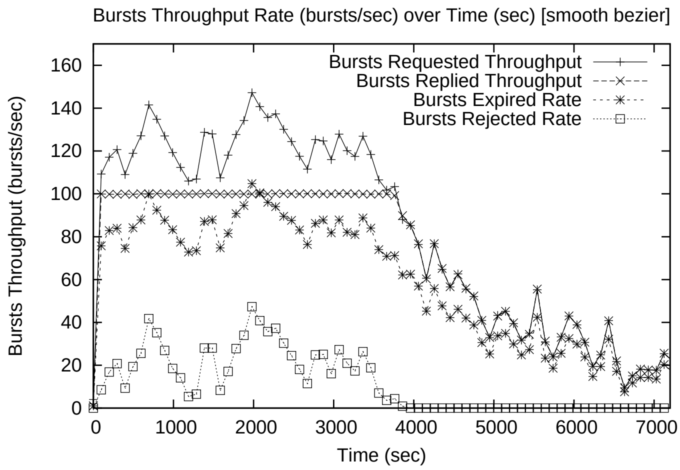
Figure 4.26: Reactive Burst-Based Admission Control (Workload 1)

Comparing the two approaches that implements admission control, we identify that the traditional policy may be directly applied to the reactive workload. Otherwise, the reactive admission control achieves a better result, once it keeps the user satisfaction and reduces the average response time significantly (more than 20%).

A similar behavior is observed for workload scenarios 2 and 3. Figure 4.27 shows throughput values obtained for each experiment. In (a), we see that for workload 2, the rate of bursts rejected varies from 0% to 40% during the server's overload. In (b), for workload 3, the rate of bursts rejected varies from 0% to 50% during the server's overload.



(a) Average Throughput (Workload 2)



(b) Average Throughput (Workload 3)

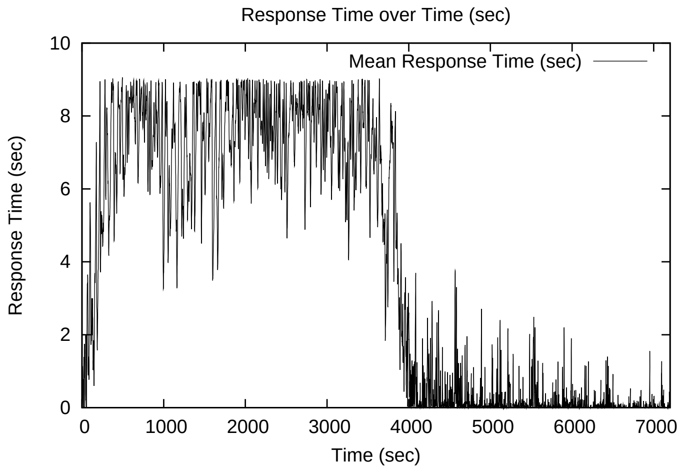
Figure 4.27: Reactive Burst-Based Admission Control (Workloads 2 and 3)

The user satisfaction rate has decreased as can be observed by the increase of burst expired rate in the picture.

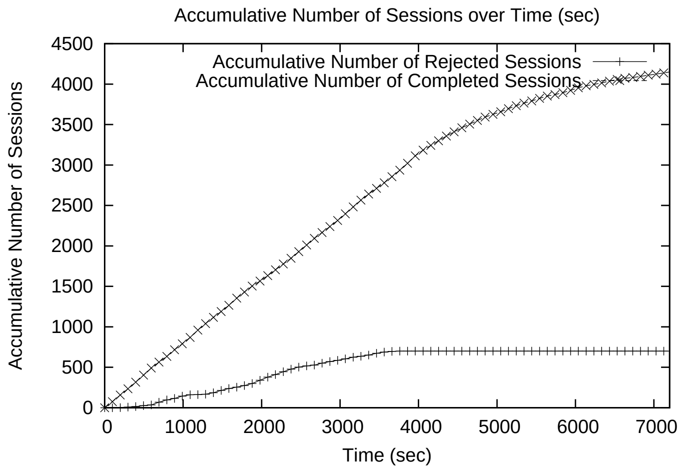
Comparing to the approach without admission control, the experiment with reactive admission control policy shows a significant improvement (from 95%) in the average response time. Comparing the two approaches that implement admission control, we identify that the reactive admission control achieves a better result again, once it reduces the average response time in more than 30% with a little better user satisfaction rate.

Session-Based Admission Control

This section shows the results for the experiments using the session-based admission control policies implemented in the *USAR-QoS* for the same workload scenarios. As the main conclusions for both workloads are the same, we show only the results for workload 1.

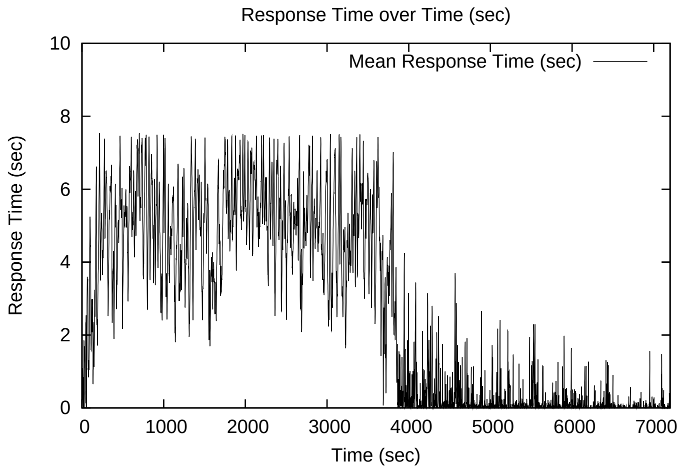


(a) Average Response Time

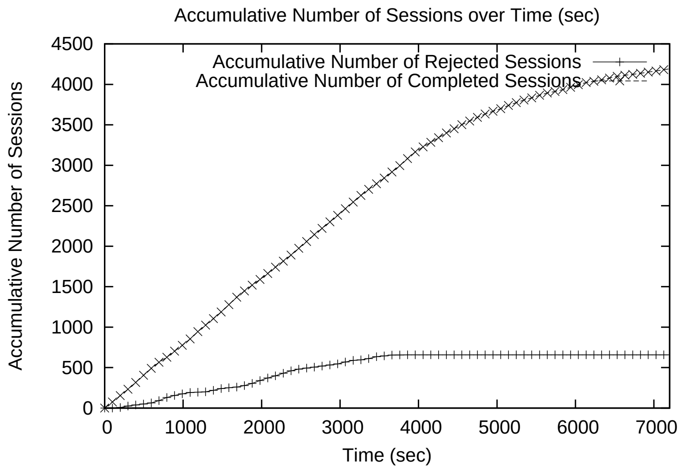


(b) Accumulative number of sessions

Figure 4.28: Non-Reactive Session-Based Admission Control (Workload 1)



(a) Average Response Time



(b) Accumulative number of sessions

Figure 4.29: Reactive Session-Based Admission Control (Workload 1)

Figure 4.28 represents the results for system with non-reactive session admission control, where (a) is the average response time and (b) is the number of sessions. The average response time varies over time, from 3,5 to 9 seconds after established. The amount of unsatisfied users is similar to the experiment without admission control. Figure 4.29 represents the results for system with reactive session admission control. The average response time varies across time, ranging from 2 to 7,5 seconds after established. Again the amount of unsatisfied users is about the same of the experiment without admission control.

We can notice that there is a trade-off here as the reactive admission control based on session has achieved a better result than the non-reactive one, reducing the average response time in at most 20%. Observing the accumulative number of sessions we can see that reactive session-based admission has rejected 10% more sessions than the non-reactive one, but it is interesting to note that this difference does not affect the user satisfaction. The server utilization during the observed time has been maximum, as expected.

Two-Level Admission Control

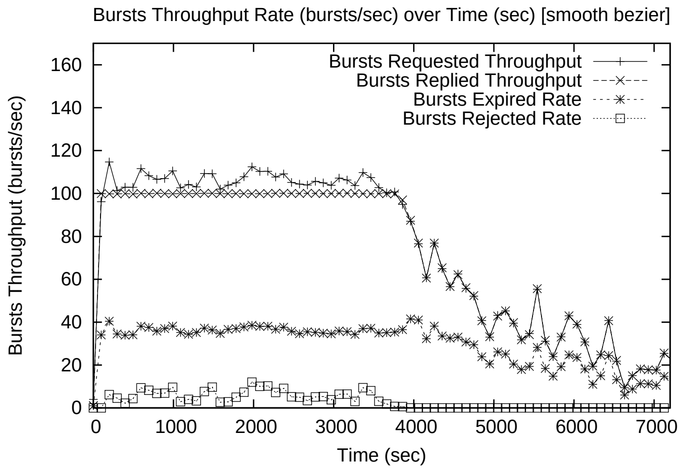
We implemented the two-level approach in the *USAR-QoS* and applied it over the same workload scenarios.

Figure 4.30 and 4.31 represent the experiment with two-level admission control policy. In Figure 4.30, (a) is the throughput and (b) is the accumulative number of sessions. Comparing it to the experiment with reactive session admission control, there is a low upper-limit value in the average response time, showing the combined policies present an interesting behavior comparing to the other policies presented. The number of rejected sessions as can be seen in (b) is lower than the values for the experiment with just the session-based approach.

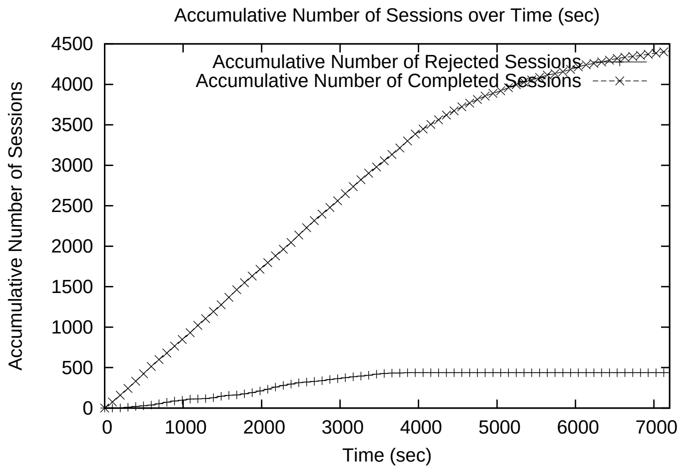
Figure 4.31 shows the average response time, that varies over time, from 2 to 6,5 seconds during server overload. This result is better than the one obtained from the session-based approach and achieves more than 30% of reduction in the response time of the non-reactive policy.

4.3.3.4 Discussion

The results show that the premises adopted to model the reactive admission control policies are effective, proving that the idea of reactive admission control minimizes the overload



(a) Average Throughput



(b) Accumulative number of sessions

Figure 4.30: Reactive Two-Level Admission Control (Workload 1)

impact. We can observe this improvement by the significant reduction in the average response time perceived by users. On the other side, the number of expired bursts of requests has reduced in the reactive policy scenario, what adds more value to the new approach proposed in this work.

The reactive admission control policies proposed in this work lead to better response time rates, preserving the user satisfaction metric. The reduction varies from 15 to 50%, with an average of 30%.

From the results we can conclude that there is a relevant improvement in the QoS of

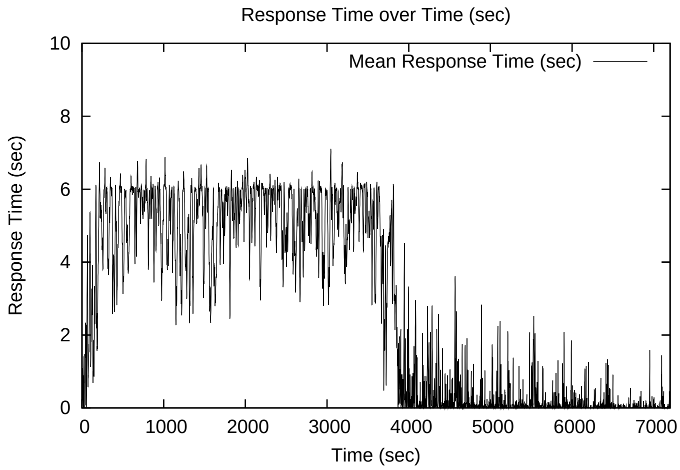


Figure 4.31: Reactive Two-Level Admission Control (Workload 1)

the reactive Internet systems through the use of reactive admission control policies. As part of ongoing work, we plan to evaluate the use of reactivity in scheduling policies and other QoS strategies.

4.3.4 Scheduling

In order to improve the QoS of web applications, reducing the impatience of users, we propose new scheduling techniques based on user reactivity. We propose two different approaches that give priority to users of different profiles of reactive behavior: the Patient-First Impatient-Next (PFIN) and the Impatient-First Patient-Next (IFPN), both with two different configurations. We evaluate them using the *USAR-QoS* simulator that allows the simulation of the proposed QoS strategies considering the dynamic interaction between client and server sides in an Internet service scenario. For sake of reproducing representative workloads, our experiments use the TPC-W-based reference benchmark. The results show the benefits obtained by the reactive scheduling approach that is effective to reduce de bursts expired rates due to the impatient behavior.

The next subsections present our proposed approaches for scheduling, the experimental methodology and results, and then the conclusions.

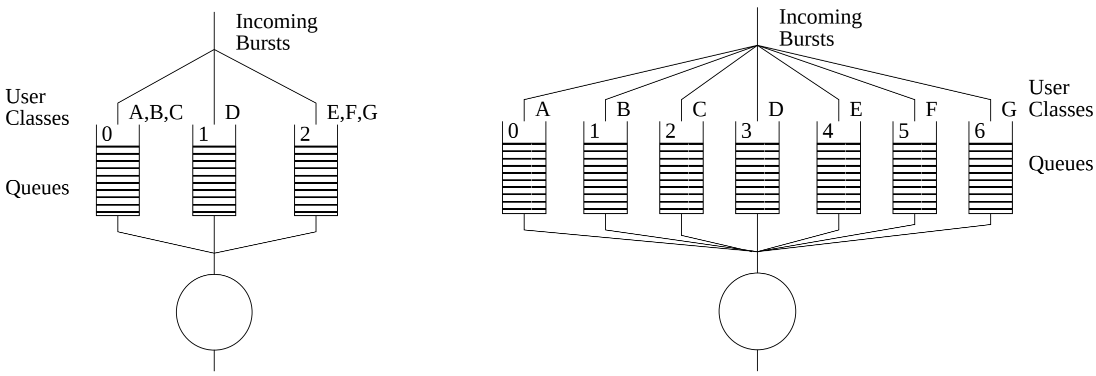
4.3.4.1 Our Proposed Approaches

In overload scenarios, web servers receives numerous requests for its services, which they cannot handle at the same time. They typically use a buffer or queue to store incoming requests awaiting for service. Requests in the queue are typically stored in the order of arrival. The web server will take the request at the front of the queue, and service it first. This is an example of First-In First-Out (FIFO) scheduling. Most existing web servers provide services based on the Best Effort FIFO scheduling [120].

There are works that propose different approaches to improve the basic FIFO approach. They present gains compared to the Best Effort approach, but they fail to consider the reactivity, an important dimension of the user interaction with the service. Using the reactive approach derived from the *USAR* model, we are able to create new policies more suitable for real scenarios.

First, it is important to recall the concepts of burst and session. Bursts consist of a request and its embedded objects for fetching a web page. A burst is submitted to the server when a user clicks on a link or requests a Web page. Bursts mimic the typical browser behavior where a click causes the browser to first request the selected Web object and then its embedded requests. A session consists of a sequence of bursts in which the time between any two consecutive bursts is below a certain threshold.

In the next subsections we present new scheduling policies based on the reactivity. The basic idea is that bursts must be classified using the *USAR* model into classes that must be placed in priority queues. The server will process the high priority queues before serving any of the low priority ones. The classification criteria is based on the reactivity model we have already described, that establishes seven user classes from A to G, used to classify each burst. The differences between each policy is the mechanism used to decide in which queue each request should be placed when the reactive scheduling mechanism is active.



(a) Three Priority Queues (b) Seven Priority Queues

Figure 4.32: Servers implementing PFIN Scheduling Approach

Patient-First Impatient-Next Scheduling Approach (PFIN)

In Patient-First Impatient-Next (PFIN) approach, requests classified with impatient classes (A, B, and C) go to the low priority queues, and those with patient classes go to the high priority ones.

Figure 4.32 presents two variations of the PFIN approach, which differs in terms of the number of priority queues used to schedule the requests. In (a) there are three queues and all the bursts identified with impatient classes are scheduled to the low priority queue, and those with patient to the high priority one. Bursts identified with classes A, B or C go to queue 0, with D go to queue 1, and finally, those classified with classes E, F and G go to queue 2. In (b) we employ seven queues, one for each user class. The most patient classes get the high priority queues. Bursts classified as class A go to queue 0, the ones with class B to queue 1, C to queue 2, D to queue 3, E to 4, F to 5 and G to queue 6.

The PFIN policy is based on the idea that when the load is increasing the users who have more patient profiles tend to present a lower load to the server than the impatient ones since after receiving its response, they take more time to proceed and submit another request. Users with impatient profiles tend to react faster, asking requests even before receiving the previous one.

This policy has the premise that under overload scenarios it may be better to give priority to users who have more chance to spend more time to submit requests, slowing down the server load and increasing the user satisfaction.

Impatient-First Patient-Next Scheduling Approach (IFPN)

In Impatient-First Patient-Next (IFPN) approach, requests classified with impatient classes (A, B, and C) go to the high priority queues, and those with patient classes (E, F, and G) go to the low ones. Like PFIN approach, two variations are proposed with three or seven priority queues.

The IFPN policy is based on the idea that when the load is increasing it is better to answer first the impatient users, in order to increase their satisfaction. The advantage on delaying answers to patient instead of the impatient ones is the fact that the satisfaction of patient users tends to take more time to degrade. In summary, this policy has the premise that it may be better to give less priority to users that have more chance to wait for the response to their requests.

4.3.4.2 Experimental Methodology

We simulate several workload scenarios using *USAR-QoS* to observe how the application server behaves under various loads. Each one is composed of a TPC-W benchmark [1, 78] workload containing different distributions of user action classes. Due to space constraints is not possible to show all the experiments. We present the results for the evaluation of a 5,000 TPC-W-based sessions workload based on the reactive workload generated in [101]. This workload presents a high load to the server as can be observed on the experiments below. The distribution of action classes of the workload is shown on Table 4.6.

A	**B**	**C**	**D**	**E**	**F**	**G**
5%	10%	10%	15%	15%	15%	30%

Table 4.6: User reaction classes distribution for the workload

4.3.4.3 Experimental Results

Using the TPC-W-based workload we simulate the new scheduling policies and the Best Effort FIFO approach, comparing the results. Next we present out main results analysis.

FIFO approach

Figures 4.33 and 4.34, and Table 4.7 present the experimental simulation using the typical Best Effort FIFO scheduling mechanism. The sessions are created in a rate of 10 sessions per second. The Figure presents the bursts throughput (a) and the response time (b). The value of the response time varies up to almost 33 seconds. This occurs due to the high load over the server and the client's behavior. Most of the time the average response time achieves more than 10 seconds. Since this value is typically considered as the upper-limit that is acceptable by the client [75], we can suppose that the server is saturated. The high response times cause the increase of the user impatience. The rate of bursts expired, i.e., the number of requests the user asks the next one before receiving the previews response, achieves a very high value, close to 38%.

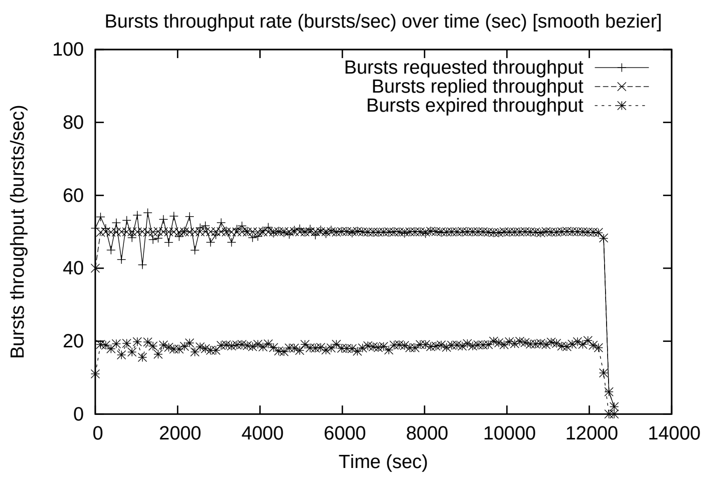


Figure 4.33: Simulation of the Best Effort FIFO scheduling approach - Bursts Throughput

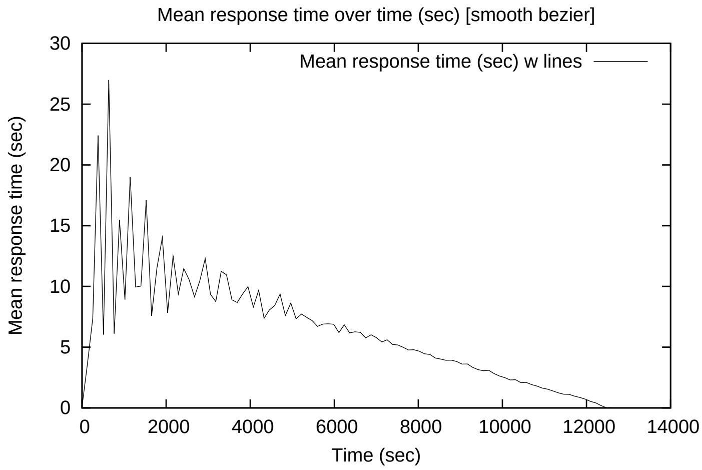


Figure 4.34: Simulation of the Best Effort FIFO scheduling approach - Average Response Time

Duration (sec)	Max R (sec)	Average R (sec)	Cum. Bursts Requested	Cum. Bursts Expired	Bursts Expired Rate
12,840	33.01	6.39	621,342	230,646	37.12%

Table 4.7: Summary for the simulation of the Best Effort FIFO scheduling approach

We simulate the new scheduling policies for the same workload trace, comparing the results with the best effort FIFO case.

PFIN approaches

Tables 4.8, 4.9 and 4.10, and Figures 4.35 and 4.36 present the experiments evaluating the PFIN scheduling approaches with three and seven queues. In the figures we observe the bursts throughput and the average response time, respectively.

For the experiment with three priority queues, the maximum response time achieves the unacceptable value of 84.07 seconds. The average response time is 6.27 seconds. The bursts expired rate is 25.35%, a better result compared to the experiment with the FIFO mechanism from last subsection. These values shows that this experiment presents a good result, since it is effective to keep a good average response time with a lower bursts expired rate.

Scheduling approach	Duration (sec)	Max R (sec)	Average R (sec)	Cum. Bursts Requested	Cum. Bursts Expired	Bursts Expired Rate
3 queues	13,002	84.07	6.27	621,342	157,537	25.35%
7 queues	13,252	338.62	6.15	621,342	100,083	16.11%

Table 4.8: Summary for the simulation of the PFIN scheduling approaches

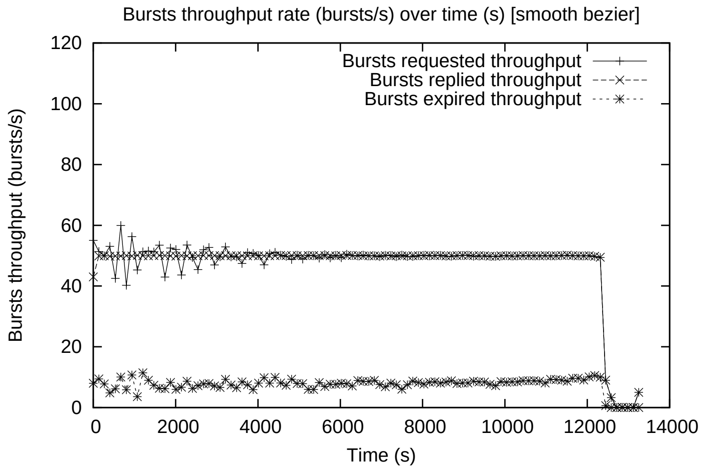


Figure 4.35: Simulation of the PFIN scheduling approach with seven priority queues - Bursts Throughput

Table 4.9 presents the response times and the bursts expired rates for each set of bursts classified according to the action class. As we observe, both measures are bigger for the impatient behavior classes compared to the patient ones. This occurs due to the scheduling mechanism that considers three queues priorizing the patient classes.

For the experiment with seven queues we observe a higher maximum response time than the experiment with three queues, that achieves 338.62 seconds. However, the average value for this measure is 6.15 seconds, an acceptable value according to the literature and subtle lower compared to the experiment with thee queues. The bursts expired rate is 16.11%, a smaller value also compared to the previews experiments. Then, this experiment presents

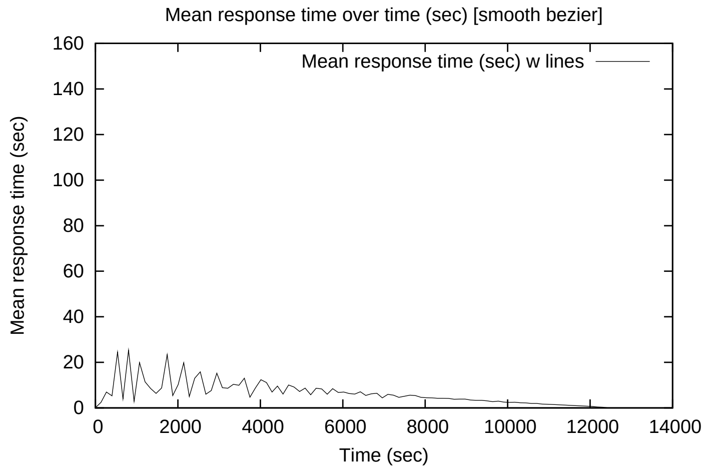


Figure 4.36: Simulation of the PFIN scheduling approach with seven priority queues - Average Response Time

a good result, since it is capable of maintaining a low average response time, with a low expiration rate.

Investigating the reason for the observed behavior, we present Table 4.10 that shows the response time and the bursts expired rate for each action class. For classes with more impatient behavior, the response time and bursts expired rates present higher values since the corresponding bursts are directed to lower priority by the scheduling mechanism. The patient classes present lower values.

IFPN approaches

Action classes	Max R (sec)	Average R (sec)	Bursts Expired Rate
All	84.07	6.27	25.35%
A, B e C	84.07	20.23	99.50%
D	20.26	0.44	2.95%
E, F e G	0.79	0.05	0.00%

Table 4.9: Summary for the simulation of the PFIN scheduling approach with three priority queues

Action classes	Max R (sec)	Average R (sec)	Bursts Expired Rate
All	338.62	6.15	16.11%
A	338.62	19.44	99.63%
B	107.55	6.90	76.21%
C	60.68	1.76	30.55%
D	18.19	0.49	2.75%
E	1.90	0.07	0.04%
F	0.52	0.04	0.00%
G	0.22	0.03	0.00%

Table 4.10: Summary for the simulation of the PFIN scheduling approach with seven priority queues

The experiments that evaluates the IFPN scheduling approach with three and seven queues are presented on the tables 4.11, 4.12 and 4.13, and in the Figures 4.37 and 4.38. In the figures we observe the bursts throughput and the average response time, respectively.

Scheduling approach	Duration (sec)	Max R (sec)	Average R (sec)	Cum. Bursts Requested	Cum. Bursts Expired	Bursts Expired Rate
3 queues	13,660	44.37	5.95	621,342	16,500	2.66%
7 queues	13,132	73.21	6.17	621,342	3,412	0.55%

Table 4.11: Summary for the simulation of the IFPN scheduling approaches

The IFPN with three queues experiment presents a maximum and average response time of 44.37 and 5.95 seconds, respectively. The bursts expired rate is 2.66%. These values are lower than the obtained for the experiments with the FIFO and PFIN approaches. Table 4.12 presents the measures for the classes. We may observe a bigger response time for more patient classes once they have lower priority. With this policy, the bursts expired rate for users with impatient behavior is close to 0% because the scheduling mechanism is capable of answering with great efficiency those users, increasing their satisfaction.

Action classes	Max R (sec)	Average R (sec)	Bursts Expired Rate
All	44.37	5.95	2.66%
A, B e C	0.19	0.03	0.00%
D	0.51	0.04	0.02%
E, F e G	44.37	9.72	4.43%

Table 4.12: Summary for the simulation of the IFPN scheduling approach with three priority queues

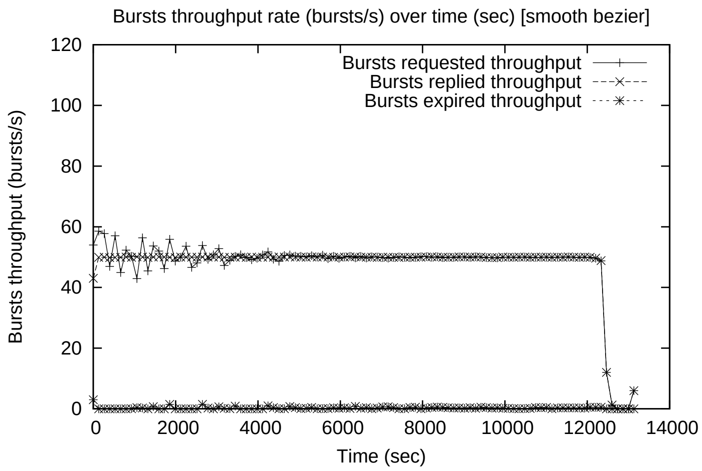


Figure 4.37: Simulation of the IFPN scheduling approach with seven priority queues - Bursts Throughput

The experiment with seven queues presented a response time maximum of 73.21 and average of 6.17 seconds. These values are subtle bigger than the obtained for the experiment with three queues. However, the bursts expired rate achieves very small values, since it is just 0.55%. This means that the number of bursts the user presents a impatient behavior is of just 0.55% of the bursts, what demonstrates the benefits of this scheduling approach. The behavior of each class summarized on table 4.13 demonstrates this. The request rate for all the classes is close to 0%. However the response time is bigger for the patient classes, since they have lower priority. As an effect of the reactivity, the bursts expired rate is not very big because these users have patience behavior tendency, waiting for servers answer.

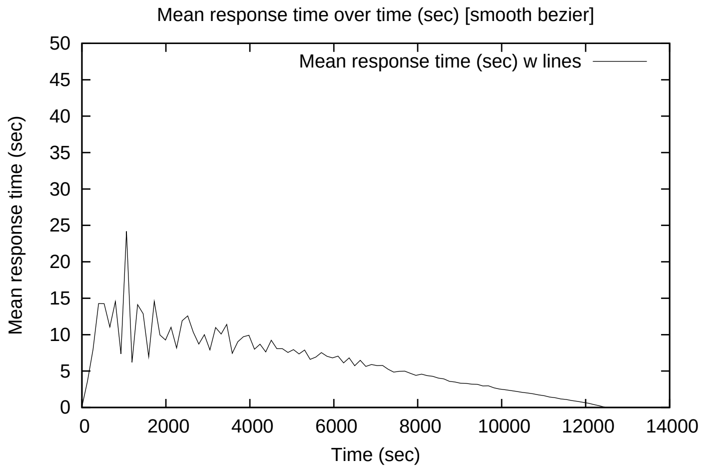


Figure 4.38: Simulation of the IFPN scheduling approach with seven priority queues - Average Response Time

4.3.4.4 Discussion

We have proposed and evaluated scheduling techniques based on the user reactivity to guarantee QoS in Internet services. We presented the PFIN (patient-first impatient-next) and IFPN (impatient-first patient-next) approaches. For each policy we have discussed the use of three or seven priority queues.

Each approach has been evaluated using the *USAR-QoS* simulator running 5,000 TPC-W-based sessions. We evaluate the new policies comparing the results with the Besf Effort FIFO approach scenario.

We observe that the experiments using the novel scheduling strategies present very low bursts expired rates, i.e., the rate of requests the user presents the impatient behavior. This is due to the scheduling mechanism that gives priority to determined behavior. The average response times also present improvements but its maximum value is very high. This occurs because some requests are addressed to the low priority queues.

The proposed policies may be unfair to certain users since they use their typical behavior to improve QoS. The benefits obtained are significant and we recommend our policies despite the subject of unfairness.

We can conclude that there is a relevant improvement in the QoS of reactive Internet

Action classes	Max R (sec)	Average R (sec)	Bursts Expired Rate
All	73.21	6.17	0.55%
A	0.11	0.03	0.00%
B	0.17	0.03	0.00%
C	0.21	0.03	0.00%
D	0.42	0.04	0.01%
E	2.38	0.06	0.02%
F	11.80	0.22	0.02%
G	73.21	17.89	1.80%

Table 4.13: Summary for the simulation of the IFPN scheduling approach with seven priority queues

systems through the use of reactive scheduling policies. As part of ongoing work we are planning to study the use of reactive scheduling and admission control strategies together in order to optimize the QoS control, improving the user satisfaction.

4.3.5 Admission Control and Scheduling

In this section we present a hybrid QoS control mechanism that combines both admission control and scheduling and considers reactivity.

4.3.5.1 Strategy

We propose a hybrid three-level approach, that combines the two admission control strategies and scheduling. The idea of this new approach is to put together the advantages of each one. Admission control is good to avoid raising the response time to unacceptable values [100]. Scheduling is adopted to control the burst's priority according to the user class, providing a reduction in the burst's expiration rate, measure related to user satisfaction.

In the session-based approach, rejection of sessions is drastically started as response time grows. In the two-level approach, first of all, burst rejection is started, before the rejection of sessions. This strategy smooths the session rejection through a previous step.

Once the burst rejection is not effective to slow down the response time, session rejection is activated. In parallel, scheduling produces a burst reordering process that can reduce the number of unsatisfied users. We define the following criteria:

- $\alpha_1 \leq R < \beta_1$: reject bursts of classes A, B and C;
- $\beta_1 \leq R < \theta_1$: reject bursts of classes A, B, C and D;
- $R \geq \theta_1$: reject bursts of all user action classes;
- $\alpha_2 \leq R < \beta_2$: reject user sessions with $USP < 4$, i.e., with average user reaction classes A, B or C;
- $\beta_2 \leq R < \theta_2$: reject user sessions with $USP < 5$, i.e., with average user reaction classes A, B, C, or D;
- $R \geq \theta_2$: reject all user sessions;
- $R > \gamma$: turn on the scheduling policy.

This strategy rejects both bursts and sessions, but according to different limit values, balancing the burst's rejection and providing a way to schedule the bursts in order to raise the user satisfaction.

4.3.5.2 Experimental Methodology

In order to evaluate the effectiveness of the proposed reactive QoS policies, we simulate them using the *USAR-QoS* simulator. We prepare a TPC-W-based synthetic workload trace file to be used as the input for the simulation.

We simulate several scenarios using different *USAR-QoS* configurations to observe how the application server behaves. We discuss in this work a scenario in which the server achieves a high throughput and the response times observed raises over the user satisfaction threshold. We determine this threshold based on [75] where the authors identify three groups regarding the response time of a system:

- 0.1 sec: the limit when a user perceives that the system is reacting instantaneously.
- 1.0 sec: the limit when the flow of thought of a user is not interrupted, although the user may notice the delay.
- 10.0 sec: the limit when a user loses attention and the interaction with the system is disrupted.

Based on these values we implement each of the proposed reactive admission control and scheduling strategies in the *USAR-QoS*. As already explained in Section 4.3.3, each reactive QoS policy has a set of values (α, β, θ and γ) that define its functioning. These values should be carefully chosen since the effectiveness of each policy depends on them. The values we choose are: $\alpha_1 = 3.0$, $\beta_1 = 5.0$, $\theta_1 = 7.0$, $\alpha_2 = 5.0$, $\beta_2 = 7.0$, $\theta_2 = 9.0$, and $\gamma = 0.0$.

We also evaluate the basic non-reactive QoS strategies on *USAR-QoS*. We implement a traditional session and burst admission control mechanisms with a threshold of 9.0 and 7.0 seconds, respectively. We implement also the Best Effort FIFO scheduling approach.

The experiments we present here consist of 5000 sessions, created in an average rate of 10 sessions per second. The server is configured to support 50 bursts per second of throughput. The trace file is based on the TPC-W benchmark [1, 78]. Each burst is identified with a different user reaction class according to the *USAR* model [94]. The overall distribution of action classes for the trace file is showed by Table 4.14.

A	**B**	**C**	**D**	**E**	**F**	**G**
5%	10%	10%	15%	15%	15%	30%

Table 4.14: User reaction classes distribution for the workload

We evaluate all combinations of reactive and non-reactive admission control and scheduling policies to compare the results and verify the most effective ones. Due to space constraints we show only the most relevant results.

4.3.5.3 Experimental Results

Id	Adm. Control		Sched.	Duration	Max	Mean	Bursts	Bursts	Burst	Rejected
	Session	Burst	Policy	(sec)	R (sec)	R (sec)	requested	expired	lost rate	sessions
1	No	No	FIFO	12,604	29.41	6.51	621,342	229,755	36.98%	0
2	React	React	FIFO	9,031	7.22	3.24	535,018	206,622	38.62%	717
3	No	No	PFIN	13,002	84.07	6.27	621,342	157,537	25.35%	0
4	No	No	IFPN	13,660	44.37	5.95	621,342	16,500	2.66%	0
5	React	React	IFPN	8,910	7.12	3.35	544,078	126,925	23.33%	640

Table 4.15: Summary of experiments

Table 4.15 presents the experimental simulation results summary. Each experiment is identified by a number from 1 to 5. The columns admission control and scheduling describes the experiment configuration in terms of the QoS strategies. The table presents the following information: session and burst admission control policy, scheduling policy, total experimental duration, higher response time value (represented as R), mean response time, total number of requests, responses and expirations of bursts, burst lost rate, and number of rejected sessions by the session admission control. The burst lost rate represents the percentage of bursts expired compared to the whole number of bursts requested to the server. It corresponds to the number of occurrences the user asks the next burst before receiving the previous response.

In experiment 1 no admission control policy is active and the scheduling performed follows the typical FIFO approach. Figures 4.39 and 4.40 present the average response time and the bursts throughput, respectively. As we observe, the mean response time achieves a value higher than the 10-seconds limit due to the great number of requests being scheduled to the server. The throughput of the server is close to the server limit and there is a great number of bursts that expires during the execution, showing the overload situation. The total simulation time for the execution of the 5,000 sessions is 13,284 seconds. We observe that the burst lost rate achieves 36.98%, representing probably a high number of user unsatisfied.

The simulation performed in experiment 2 uses the reactive admission control policies

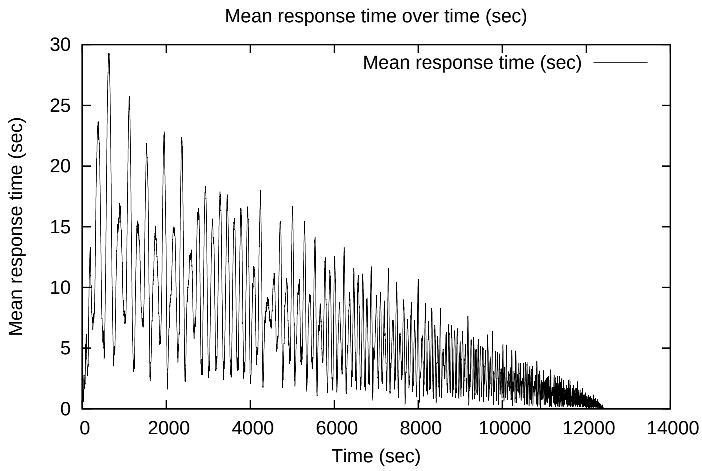


Figure 4.39: Experiment 1: no admission control and FIFO scheduling - Average Response Time

for session and burst. The scheduling approach is set to run the typical FIFO approach. The response time achieves lower values than experiment 1, showing the experiment configuration is effective to guarantee a better QoS level. Table 4.15 shows the maximum response time of 7.22 seconds, almost the lower value compared to the other experiments. This is due to the rejection of sessions and bursts by the admission control mechanism. Figures 4.41 and 4.42 show the response time achieved and the cumulative throughput, respectively. It is important to notice that there is a significant difference between the cumulative bursts requested throughput and the cumulative number of bursts replied, as we can see in (b). Their difference corresponds to the number of bursts rejected by the reactive mechanism of admission control. Since a great number of bursts and sessions are rejected the load under the server decreases and the response time achieves lower values. Despite the low response times, the user satisfaction is not fulfilled and the burst lost rate achieves 38.62%, showing that a scheduling mechanism may improve the results.

In experiment 3 and 4 there is no admission control active but the scheduling mechanism is set to perform the PFIN an IFPN approaches, respectively. For experiment 3 the busts lost rate achieves 25.35%, a lower value than experiments 1 and 2, but the response time

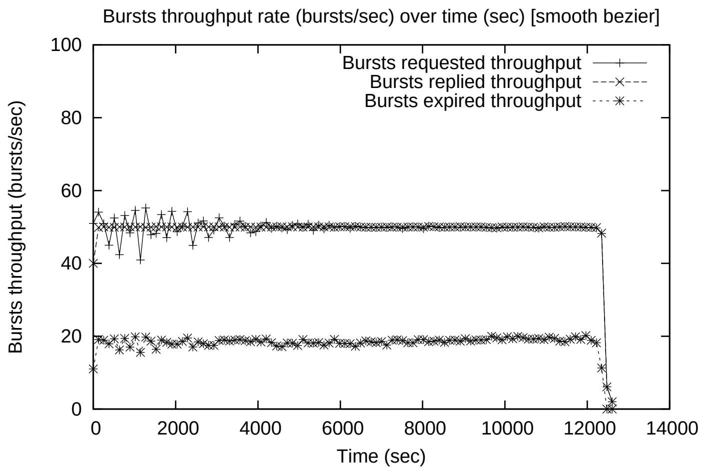


Figure 4.40: Experiment 1: no admission control and FIFO scheduling - Bursts Throughputs

achieves 84.07 seconds showing that the scheduling may cause high delays. For experiment 4, the response time achieves 44.37 seconds, but the lost rate is very low (2.55%) showing that the IFPN scheduling is very effective to provide a response time according to the user tolerance to QoS. It is worth to remember that such mechanism gives higher priority to requests of impatient users. Those requests are answered first by the server and the impatience level decreases, impacting the burst lost rate.

In order to improve the results, experiment 5 evaluates the hybrid strategy. In this experiment the reactive mechanism of admission control for session and burst and the reactive IFPN scheduling approach are active. As we observe, the maximum response time values observed on Table 4.15 present the lower value compared to the other experiments. The experiment presents the second lower value for the burst lost rate (21.67%). Figures 4.43 and 4.44 present the response time and cumulative throughput, respectively. We observe the low response time values and a significant percentage of bursts rejected that impacts the burst lost rate value.

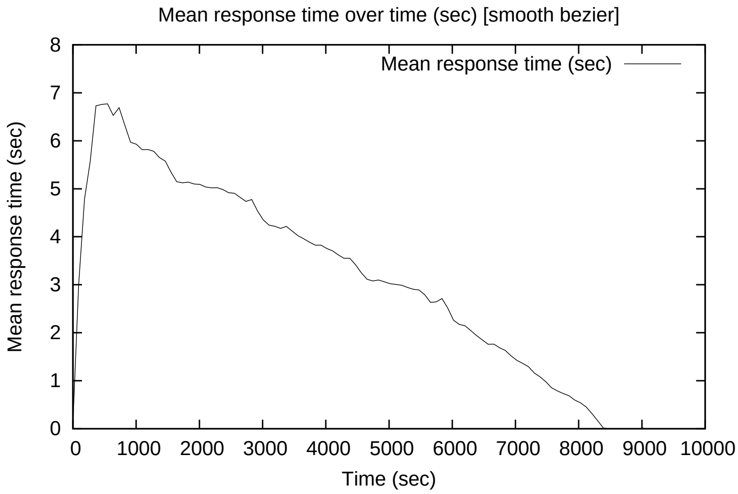


Figure 4.41: Reactive admission control and FIFO scheduling - Average Response Time

4.3.5.4 Discussion

As we may observe, the experiments running the reactive strategies achieve better response times and burst lost rates compared to the experiment without any additional mechanism. Experiment 4, running just the IFPN reactive scheduling, achieves the best burst lost rate, otherwise the response time behavior is not the better one. Experiment 2 achieves better response times due to the effectiveness of the rejection of bursts and sessions, despite their high burst lost rate. Experiment 5, running the reactive admission control and scheduling, presents the best result, achieving an equilibrium in terms of response time and burst lost rate values.

Considering these results, we conclude that the reactive QoS approaches present significant improvements, despite the gains are different according to each experiment configuration. Moreover, it is a task of the systems engineer to choose the best approach in order to provide each application demands.

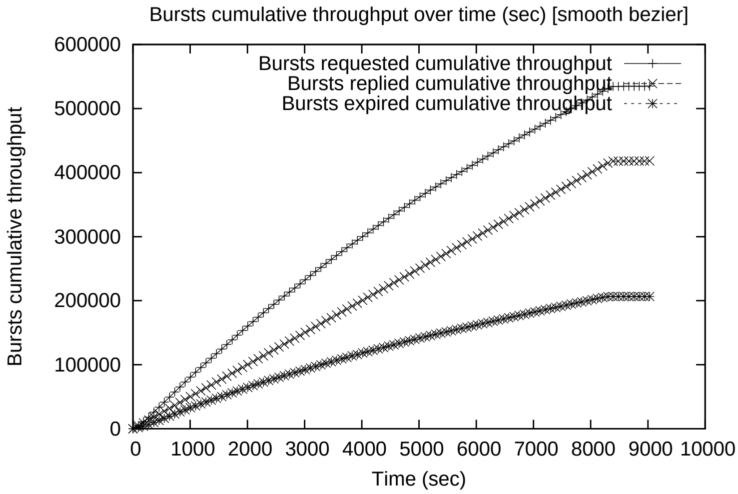


Figure 4.42: Reactive admission control and FIFO scheduling - Cumulative Throughput

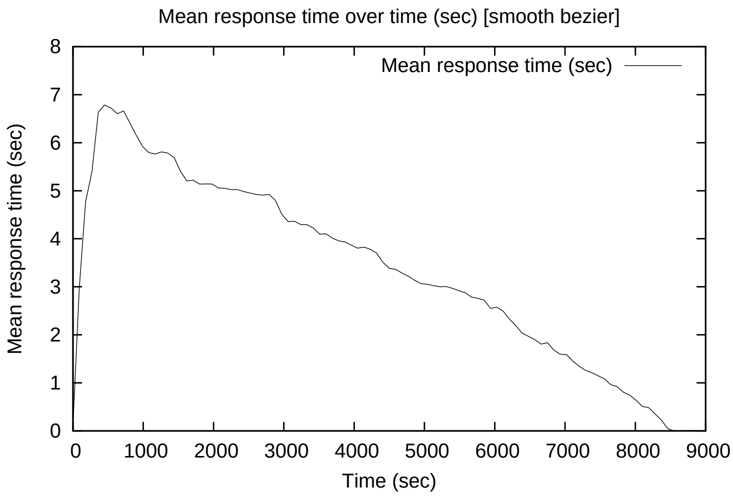


Figure 4.43: Reactive admission control and IFPN scheduling - Average Response Time

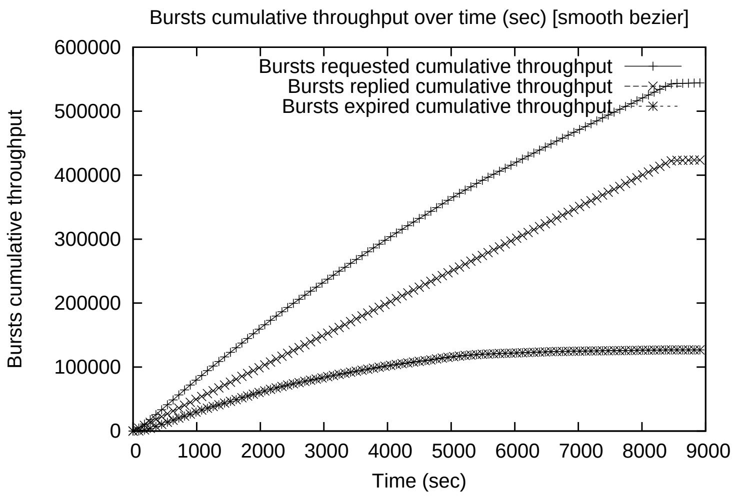


Figure 4.44: Reactive admission control and IFPN scheduling - Cumulative Throughput

4.4 Summary

In this chapter we presented a case study where we focus on the performance dimension of the reactivity. The main topics of this case study were the characterization of the reactivity, the use of it in workload generation, the *USAR-QoS* simulator, and the use of reactivity in QoS control mechanisms. We describe for each of these topics the main results and conclusions.

We have proposed *USAR*, a hierarchical workload characterization model. It comprises a validation methodology through simulation of user reactions. We consider behavior as how the interaction of users with Web applications is affected by variable latencies, measured in terms of response time. In order to demonstrate and validate the *USAR* model, we present its application and validation using actual data from a proxy-cache server.

We have analyzed the impact of reactive workloads on the performance of Web applications. In order to generate these workloads we have designed a new version of *httperf* workload generator to create user sessions with dynamic inter-arrival time. The new version of *httperf* were used to generate workloads based on the TPC-W benchmark in a experimental study. We evaluate the performance of a Web application using three distinct scenarios of load, focusing on the analysis of the following metrics: throughput, response time, active bursts and active sessions for each scenario.

The results of our experiments show the changes in the performance of the Web server when the workload with reactivity is applied to it. The graphics show changes both in the values of each variable in time as in the form of each curve. Both the throughput and response time values have significant variations.

In order to provide a easy and controlled way to validate our QoS approaches, we have developed a simulator. The main reason is that with simulation we avoid many environmental conditions that can impact negatively the results and obtain results easier and faster.

From the results we observe different gains according to each QoS approach. The mechanism based only on reactive scheduling achieves the best burst lost rate, since its mechanism gives priority to requests classified with impatient classes. The mechanisms that adopt the admission control are effective to reduce the response time, but may cause the increase in the burst lost rate, due to the increase in the amount of users rejecting bursts or sessions. The hybrid mechanism presents an equilibrium, reducing both the response time values and the burst lost rate.

Is important to observe that the proposed policies may be unfair to certain users since their typical behavior is used to improve QoS. The benefits obtained are significant and we recommend our policies despite the subject of unfairness. There is a relevant improvement in the QoS of reactive Internet systems through the use of reactive approaches.

Chapter 5

Case Study - e-Commerce

This chapter present an e-business case study, that consists of the study of a popular e-commerce application: online auctions. The chapter is divided in five main parts. Section 5.1 presents an overview of the case study scenario. Section 5.2 describes the preliminary study about reactivity in this context of e-business, also presenting some important modeling issues. Section 5.4 explains how we model and characterize this case study. Section 5.5 describes how we model reactivity in this scenario of e-business. Finally, Section 5.6 summarize the main aspects and results of this chapter.

The meaning of the term "electronic commerce" has changed over time. Originally, "electronic commerce" meant the facilitation of commercial transactions electronically, usually using technology like Electronic Data Interchange (EDI, introduced in the late 1970s) to send commercial documents like purchase orders or invoices electronically.

Later it came to include activities more precisely termed "Web commerce" – the purchase of goods and services over the World Wide Web via secure servers (note HTTPS, a special server protocol which encrypts confidential ordering data for customer protection) with e-shopping carts and with electronic pay services, like credit card payment authorizations.

The online auction business model is one in which participants bid for products and services over the Internet. When one thinks of online auctions they typically think of eBay, the largest online auction site. Like most auction companies, eBay does not actually sell goods that it owns itself. It merely facilitates the process of listing and displaying goods, bidding on items, and paying for them. It acts as a marketplace for individuals and businesses that use the site to auction off goods and services.

Several types of online auctions are possible. In an English auction the initial price starts low and is bid up by successive bidders. In a Dutch auction the price starts high and is reduced until someone buys the item. eBay also offers fixed price listings.

The strategic advantages of this business model are:

1. No time constraints. Bids can be placed at any time, 24 / 7 . Items are listed for a number of days (normally between 1 and 10) (at the discretion of the seller), giving purchasers time to search, decide, and bid. This convenience increases the number of bidders.

2. No geographical constraints. Sellers and bidders can participate from anywhere that has Internet access. This makes them more accessible and reduces the cost of "attending" an auction. This increases the number of listed items (i.e., number of sellers) and the number of bids for each item (i.e., number of bidders). The items do not need to be shipped to a central location, reducing costs, and reducing the seller's minimum acceptable price.

3. Intensity of social interactions. The social interactions involved in the bidding process are very similar to gambling. The bidders wait in anticipation hoping they will "win" (eBay calls the successful bidder the "winner"). Much like gambling addiction, many bidders bid primarily to "play the game" rather than to obtain products or services. This creates a highly loyal customer segment for eBay.

4. Large number of bidders. Because of the potential for a relatively low price, the broad scope of products and services available, the ease of access, and the social benefits of the auction process, there are a large numbers of bidders.

5. Large number of sellers. Because of the large number of bidders, the potential for a relatively high price, reduced selling costs, and ease of access, there are a large number of sellers.

6. Network economies. The large number of bidders will encourage more sellers, which, in turn, will encourage more bidders, which will encourage more sellers, etc., in a virtuous spiral. The more the spiral iterates, the larger the system becomes, and the more valuable the business model becomes for all participants.

7. Captures consumers' surplus. Auctions are a form of first degree price discrimination. As such, they attempt to convert part of the consumers' surplus (defined as the area above the market price line but below the firm's demand curve) into producers' surplus. On-line auctions are efficient enough forms of price discrimination that they are able to do this.

5.1 Scenario: eBay

5.1.1 The eBay Auction Site

eBay was founded in 1995. eBay boosters have claimed that in terms of revenue growth, eBay is among the fastest-growing companies of all time.

Millions of collectibles, appliances, computers, furniture, equipment, vehicles, and other miscellaneous items are listed, bought, and sold daily. Some items are rare and valuable, while many others are dusty gizmos that would have been discarded if not for the thousands of eager bidders worldwide, proving that if one has a big enough market, one will find someone willing to buy anything. It is fair to say that eBay has revolutionized the collectibles market by bringing together buyers and sellers internationally in a huge, never-ending yard sale and auction. As of June 2005, there were over 15,000 members in the eBay Developers Program, comprising a broad range companies creating software applications to support eBay buyers and sellers as well as eBay Affiliates.

eBay [44] is a global phenomenon - the world's largest garage sale, online shopping center, car dealer and auction site with 147 million registered users in 30 countries as of March 2005. You can find everything from encyclopedias to olives to snow boots to stereos to airplanes for sale. And if you stumble on it before the eBay overseers do, you might even find a human kidney or a virtual date.

eBay is, first and foremost, an online auction site. You can browse through categories like Antiques, Boats, Clothing & Accessories, Computers & Networking, Jewelry & Watches and Video Games. When you see something you like, you click on the auction title and view the details, including pictures, descriptions, payment options and shipping information. If you have a pretty good idea of what you're looking for, you can search for it using simple keywords, such as “Apple iPod”, or using more advanced search criteria that helps narrow the results, such as keywords to exclude, item location, price range and accepted payment methods.

If you place a bid on an item, you enter a contractual agreement to buy it if you win the auction. All auctions have minimum starting bids, and some have a reserve price - a secret minimum amount the seller is willing to accept for the item. If the bidding doesn't reach the reserve price, the seller doesn't have to part with the item. In addition to auctions, you can find tons of fixed-price items on eBay that make shopping there just like shopping at any other online marketplace. You see what you like, you buy it, you pay for it and you wait for it to arrive at your door. There are also auction listings that give you the option to “Buy it Now” for a price that's typically higher than the auction's start price. If you choose to buy the item for the ”Buy it Now” price instead of bidding on it, the auction ends instantly and the item is yours.

eBay [105] is a “second price” auction, in which the winner pays out not their own winning bid, but the second highest bid (plus a small amount extra, which is the smallest price increment allowed in the bidding). It also uses an automatic bidding system, named “proxy-bidding”, to make bidding on auctions more convenient and less time-consuming for bidders. There is nothing you have to set up in order to bid in this way.

Figure 5.1 presents an eBay screenshot. eBay generates revenue from sellers, who

Figure 5.1: eBay Screenshot

pay a fee based on the selling price of each item, a fee based on the starting price, and from advertising. eBay does not handle the goods, nor does it transact the buyer-seller payments, except through its subsidiary PayPal. Instead, much like newspaper want-ads, sellers rely on the buyers' good faith to make payment, and buyers rely on the sellers' good faith to actually deliver the goods intact. To encourage fidelity, eBay maintains, rates, and publicly displays the post-transaction feedback from all users, whether they buy or sell. This way, the buyer is encouraged to examine the sellers' feedback profile before bidding to rate their trustworthiness. Sellers with high ratings generally have more bids and garner higher bids. However, it is possible for sellers to make their feedback private and just leave the numbered rating (number of positive, negative and neutral feedback with a positive feedback percentage), which means that bidders and sellers cannot see the comments other users have left. eBay also has a significant affiliate program, and eBay affiliates can, for example, place live eBay product images and links on their web sites. eBay claims that statistically fewer than 1 in 200 transactions fail.

There are some controversial practices of users identified by eBay:

- Bid sniping: is placing a high bid during the last few seconds of an auction such that no time remains for other users to counterbid. This practice is allowed on eBay, however many other auction sites, such as Yahoo! Auctions, prevent this by automatically extending the auction by five minutes when a last-minute bid is placed.

- Shill bidding: is the deliberate use of secondary registrations, aliases, family members, friends, or associates to artificially drive up the bid price of an item. (This is also known as "bid padding".) Shill bidding is not allowed on eBay. [11] Furthermore, shill bidding is a crime in many jurisdictions, and can be prosecuted under United States wire fraud laws.

5.1.2 Dataset Description

We use data from eBay site in this case study. The data consists of the auction information of three different products: Nintendo GameCube, Sony PlayStation 2, and Microsoft Xbox System. The data was collected from 05/25/2005 to 08/15/2005, almost three months. Table 5.1 presents general information about the auction data. As we can see there are a significant number of auctions and bids to analyze.

	Product			
Information	**Nintendo**	**Sony**	**Xbox**	**Total**
# Auctions	8855	17234	9928	36107
# Bids	85803	179057	120021	384881

Table 5.1: General information about the eBay dataset

For each auction, we have the information presented in Table 5.2. The information about each auction bid is showed in Table 5.3. As can be seen, it is a rich dataset, that can be deeply analyzed in order to identify important negotiation rules.

We have decided to use the data from three auction products: Nintendo, Sony and Xbox. In the next section we will describe some important statistics about these auctions.

5.1.3 Dataset Analysis

We analyze the eBay datasets in order to better understand this important e-business market. Table 5.4 presents general statistics, concerning the number of buyers and sellers that participates in the auctions, the auction pricing, the average number of bids per auction, and the average number of unique bidders per auction.

From these statistics, we can conclude that:

- The number of successful auctions vary from 63% to 69%.
- The number of distinct sellers is high, showing that there is no concentration of auctions in a small number of sellers. The top seller has created 186 Nintendo auctions, 223 for Sony, and 116 for Xbox.
- The number of distinct buyers is high as well, enforcing the competition in this e-market. On the other hand, from this set of buyers, there exists a reduced number of them who become winners.
- The mean variation of price between new and used products is small, however the standard deviation of the prices is very high.
- There are a significant number of bids per auctions, which indicates the level of competition during the negotiation. This information is confirmed once the average number of unique bidders per auction is more than 5.

Table 5.5 shows statistics about the distribution of auctions among buyers and sellers. The objective of this analysis is to understand the participation frequency of buyers and

#	Field	Description
1	auid	Auction ID (generated by eBay)
2	rsmt	Reserve price met or not
3	rsus	Reserve price set or not
4	endt	Auction ending date and time
5	stdt	Auction starting date
6	stdt2	Auction starting time
7	nobi	Number of bids attracted
8	curr	Currency used for auction
9	wibi	Winning bid price
10	payp	Winning paid price
11	wiid	Winner's ID
12	bira	Bidder's feedback scores in eBay
13	seid	Seller's ID
14	sera	Seller's feedback score in eBay
15	svpg	Page saved or not (for programming purpose)
16	pofb	The percentage of seller's positive feedback
17	item	Auction title which describes the item in condition etc.
18	byno	Buy-it-now price that seller sets
19	mesi	The date since seller became eBay membership
20	stbi	Starting bid that seller sets to attract bids
21	melo	Seller' location as eBay membership(e.g., Hartford, CT)
22	abme	Whether seller uses eBay' About Me option
23	pose	Whether seller is an eBay Power Seller
24	cond	Item's condition (New or Used)
25	sest	Whether seller has eBay store
26	gift	Whether seller provides Gift Wrap Service to buyers
27	pabp	Whether seller uses PayPal Buyer Protection program
28	idve	Whether seller's member ID is verified by the third party.
29	location1	Seller's location in town and state
30	location2	Seller's location in country
31	shipto	Places where Item can be shipped to (US, or Worldwide etc.)
32	shipment	Shipment fee (fixed fee, or free or calculation upon distance)
33	pic	Number of pictures used in the auction)
34	filesize	The auction file size (excluding pictures)
35	returndummy	Whether the auction item can be returned for refund
36	insurance	Whether seller offers shipment insurance
37	pame1	The first payment method that seller can accept (e.g. PayPal)
38	pame2	The second payment method that seller can accept (Visa etc.)
39	pame3	Other payment methods
40	pame4	Other payment methods
41	pame5	Other payment methods
42	pame6	Other payment methods
43	category	Auction product type category

Table 5.2: Auction Data Fields

#	Field	Description
1	auid	Auction ID (generated by eBay)
2	biid	Bidder ID
3	biam	Bidding amount, i.e., buyer's bidding price
4	bira	Bidder's feedback scores in eBay
5	bidt	Bidding date and time

Table 5.3: Auction Bid Data Fields

sellers in this e-market.

From this analysis it is possible to find out that the number of auctions per seller presents a huge variation, but in average there are a small number of auctions per seller. In the case of auctions per buyer, we can identify that buyers participate in average in more than four auctions with a huge standard deviation. This information and the other statistics related to number of bids and bidders per auction suggest that there is competition in auctions and an opportunity to study the behavior of bidders in this e-market.

Table 5.6 presents statistics about the auction overlapping, that represents the level of concurrency between auctions. In the table we present the minimum, maximum, the average and standard deviation (STD) of this measure.

As can be seen, there exists a significant number of concurrent auctions over time, which gives more options to buyers, concerning when to bid and which price to offer.

Figures 5.2, 5.3 and 5.4 represent the histogram of auction duration for Nintendo, Sony and Xbox, respectively. The three set of auctions present similar duration behavior. Nintendo has more than 40% of auctions during a week and around 15% of them during one day. For Sony and Xbox, the number of auctions with 7-day duration is 40% and 1-day is 20%. The other durations are similar: 20% of three days long and 15% of five days long. Around 5% of the auctions has durations of 2, 4, 6 and 10 days. As can be seen, clearly the duration of auctions has a predominance of odd number of days.

Statistics	**Product**		
	Nintendo	**Sony**	**Xbox**
# Auctions	8855	17234	9928
Number of auctions with winner	6103	10884	6466
Number of unique sellers	5453	9340	6466
Number of unique buyers that win	735	795	548
Number of unique buyers	18073	39026	26358
Average winner price (general)	US$ 32.04	US$ 44.21	US$ 49.63
Average winner price STD (general)	37.04	51.36	58.87
Average winner price (new)	US$ 35.32	US$ 48.71	US$ 52.97
Average winner price STD (new)	37.53	60.26	66.10
Average winner price (used)	US$ 31.90	US$ 41.71	US$ 50.34
Average winner price STD (used)	38.14	49.09	58.12
Average number of bids per auction	11.59	12.38	14.13
Average number of bids per auction STD	9.43	10.82	11.27
Average number of unique bidders per auction	5.39	5.71	6.48
Average number of unique bidders per auction STD	3.58	4.23	4.57

Table 5.4: General statistics about the eBay dataset

Figures 5.5, 5.6 and 5.7 show the graphs that represents the number of bids versus the auction duration for Nintendo, Sony and Xbox, respectively. We can see that the number of bids is normally small in all of them but there exists a huge variation in auctions with duration of 1, 3, 5, 7 and 10 days. These observations can be explained using the auction duration histogram, that shows there are many auctions with these duration values. And for the auctions with the duration of 10 days, a higher number of bids was expected, once the probability of giving bids is naturally higher since there is more time to do it.

Figures 5.8, 5.9 and 5.10 illustrate the histogram of auctions per buyer for Nintendo, Sony and Xbox, respectively. The histograms have a descendant behavior, showing the

Statistics		**Product**		
		Nintendo	**Sony**	**Xbox**
Number of auctions per seller	MIN	1	1	1
	MAX	256	260	136
	AVG	1.62	1.84	1.54
	STD	6.06	6.24	2.87
Number of auctions per buyer	MIN	1	1	1
	MAX	269	308	187
	AVG	4.75	4.59	4.55
	STD	7.47	7.86	6.85

Table 5.5: General information about the eBay dataset - Auction Distribution

Statistics		**Product**		
		Nintendo	**Sony**	**Xbox**
Concurrent auctions	MIN	72	159	68
	MAX	5863	16606	5965
	AVG	1206.68	2088.55	1230.34
	STD	321.75	620.44	340.36

Table 5.6: General information about the eBay dataset - Auction overlapping

number of bidders that participate in only one auction represent 50% (Xbox) to 70% (Sony). Most of the bidders participate in up to 20 auctions.

Figures 5.11, 5.12 and 5.13 show the histogram of winner bid value for Nintendo, Sony and Xbox, respectively. The histograms have a descendant behavior, showing that there is a concentration of auctions with a low winner bid (30% to 50%). Most auctions have a winner price smaller than U$100.

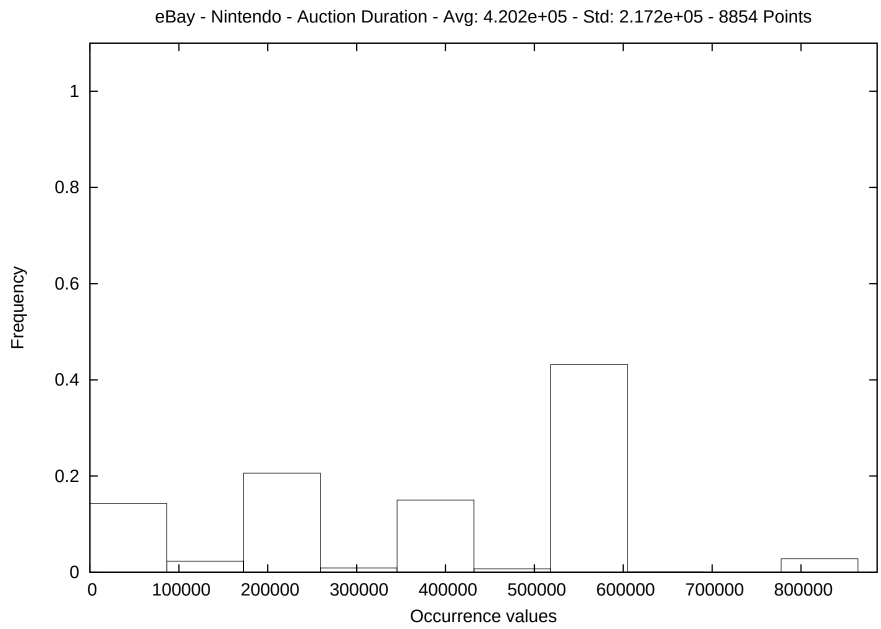


Figure 5.2: Auction Duration - Nintendo

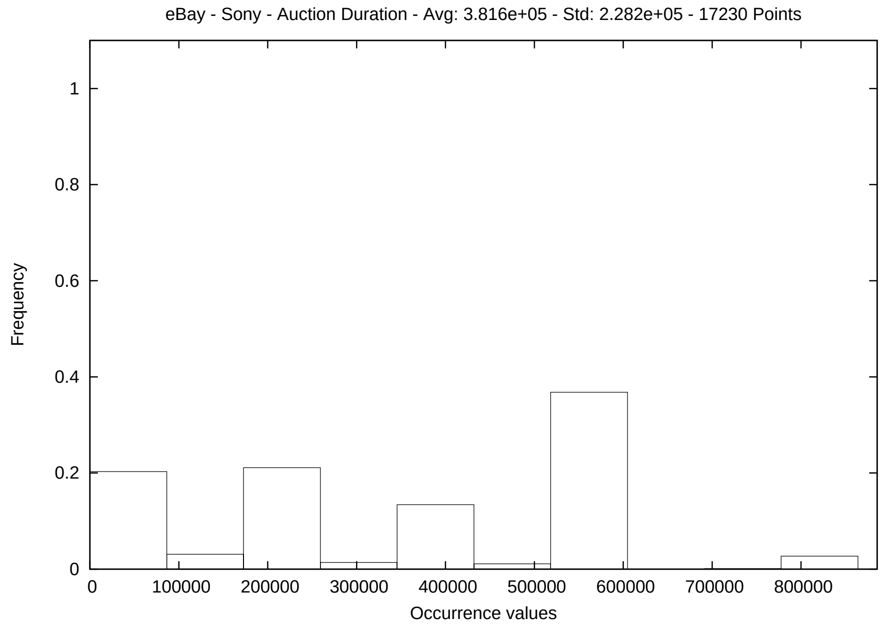


Figure 5.3: Auction Duration - Sony

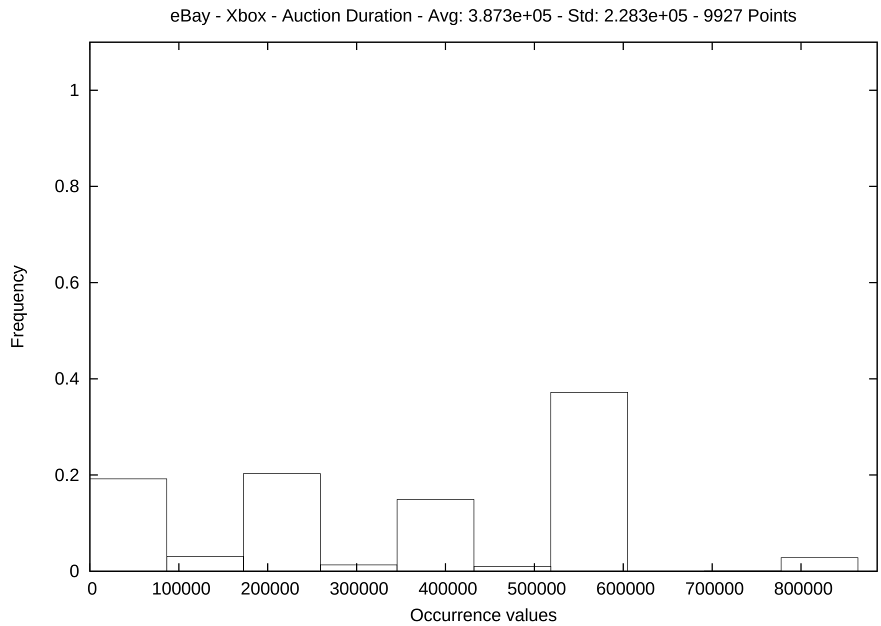


Figure 5.4: Auction Duration - Xbox

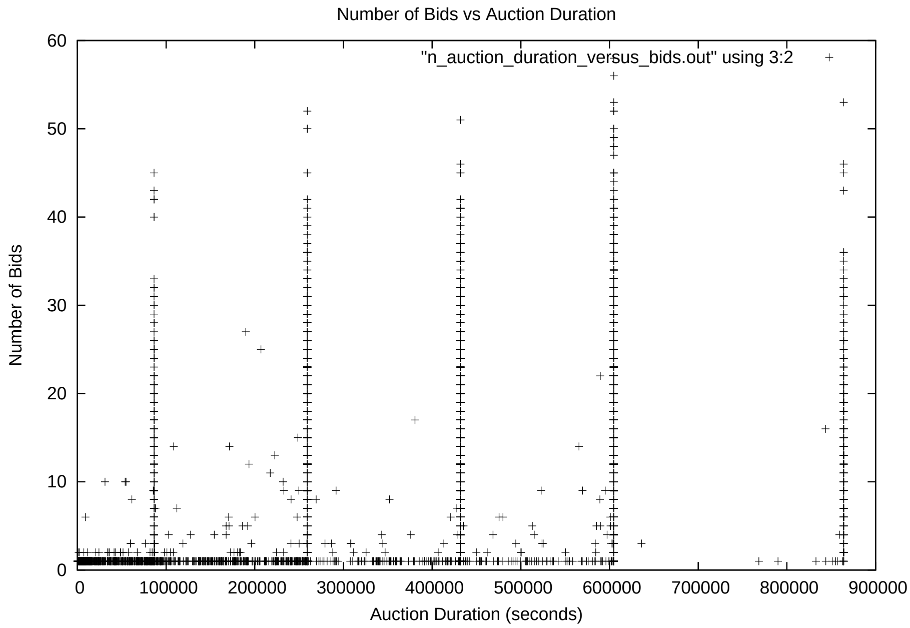


Figure 5.5: Auction Bids versus Duration - Nintendo

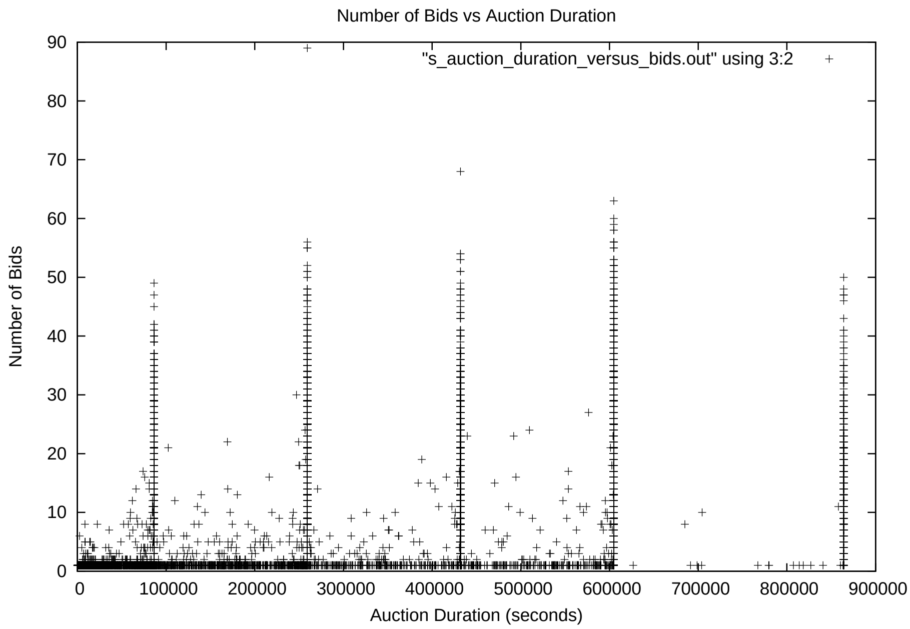


Figure 5.6: Auction Bids versus Duration - Sony

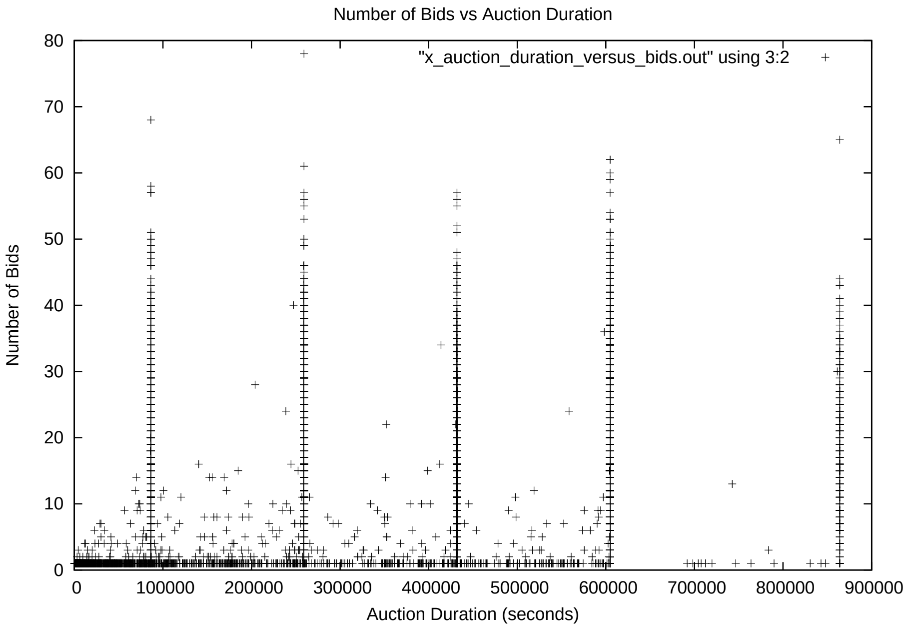


Figure 5.7: Auction Bids versus Duration - Xbox

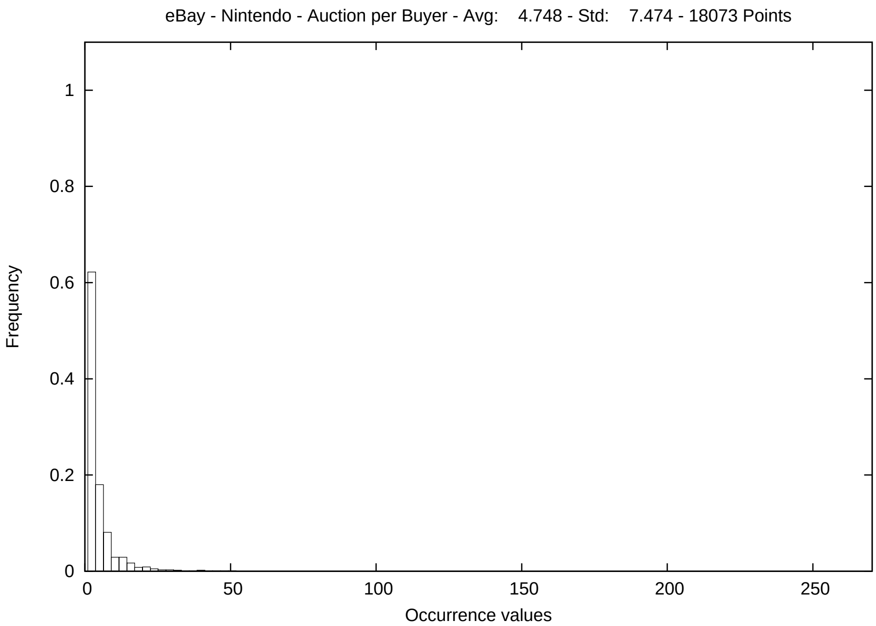


Figure 5.8: Auction per Buyer - Nintendo

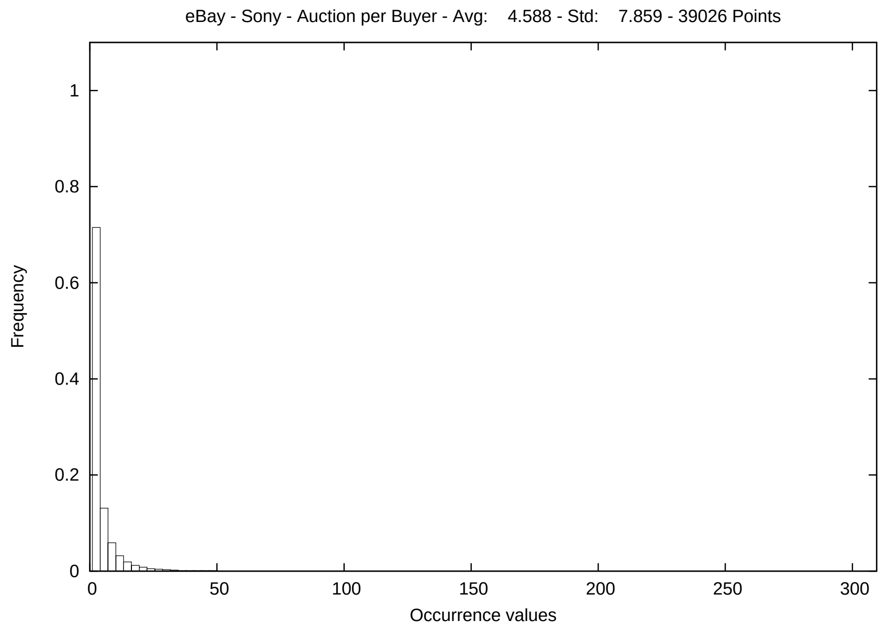


Figure 5.9: Auction per Buyer - Sony

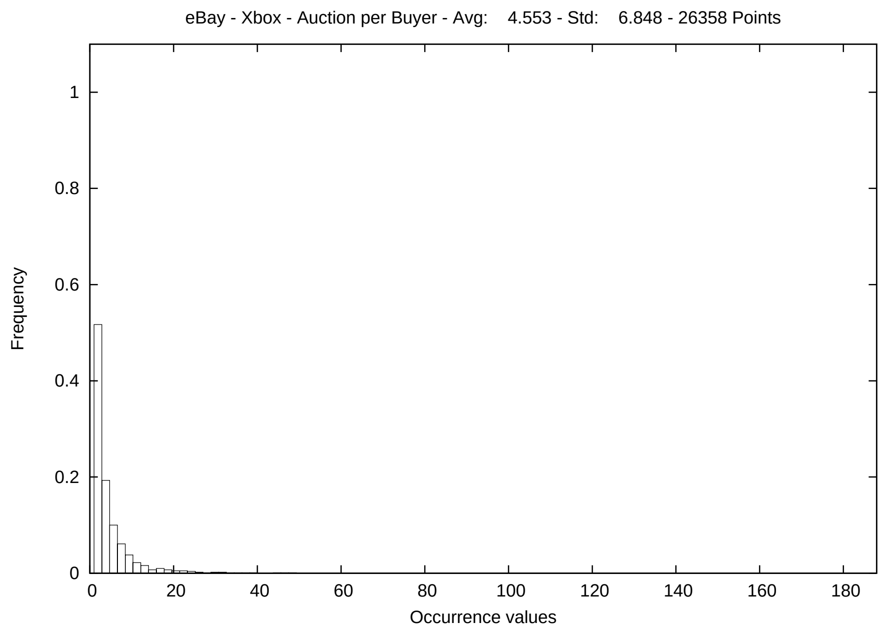


Figure 5.10: Auction per Buyer - Xbox

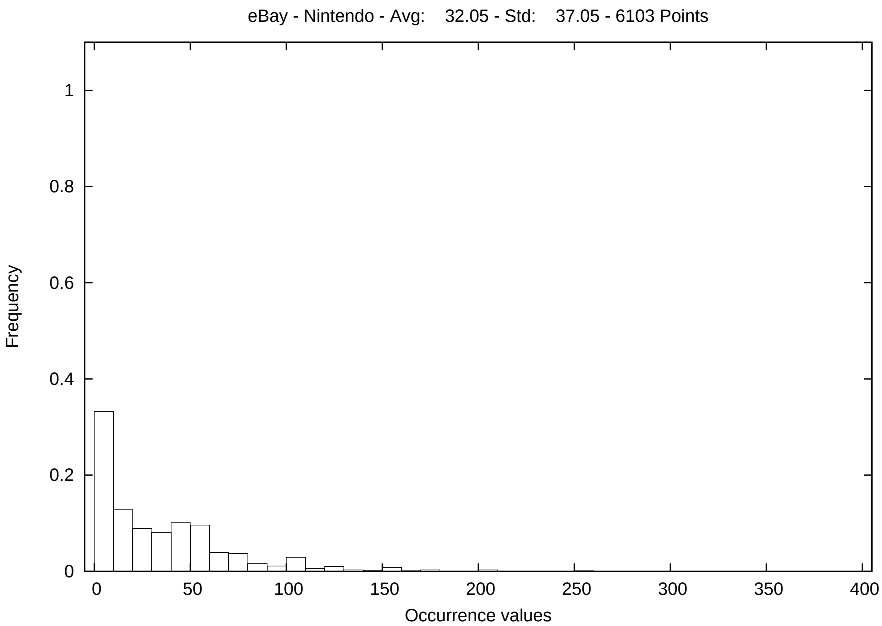


Figure 5.11: Winner Bid - Nintendo

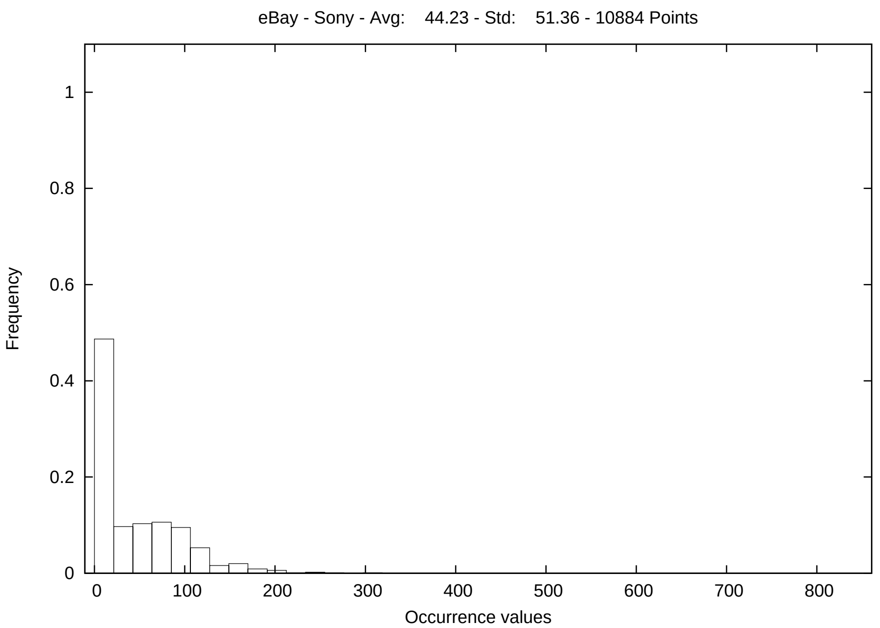


Figure 5.12: Winner Bid - Sony

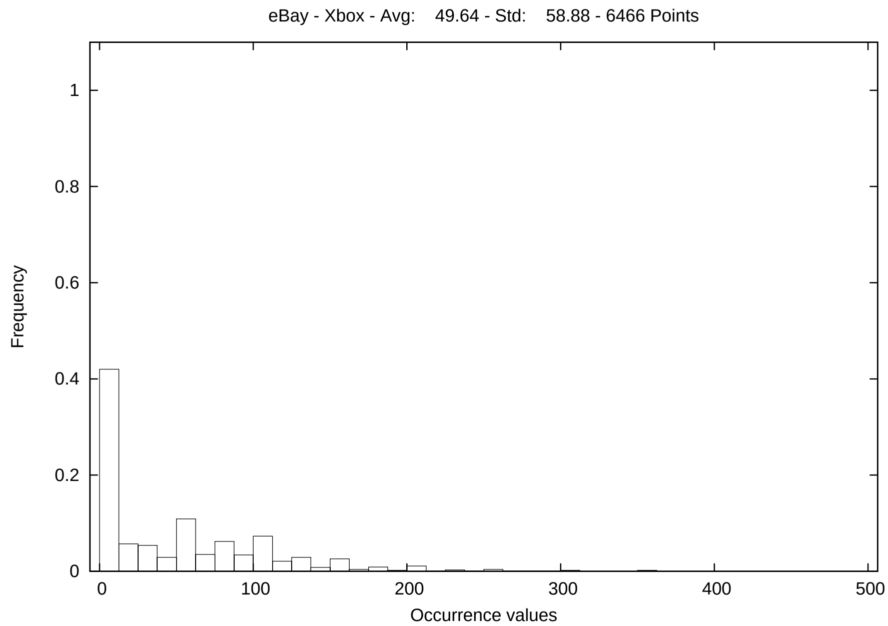


Figure 5.13: Winner Bid - Xbox

5.2 Reactivity - Preliminary Study

This section presents our preliminary study about reactivity. This reactivity analysis is important in order to provide more insights about this e-business case study. Moreover, this section also presents some modeling issues.

First we identify that it is important to analyze two aspects:

1. The inter-bidding time (IBT): the time between two consecutive bids.

2. The price difference: the difference of price value between two consecutive bids.

Figures 5.14, 5.15 and 5.16 show the difference of price and time between each consecutive bids for Nintendo, Sony and Xbox, respectively. As can be seen, there is a huge concentration of points where the IBT and the price difference are low. However the variation of these two metrics shows that there is a significant variation between them. In the case of IBT, this variation is very high. Moreover, it is not possible to identify a clear correlation between these two metrics. This analysis reaches similar results for the three different auction products.

In order to better understand the dynamics of the auction, we create a dataset for each bidder with some important information. This dataset dictionary is presented in Table 5.7.

Figures 5.17, 5.18 and 5.19 present some results from the datasets for Nintendo, Sony and Xbox, respectively. We have analyzed five aspects:

- BIDDER BIDS REL: the relative quantity of bidder bids.

- ACTIVE TIME REL: the relative time that each buyer stays active in the auction, that is, the time between the first and last bid.

#	**Field**	**Description**
1	AUID	Auction Identification
2	BIDDER_ID	Bidder Identification
3	BIDDER_BIDS	Quantity of Bidder Bids
4	BIDDER_BIDS_REL	Relative Quantity of Bidder Bids
5	BIDDER_MIN_BID	Min Bid Value
6	BIDDER_MAX_BID	Max Bid Value
7	BIDDER_BIDS_AVG	Average Bid Value
8	BIDDER_BIDS_STD	Standard Deviation Bids Value
9	BIDDER_BIDS_AVG_REL	Relative Average Bid Value
10	ACTIVE_TIME	Period of Time between the First and Last Bidder's Bid
11	ACTIVE_TIME_REL	Relative Time between the First and Last Bidder's Bid
12	START	Auction Start Time
13	END	Auction End Time
14	DURATION	Auction Duration
15	MIN_BID_DATE	First Bid Date
16	MAX_BID_DATE	Last Bid Date
17	BID_ENTRANCE_REL	Relative First Bid Date
18	BID_LEFT_REL	Relative Last Bid Date
19	AVG_DELTA_TIME	Average time difference between two bids of the same bidder in a auction.
20	MIN_DELTA_TIME	Minimum time difference between two bids of the same bidder in a auction.
21	MAX_DELTA_TIME	Maximum time difference between two bids of the same bidder in a auction.
22	AVG_DELTA_TIME_REL	Relative average time difference between two bids of the same bidder in a auction.
23	MIN_DELTA_TIME_REL	Relative Minimum time difference between two bids of the same bidder in a auction.
24	MAX_DELTA_TIME_REL	Relative Maximum time difference between two bids of the same bidder in a auction.
25	AVG_DELTA_PRICE	Average price difference between two bids of the same bidder in a auction.
26	MIN_DELTA_PRICE	Minimum price difference between two bids of the same bidder in a auction.
27	MAX_DELTA_PRICE	Maximum price difference between two bids of the same bidder in a auction.
28	AVG_DELTA_PRICE_REL	Relative average price difference between two bids of the same bidder in a auction.
29	MIN_DELTA_PRICE_REL	Relative minimum price difference between two bids of the same bidder in a auction.
30	MAX_DELTA_PRICE_REL	Relative maximum price difference between two bids of the same bidder in a auction.
31	IS_WINNER	Flag that identify if the bidder is the winner.

Table 5.7: Bidders Dataset

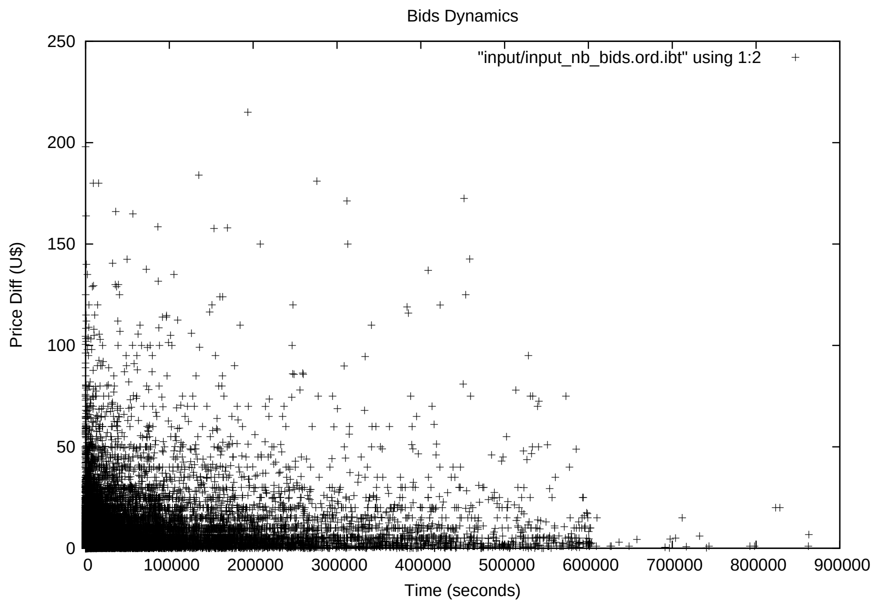


Figure 5.14: Bid Dynamics - Nintendo

- BID ENTRANCE REL: the relative time where the buyer initiates her/his participation in the auction - first bid.
- BID LEFT REL: the relative time where the buyer finishes her/his participation in the auction - last bid.
- IS WINNER: identify if the bidder is the winner of the auction.

Analyzing these aspects, we can conclude that:

- There is a significant number of winners that makes the totality of bids (one or more) in the auctions. The other winners present a relatively small number of bids or less than 50%.
- We observe a relatively small active time.
- In most auctions, the winner puts the bid near the end of the auction.
- The winners' last bid is always placed close to the end of the auction negotiation.

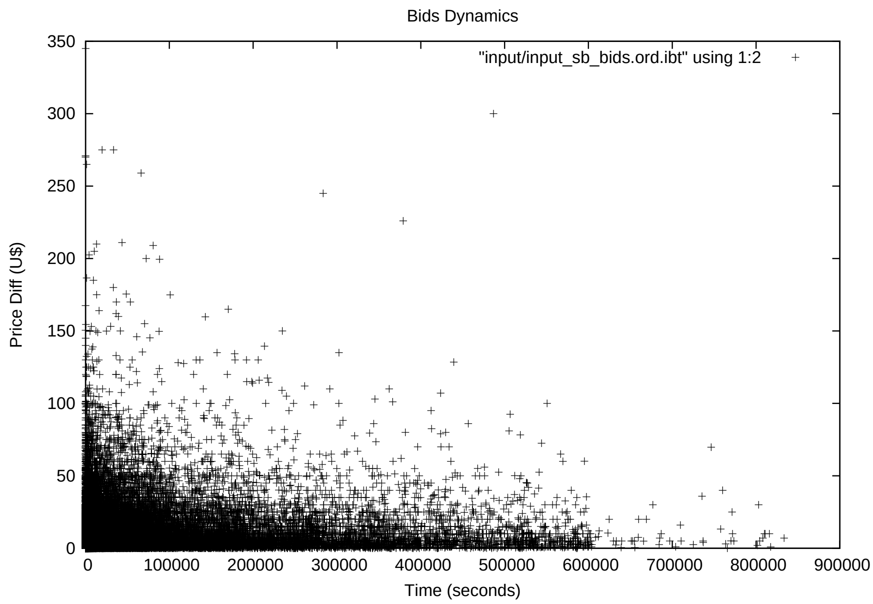


Figure 5.15: Bid Dynamics - Sony

These results that can be seen in the pictures of Figure 5.17 are similar in the other two (Figures 5.18 and 5.19).

In order to analyze the bidders' behavior during auctions, we classify them by isolating the winner attribute to evaluate how some attributes affect the result of the auction. Initially we identify the most relevant attributes using an attribute selection algorithm. We use the following attributes from the dataset presented in Table 5.7: BIDDER_BIDS_REL, ACTIVE_TIME_REL, BID_ENTRANCE_REL, BID_LEFT_REL, AVG_DELTA_TIME_REL, AVG_DELTA_PRICE_REL, and IS_WINNER.

Figure 5.20 presents the classification tree for Nintendo auctions dataset. It is possible to identify some interesting results:

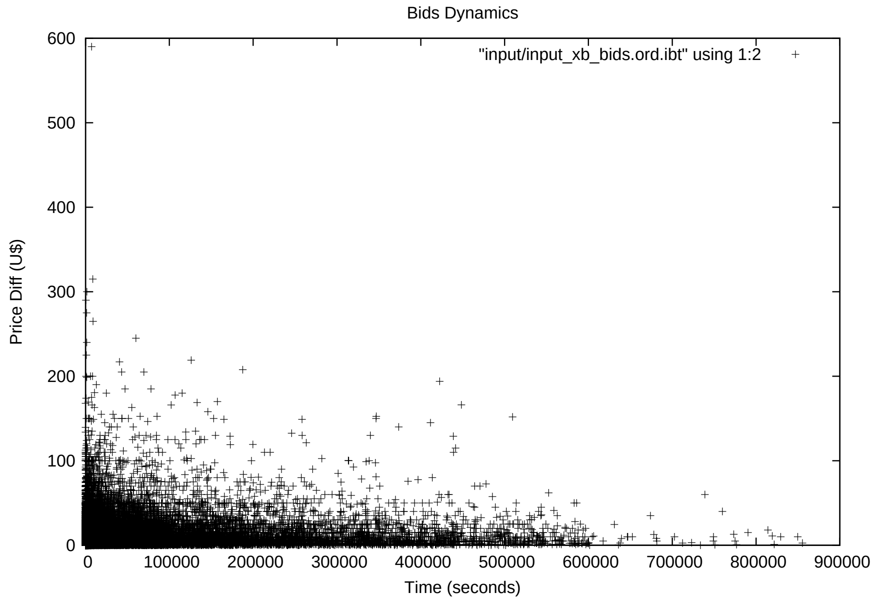


Figure 5.16: Bid Dynamics - Xbox

- 43% of winner bidders submitted the last bid almost at the end of the auction, their bids represent less than 90% of auction bids, and their average relative delta price consists of small values.

- 34% of winner bidders have made more than 90% of the bids of the auction.

- 9.3% of winner bidders present a small average relative price difference between bids (less or equal than 0.02), a small average relative delta time between bids (less or equal than 0.0145).

- 9.3% make less bids than 38.5% of the total number of the auction bids, and make the last bid almost at the end of the auction (BID_LEFT_REL > 0.9993).

- 8.5% of winner bidders present an average relative delta time between bids greater or equal than 0.0145, make less bids than 38.5% of the total number of the auction bids, and make the last bid almost at the end of the auction (BID_LEFT_REL > 0.9996).

- 4.5% of winner bidders present a small average relative price difference between bids

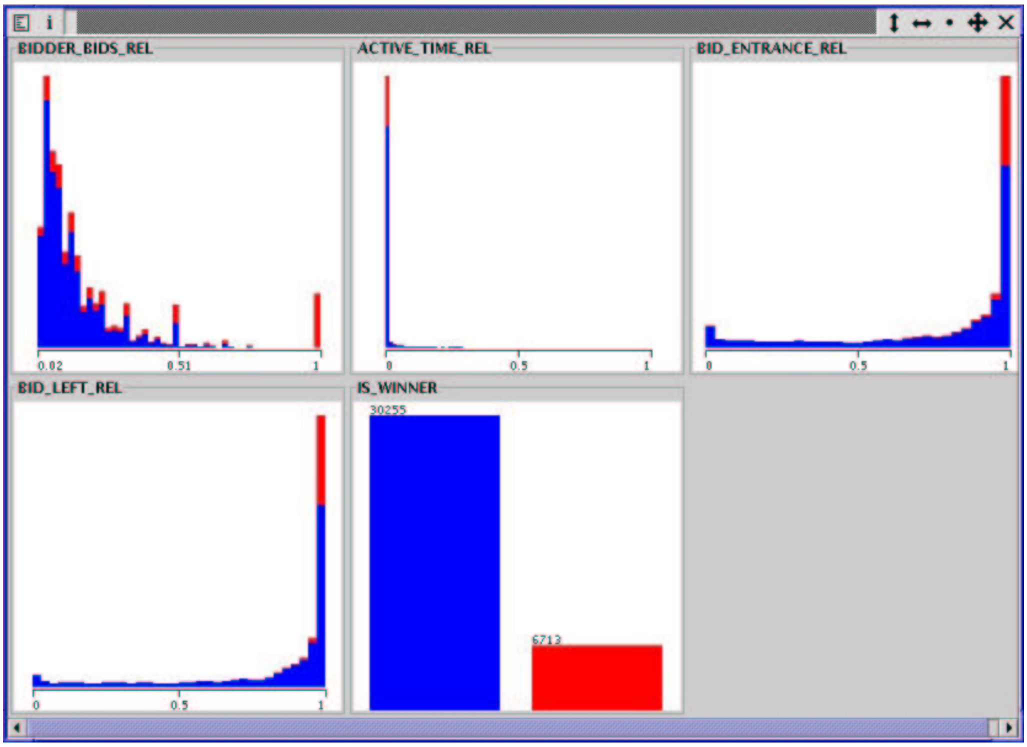


Figure 5.17: Dataset Analysis - Nintendo

(less or equal than 0.0227), makes 38.5% to 89.5% of the total number of the auction bids, and place the last bid almost at the end of the auction (BID_LEFT_REL > 0.9993).

The results of the classification for Sony and Xbox auctions are similar to Nintendo. From the analysis of these auctions we identify that it is interesting to divide the auction period in parts to analyze them separately. These will clarify the factors that directly affect the result of the auction.

The idea of breaking the auction negotiation in parts was motivated by the fact that reactivity is not an aggregated measure, therefore a hierarchical model for online auctions would be necessary to provide a representation to investigate reactivity.

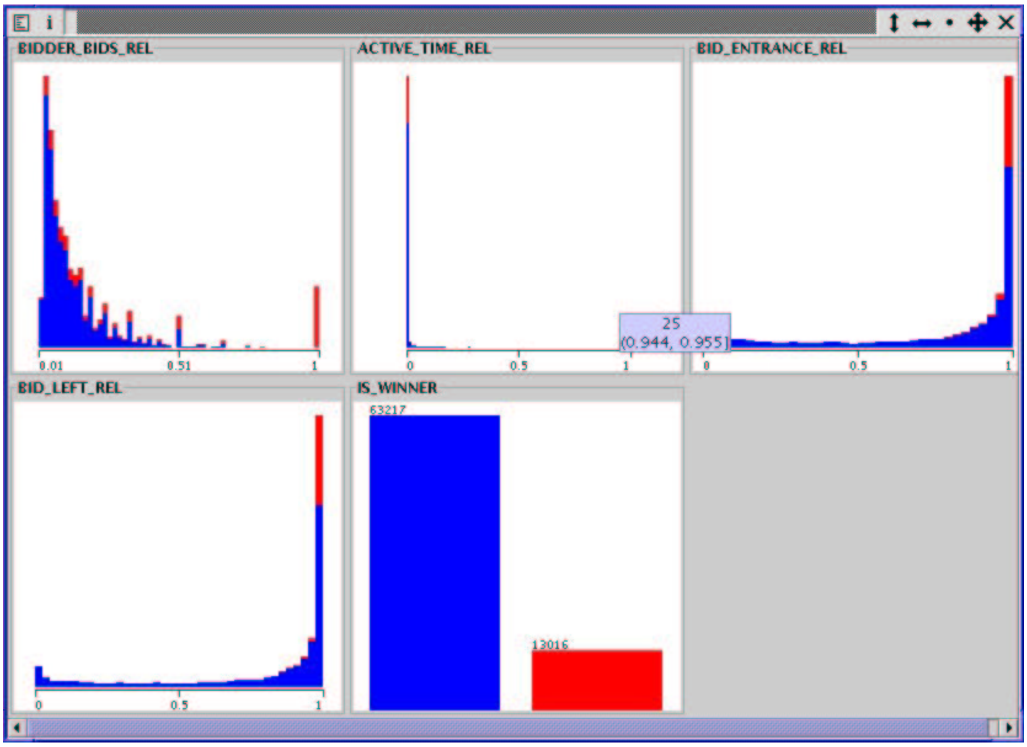


Figure 5.18: Dataset Analysis - Sony

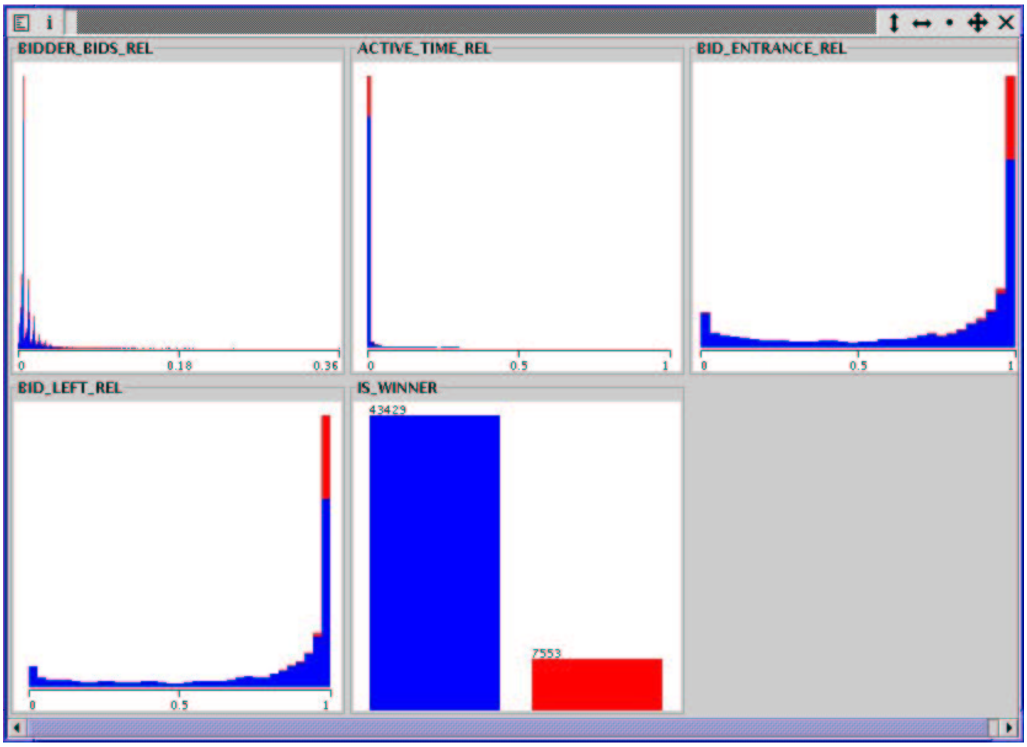


Figure 5.19: Dataset Analysis - Xbox

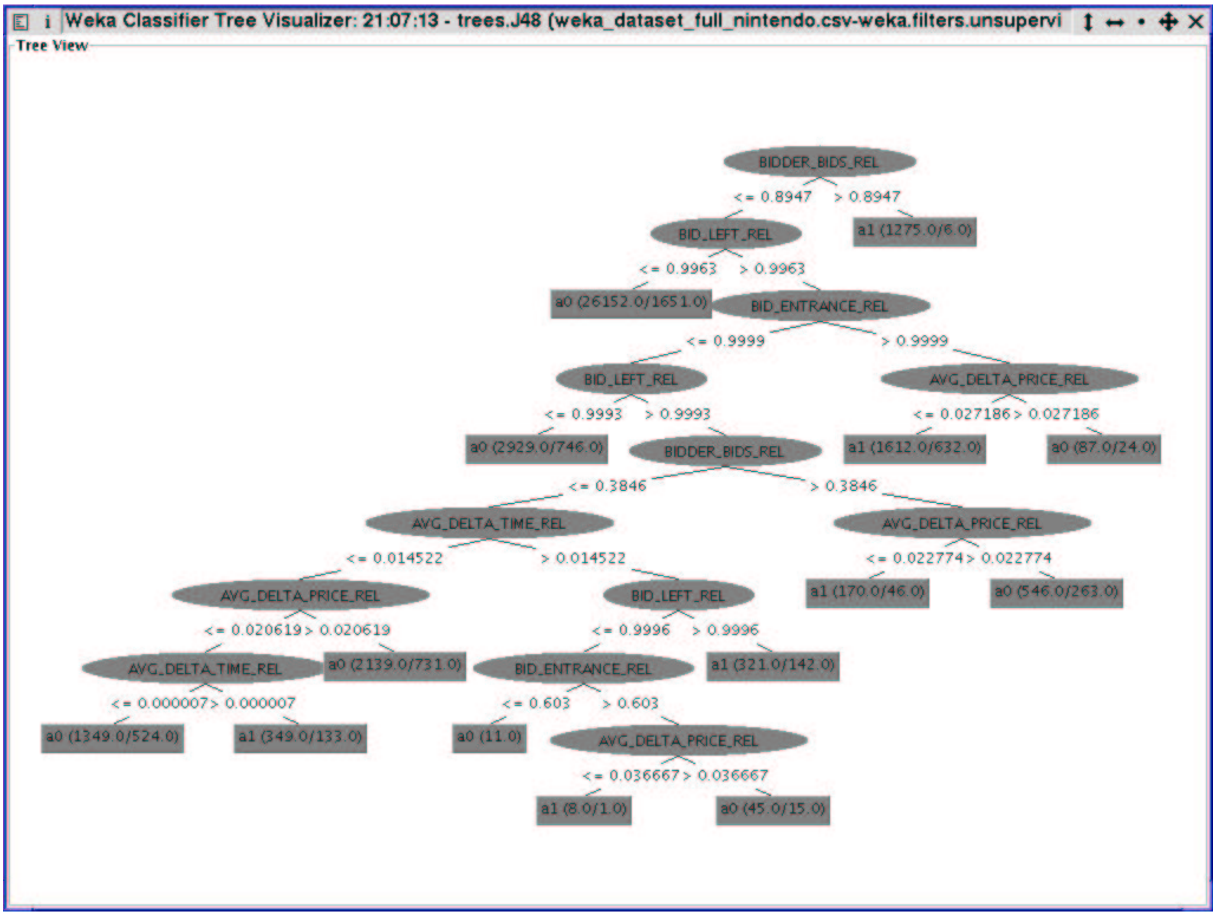
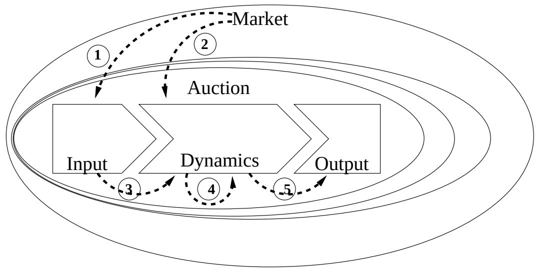


Figure 5.20: Classification Tree - Nintendo

5.2.1 Modeling Issues

In this section we present some key research questions we want to answer, showing how the ideas presented in this work can help us to achieve our main objective, which is the modeling of reactivity.

An auction is a very complete e-business application to characterize because it has many attributes and is very dynamic. The auction has an initial state defined by its input parameter, such as the beginning date, the starting bid, and payment method. The intermediary state is the phase when the negotiation occurs through the bids. The final state occurs after the negotiation and is characterized by its output parameters, such as the winning price, the winner bidder, and the number of bids.



1) How does the market affect the input?
2) How does the market affect the auction dynamics?
3) How does the input affect the auction dynamics?
4) How does the auction dynamics affect itself?
5) How does the auction dynamics affect the output?

Figure 5.21: Auction Dynamics - Reactivity

Figure 5.21 shows a high-level representation of the auction. There are five different dimensions to analyze reactivity in this case, associated with the numbered questions. We realized that the set of criteria that may be used to model reactivity is associated to one of the following entities:

- Auction: may affect the dynamics of the negotiation. The information about the auction as an input (See Figure 5.21 - number 3) and as dynamics (4) causes direct impact on dynamics.

- Bidder: participates in the auction during the negotiation phase affecting the dynamics (4).

- Bid: is the result of the bidder's action and also affects the dynamics (4).

- Seller: is the owner of the auction and her/his information (input) may affect the dynamics of the auction. This is represented by number 3.

- Market: is the set of all auctions and can affect the input definition of the auction, when the seller defines the auction parameters (1), and the dynamics, when the bidder decides when and how to make a bid (2).

It is important to explain that the output of the auction is related to all aspects that can change the dynamics, therefore all entities (and their characteristics) may have an impact on the output (Figure 5.21 - number 5).

These research questions are important in order to clarify why it is important to investigate reactivity in e-business applications, such as online auctions. The next section describes how we characterize and model online auctions.

5.3 Reactivity: Conceptual and Definition Levels

An auction is the process of buying and selling things by offering them up for bid, taking bids, and then selling the item to the bidder who gave the highest bid. The English auction is the most well known model. Participants bid openly against one another, with each bid being higher than the previous bid. The auction ends when no participant is willing to bid further, or when a pre-determined "buy-out" price is reached, at which point the highest bidder pays the price. The seller may set a "reserved" price and if the auctioneers fail to raise a bid higher than this reserved price the sale may not happen.

In an auction there are two entities, the buyer and the seller. The English auction is an ascending-price auction. Each auction instance represents the session of the auction engine. There are many states that an auction session may be assigned to: *Active*, *Active with Bids*, *Active with Buy-it-now Option*, *Active with Bids and Buy-it-now Option*, *Cancelled*, *Ended with Buy-it-now Option*, *Ended with Sale*, and *Ended without Sale*. Buy-It-Now is the simplest way to buy on eBay. It allows you to buy an item when you want it, at a known set price. When this option is set, the seller gives you the opportunity to purchase the item right away without waiting for an online auction to end for the defined price. The seller may create an auction session, cancel it, and set the “Buy it now” option. The buyer may place a bid, the most common action, or perform a “Buy it now” offer. There are several attributes that may affect the buyer’s action, such as: the number of bids, the current price, the seller’s feedback, and the payment method. Despite this, we have analyzed thoroughly several attributes in order to identify which of them are more semantically relevant to capture the user’s perception in terms of e-business. We identify some attributes that are more related to the e-business negotiation, such as the *time of the negotiation*, *winning price*, *who is winning the auction*, and *type of competition*. Therefore we have decided to adopted these attributes as perception criteria for our research in online auctions. These attributes will be described in the case study of online auctions, presented in Chapter 5.

Applying the reactivity model in this scenario we get the result presented in Table 5.8.

5.4 Characterizing Reactivity

5.4.1 Hierarchical Model

This section presents the basic components of our hierarchical model and characterization methodology, whose initial elements were presented in [96] and [97]. The premise of the characterization is to capture the relevant information about the auction negotiation features to understand its dynamics. These criteria describe the bidding at various levels of granularity and are organized as a hierarchy.

A reactivity model can be defined through *agents* who react to *events* through *actions* [95], similar to what has been presented in Chapter 3. In the online auctions environment, agents are bidders, their actions are their bids, and events are bids from other bidders. Another important concept in the reactivity model is the *time to react*. In online auctions, because of its long duration, the intermittent nature of the bidders’ visits to the

Element	**Value**
S	Created Active Active with Bids Active with Buy-it-now (BIN) Option Active with Bids and Buy-it-now (BIN) Option Cancelled Ended with Buy-it-now (BIN) Option Ended with Sale Ended without Sale
E	Buyer Seller
Ac	Buyer: MakeBid, MakeBuyItNowOffer Seller: Cancel, SetBuyItNow
P	Created - Seller - Cancel Active - Buyer - MakeBid Active - Seller - SetBuyItNow Active with Bids - Buyer - MakeBid Active with Bids - Seller SetBuyItNow Active with BIN Option - Buyer - MakeBid Active with BIN Option - Buyer - MakeBuyItNowOffer Active with BIN Option - Seller - Cancel Active with Bids and BIN Option - Buyer - MakeBid Active with Bids and BIN Option - Buyer - MakeBuyItNowOffer Active with Bids and BIN Option - Seller - Cancel
PC (related to Buyer)	Negotiation Time Winning Price Winning Bidder (who is winning?) Competition Type

Table 5.8: Reactivity Model - Auction

sites, and their own behavior characteristics, the time to react induces the occurrence of both synchronous and asynchronous reactivity periods. In the characterization model we present below, we have incorporated these ideas in the definition of the basic constructs of the model. The important objectives are to isolate, measure and explain periods of synchronous and asynchronous reactivity, how reactivity leads to competitiveness, and the instantaneous impact on the auction winning price, and the winner. These are all components necessary to understand the auction dynamics.

As illustrated in Figure 5.22, our characterization model for online auctions is organized as a five-level hierarchy: bid, session, sequence, auction, and market. The bid (represented by rectangles containing a number that identifies the bidder) is the lowest level, representing the bidder's action. A session is a group of one or more bids from the same bidder, in which the time interval between any two consecutive bids is below a threshold θ_{ses}. The session delimits activity intervals for each bidder. The sequence is a group of one or more sessions, where the inactivity period between two consecutive sessions is below a threshold θ_{seq}. Notice that $\theta_{ses} \leq \theta_{seq}$. Each auction is modeled by a group of sequences and the market is the set of all auctions. In Figure 5.22, the labels $T1$ and $T3$ represent the beginning of two auctions, and the end is defined by $T2$ and $T4$. In the case study presented in this work, we adopt θ_{ses} and θ_{seq} as 90 minutes.

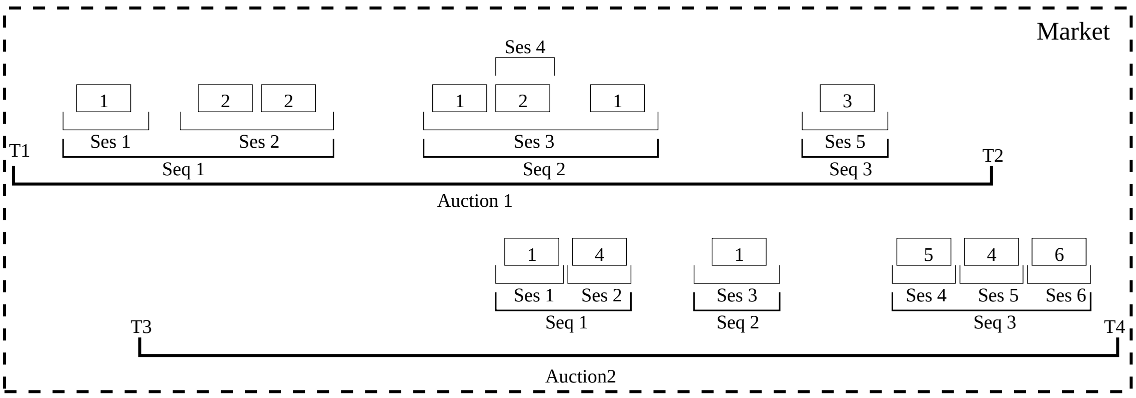


Figure 5.22: Auction Hierarchy

Previous auction characterizations [22, 21, 105] consider mainly the static aspects, such as the characteristics that define the auction's input, and some information about the negotiation and its results, such as the number of bidders, amount of bids, and winner price. The main novelty of our model is to introduce a hierarchy that captures detailed and rich auction features that enable us to study reactivity in a dynamic context, as showed by Figure 5.22.

We propose a four-level characterization: bid, session, sequence, and auction, as shown in Table 5.9. The bid is the finest grain level, representing the bidder's action. It is characterized by the time it is placed, the bid value and the bidder. The session attributes are intended to capture the reactivity the bidder exhibited towards the auction within the defined session interval: number of bids placed, the existence of competition, if the session impacted the sequence it is inserted in, and if the bidder is a recurrent one from previous sessions. We quantify the following sequence attributes that are related to reactivity: the time locality in terms of overall auction span, the amount of competition and whether the sequence resulted in a winner change. The auction is composed of one or more sequences.

Table 5.9 presents the hierarchical characterization. Based on the sequence attribute values, there are 15 valid combinations (since the 3 possibilities of *Time-Locality* = I and *Winner's Impact* = w are not valid) to describe patterns of auction's sequences. Considering the session's characterization, all the 32 possible patterns are valid. In order to simplify the sequence and session patterns representation, we adopt letters (lowercase or uppercase) as labels. For example, the sequence pattern IZW (initial sequences, with zigzag competition and winner changing) and the session pattern OStrW (session with one bid, serial competition, non-trigger, non-recurrent, and winner changing). It is important to emphasize that each criterion of both the sequence and session has mutually exclusive values.

To illustrate our characterization of sessions and sequences, consider one sequence of an auction that has two sessions and is the third sequence out of five. The first session has two bids from a new bidder in this auction. The first bid of the session does not change the winner, so the bidder decides to place a new bid, becoming the current winner of the auction.

This first session is classified to MSTrW according to our hierarchy. The second session has only one bid that changes the winner, from a bidder who has already participated in this auction negotiation. This second session has the pattern OStRW. Both of these sessions belong to a sequence that has the pattern MSW, that is, an intermediary sequence that has a successive competition and changes the winner of the auction negotiation.

This detailed characterization of bids, sessions, sequences and auctions provides a new approach for understanding the negotiation patterns and bidding behavior. By focusing on the sequences that comprise an auction, we can better understand the negotiation patterns that evolved in the auction. Similarly, when we focus on the sessions that an individual bidder participated in throughout an auction, his bidding behavior in that auction emerges. The next subsections present the characterization methodology for auction negotiation and bidding behavior.

The two next sections present the characterization methodology for Auction and Bidder, respectively.

Auction	Starting Price	The minimum bid value previously set by the seller.	
	Duration	The auction negotiation duration.	
	Seller	The owner of the auction.	
Sequence	Time-Locality	Initial (I)	The first sequence of the auction.
		Intermediary (M)	An intermediary sequence of the auction.
		Final (F)	The last sequence of the auction.
	Competition	No competition (N)	Does not present competition, only one bidder's session.
		Successive competition (S)	There is a competition, but no overlap between bidders' actions.
		Zigzag competition (Z)	Characterizes a more direct competition, where two or more bidders compete with each other in more than one occasion in the sequence.
	Winner's Impact	Do not change winner (w)	The sequence does not change the last winner bidder.
		Change winner (W)	The sequence changes the last winner bidder.
Session	Size	One (O)	Has just one bid.
		More (M)	Has more than one bid.
	Competition	Serial (S)	Does not overlap with any other one.
		Parallel (P)	Is concurrent with other(s), defining a parallelism.
	Activity	Non-Trigger (t)	Does not initiate the sequence's activity.
		Trigger (T)	Initiates the sequence's activity .
	Recurrence	Non-Recurrent (r)	The session is from a bidder who has not bid before in this auction.
		Recurrent (R)	The session is from a bidder who has already bid in this auction.
	Winner's Impact	Do not change winner (w)	The session does not change the last winner bidder.
		Change winner (W)	The session changes the last winner bidder.
Bid	Time	The time that each bid is placed during auction negotiation.	
	Price	The bid value.	
	Bidder	The participant who places this bid.	

Table 5.9: Description of Auction Model Hierarchy

5.4.2 Characterization Methodology for Auction Negotiation

As previously mentioned, each auction is composed of a set of one or more sequences. Each auction can therefore be described by a vector, whose 15 components are the types of possible sequences, and the values are the relative frequency of each sequence pattern (see Table 5.9). For example, if an auction has two sequences, out of which one is IZW and the other one is ISw, then this auction is represented by 50% of IZW, 50% of ISw and 0% of other sequence patterns.

To identify auction negotiation patterns, we group together vectors that exhibit similar distribution of sequence patterns by applying clustering algorithms [26]. The ideal number of clusters is determined through the metric beta-CV, as described in [79, 80]. We employed k-means [55] based on its computational efficiency[1]. Beta-CV denotes the intra-CV/inter-CV ratio for the clusters. While the intra-CV measures the coefficient of variation for the similarities intra-cluster, the inter-CV measure the similarities between different clusters. Thus, beta-CV is a measure of the quality of the clusters generated. The more stable the beta CV the better quality in terms of the grouping obtained. The ratio between the intra and intercluster variance, denoted beta-VAR is also useful in determining the quality of the clustering process (the smaller its values, the better the clustering set).

The frequency distribution of sequence patterns allows us to understand the overall auction negotiation patterns, however it is not possible to analyze the temporal aspects of the negotiation, that is, how the auction develops across time. This aspect is important to complement the analysis and also to allow the possibility to generate an online auction synthetic workload.

In order to provide a way to do this complementary analysis, for each identified cluster of auctions, we create an Auction Model Graph (AMG). AMG, that is based on Customer Behavior Model Graph (CBMG) [79, 80]. This is a state transition graph that has one node for each possible sequence pattern and the edges are transitions between these sequences.

[1] There may be better algorithms for determine the desired clusters, but we leave the investigation of the best algorithm as a future work direction.

A probability is assigned to each transition between two nodes, representing the frequency at which these two sequences occur consecutively in a cluster.

5.4.3 Characterization Methodology for Bidding Behavior

The characterization of online auctions organized as a hierarchy provides a new approach for understanding the negotiation patterns and the bidding behavior, two fundamental aspects to analyze the impact of reactivity. In this section we exploit a portion of the auction characterization to understand the bidding behavior.

Our overall strategy is to determine groups of bidders who are similar regarding their behavior, a typical clustering task. We may divide the strategy into two steps. In the first step we define the relevant dimensions for performing the clustering. In the second step we define the similarity metric that will be used in identifying behaviors that look alike.

In order to determine the relevant dimensions, we employ the sequence patterns that compose an auction in order to characterize auction negotiation. Such characterization is performed through the analysis of the nature of the sequences that compose a pattern and the causality among these sequences [97]. We also use the session patterns for each bidder. Our hypothesis here is that we should quantify the bidder behavior variability so that we are able to correlate the possible behaviors both to the scenarios where they happened and to the auction outcome.

The bidding behavior is then characterized by the distribution of session and sequence patterns, that is, the frequency of occurrence of each valid session and sequence pattern. There are 16 session patterns associated with the combination of the four attributes (*Size*, *Activity*, *Recurrence*, and *Winner's Impact*) described in Table 5.9 and 9 values associated with the sequence in which the session is (considering *Time-Locality* and *Competition*). Combining these 16 session patterns and these 9 session attributes inherited from its sequence, there are 144 possible session and sequence patterns. We also consider two additional variables: *ToE* and *ToX*. These two variables were used in [22] and stand for the time of entry and time of exit of the bidder in the auction. They are measured through the timestamp of the first and last bid respectively and normalized with respect to the auction duration, being mapped to a value between 0 and 1. We decide to consider these timing attributes, since it is very important to identify in which part of the auction negotiation a bidder starts and ends her/his participation. The bidding behavior exhibited by each bidder in an auction is represented by the 146-entry vector just described.

Since all values are numeric, our similarity metric is just the Euclidean distance between the vectors. It is important to note that we expect the vectors to be quite sparse, what helps in determining groups.

After we determine the vectors, we give them as input to a clustering technique to determine groups of bidding behaviors that are similar.

5.4.4 Applying Methodology to Overall Characterization

In this section we apply our model to an actual dataset that consists of 8855 eBay auctions comprising of 85803 bids for Nintendo GameCubes from 05/25/2005 to 08/15/2005. eBay [44, 11, 18] employs a non-trivial mechanism of second price auction, hidden winner, and hard auction closing, in a typical complex online auction environment that demands our characterization model.

A statistical analysis of this data shows that: the number of successful auctions is 75.7%; the number of unsuccessful auctions with bids is around 10%; the number of distinct sellers is high (5453), showing that auctions are not concentrated among a small number of sellers. The number of distinct bidders is also high (18073), indicating high level of competition. On the other hand, from this set of bidders, just very few of them become winners (735); the mean variation of price between new and used products is small, however the standard deviation of the prices is very high; and there is a significant number of bids per auctions (11.59), which indicates the level of competition during the negotiation. This is confirmed by the average number of unique bidders per auction, which is greater than 5.

Table 5.10 presents some auction information that is important to better understand its dynamics. We call successful an auction that has bids and sells the item, and the opposite outcome defines the unsuccessful group. Unsuccessful auctions are almost 10% of the auctions with bids.

Auctions	#	**#Seq**	**#Ses**	**#(Ses/Seq)**	**#(Seq/Auct)**	**#(Ses/Auct)**	T_{act}	T_{inact}
Success	6707	29575	45201	1.53	4.41	6.74	1.72%	98.28%
Unsuccess	640	2301	3187	1.38	3.59	4.98	1.00%	99.00%

Table 5.10: Auction Characterization - Overall Statistics

From Table 5.10 we can see that the average number of sessions per sequence is small, just 1.53, since it is common to find one or more sequences with one session in all auctions. On the other hand, the average number of sequences per auction shows that the dynamics of the negotiation is rich, which motivates our analysis. Another aspect we analyze is the active and inactive times of the auctions. The active time is the total time the auction has activity, that is the sum of the sequence times. We expected a short active time per auction, since there are usually long intervals between sets of bids, but an active time of just 1.72% is beyond our expectations. Further, the group of unsuccessful auctions has a smaller average number of sessions per sequence, sequences per auction, and average inactive time, which is just 1%.

Figures 5.23 and 5.24 show the histograms of patterns for auction sequences and sessions, respectively, for the successful auctions. As described by Table 5.9, sequences are represented by 3 letters and sessions by 5 letters. As can be seen, all fifteen sequence patterns have occurred. For session patterns, just 3 patterns (identified as OPTrw, OPTRw, and OPTRW) have not been observed in the auction's dataset. As can be seen in the histogram of auction sequences - Figure 5.23 - the most popular patterns for successful auctions are: MNW, MNw, INW, MSW, and FNW. The pattern INW is an initial sequence. The patterns MNW, MNw and MSW are intermediary sequences. And the patterns FNW is a final one. In terms of competition, from this group only the pattern MSW is characterized by competition. Regarding the changing winner aspect, only MNw do not cause changes in the winner bidder.

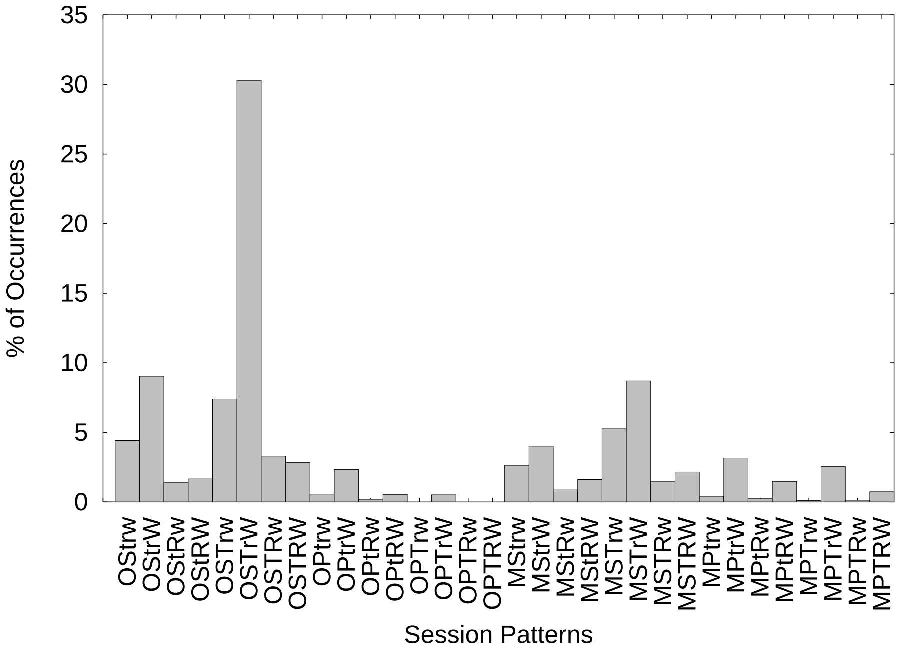


Figure 5.23: Histograms of Auction Sequences

Table 5.11 shows the frequency of occurrence of these most popular sequence patterns. We can note that these patterns represent almost 80% of the total number of sequences for both classes of auctions. Analyzing them to compare the successful and unsuccessful auctions, we reach some conclusions: the difference between the patterns MNW and MNw (7%), shows that unsuccessful auctions present a smaller probability of changing the winner bidder. The differences for the pattern MSW show that there is less competition in unsuccessful auctions, even when there is change in the winner bidder.

	Sequence's Pattern ID				
	MNW	**MNw**	**INW**	**MSW**	**FNW**
Successful Auctions	7540 (25.5%)	4857 (16.4%)	4493 (15.2%)	3685 (12.5%)	2676 (9.1%)
Unsuccessful Auctions	487 (21.16%)	439 (19.08%)	369 (16.04%)	225 (9.78%)	314 (13.65%)

Table 5.11: Auction's Sequences: Most Popular Patterns

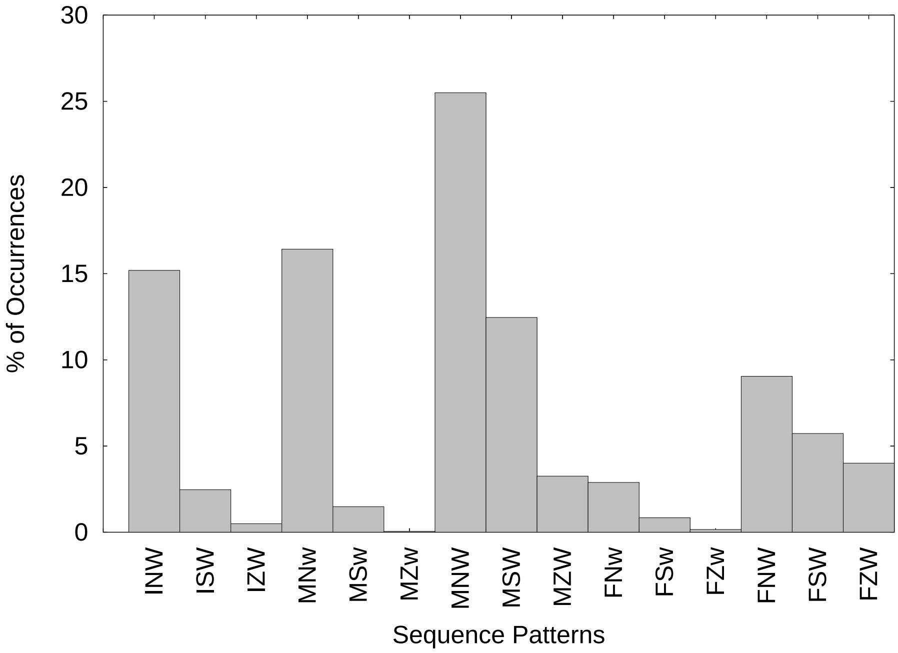


Figure 5.24: Histograms of Auction Sequences

Table 5.12 presents the statistics for auction's sequences classification. As can be seen, the group of successful auctions has 18%, 59%, and 23% of initial, intermediary, and final sequences, respectively. From this, we identify that 5% of the sequences are both initial and final, that is, the sequence is unique in the auction. In terms of competition, 69% of the auction's sequence have no competition. And 78% of the sequences cause change in the winner bidder. Comparing to the group of unsuccessful auctions, we observe a higher percentage of auctions that contain unique sessions (9%), less competition (76% of sequences have no competition), and smaller probability of changing the winner bidder.

	Time			**Competition**			**Change W**	
	I	**M**	**F**	**N**	**S**	**Z**	**w**	**Y**
Successful Auctions	18%	59%	23%	69%	23%	8%	22%	78%
Unsuccessful Auctions	19%	53%	28%	76%	21.5%	2.5%	28%	72%

Table 5.12: Auction's Sequences: Classification

In Figure 5.24, we can see that the most popular session's patterns for successful auctions are: OSTrW, OStrW, MSTrW, OSTrw, and MSTrw. The patterns OSTrW, OStrW

	Size		**Competition**		**Trigger**		**Recurrent**		**Change W**	
	O	**M**	**S**	**P**	**t**	**T**	**r**	**R**	**w**	**W**
Successful Auctions	64%	36%	87%	13%	35%	65%	81%	19%	28%	72%
Unsuccessful Auctions	60%	40%	95%	5%	28%	72%	90%	10%	36%	64%

Table 5.13: Auction's Sessions: Classification

and OSTrw correspond to sessions of size 1. MSTrw and MSTrW are patterns for sessions that have two or more bids. All of these patterns are characterized by non-overlapping with other sessions and no recurrence. Just pattern OStrW is composed of non-triggered sessions. Further, OSTrW, OStrW and MSTrW change winner.

Table 5.13 presents the classification statistics of auction's sessions. The group of successful auctions has 64% of one-sequence sessions. There is overlapping in 13% of sessions, 65% trigger, 19% present recurrence, and 72% of them change winner. Comparing to the group of unsuccessful auctions, we observe that there are less sessions with one bid in this group, despite it is also high (60%). The overlapping between sessions is rare, occurs in just 5% of sessions. As expected, there is less recurrence and changes in the winner bidder in unsuccessful auction's group. Finally, the percentage of trigger sessions is 10% higher than in the successful set of auctions.

Another important aspect to analyze is the mean second price variation for each auction sequence pattern, showed by Figure 5.25. Again, the patterns are represented by three letters: time-locality (I, M, F), competition (N, S, Z), and winner's impact (w, W). Analyzing these variations, we can understand part of the auction's dynamics, explaining how the second price varies according to different situations that our methodology enables to model. We can observe an ascending variation, like a stair shape, for the three initial groups of three patterns, showing a typical profile of competition versus price variation for the initial and intermediary sequences. The final sequences (the last six patterns in the graph) presents a different behavior: for the sequences that do not change de winner, the competition sequences are quite similar and have higher mean price variation than the other that

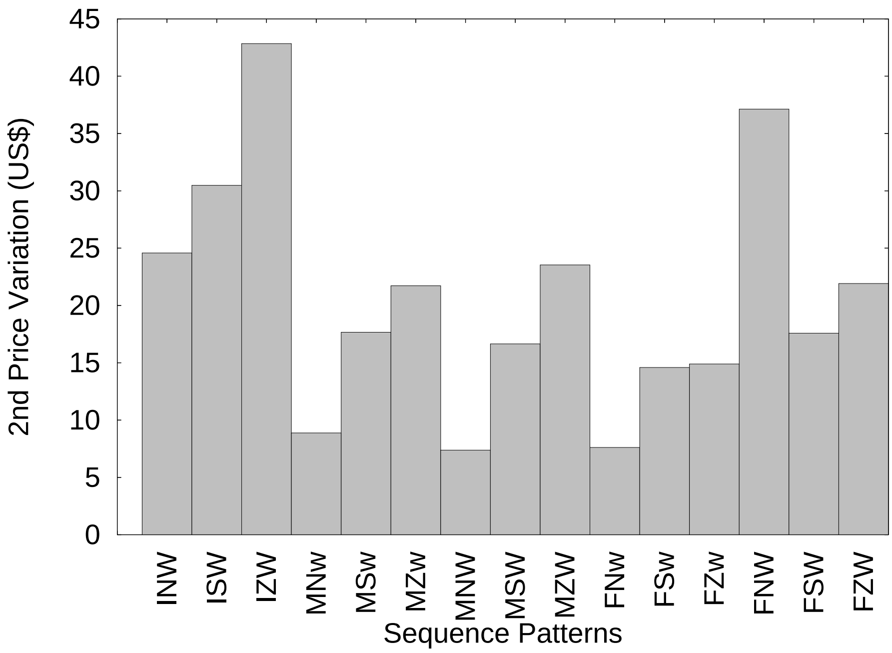


Figure 5.25: Auction Sequence Patterns: Mean 2nd Price Variation

do not have competition (FNw). In the case of the final sequences that change the winner (the three last sequence patterns in the picture), the result is completely different, as a consequence of the behavior of snipers, an important component of eBay auction [105]. As can be seen, the highest variations occur for patterns *IZW* and *FNW*, where exists zigzag competition and sniping behavior, respectively. On the other hand, the lowest variations happen for patterns *MNw*, *FNw* and *MNW*, characterized by absence of competition. This analysis is an example of how our model can help understanding the auction dynamics and is the basis for reactivity modeling, that we explain in Section 5.

Our preliminary analysis also challenges some research results that overemphasize the role of snipers in eBay auctions [105, 20]. According to these researchers, eBay activity between the start of the auction and the few moments before the end of the auction, tend to have limited effect on the final price. They have even proposed a “sealed bid” model to explain the sniping activity. Our results point out that reactivity in the initial and intermediary stages do matter.

The current bidding characterization around clusters of bidding strategies (e.g., jump bidders, evaluators, snipers, participators [21, 105]) have been defined based on static or aggregate measures, such as the total number of bids, time of first bid, time of last bid etc. We have found out that there is much more to bidders' strategies in terms of the dynamics of the competition, which can be captured through the reactivity concepts. For example, we have observed through our constructs the same bidder engaging in different strategies at different points in the same auction, including so-called snipers who have engaged in bidding sessions in initial stages of the auction.

The application of our characterization model using an eBay dataset demonstrates that our proposal provides a way to open the auction dynamics's "black box". We are aware that these results must be validated against a larger dataset and realize that we should characterize auctions using sequences, and bidders through sessions. By doing this, we will have a better semantic characterization, once the current work shows that it is difficult to explain some specific behavior through a general data analysis. The next sections apply the characterization methodologies for auction negotiation patterns and bidding behavior profiles.

5.4.5 Applying Methodology to Identify Auction Negotiation Patterns

This section presents the characterization of the auction negotiation. Following our methodology, we apply k-means clustering to identify similar auctions. To determine the number of meaningful clusters representing typical negotiation patterns in the dataset, we computed the dissimilarity measures beta-CV e beta-VAR in addition to the k-means error coefficient and visual inspection. The analysis pointed out 7 as the best number of clusters for auctions.

Table 5.14 shows the frequency distribution of the 15 possible sequences for the clusters. The last row of the table shows the percentage of auctions that falls in each cluster. We can describe each cluster as:

- **A0:** auctions with very small number of sequences, almost all of them unique and

Sequence	**Clusters**						
	A0	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**
1 (I-N-W)	0.00	0.00	19.99	47.17	0.00	15.46	15.56
2 (I-S-W)	0.00	40.82	0.66	0.00	0.00	1.09	1.13
3 (I-Z-W)	0.76	0.00	0.41	0.00	8.68	1.42	0.21
4 (M-N-w)	0.00	3.34	8.63	1.05	0.00	36.32	10.07
5 (M-S-w)	0.00	0.68	1.06	0.84	0.00	1.90	1.24
6 (M-Z-w)	0.00	0.00	0.00	0.00	0.00	0.04	0.09
7 (M-N-W)	0.00	5.78	13.99	1.09	0 .00	16.39	44.2
8 (M-S-W)	0.00	5.56	31.59	0.00	0.00	5.55	7.71
9 (M-Z-W)	0.00	2.98	2.62	2.68	0.46	2.66	2.88
10 (F-N-w)	0.00	6.37	2.75	11.55	0.00	3.01	2.18
11 (F-S-w)	0.00	2.23	1.04	1.53	0.00	1.12	0.76
12 (F-Z-w)	0.00	0.53	0.31	0.31	0.00	0.13	0.11
13 (F-N-W)	99.24	10.53	5.13	16.56	0.00	4.74	4.83
14 (F-S-W)	0.00	12.73	6.52	11.09	90.87	5.10	5.40
15 (F-Z-W)	0.00	8.44	5.31	6.12	0.00	5.06	3.62
Frequency (%)	19	4	16	12	1	20	28

Table 5.14: Distribution of Cluster Sequences

with no competition. All of them change the winner, as expected, once the first sequence always changes the winner in eBay. This is the third most frequent negotiation pattern in our dataset.

- **A1:** consists of auctions with medium level of activity, most of them with two activity moments. These auctions present a high competition level (74%), from which 84% is successive (see competition types in Table 5.9), and change the winner in 86.8% of their sequences. This pattern is rare, occurring in just 4% of auctions.

- **A2:** auctions with high activity, a significant amount of competition (49.5%), with predominance of successive type (8.3 in each 10 sequences with competition). Moreover, 86.2% of auction sequences changes the winner. 16% of the auctions have this pattern.

- **A3:** a set of auctions with characteristics similar to *A1* in terms of the number of auction sequences and winner changing. However, most of their sequences do not present competition (77.4%). Accounts for 12% of the dataset auctions.

- **A4:** represents auctions with predominance of low level of activity (sequences). Most auctions in this cluster have just one sequence. The competition level is maximum, with 90.9% of successive and 9.1% of zigzag competition types. As expected, all sequences change the winner. Only 1% of auctions present this pattern.

- **A5:** represents auctions with a high number of sequences with the presence of initial (I), intermediary (M) and final (F) patterns. They have only 24.1% of competition, divided in 61% of successive and 39% of zigzag competition. Almost 57.5% of sequences change the winner. It is the second most popular auction negotiation pattern (20%).

- **A6:** group of auctions with a large number of sequences, but with low competition level (23.2%). It is similar to *A5*, however the number of sequences that changes the winner is much higher, almost 86%. It is the most popular pattern, with 28% of auctions.

Aspects		Clusters						
		A0	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**
Inputs	Starting Price (US$)	71.4	36.4	20.9	47.1	43.3	16.9	17.7
	Duration (days)	2.7	4.5	5.1	4.9	4.9	5.7	5.8
Outputs	#Bids	1.1	9.7	16.3	4.8	5.4	15.8	17.1
	#Bidders	1.0	5.0	7.5	2.7	3.0	7.1	7.5
	1st Price (US$)	72.1	67.3	73.5	64.3	57.3	82.7	81.2
	2nd Price (US$)	71.9	65.6	71.8	59.9	56.2	80.3	79.3

Table 5.15: Auction analysis

Once determined the seven auction clusters, we analyze the relationships between auction inputs and outputs with the negotiation. Table 5.15 shows some important aspects for each cluster. It presents two auction negotiation inputs (aspects defined before the negotiation initiates - starting bid and duration) and four outputs (aspects determined after negotiation ends - number of bids and number of bidders, 1st Price and 2nd Price). Every measurement represents the average value of the attribute.

A0 has the highest *starting price* and the shortest *duration*. In terms of its dynamics, we previously identified low activity and competition. However, it is interesting to note that these auctions achieve a high winner price (the *AVG 2nd price* is US$71.9). These can be explained by the fact that they present a very high *starting price*, very close to the final price obtained.

A3 and *A1* have similar characteristics, but different behavior in terms of competition profile, as previous explained. It is important to note that they produce different results: the average number of bids and bidders for *A3* is almost half of *A1*, which can be demonstrated by the competition level. Moreover, the final negotiation price is almost 10% higher for auctions of *A1*.

A4 has an average duration of almost 5 days and an average starting price of US$43.30. In these set of auctions we identify very low activity during negotiation and very high competition level, that is, there are small number of sequences, where competition dominates. The mean number of bids is only 5.4 (associated to 3 bidders). This cluster has the lowest *2nd price*, which can be explained in part by its low activity, that occurs mainly at the end of the negotiation.

A5 and *A6* present similar characteristics in terms of inputs and outputs, despite the differences in their typical dynamic aspects as presented in Table 3. *A2* also has some similarities with *A5* and *A6*, despite the starting bid 15% higher and duration 15% shorter than them. It is interesting to note that the final prices of *A2* are 10% smaller than *A5* and *A6*. We believe that what determines these results are directly related to the bidding behavior of the agents that participate in these auctions. We will investigate these aspects again in next sections, correlating bidding behavior to auction negotiation characteristics.

5.4.6 Applying Methodology to Identify Bidding Behavior Profiles

This section describes our bidding behavior characterization that follows the proposed methodology already presented in this doctoral dissertation.

According to our methodology, we apply a clustering technique to identify similar bidding behavior profiles based on the distribution of session patterns. We also consider the time-locality and competition aspects from the sequence in which the session is inserted to characterize the bidders. As previously explained, we also employ the attributes *ToE* and *ToX* to our feature vector.

Figure 5.26 shows beta-CV and beta-VAR for bidding behavior clusters. Beta-CV denotes the intra-CV/inter-CV for the clusters. Whereas the intra-CV measures the coefficient of variation for the similarities intra-cluster, the inter-CV measure the similarities between different clusters. Thus, beta-CV is a measure of the quality of the clusters generated. The more stable the beta CV the better quality in terms of the grouping obtained. The ratio between the intra and intercluster variance, denoted beta-VAR is also useful in determining the quality of the clustering process (the smaller its values the better is the clustering set). We analyze each number of clusters using these measures and also the k-means error coefficient as well as visual inspection. The analysis pointed out 16 as the best number of clusters for bidding behavior.

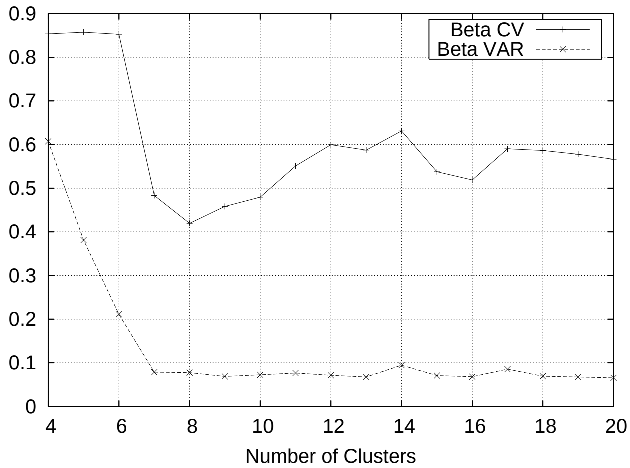


Figure 5.26: Bidding Behavior Clustering - beta-CV and beta-VAR

Next we present an analysis of each of the cluster, where we adopt the notation *Bn* to identify each bidding behavior cluster, where n varies from 0 to 15. Following our analysis, we present the characterization of each of these groups of bidders:

- **B0**: bidders who act in initial (53%) or intermediary (44%) negotiation sequences, in the presence of competition in 70% of the sequences, from which more than 80% have successive type. Most of their sessions have just one bid (57%), are triggered (63%), non-recurrent (74%), and change winner (76%). They act during the earliest stages of the auction negotiation (24-33% of duration time), placing 2.6 bids in average. These bidders represent 4.3% of bidding behaviors.

- **B1**: these bidders mainly act in intermediary sequences of the auction (91%) and in competitive situations (91%) with successive type predominance. Most of their sessions have more than one bid (89%), are triggered (91%), non-recurrent (76%), and change winner (92%). They act late in the auction negotiation (85-88% of duration time), placing 3.4 bids in average. They represent only 1.8% of the bidders.

- **B2**: bidders who act in the last auction sequences, 74% with competitive situation with 40% of zigzag competition type. Their sessions have more than one bid in 68%, are triggered in 76% and change winner in 65%. Moreover, they are bidders who have not participated in the negotiation yet, called non-recurrent bidders. They represent 11.5% of the bidding behaviors, act after 99% of auction negotiation timing, placing 2.4 bids in average.

- **B3**: bidders that act typically in intermediary sequences (93%), in scenarios with no competition (85%). In general, their sessions have only one bid (89%), are triggered (88%), and change winner (89%). Only 28% of them have already participated in the current negotiation before (recurrent). They act typically after the middle of the auction negotiation, from 72 to 78% of negotiation timing duration. They are a popular class, occurring in 12.9% of the bidding behaviors. They place 1.7 bids in average.

- **B4**: group of bidders that act typically in the intermediary (48%) and final (36%) sequences, in competition situations (95%), with predominance of zigzag competition type (60%). Their sessions have one bid in 52% of occurrences (when normally successive competition takes place) and more than one bid in 48% (when zigzag is observed). Also, 68% of their session are non-triggered and they have not participated yet in negotiation in 87%. They change winner in 62% of their sessions. They place on average 2.3 bids and act during the final moments of the auction duration (after 95% of timing duration). These group is the most frequent one, happening in 16.2% of the behaviors.

- **B5**: bidders who act typically in the first sequence (87%) of the auctions and never in the last one, in situation with no competition (96%), placing only one bid (94%) per session. As expected from the previous characteristics, their sessions are triggered. Moreover, they are non-recurrent in 87%, change winner in 94%, and place only one bid. This behavior happens in 5.2% and during the initial phase of the negotiation (from 23 to 25% of auction timing duration).

- **B6**: bidders that act typically in intermediary sequences (96%), in scenarios with no competition (94%). In general, their sessions are triggered (96%), however they do not change winner (91%). Only 18% of them have already participated in the current negotiation before (recurrent). They act typically after the middle of the auction negotiation, from 71 to 75% of negotiation timing duration. This behavior happens in 4.7%. Different from bidders from *B3*, they place 3.6 bids in average.

- **B7**: bidders that act typically in intermediary sequences (97%), in scenarios with no competition (95%). Similar to *B6*, their sessions are triggered (97%) and they do not change winner (94%). Only 13% of them have already participated in the current negotiation before (recurrent). They act typically after the middle of the auction negotiation, from 73 to 75% of negotiation timing duration. This behavior happens in 6.4%. Different from bidders from *B6*, their sessions are typically of size 1 and they place 1.2 bids in average in auction negotiation.

- **B8**: these bidders mainly act in initial (38%) and intermediary (56%) sequences of the auction in activity moments where no competition is typical (77%). Most of their sessions have one bid (73%), are triggered (84%), recurrent (62%), and change winner (74%). They act through auction negotiation, participating from 27 to 84% of the timing duration of the negotiation, placing typically 4.3 bids in average. Moreover, the bidders from this group have, in average, 2.6 sessions per auction, the highest value. They represent only 1.5% of the bidders.

- **B9**: these bidders mainly act in intermediary sequences of the auction (92%) and in situations where no competition predominates in 80%. Most of their sessions have more than one bid (86%), are triggered (85%), non-recurrent (68%), and change winner (86%). They act late in the auction negotiation (75-81% of duration time), placing typically many bids, 4.2 bids in average. They represent 6.4% of the bidders.

- **B10**: bidders who place bids in intermediary sequences (94%) and the rest in final ones, and their sequences are typically with successive competition (95%). Their sessions have one bid (95%), are non-triggered (93%), non-recurrent (90%), and do not change the winner (95%). They act very late in the auction negotiation (between 90 and 92% of duration time), placing typically 1.2 bids in average. They represent 2.6% of the bidders.

- **B11**: bidders who act mainly in the first sequence (75%) of the auctions, in situation with no competition (82%), placing only one bid (89%). As expected from the previous characteristics, their sessions are triggered. Moreover, they are non-recurrent in 75%, change winner in 90%, and place only 1.6 bids in average. This behavior happens in 4.9% and during the last phase of the negotiation (from 83 to 87% of auction timing duration).

- **B12**: bidders who place bids in intermediary sequences (84%) and in final ones (16%), from which their sequences are typically with competition (87% of successive and 9% of zigzag competition pattern). Their sessions have more than one bid (90%), is non-triggered (90%), non-recurrent (75%), and change the winner (90%). They act very late in the auction negotiation (between 92 and 94% of duration time), placing 3.6 bids in average. They represent 2.6% of the bidders.

- **B13**: group of bidders that participate in intermediary sequences (89%) and in final ones (11%), from which their sequences are typically with competition (94%), with high predominance of successive type. Their sessions have one bid (91%), present a balanced behavior in terms of trigger, are non-recurrent (72%), and change the winner (92%). This group acts very late in the auction negotiation (between 91 and 93% of duration time), placing only 1.5 bids in average. They represent 9.0% of the bidders.

- **B14**: bidders who act in the last auction sequences, 67% in sequences without competition. Their sessions have always only one bid, are triggered in 2/3, and always change winner. As expected, all of them is non-recurrent. They represent 8.5% of the bidding behaviors, act very late in the auction, close to 99% of auction negotiation timing, placing 1.0 bid in average, as expected.

- **B15**: bidders who place bids in intermediary sequences (98%) and their sequences are typically with successive competition (96%). Their sessions has one bid (98%), are triggered (98%), non-recurrent (94%), and do not change the winner (97%). They act very late in the auction negotiation (around 89% of duration time), placing typically 1.1 bids in average. They represent 1.6% of the bidders.

In order to summarize the characteristics of each bidding behavior cluster, we present in Figure 5.27 a hierarchical organization in which each level represents a bidder session attribute, in top-down order by information gain [72]. As can be seen, our bidding behavior characterization provides many details useful for sake of investigation and this hierarchical organization facilitates the analysis.

In Table 5.16, the *Aspects* columns are measures of outcomes for each cluster and *F* presents the percentage of occurrence for each one. Comparing the results from Figure 5.27 and Table 5.16, it is possible to understand some interesting aspects about auction negotiation and bidding behavior. *BR* measures the bidder reputation in terms of experience in eBay auctions, *NoB* is the number of bids, δ *1st* is first price variation, δ *2nd* is second price variation, *ST* is the bidder session time (average duration), *S/B* is the average number of sessions per bidder and, finally, *W* is the percentage of bidders that are winners. The attributes *ToE* and *ToX* have already been explained.

In order to illustrate the analysis potential of our new approach, we are going to present an example of a specific analysis of bidding behavior and then some conclusions about the bidding behavior profiles and their relations to some auction negotiation characteristics that we have also analyzed in our online auctions research.

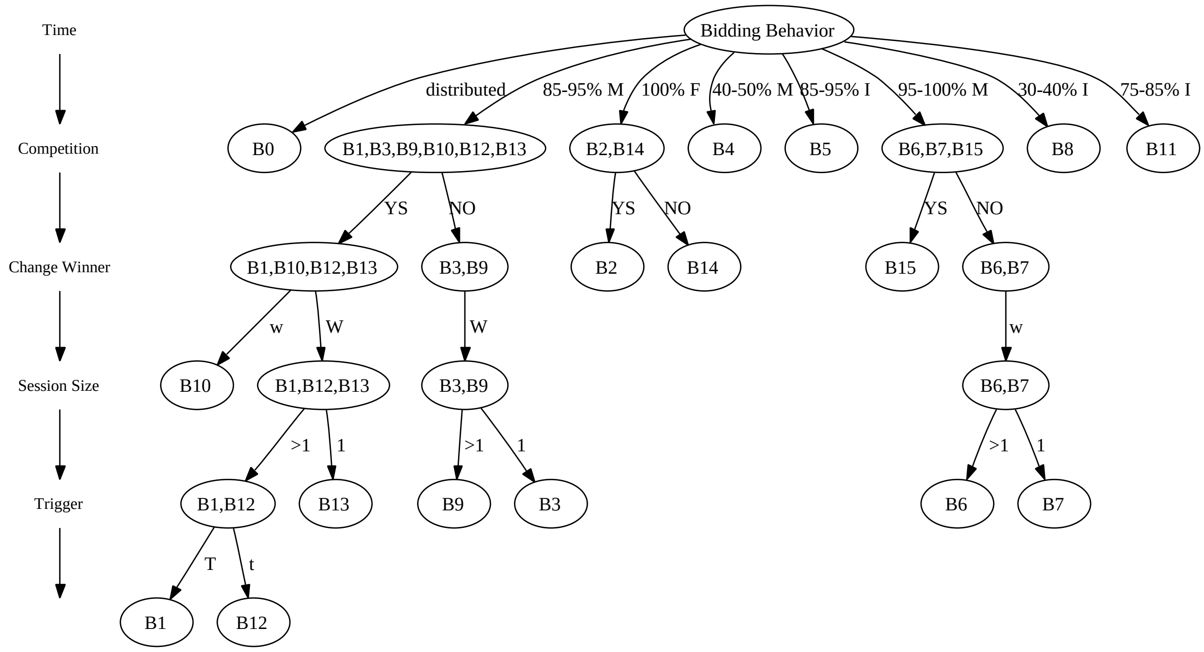


Figure 5.27: Bidding Classification Tree

Bidders from clusters *B2* and *B14* have similar behavior in terms of time-locality, but they have a completely different competition profile (see Figure 5.27 and Table 5.16).

Bidders from *B2* have more than 70% of sessions with competition, from which 60% are successive. *B14* has bidders with 60-70% of sessions without competition. Both groups presents similar characteristics in terms of *ToE* and *ToX* and without our new detailed approach it would not be possible to differentiate between them.

The bidders' experience of *B2* is more than three times the value of *B14*. Bidders from *B2* have placed 140% more bids than *B2*. In terms of 1st price variation they have similar behavior, but bidders from *B14* have an average 2nd price variation almost three times the value of *B2*. This fact suggests a willingness to pay more than it is necessary to win. Bidders from *B14* usually place only one bid close to the end of auction negotiation. Considering the auction winning rate, bidders from *B2* present better conditions to win, 27%, against 8% from *B14*. It is important to emphasize how important may be the correlation analysis of bidder behavior and auction negotiation patterns, once there are many aspects that we can not clearly explain without considering the scenario where the bidder is acting, what suggests the importante of reactivity investigation.

Id	**Aspects**									**F**
	BR	*NoB*	**δ1st**	**δ2nd**	**ST**	**S/B**	**W**	*ToE*	*ToX*	(%)
B0	82.8	2.6	10.8	10.1	178	1.3	4	2.43	3.34	4.3
B1	44.7	3.4	7.9	9.1	210	1.3	6	8.56	8.85	1.8
B2	116.5	2.4	6.0	10.9	218	1.0	27	9.90	9.91	11.5
B3	160.7	1.7	11.3	5.8	54	1.4	9	7.22	7.78	12.9
B4	142.6	2.3	5.6	9.97	381	1.1	14	9.60	9.69	16.2
B5	174.1	1.3	11.4	13.4	11	1.1	2	2.33	2.55	5.2
B6	57.3	3.6	0.6	11.0	180	1.2	2	7.14	7.42	4.7
B7	150.9	1.2	0.4	6.9	21	1.1	1	7.32	7.52	6.4
B8	63.8	4.3	6.9	9.7	74	2.6	13	2.77	8.44	1.5
B9	51.7	4.2	8.1	11.7	242	1.5	9	7.56	8.12	6.4
B10	123.2	1.2	0.4	5.3	27	1.1	2	9.07	9.16	2.6
B11	119.0	1.6	6.6	28.9	52	1.3	13	8.33	8.67	4.9
B12	47.7	3.6	7.3	9.6	299	1.3	13	9.24	9.42	2.6
B13	112.4	1.5	9.5	4.6	38	1.3	10	9.11	9.33	9.0
B14	491.8	1.0	7.0	29.1	0	1.0	8	9.89	9.89	8.5
B15	92.6	1.1	0.2	5.2	13.1	1.1	1	8.90	8.97	1.6

Table 5.16: Bidding Behavior Analysis

Analyzing Table 5.16, we see that the most frequent clusters are *B4* (16.2%), *B3*

(12.9%), *B2* (11.5%), *B13* (9%) and *B14* (8.5%). Together they correspond to almost 60% of the amount of bidders. The rarest bidding behavior profiles are *B8*, *B15* and *B1*, with 1.5%, 1.6%, and 1.8%, respectively.

In terms of auction's winner, *B2* has the highest rate (27%). Analyzing session duration, we identify that bidders from *B4* have longer sessions, around 6 minutes in average. The bidders from *B12* also stay longer active, around 5 minutes in average, followed by *B9*, who spends around 4 minutes. As expected, the shortest sessions, shorter than 1 minute, belongs to bidders who usually have session with only one bid, such as *B3*, *B5*, *B7*, *B10*, *B13*, *B14*, and *B15*.

In terms of second price, which defines the winner price to pay in eBay, bidders from *B14* have the highest inter-bidding price (US$29.1), followed by *B11* (US$28.9).

It is very interesting to note that bidders that would be mapped to the same categories (e.g., *Opportunists*) defined by the taxonomy of [22], have different characteristics, which emphasizes the benefits of our bidding behavior clustering approach. This fact can be clearly observed comparing *B2* and *B13*. In Figure 5.27 we can see that they are different in terms of time-locality and competition. Both of them would be considered *Opportunists* by [22]. However, our characterization identifies that bidders from *B13* participate in almost 90% of intermediary sequences, not the last auction competition, while *B2* represents bidders who have chosen exactly the last sequence of the auction to place bids and to present a high competition profile.

This analysis shows the completeness and accuracy of our bidding behavior characterization. Moreover, our approach has uncovered two additional important facts that motivate us to further correlate bidding behavior with auction negotiation patterns:

- There are bidders who belong to the same categories proposed in the literature, who present different behavior profiles using our new hierarchical model.
- There are bidders who present different behaviors in different auctions.

We emphasize two important aspects of the analysis so far: (1) using the attributes from our hierarchical model in a detailed data mining process results in more detailed and accurate bidding behavior clusters; and (2) the bidding behavior profiles we achieve with our approach are able to capture dynamic behavior and nuances that are ignored by the literature, such as the taxonomy of [22].

Through our case study, it is demonstrated that the criteria adopted by our characterization methodology are efficient to differentiate the bidding behavior profiles, considering the proposed clustering strategy. This is demonstrated by the classification tree showed in Figure 5.27.

Despite the differences of each group of bidders, in general we identify groups with predominance of competition or no competition. The same is observed for the changing winning attribute. In terms of the time of bidding, it varies a lot. There are bidders that act in the beginning of the auction negotiation (such as *B0* and *B5*), other ones who act between the middle and the last stage of the auction (such as *B3*, *B6*, *B7* and *B9*), and others who act in the last stage (such as *B1*, *B4*, *B10*, *B11*, *B12*, *B13* and *B15*). Moreover, there are bidders who place bids in the last-minute stage, known as “snipers” by the literature (such as groups of bidders *B2* and *B14*). We identify also bidders who place bids from the beginning to the end of the auction (such as *B8*).

Analyzing all these bidding behaviors, we try to identify how to explain a behavior, what causes the bidder to act in one way or another one. We conclude that some bids may be explained by auction parameters (such as the *starting price*), other are determined by the competition scenario of one auction sequence. Nevertheless, we know that part of the bidding behavior is reactive (e.g., a bidder reacts to a competitor’s bid which overcome his/her winner bid, placing another bid in the same auction sequence, or a bidder who reacts to the last-minute chance, trying to win the auction negotiation) and may be modeled and the other one is not. The next challenge is to work on the correlations between bidding behavior and auction negotiation patterns with the objective of identifying these correlations.

Our analysis also challenges some research results that overemphasize the role of snipers in eBay auctions [105, 20]. According to these researchers, eBay activity between the start of the auction and the few moments before the end of the auction, tend to have limited effect on the final price. They have even proposed a “sealed bid” model to explain the sniping activity. Our results point out that reactivity in the initial and intermediary stages do matter.

5.4.7 Correlating Auction Negotiation Patterns and Bidding Behavior

This section presents an analysis of the correlation between the auction negotiation patterns (identified in Section 5.4.5) and bidding behavior (characterized in Section 5.4.6).

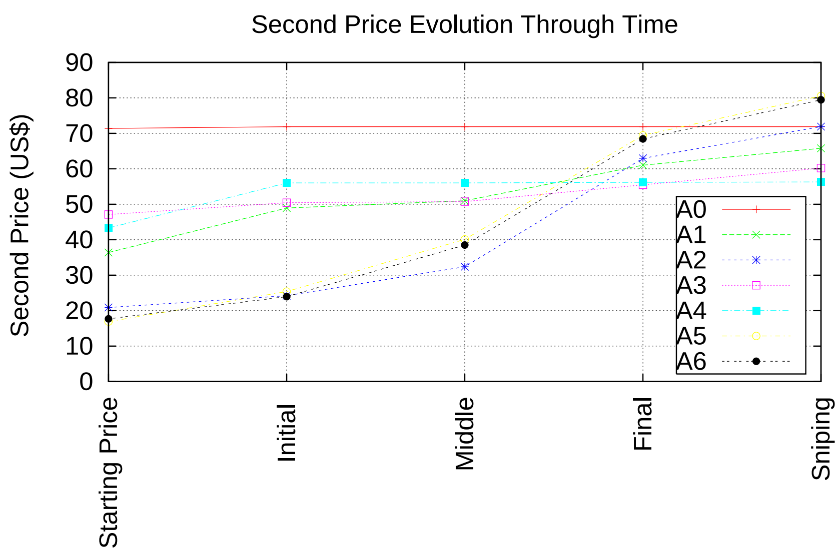


Figure 5.28: 2nd Price Evolution Through Time

Figure 5.28 shows the 2nd price evolution through time for each auction cluster. There are five time parts we divided in this work, that we call checkpoints: Starting Price (first auction state), Initial, Middle, Final (before sniping - 10 minutes to the end), Sniping (last auction state). Cluster 0 has a singular tendency. Clusters 2, 5 and 6 have a similar behavior, that we will try to explain using the bidding behavior observed in each of them.

Id	**Auctions**						
	A0	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**
B0	1.2	7.2	3.8	0.7	0.9	4.8	4.6
B1	0.0	0.7	4.0	0.0	0.0	1.2	1.5
B2	1.5	15.2	10.1	27.6	36.2	11.5	9.8
B3	0.0	2.4	6.8	0.7	0.0	10.8	22.9
B4	4.0	57.0	15.2	20.6	41.6	15.3	13.7
B5	0.0	0.0	4.8	4.0	0.0	6.4	6.0
B6	0.0	0.8	2.1	0.6	0.0	11.4	3.1
B7	0.0	1.2	3.3	0.3	0.0	14.2	4.6
B8	0.0	0.0	1.2	0.3	0.0	2.0	2.0
B9	0.0	1.3	3.3	0.5	0.0	4.4	11.3
B10	0.0	1.7	5.2	0.4	0.0	2.2	1.8
B11	0.0	0.0	5.6	29.5	0.0	2.8	3.3
B12	0.0	1.0	6.1	0.0	0.0	1.6	2.0
B13	0.0	4.0	21.5	0.0	0.0	5.6	7.6
B14	93.2	6.6	4.1	14.2	21.3	4.5	4.4
B15	0.0	0.8	2.9	0.5	0.0	1.4	1.3

Table 5.17: Bidding Behaviors versus Auction Negotiations

Table 5.17 presents the relative number of occurrences of each bidding behavior profile over the auction negotiation patterns of eBay case study.

In *A0* we observe predominance of *B14*, which explains the typical small number of sequences in the auctions e the predominance of them in the last minutes of negotiation. The high starting price (US$71.4) and short duration (2.7 days) can explain why this bidding behavior is identified. It is interesting to note that the winning price is very close to the starting price, only a variation of US$0.5. This auction pattern attracts bidders who are determined to win and have a good idea of what price to offer. Moreover, a few number of bidders from *B4* (4%) participates in these auctions, trying to win the auction in the last stages. They compete with bidders from *B14*, in zigzag and successive competition types. As the starting price is so high, it repeals other bidding behaviors.

Auctions from *A1* have major participation of *B4* and *B2*, which together represents around 72% of the bidders in this type of auction. The other bidding profiles are *B0* (7.2%) who probably were attracted by a winning price under the market value during the intermediary part of the negotiation, and *B14* (6.6%) and *B13* that participate in the last minutes of auction negotiations, usuaaly with a one-chance bid. These auctions have high level of competition, which is explained by the predominance of bidders from *B4* and *B2*. The competitive price attracts a significant number of bidders (5), who contribute with 9.7 bids per auction in average. The winning price achieves US$65.6, 5% less than the average winning value of all negotiations (U$69.3).

In *A2* we observe the predominance of *B13* (21.5%) and *B4* (15,2%), which explain the typical successive competition observed in almost half of the negotiations. This group of auctions attracts the highest number of bidders (7.5), who place 16.3 bids per auction together. These can be explained by the low starting price (US$20.9) and the long duration (5 days). The winning price is a little more than the average winning price, almost the same of *A0*, despite the completely different negotiation patterns and bidding profiles.

Auctions from *A3* have participation of groups of bidders who typically place bids at the end of the negotiation. There is a group who participates in the last minutes (*B2*, *B4*, and *B14* and represents more than 62% of the participants. The other group is represented by bidder from *B11*, who activates the auction's activity in the beginning of the final stages. In general, each auction has 3 bidders, placing one or two bids each one. These small number of bidders per auction explains the low competition level and, consequently, the low wining price (US$59.9) - 14% lower than the average winning price.

Auctions from *A4* are very similar to *A3* in terms of starting price, duration, number of bidders and bids, and winning price (both present a variation of almost US$23 from starting and winning prices). In terms of participants, both of them presents significant amount of bidders from groups *B2*, *B4* and *B14*, that participate in the final stages of the auction negotiation. In *A4* these groups are more frequent, achieving 99%, 50% more than *A3*. The difference is that in *A3* there is almost 30% of *B11*. The presence of more bidders from *B4* in *A4* explains why it presents 10% more bids and bidders than *A3*. The occurrence of *B11* in *A3* can explain why these auctions achieve a 1st price 10% higher and a 2nd price 6% higher than *A4*, since these bidders participate earlier than the other frequent groups in the auction negotiation. In this auction we verify the lowest winning price of (US$56.2), almost 20% less than average winning price.

A5 and *A6* represent auctions with similar characteristics in terms of starting price (US$16.9 and US$17.7), which are the lowest starting prices comparing to the other groups, duration (5.7 and 5.8 days), which are the longest durations among all groups, high number of bidders (7.1 and 7.5) and bids per auction (15.8 and 17.1), and high winning prices (US$80.3 and US$79.3), which are the highest values. Moreover, these groups are very similar in terms of number of sequences and competition patterns (around 25% of competition). These groups are very popular; they represent together almost half of the auction negotiations. The difference between them relates to the number of sequences that changes the winner. Auctions from *A5* change the winner in 57.5% of sequences, while in *A6* the percentage is 86%. Despite this, the outcomes are similar and the price evolution too, as previously showed.

Our challenge is how to explain the difference in winner changing, despite all similarities presented. This explanation comes from analyzing the bidding behavior of these clusters.

We observe, according to Table 5.17, that 8 from 16 bidding behavior profiles are more frequent in *A5* and *A6*: *B8* (11.5% and 9.8%), *B3* (10.8% and 22.9%), *B4* (15.3% and 13.7%), *B5* (6.4% and 6.0%), *B6* (11.4% and 3.1%), *B7* (14.2% and 4.6%), *B9* (4.4% and 11.3%), and *B13* (5.6% and 7.6%). Together they represent around 80% of the bidder's actions. Analyzing them we find out that the main differences in terms of bidding behavior are in pair of clusters *B3* and *B7*, and *B6* and *B9*.

Comparing *B3* and *B7* (see Table 5.16), we can see that bidders from *B3* causes a US$11.3 variation in first price, while the ones from *B7* vary it just US$0.4. Also, comparing *B6* and *B9*, we can see that bidders from the first one vary the first price in US$0.6, while *B9* presents a US$8.1 variation. These observations confirm why *A5* changes winner less frequently than *A6*, once the group of bidders from *B3* and *B7* are less frequent in auctions of *A5*, and bidders from *B9* are more frequent in auctions of *A6* than *A5*. However, how these difference price variations culminate in similar winning prices still remains unexplained. This motivates us investigate more details about the clusters. We conclude from this analysis that, despite the different first price variations observed in the bidding behaviors of *A5* and *A6*, which explains the winner changing difference, the second price variation of them is similar, leading to the same outcome.

In order to complete this analysis, we have to compare them with *A2*. Analyzing Figure 5.28 explanations, we realized that *A2* has less frequency of *B5* and *B0*, while *A5* and *A6* have almost the same amount of them. This explains why the second price of *A2* rises less than *A5* and *A6* until *Middle* time checkpoint. After that, they present similar proportion in terms of price evolution.

As can be seen through our case study, our approach considers detailed aspects related to auction negotiation dynamics, and also identifies the bidding behavior with a wealth of details. Furthermore, we can correlate the auction negotiation patterns with bidding behavior profiles, providing new ways of understanding how the negotiation characteristics affect the bidders and vice-versa.

In the next section we present how we model reactivity, using the concept of reactive transitions, that will be presented.

5.5 Reactivity - Modeling Level

This section describes how we model reactivity in online auctions. The Conceptual and Definition parts of the model have already been described in Section 5.3 and the last one, called Modeling level, is going to be presented in the next sections. The study of reactivity provides many interesting analyses of online auctions, as showed in the last sections, in particular the new characterization methodologies that were developed.

However, the most critical challenge we face in this research on online auctions is how to specifically model reactivity and identify interesting reactivity patterns. In the beginning we tried to adopt some aggregate measures of auction negotiations, which led us to conclude that it is probably impossible to do it. Therefore, on the other hand, we have tried to model it using the finest grain level (criterion associated to each bid individually) and the results were also disappointing.

Examples of the use of aggregate measures can be found in Section 5.2, where we try to adopt many average measures of auction attributes and characteristics. An example of how we try to adopt a finest grain level associated to each bid in as individual measure is the *Entropy* concept we have applied in this case study.

We adopted this concept in order to classify the auction state, considering its level of activity. We denoted this concept entropy, as an analogy to the auction temperature (activity). We consider two variables to measure this level of activity in online auctions:

1. The inter-bidding time (IBT): is the time between two consecutive bids;
2. The inter-bidding price (IBP): the difference of price between two consecutive bids.

The idea to adopt the auction entropy concept is to correlate these measurements following an intuitive analysis. The slower is the IBT, the higher is the auction activity in terms of bids, the higher is the auction entropy. And the higher is the IBP, higher tends to be the auction activity in terms of price evolution, the higher is the auction entropy. We define entropy of these two measurements as the harmonic mean of them. Typically, the harmonic mean is appropriate for situations when the average of rates is desired. The formula for the harmonic mean of two measures is to multiply them, and divide that quantity by their arithmetic mean. In mathematical terms:

$$\frac{2 \times (IBP \times IBT^{-1})}{IPB + IBT^{-1}}$$

Figures 5.29, 5.30 and 5.31, show the auction entropy for three different auctions. Auction of 5.29 has 10 bidders, who made 19 bids. The duration was a week and the winner price was US$83.52. The entropy, that varies from 0 to 0.7, has an ascending behavior and a peak near the end of the auction.

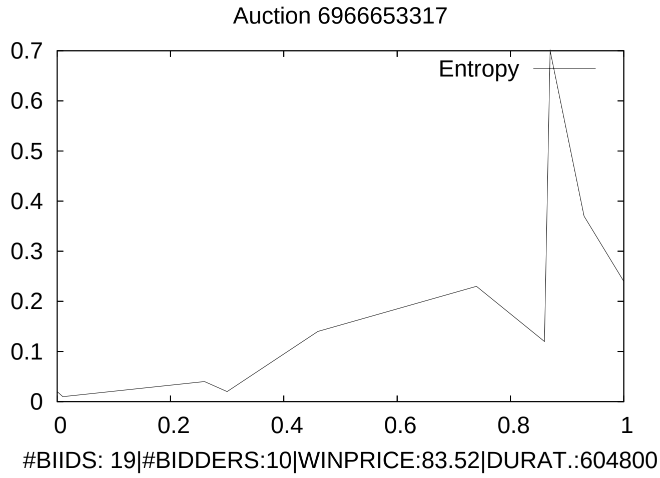


Figure 5.29: Auction Entropy

Auction of 5.30 has 10 bidders, who made 37 bids. The duration was also a week and the winner price was US$168.520. The entropy, that varies from 0 to 0.4, has an irregular profile and a peak near the end of the auction.

Auction of 5.31 has 7 bidders, who made 12 bids. The duration was three days and the winner price was US$50.05. The entropy, that varies from 0 to almost 1, has a descending

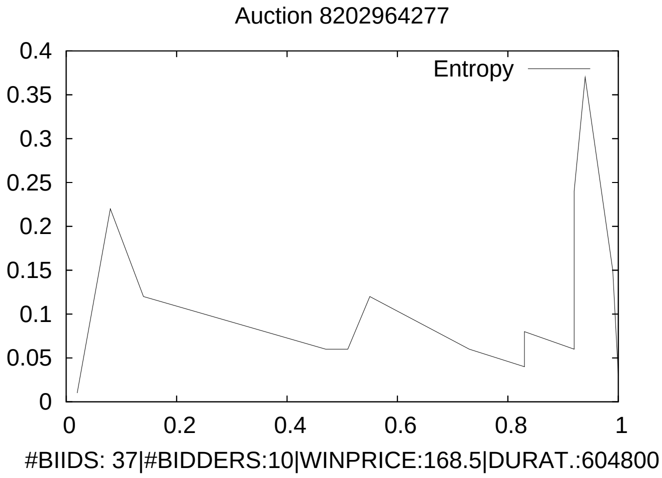


Figure 5.30: Auction Entropy

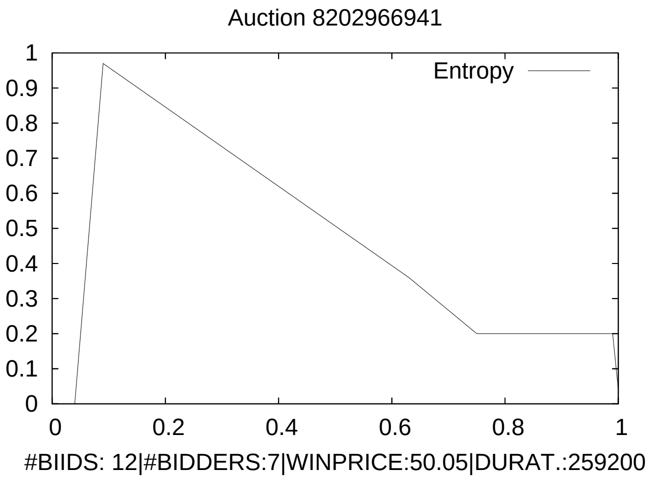


Figure 5.31: Auction Entropy

behavior. In the begining of the auction the entropy achieved the highest value, that can be explained by the high starting bid value set by the seller that cause the negotiation price to increase.

Comparing the first two auctions we can see they have a completely different behavior during 90% of the auction. After this, both present a peak, as a result of more activity before the end of the negotiation.

We also have tried to consider entropy as an aggregated measure, what we call the cumulative entropy, which measures the total entropy of an auction. It is calculated by the area under the curve, using the *Simpson's rule.* Again, we did not achieve any consistent result.

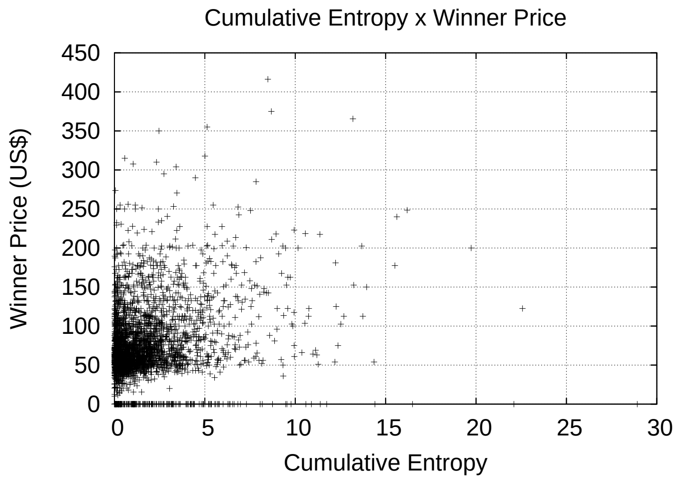


Figure 5.32: Cumulative Entropy - Winner Price

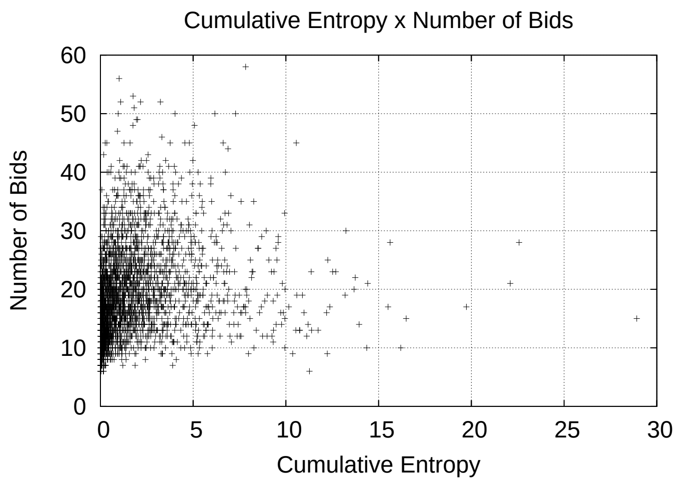


Figure 5.33: Cumulative Entropy - Bid

We have tried to compare the cumulative entropy with some auction characteristics, such as winner price (see Figure 5.32), number of bids in the auction (see Figure 5.33), quantity of participants (see Figure 5.34), and duration (see Figure 5.35). As can be seen in these figures, none direct correlation was identified.

We also study some social network concepts [118], trying to model the auction through them. A social network [68] is a social structure made of nodes which are generally individuals or organizations. It indicates the ways in which they are connected through various social familiarities ranging from casual acquaintance to close familial bonds. According to Newman [87, 88], a social network is a set of people or groups of people connected through patterns of social interaction, which can be represented as nodes and links, respectively, in a graph.

Basically we adopted some inherited classic properties of social networks in our analysis, as follows:

- Winner-Bidder-Out Degree: represents the bidders as their competition profile. It can be represented generating the histogram of times that each bidder becomes winner during the auction negotiation. Also, the inverse CDF can be used to analyze the distribution degree.

- Competition Reciprocity: denotes the competition relation between each pair of bidders who interact with each other. It shows the reciprocity and also quantifies how is the relation between each relation pair of bidders.

- Asymmetry Set: is the difference of its in and out sets. As lower is the asymmetry set, as higher is the competition between the set of bidders in an auction.

- Clustering coefficient is defined as the probability of any two of its neighbors being neighbors themselves. This measure represents how the competition is distributed among the set of bidders who are potential winners in an auction. It is a measure of the likelihood that two associates of a node are associates themselves. A higher clustering coefficient indicates a greater "cliquishness" - similar competition profile between the evaluated set of bidders.

Our idea using this approach was to correlate these properties with different groups of auctions and also establish a correlation between these properties and reactivity. One more time we conclude that there does not exist a clear and consistent correlation between these aspects.

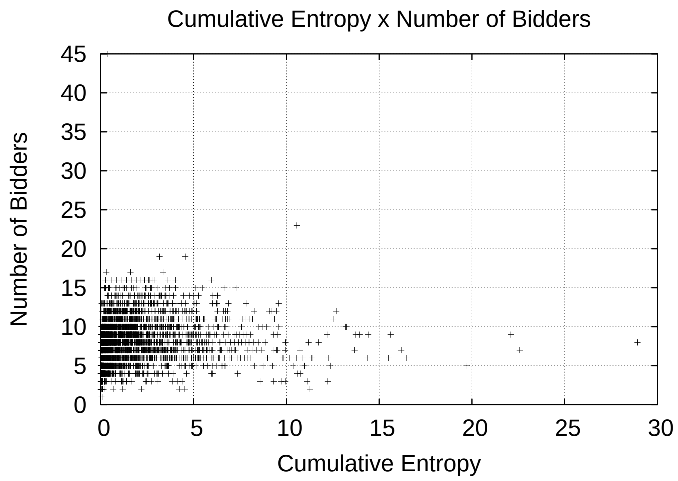


Figure 5.34: Cumulative Entropy - Bidder

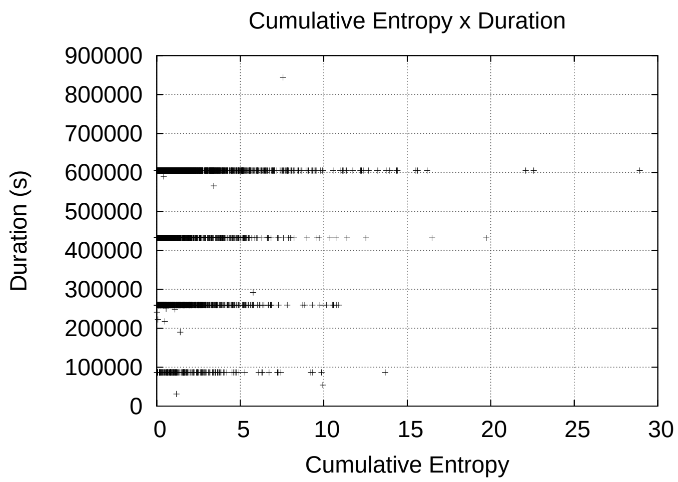


Figure 5.35: Cumulative Entropy - Duration

5.5.1 Reactive Transitions

In this section we describe the reactive transitions, that is, the best way we find in this research to model reactivity in online auctions.

Initially, the idea was to create a graph to represent the auction negotiation. A directed graph or digraph G is an ordered pair $G = (V, A)$ where V is a set of vertices or nodes, and A is a set of ordered pairs of vertices, called directed edges, arcs, or arrows. An edge $e = (x, y)$ is considered to be directed from x to y; y is called the head and x is called the tail of the edge. An auction can be represented by a graph where each auction state is a vertex and each bid, that changes the auction state, is an edge. An Auction State is defined as set of characteristics of the auction, such as the winning price and number of bids, in a specific time.

Later, we identify that each time a bidder places a bid, there is a perception criteria she/he potentially has observed and the bid represents the action. However, this action does not cause the state of the negotiation where the next action will occur. Observe that the next action will happened with a new state, when a new perception criteria will be valid. Consequently, we conclude the auction can not be represented by a typical graph for our purpose of modeling reactivity.

Therefore, based on the knowledge we have acquired from our previous approaches to model reactivity in online auctions and this conclusion, we decide to model reactivity at the finest grain level (each bid), considering the bidder profile, the auction negotiation pattern, and also both together. We call this approach reactive transitions, since the idea is to identify reactive patterns in each interaction that occurs in the auction.

The general idea of modeling reactivity, as we have previously presented in Chapter 3, is to identify the set of perception aspects that can affect the bidder behavior and also some characteristics that can be measured as a consequence of the bidder's action in the auction negotiation.

After doing many different experiments, we decide to adopt the following aspects to model each reactive transition, that represent the perception criteria:

- Negotiation Time (*Time*): represents the relative time of the auction negotiation perceived by the bidder when placing the bid.

- Winning Price (*WinP*): is the relative winning price of the auction negotiation perceived by the bidder when placing the bid.

- Winning Bidder (*WinB*): represents the perception of the bidder when acting in terms of who is winning the negotiation at this moment (it is me - 1 - or not - 0).

- Competition Type (*Comp*): defines the competition type (no competition - N, successive competition - S, zigzag competition - Z) the bidder perceives when placing the bid.

We have consider two consequences in the auction negotiation as resulting of the action (bid):

- Inter-Bidding Price - (*IBP*): is the variation of price the current bid causes in the auction negotiation.

- Change Winner (*ChW*): indicates if the action changes the winner (1) or not (0).

We discretize some of these measures. The negotiation time can assume the following values:

- I: initial phase of the auction negotiation (up to 0.3 - that is, up to 30% of the negotiation).

- M: middle phase of the auction negotiation (from 0.3 to 0.8).

- F: final phase of the auction negotiation (from 0.8 to 0.99).

- S: sniping phase of the auction negotiation (more than 0.99), that is to capture the last second bidding behavior.

It is important to clarify that these thresholds were established for a single product of this eBay dataset. The winning price assume the following values:

- L: low (less than US$35)
- M: medium (from US$35 to US$75)
- H: (more than US$75)

The *IBP* assumes the following values:

- VL: very low (less than US$1)
- L: low (from US$1 to US$5)
- M: medium (from US$5 to US$10)
- H: high (more than US$10)

It is important to say these values are based on the statistical analysis of our dataset. The attributes *winning bidder* and *change winner* assume boolean values (0 or 1). The *competition type* has already been specified in categories. Therefore each reactive transition can be represented by 4 perception attribute values and 2 attributes obtained from the action.

The set of perception criteria can assume 72 (4*3*2*3) possibilities and there are 8 (4*2) options in terms of action's consequence in the auction negotiation.

We then apply a data mining technique named association rules. The problem of mining association rules was introduced in Agrawal et at. 1993 [4, 5]. The aim of association rule mining is to find interesting and useful patterns in a transaction database. The database contains transactions which consist of a set of items and a transaction identifier (e.g., a market basket). Association rules are implications of the form $X \rightarrow Y$ where X and Y are two disjoint subsets of all available items. X is called the antecedent or LHS (left hand side) and Y is called the consequent or RHS (right hand side). Association rules have to satisfy constraints on measures of significance and interestingness. We call these measures *Support* and *Confidence*.

In the next section we present the results of modeling reactivity in online auction using reactive transitions.

5.5.2 Results

In this section we present the results we achieve using the idea of reactive transitions. We divide the results in general (applied to the whole dataset without any filtering), aggregated by bidder, aggregated by auction, and also the analysis considering the bidder in auction matching.

5.5.2.1 General Results

Applying the methodology to generate the reactive transitions for the role dataset, setting a very low minimum support (0.05%) and a minimum confidence of 50%, we obtain the results presented in Table 5.18.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
63.6	0.5	M	VL	1	N	→	VL	0
63.1	0.3	F	M	0	Z	→	L	0
60.9	0.3	I	VL	1	N	→	VL	0
60.5	4.0	S	M	0	S	→	L	0
60.3	1.9	S	L	0	S	→	L	0
57.0	2.7	F	L	0	S	→	L	0
56.9	1.5	F	M	0	S	→	L	0
56.4	0.4	F	L	1	N	→	L	0
55.8	0.7	S	H	0	Z	→	L	0
55.5	1.1	S	H	0	S	→	L	0
54.1	1.8	S	M	0	Z	→	L	0
53.6	0.7	S	VH	0	Z	→	L	0
53.0	0.3	M	L	1	N	→	L	0
51.9	0.3	M	VL	0	S	→	VL	0
51.8	0.3	I	VL	0	S	→	VL	0
51.2	0.3	F	VL	0	S	→	VL	0
51.5	0.5	M	L	0	S	→	L	0
50.5	0.5	S	L	0	Z	→	VL	0
50.1	0.4	F	L	0	Z	→	L	0
50.1	5.8	F	L	0	N	→	L	0

Table 5.18: Reactive Transition - General Dataset

For example, the first transition *M VL 1 N* $\rightarrow$ *VL 0* indicates that in 63.6% with a support of 0.5% a bidder in the middle of the negotiation time, observing a very low winning price, who is winning the auction negotiation, without any competition, places a bid causing a very low inter-bidding price and no change in the winner bidder. As can be seen, these rules representing the reactive transitions are not representative.

It was expected that this general analysis would not be representative, once is very difficult to identify reactivity without grouping any similarities of bidding behavior, auction negotiation or both. Continuing the analysis in this direction, the next sections present the results aggregated by bidder, auction and both of them, respectively.

5.5.2.2 Bidder Results

In this section we present the main reactive transitions according to the bidding behavior profile. We have chosen some groups of bidders to show the results and also present only the main relevant ones for each group.

For each presented result, the labels in the tables mean: *Conf.* is confidence, *Sup.* is the support, *Time* is the relative time where the auction negotiation is when the bidder acts, *WinP* is the relative winning price of the negotiation, *WinB* is an attribute that indicates wether the bidder is the current winner (1) or not (0), *Comp* indicates the competition type (N for no competition, S for successive, Z for zigzag) perceived by the bidder, *IBP* is the price variation the bid causes, and *ChW* is an attribute that indicates wether the winning bidder changes (1) or not (0).

Table 5.19 shows the results for group *B2*. As can be seen, for this bidding behavior profile, we identify that during sniping phase (S), in negotiations where the winning price is medium (M), the bidder who acts is the current winner, in a scenario without competition, the action of the bidder causes a very low (VL) increase in winning price and keeps the bidder as winner at that moment. This typical transition occurs with a Confidence of 60.78% and a Support of 0.62%. Almost the same transition is observed for the case of a zigzag competition. This behavior can be explained as a bidder who is winning the auction negotiation, perceiving that it has been finishing try to guarantee the result, causing a small price variation.

It is interesting to remember that these bidders typically act in the last auction se-

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
60.8	0.6	S	M	1	Z	→	VL	0
60.5	0.5	S	M	1	N	→	VL	0

Table 5.19: Reactive Transition - Bidding Behavior - Cluster B2

quences, 74% with competitive situation with 40% of zigzag competition type. Their sessions have more than one bid in 68% and they have not participated in the negotiation yet, called non-recurrent bidders. They represent 11.5% of the bidding behaviors, act after 99% of auction negotiation timing, placing 2.4 bids in average.

Table 5.20 shows the results for group *B5*. The most relevant reactive transitions indicate that during the initial and intermediary negotiation phase (I or M), in negotiations where the winning price is medium or over market price (M ou H), bidders who are not current winners in a scenario without competition act causing a high (H) variation in winning price and become winner. This typical transitions happened with a Confidence higher than 80%.

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
80.6	0.4	M	H	0	N	→	H	1
84.0	1.6	I	H	0	N	→	H	1
82.0	6.0	I	M	0	N	→	H	1

Table 5.20: Reactive Transition - Bidding Behavior - Cluster B5

Remembering *B5* profile, they are bidders who act typically in the first sequence (87%) of the auctions and never in the last one, in situation with no competition (96%), placing only one bid (94%) per session. This behavior happens in 5.2% and during the initial phase of the negotiation (from 23 to 25% of auction timing duration). These bidders can be labeled as *Early Evaluators*.

Table 5.21 shows the result for group *B8*. The main reactive transition indicates that during the initial negotiation phase (I), in negotiations where the winning price is medium (M), bidders who are not current winners in a scenario without competition act causing a high (H) variation in winning price and become winner. This typical transitions happened with a Confidence higher than 90%.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
90.9	1.6	I	M	0	N	→	H	1

Table 5.21: Reactive Transition - Bidding Behavior - Cluster B8

As previously characterized, these bidders mainly act in initial (38%) and intermediary (56%) sequences of the auction in activity moments where no competition is typical (77%). Most of their sessions have one bid (73%), are triggered (84%), recurrent (62%), and change winner (74%). They act through auction negotiation, participating from 27 to 84% of the timing duration of the negotiation, placing typically 4.3 bids in average. They are bidders very active and present a well defined reaction in the beginning of the auction negotiation.

Table 5.22 shows the results for group *B11*. As can be seen, for these bidding behavior profile, we identify that after the middle of the auction negotiation (M, F), in negotiations where the winning price is medium in general (M), and in scenarios without competition, if the bidder who acts is not the current winner than she/he causes a high (H) variation in winning price and becomes winner. Else, if the bidder is the current winner and the negotiation is in quite closed, she/he raises a little the winning price, trying to guarantee the result. These transitions occur with a Confidence higher than 77%.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
94.3	6.0	M	M	0	N	→	H	1
83.3	1.3	M	H	0	N	→	H	1
77.7	13.1	F	M	0	N	→	H	1
77.4	0.8	S	M	1	N	→	VL	0

Table 5.22: Reactive Transition - Bidding Behavior - Cluster B11

These reactive behaviors happen for bidders who typically act in the first sequence (75%) of the auctions, in situation with no competition (82%). They change winner in 90%, and place only 1.6 bids in average. This behavior happens in 4.9% and during the last phase of the negotiation (from 83 to 87% of auction timing duration).

Table 5.23 presents the most frequent reactive transitions for group *B14*. From the results, we can see that after the medium of the auction negotiation (M, F, or S), in negotiations where the winning price is high (over the average market price), and in scenarios without competition, and the agents (bidders) are not current winners. They act causing a high (H) variation in winning price and becoming winners. These transitions occur with a Confidence higher than 70%.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
93.3	0.5	M	H	0	N	→	H	1
72.2	13.8	S	H	0	N	→	H	1
70.0	1.1	F	H	0	N	→	H	1

Table 5.23: Reactive Transition - Bidding Behavior - Cluster B14

As previously analyzed, bidders from *B14* act in the last auction sequences, 67% in sequences without competition. Their sessions have always only one bid, are triggered in 2/3, and always change winner. As expected, all of them is non-recurrent. They represent 8.5% of the bidding behaviors, and typically act very late in the auction, placing 1.0 bid in average, as expected.

These results are only a small part of the analysis of reactive transitions per bidder. As can be observed from these results, the relevance of the analysis is much better than the general analysis of reactive transitions, as presented in the last section. These can be explained by the fact that reactivity is not a generic phenomenon, it acquires representativity or accuracy when it is associated with a typical scenario, such as the bidder's profile, as showed in this section. Also, in the next sections we present this analysis aggregated by auction negotiation patterns and then by bidder in auction pairs.

5.5.2.3 Auction Results

In this section we present the main reactive transitions according to the auction negotiation pattern. We have chosen some groups of auctions to show the results and also present only the main relevant reactive transitions (considering their Confidence) for each group. For each presented result, the labels in the tables mean the same that we have already presented in the last section.

Table 5.24 shows the results for group of auctions *A0*. Remembering the auction characterization we have already presented in this document, the auctions have typically a very small number of sequences, almost all of them unique and with no competition. Almost all their sequences change the winner. This is the third most frequent negotiation pattern in our dataset.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	25.6	S	H	0	N	→	H	1
100.0	1.0	M	H	0	N	→	H	1
100.0	0.4	M	M	0	N	→	H	1
100.0	0.3	I	M	0	N	→	H	1
98.6	38.8	S	M	0	N	→	H	1
96.7	2.0	F	H	0	N	→	H	1
95.4	5.9	F	M	0	N	→	H	1

Table 5.24: Reactive Transition - Auction Negotiation - Cluster A0

As can be seen, for these auctions, the reactive transitions show there are interactions where bidders who are not current winners, perceiving a scenario with no competition, act causing a high variation in winning price and become winners. These interactions occur typically in sniping phase (as can be identified by a support higher than 60%) in negotiations with medium and high winning prices. This typical reactive transition rule occurs with a Confidence higher than 95% (100% in some cases) and contemplates more than 73% of the actions (bids) of these auctions.

Table 5.25 shows the results for auctions from *A1*. From auction characterization we have done, these are auctions with medium level of activity, most of them with two activity moments. These auctions present a high competition level (74%), from which 84% is successive, and change the winner in 86.8% of their sequences. This pattern is rare, occurring in only 4% of auctions.

The most relevant reactive transition (in terms of *Confidence* - 95.7%) identifies a typical

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
95.7	0.8	I	H	0	S	→	H	0
81.0	0.6	S	M	1	Z	→	VL	0
71.4	0.7	S	M	1	S	→	VL	0
68.4	0.5	S	M	1	N	→	VL	0

Table 5.25: Reactive Transition - Auction Negotiation - Cluster A1

behavior where a bidder acts in initial phase of the auction negotiation, that presents high winning price, perceiving a successive competition and also that she/he is not the current winner of the auction, causing a high variation in the winning price, but not enough to change the winner.

The other significant reactive transitions represent bidders who are current winning the auction negotiation, that act in sniping phase, perceiving all kinds of competition situations (no competition, successive and zigzag), causing a very small increase in price with objective to guarantee the win.

Table 5.26 shows the results for group of auctions *A4*. These negotiations present predominance of low level of activity (sequences), most of them have just one sequence. The competition level is very high, with 90.9% of successive and 9.1% of zigzag competition types.

For this auction negotiation pattern, we have many reactive transitions with 100% of confidence. A typical behavior observed is the bidders acting in the last seconds of the auctions, in scenarios where they are winning and perceive competition or not, raising the winning price to try to guarantee the win.

These results are also only a small part of the analysis of reactive transitions per auction negotiation patterns. As we have observed with the analysis from bidder's group, the relevance of the analysis is much better than the general analysis of reactive transitions.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	1.5	S	M	1	S	→	VL	0
100.0	0.5	S	M	1	N	→	VL	0
100.0	0.3	S	L	1	Z	→	VL	0
100.0	0.3	F	L	1	N	→	VL	0
100.0	0.3	M	M	0	N	→	H	1
100.0	0.3	S	H	1	Z	→	L	0
100.0	0.3	F	H	0	S	→	L	0
100.0	0.3	M	H	1	S	→	H	0
100.0	0.3	F	M	0	Z	→	M	1

Table 5.26: Reactive Transition - Auction Negotiation - Cluster A4

5.5.2.4 Bidder in Auction Results

This section presents some results in the context of bidder in auction. In our case study, we have identified 16 different groups of bidders and 7 groups of auctions. From these groups, we can have 112 different scenarios to analyze, however just 82 has been occured in the dataset.

The number of reactive transitions identified in these 82 bidder in auction scenarios are very high, suggesting the accuracy of the approach. There are 1251 reactive transition rules with a confidence higher than 50%. Considering a 100% confidence, we have 359 rules.

We choose some results about the reactive transitions considering the analysis of bidder in auction sets. This aggregation is the most restrictive analysis that we can perform. On the other hand, we can say that it is the most confident analysis, once it has the most complete set of knowledge about the environment (auction negotiation) and the agent (bidder behavior).

We present here two different bidding behaviors participating in auctions from cluster *A0*. These auctions have very small number of sequences, almost all of them unique and with no competition. All of them change the winner, as expected, once the first sequence always changes the winner in eBay. This is the third most frequent negotiation pattern in our dataset.

Table 5.27 shows some results with 100% of confidence for bidders from *B0*, acting in auctions from *A0*. We have already identified that bidders from *B2* act typically in the last auction sequences and they have not participated in the negotiation yet, called non-recurrent bidders. They usually act after 99% of auction negotiation timing, placing 2.4 bids in average.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	3.8	S	M	0	Z	→	L	0
100.0	3.8	F	L	1	N	→	VL	0
100.0	1.9	S	L	1	N	→	VL	0
100.0	13.2	F	M	0	N	→	H	1
100.0	5.7	S	M	0	N	→	H	1
100.0	5.7	F	H	0	N	→	H	1
100.0	1.9	M	L	0	N	→	H	1
100.0	3,8	S	M	0	S	→	VL	1

Table 5.27: Reactive Transition - Bidder in Auction - Cluster A0B2

According to these results, we identify some typical behaviors of these bidders when acting in these kind of auctions. The most common reaction of them is to act at the end of the negotiations, perceiving different winning prices (low, medium, or high) when they are not winners and there are no competition, causing a high winning pricing variation and becoming winners. Also there is the case when they are winning the auction with a low winning price, however they place a new bid, causing a very low increase in the winning price who themselves are going to pay. Also, the results show that in some competition situations in sniping phase, they also act, trying to become the winner, having success sometimes, but failing in others.

Table 5.28 shows some results with 100% of confidence for bidders from *B14*, acting in auctions from *A0*. We have already identified that bidders from *B14* usually act in the last auction sequences, 67% in sequences without competition. Their sessions have always only one bid and always change winner. They represent 8.5% of the bidding behaviors, act close to 99% of auction negotiation timing, placing 1.0 bid in average.

From the analysis of these reactive transitions results, we identify, in more than 75%

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	45.0	S	M	0	N	→	H	1
100.0	30.0	S	H	0	N	→	H	1
100.0	5.7	F	M	0	N	→	H	1
100.0	2.1	F	H	0	N	→	H	1
100.0	1.1	M	H	0	N	→	H	1
100.0	0.4	M	M	0	N	→	H	1

Table 5.28: Reactive Transition - Bidder in Auction - Cluster A0B14

of the cases with 100% of confidence, that these bidders act in sniping phase, perceiving medium or over market winning prices and no competition, becoming winners and causing a high increment in the auction's winning price.

We also present three different analysis of bidding behaviors acting in auctions from cluster *A3*. These auctions present medium level of activity, most of them with two activity moments. These auctions do not present competition in almost 80% and change the winner quite always (90%). They represent 12% of the auctions from the dataset.

Table 5.29 shows some results with 100% of confidence for bidders from *B0*, acting in auctions from *A3*. These reactive transitions typically represent two different behaviors from these bidders: they trying to keep winning the auction negotiation raising the winning price, and they becoming winners in the middle and sniping phases.

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	4.0	M	M	1	N	→	VL	0
100.0	2.0	F	H	1	N	→	M	0
100.0	4.0	M	M	0	N	→	H	1
100.0	2.0	M	H	0	N	→	H	1
100.0	2.0	S	L	0	N	→	L	1
100.0	4.0	M	L	1	N	→	L	0
100.0	2.0	M	H	1	N	→	H	0

Table 5.29: Reactive Transition - Bidder in Auction - Cluster A3B0

Table 5.30 shows some results with 100% of confidence for bidders from *B3*, acting in auctions from *A3*. These reactive transitions represent almost 30% of their acts, showing the bidder trying to become winner over different perceptions. In the first one (more frequent) the bidder act at the end of the auction, without competition and observing a low winning price, becoming the winner and causing a small increase in winning price. In a similar scenario, but perceiving a over market winning price (H - high), they act becoming the winner and causing a high increase in winning price. The first behavior is also observed during the middle of the auction negotiation.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	19.1	F	L	0	N	→	VL	1
100.0	4.8	M	L	0	N	→	VL	1
100.0	4.8	F	H	0	N	→	H	1

Table 5.30: Reactive Transition - Bidder in Auction - Cluster A3B0

Table 5.31 presents the reactive transitions with 100% of confidence for bidders from *B5*, acting in auctions from *A3*. These reactive transitions represent more than 45% of these bidder's actions. These are bidders who act typically in the first sequence (87%) of the auctions and never in the last one, in situation with no competition (96%), placing only one bid (94%) per session. They are non-recurrent in 87%, change winner in 94%. This behavior happens in 5.2% and during the initial phase of the negotiation (from 23 to 25% of auction timing duration).

From the results, we identify that these bidders act in these kind of auctions in the beginning, perceiving no competition and different winning price (medium in most times), causing a high increase in the winning price of the negotiation and becoming the winner at that occasion. Also, we identify a rare behavior of them, acting during the initial phase without competition, raising the winning price she/he has offered before with a high increment of price.

As can be seen in this section, the reactive transitions is an approach that provide a intuitive way to understand the interactions in an online auction, enriching the analysis and also guaranteeing a good accuracy in terms of confidence.

It is important to emphasize that these results become possible only after detailing our result set. We consider in this last analysis the bidder in auction sets, that is, the most

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	1.2	I	L	1	N	→	H	0
100.0	21.8	I	M	0	N	→	H	1
100.0	10.3	M	M	0	N	→	H	1
100.0	6.9	I	H	0	N	→	H	1
100.0	4.6	M	H	0	N	→	H	1

Table 5.31: Reactive Transition - Bidder in Auction - Cluster A3B0

focused approach that we can have in this study. As expected, we achieve a confident result. This can be explained by the fact that we use the detailed characteristics about the environment (auction negotiation and its state) and the agent (the bidder considering her/his characteristics), therefore this context presents the most complete set of knowledge to analyze reactivity in this case study.

The complete set of results of this case study can be found in Appendix C.

5.6 Summary

In this chapter we presented a case study where we focus on the business dimension of the reactivity. The main topics of this case study were the modeling and characterization of the reactivity, the use of it in a real case study of an online auction's service, the analysis of correlation between auction negotiation and bidding behavior, and the use of data mining to identify reactivity patterns. For each of these topics, we describe the main results and conclusions.

Modeling reactivity in online auction can contribute to understand the business dynamics and to design more complete automatic agents.

We propose a new characterization model for online auctions, which provides novelties in order to model and understand the factors that characterize and explain the auction dynamics. We apply our model to a real case study, using an eBay dataset, and study preliminary findings towards the understanding of reactivity patterns in e-business.

Our experimental analysis demonstrates that our proposal provides a way to open the auction dynamics's "black box". From this part of the work, we conclude the importance to characterize auctions using sequences, and bidders through sessions. Doing this, we will have a better semantic characterization, since the current work shows that it is difficult to explain some specific behavior through a general data analysis.

Applying a data mining technique called clustering, which sorts the analyzed data into clusters of similar data, we found auction negotiation clusters in our datasets. This result is very interesting, showing different negotiation patterns for each group of auctions considering not only static and aggregated measures, but the auction dynamics, a novelty for online auctions research. We also introduce the Auction Model Graph (AMG) to analyze the temporal aspect of the negotiation. This aspect is important to complement the analysis and can also be used to generate an online auction synthetic workload.

As can be seen through our case study, our approach considers aspects related to auction negotiation dynamics, identifying the bidder acting during the auction. If we consider just the approaches from literature, the bidding behavior profiles are limited to just few attributes, which does not allow to model the bidder acting in terms of reactivity. Moreover, we provide a hierarchical model and a methodology to characterize online auctions that is generic and may be applied to each dataset to identify with accuracy the bidding behavior profiles. Through this characterization of bidder actions we have the basis to investigate the cause-effect relations to model reactivity.

The current bidding characterization around clusters of bidding strategies (e.g., jump bidders, evaluators, snipers, participators [21, 105]) have been defined based on static or aggregate measures, such as the total number of bids, time of first bid, time of last bid etc. We have found out that there is much more to bidders' strategies in terms of the auction negotiation dynamics, activity and competition, which may be captured applying our new characterization model. For example, we have observed throughout our analysis that the same bidder engages in different strategies at different moments in the same auction, including the so-called snipers who have engaged in bidding sessions in initial stages of the auction.

Our methodology to characterize Internet auctions is generic and may be applied to different datasets to identify with accuracy the bidding behavior profiles, considering the auction negotiation dynamics. Through this characterization of bidding behavior we have the basis to investigate the cause-effect relations to model reactivity.

After applying the methodology to characterize auction negotiation and bidding behavior, we analyze the correlation between them in online auctions, explaining the observed correlations through a wealth of details that our approach is able to provide. We capture the necessary attributes that characterize reactivity in terms of competition, price changes, winning status, etc., from the bidder's perspective and the overall auction's perspective. The characterization of the relevant reactivity periods is used as input to the identification of patterns of auction negotiation as well as bidding behavior. Using actual bidding data from eBay, we demonstrate the efficacy of our approach in explaining bidding dynamics and the effect on auction outcomes by correlating auction patterns to bidding behavior.

In the last section we explain how we model reactivity in online auctions, that is the most challenge we have faced in this research on online auctions. In the beginning we have tried to adopted some aggregated measures of auction negotiation, which let us to conclude that it is probably impossible to do it. On the other hand, we have tried to model it using the finest grain level (criterion associated to each bid individually) and the results were also disappointing. Finally, we propose the idea of reactive transitions, that is, the best way we find in this research to model reactivity in online auctions. As can be seen by the results, the reactive transitions is an approach that provide an intuitive way to understand the interactions in an online auction, enriching the analysis and also guaranteeing a good accuracy.

These results can be applied to provide personalization to different bidder profiles, defining new semantic rules to design e-business agents, or in order to design decision support tools for e-commerce, for example.

Chapter 6

Conclusions and Future Work

This chapter presents the conclusions and future work, listing the main activities performed and some research contributions achieved in this doctoral dissertation.

6.1 Conclusions

Interactive computer systems have become popular in our society. From bank transactions to cell phones, we are continuously interacting with systems and even with other users through these systems. Internet systems are examples where this interaction takes place. These interactions are usually complex and intriguing. It is quite hard to determine exactly the factors that led a user to behave as observed. It is remarkable to note that the interactions are not isolated, but successive interactions become a loop feedback mechanism, where the user behavior affects the system behavior and vice-versa.

The concept of *Reactivity* denotes that the user action varies according to some perception criterion related to the interaction with the system. In this work we have identified two dimensions of reactivity: performance and business rules. We present a way to model reactivity and an explanation of how to apply it in some applications, such as Web sites and e-business services.

There is strong evidence that much of the user behavior is reactive, that is, the user reacts to the instantaneous conditions at the action time. The main hypothesis of this dissertation is that the real world is reactive, that is, the user behavior varies in part according to some factors related to the server and the application that have been provided.

The thesis of this dissertation is characterize and model reactivity in interactive systems, in particular Internet-based systems. Understanding users' reactivity has applicability to several scenarios, from system evaluation to system improvement, such as personalization. The main challenge is to identify what is the limit of reactivity characterization.

The main motivation of this work is to understand how users interact with the system, allowing to provide services with better performance and to understand better the e-business applications.

This work achieves the objectives previously defined for it, providing a way to characterize and model reactivity in interactive computer systems. Moreover, the case studies demonstrate how the theory can be applied in real scenarios and the benefits of it.

The contributions of this work are the formalization of reactivity concept, the specification of a reactivity multi-level model, the specification of a framework for modeling user reactivity, and the application of the framework to relevant scenarios such as Internet services and electronic auctions, generating reactivity models of the users.

We believe that the formalization of reactivity and the procedure to investigate it in interactive systems are generic enough to be applied to any scenario of a computer system where there is user-system interaction. In spite of this, it is important to say that our approach is flexible, since there are a variety of criteria to be chosen in order to qualify the reactivity presented in a set of interactions. Considering this, the limitation of our work is that, despite we follow the methodology and validate it using real case studies, we can not guarantee that we apply the model with the best criteria.

We have presented a case study where we focus on the performance dimension of the reactivity. The main aspects of this case study were the characterization of the reactivity, the use of it in workload generation, the *USAR-QoS* simulator, and the use of reactivity in QoS control mechanisms. We describe for each of these topics the main results and conclusions.

From the results we observe different gains according to each QoS approach. The mechanism based only on reactive scheduling achieves the best burst lost rate, since its mechanism gives priority to requests classified with impatient classes. The mechanisms that adopt the admission control are effective to reduce the response time, but may cause the increase in the burst lost rate, due to the increase in the amount of users rejecting bursts or sessions. The hybrid mechanism presents an equilibrium, reducing both the response time values and the burst lost rate. The benefits obtained are significant, demonstrating a relevant improvement in the QoS of reactive Internet systems through the use of reactive approaches.

We also have presented a case study validating our approach on business dimension of reactivity, where we describe how to model and characterize reactivity, the use of it in a real case study of an online auction's service, the analysis of correlation between auction negotiation and bidding behavior, and the use of data mining to identify reactivity patterns.

Modeling reactivity in online auction can contribute to understand the business dynamics and to design more complete automatic agents, that can act considering environment's conditions. Also, studying reactivity can benefit the economic analysis of Web-based environments, such as marketplaces.

We have proposed a new characterization model for online auctions, which provides novelties in order to model and understand the factors that characterize and explain the auction dynamics. We apply our model to a real case study, using an eBay dataset, and study preliminary findings towards the understanding of reactivity patterns in e-business.

Our experimental analysis demonstrates that our proposal provides a way to open the auction dynamics's "black box". From this part of the work, we conclude the importance to characterize auctions using sequences, and bidders through sessions. Doing this, we will have a better semantic characterization, since the current work shows that it is difficult to explain some specific behavior through a general data analysis.

Applying a data mining technique called clustering, which sorts the analyzed data into clusters of similar data, we found auction negotiation clusters in our datasets. This result is very interesting, showing different negotiation patterns for each group of auctions considering not only static and aggregated measures, but the auction dynamics, a novelty for online auctions research. We also introduce the Auction Model Graph (AMG) to analyze the temporal aspect of the negotiation. This aspect is important to complement the analysis and can also be used to generate an online auction synthetic workload.

Our research considers aspects related to auction negotiation dynamics, identifying the bidder acting during the auction. If we consider just the approaches from literature, the bidding behavior profiles are limited to just few attributes, which does not allow to model the bidder acting in terms of reactivity. Moreover, we provide a hierarchical model and a methodology to characterize online auctions that is generic and may be applied to each dataset to identify with accuracy the bidding behavior profiles. Through this characterization of bidder actions we have the basis to investigate the cause-effect relations to model reactivity.

Our methodology to characterize Internet auctions is generic and may be applied to different datasets to identify with accuracy the bidding behavior profiles, considering the auction negotiation dynamics. Through this characterization of bidding behavior we have the basis to investigate the cause-effect relations to model reactivity.

We analyze the correlation between them in online auctions either, explaining the observed correlations through a wealth of details that our approach is able to provide. Using actual bidding data from eBay, we demonstrate the efficacy of our approach in explaining bidding dynamics and the effect on auction outcomes by correlating auction patterns to bidding behavior.

The most challenge we have faced in this research is how to model reactivity and identify interesting reactivity patterns in e-business applications, such as online auctions. In the beginning we have tried to adopted some aggregated measures of auction negotiation, which led us to conclude that is probably impossible to do it. On the other hand, we have tried to model it using the finest grain level (criterion associated to each bid individually) and the results were also disappointing. Finally, we have proposed the concept of reactive transitions, which we believe is the best way to model reactivity in online auctions.

As can be seen by the results, the reactive transitions is an approach that provide an intuitive way to understand the interactions in an online auction, enriching the analysis and also guaranteeing a good accuracy. We have achieved the best results analyzing reactivity in the detailed datasets, that is, considering the results obtained with the characterization methodology. Considering the bidder in auction sets, that is, the most restrictive approach that we can have using our approach, we achieve our most confident result. This can be explained by the fact that we use the detailed characteristics about the environment (auction negotiation and its state, that is defined by other bidders) and the agent who acts at each time (the bidder that is placing bids, considering her/his behavior's characterization), thus this context presents the most complete set of knowledge to analyze reactivity in research.

It is important to emphasize that the main contribution of this research is the novel approach for characterizing and modeling interactive systems considering the concept of reactivity, showing how it can be evaluated in the different dimensions, such as performance of computer systems (physical layer) and e-business (logical layer).

However we know that our approach can be improved to consider other evaluation criteria and also to make easier its application in other scenarios.

As future work, there are other research aspects to investigate, such as:

- Propose new attributes to consider in order to model reactivity using our model and characterization methodology.
- Consider the user learning through time, that is, the evolution of user behavior profile throughout time.
- Model and characterize the inter-auction relationships in online auctions.
- Apply the characterization methodology in other case studies, such as the e-business adopted by Sams Club [36] and the Brazilian government reverse auction [42].

6.2 Publications

This work has already generated, in terms of publications, the following contributions:

- LAWEB04
  CONFERENCE: 2nd Latin American Web Congress
  TITLE: "A Hierarchical Characterization of User Behavior"
  AUTHORS: Adriano Pereira, Gustavo Gorgulho, Leonardo Silva, Wagner Meira Jr.
  DATE: October, 12-15 - 2004.
  PLACE: Ribeirão Preto, SP - Brazil
  BRIEF DESCRIPTION: Understanding the characteristics of Internet services workloads is a crucial step to improve the quality of service offered to Web users. This paper presents a hierarchical and multiple time scale approach based on [81], which propose a characterization at the session, function, and request levels. This work extends it, adding new insights to these three levels and characterizing a new level that comprises the user behavior. The approach is illustrated by presenting a characterization of a proxy-cache server from one of the biggest Brazilian federal universities. This new level of characterization is the novelty of this research area, once it considers the interaction between users and servers, that answer their requests, in order to model user behavior. This modeling shows the reaction of the users to the system performance and clarify how the quality of the offered service affects their interaction with the computer system.

- WWC04
  CONFERENCE: 7th Annual IEEE Workshop on Workload Characterization
  TITLE: "The USAR Characterization Model"
  AUTHORS: Adriano Pereira, Gustavo Gorgulho, Leonardo Silva, Wagner Meira Jr., Walter Santos
  DATE: October, 25 - 2004
  PLACE: Austin, Texas - USA
  BRIEF DESCRIPTION: Understanding the user behavior is a need to analyze the performance and the scalability of web servers. This knowledge is used, for instance, to build workload generators that help evaluating the performance of those servers. Current workload generators are typically memory-less, being unable to mimic actual user interaction with the system. In this work we propose a hierarchical characterization and simulation model focused on the user behavior, named *USAR*. We use the

latency and inter-arrival time of the requests to model user actions, which are the basis of our model. We validate this model through a proxy-cache server case study, where we perform the characterization and construct a user behavior simulator. We foresee from the results the possibility to generate more realistic workloads.

- SAINT06
  CONFERENCE: The 2006 International Symposium on Applications and the Internet (Co-sponsored IEEE)
  TITLE: "Assessing Reactive QoS Strategies for Internet Services"
  AUTHORS: Adriano Pereira, Leonardo Silva, Wagner Meira Jr., Walter Santos
  DATE: January 23-27, 2006
  PLACE: Phoenix, Arizona, USA
  BRIEF DESCRIPTION: The design of systems with better performance is a real need to fulfill user demands and generate profitable Web applications. Understand user behavior and workload they produce on the server is fundamental to evaluate the performance of systems and their improvements. User reactivity, that is, how the users react to variable server response time, is usually neglected during performance evaluation. This work addresses the use of reactivity to improve QoS of Internet services. We propose and evaluate new admission control policies. We designed and implemented the *USAR-QoS* simulator that allows the evaluation of the new QoS strategies considering the dynamic interaction between client and server sides in Internet services. The simulation uses a TPC-W-based workload and shows the benefits of the reactive policies, which can result in better QoS. The experiments show the proposed reactive admission control policies lead to better response time rates, with a reduction from 15 to 50%, preserving the user satisfaction metric.

- ISPASS06
  CONFERENCE: IEEE International Symposium on Performance Analysis of Systems and Software
  TITLE: "Assessing the Impact of Reactive Workloads on the Performance of Web Applications"
  AUTHORS: Adriano Pereira, Leonardo Silva, Wagner Meira Jr., Walter Santos
  DATE: March 19-21, 2006
  PLACE: Austin, Texas, USA
  BRIEF DESCRIPTION: Being able to mimic user behavior and the workload they

generate on the servers is fundamental to evaluate the performance of systems and their improvements. One aspect that is usually neglected by workload generators is the user reactivity, that is, how the users react to variable server response time. Further, it is not clear how the reactivity-related changes in the user generated workload affect the server and how these dependences converge. This paper addresses this problem by proposing, implementing, and validating a workload generator that accounts for reactivity while interacting with servers. Our workload generator is used, for instance, to generate workloads based on a TPC-W benchmark. These workloads are used to assess the impacts of reactivity on the performance of a Web application.

- ROTW06
  CONFERENCE: Workshop 'Reactivity on the Web' at the International Conference on Extending Database Technology (EDBT 2006)
  TITLE: "Reactivity in Online Auctions"
  AUTHORS: Adriano Pereira, Fernando Mourao, Paulo Goes, Wagner Meira Jr.
  DATE: March 31, 2006
  PLACE: Munich, Germany
  BRIEF DESCRIPTION: Interactive computer systems, that is, systems in which users cyclically interact by getting and providing information, have already a widespread and increasing use in all areas of our society. One characteristic of such systems is that the user behavior affects the system behavior and vice-versa. There is strong evidence that much of the user behavior is reactive, that is, the user reacts to the instantaneous conditions at the action time. This paper presents the reactivity concept and describes a framework to model it in interactive systems, in particular Internet-based systems. We analyze an online auction within the framework. Based on *eBay* data, we identify attributes that affect the winner bidders' behavior, such as the auction time to finish. This paper presents the first findings towards the formal description and understanding of reactivity patterns in an e-commerce application, which will be useful in improving the application and building novel mechanisms.

- IPCCC06
  CONFERENCE: 25th IEEE International Performance Computing and Communications Conference IPCCC 2006
  TITLE: "Evaluating the Impact of Reactive Workloads on the Performance of Web Applications"

AUTHORS: Adriano Pereira, Leonardo Silva, Wagner Meira Jr.
DATE: April 10-12, 2006
PLACE: Phoenix, Arizona, USA
BRIEF DESCRIPTION: The great success of the Internet has raised new challenges in terms of applications and the satisfaction of their users. In fact, there is strong evidence that a significant part of the user behavior depends on its satisfaction. Users reactions may affect the load of a server, establishing successive interactions where the user behavior affects the system behavior and vice-versa. It is important to understand this interactive process to design systems more suited to user requirements. In this work we study and explain how this reactive interaction is performed by users and how it affects the system's performance. We perform experiments using a real server under a TPC-W-based workload generated using a reactive version of *httperf*. We also simulate different workload configurations in order to evaluate the effects on the system's load. The results show that accounting for reactivity causes a significant impact on the server's performance in terms of throughput and response time, raising the possibility of performance improvement of Web systems by considering reactivity.

- LAWEB06
  CONFERENCE: 4th Latin American Web Congress - LAWEB
  TITLE: "Reactivity-based Scheduling Approaches For Internet Services"
  AUTHORS: Leonardo Silva, Adriano Pereira, Wagner Meira Jr.
  DATE: October 25-27, 2006
  PLACE: Universidad de las Americas, Puebla Cholula, Mexico
  BRIEF DESCRIPTION: Understanding the characteristics of Internet services workloads is a crucial step to improve the quality of service offered to Web users. One aspect that is usually neglected during a performance evaluation is the user reactivity, i.e., how the users react to variable server response time. This work addresses the use of reactivity to improve the quality of service (QoS) of Internet services. We propose and evaluate two new scheduling approaches: the Patient-First Impatient-Next (PFIN), and the Impatient-First Patient-Next (IFPN). We design and implement the *USAR-QoS* simulator that allows the evaluation of QoS strategies considering the dynamic interaction between client and server sides. We simulate the new strategies using a TPC-W-based workload. The experiments show the benefits of the reactive policies which can result in better QoS for Internet Services, improving the user

satisfaction.

- SAINT07
  CONFERENCE: 4th Latin American Web Congress - LAWEB
  TITLE: "Reactivity-based Quality of Service Strategies for Web Applications"
  AUTHORS: Leonardo Silva, Adriano Pereira, Wagner Meira Jr.
  DATE: January 15-19, 2007
  PLACE: Hiroshima, Japan
  BRIEF DESCRIPTION: The great success of the Internet has raised new challenges in terms of applications and user satisfaction. Web applications demand requirements, such as performance and scalability, in order to guarantee quality of service (QoS) to users. Due to these requirements, QoS has become a special topic of interest and many mechanisms to provide it have been proposed. Those mechanisms fail to consider aspects related to reactivity, i.e., how the users react to variable server response time. This work addresses the use of reactivity to provide new strategies. We design and evaluate a reactivity-based scheduling mechanism that gives priority according to user behavior. We also propose a hybrid admission control and scheduling mechanism that combines both reactive approaches. The results show benefits in terms of response time and user satisfaction.

- ICISTM07
  CONFERENCE: International Conference on Information Systems, Technology and Management
  TITLE: "A HIERARCHICAL MODEL AND CHARACTERIZATION METHODOLOGY FOR ONLINE AUCTIONS"
  AUTHORS: Adriano Pereira, Leonardo Rocha, Fernando Mourão, Paulo Goes, Thiago Torres, Wagner Meira Jr.
  DATE: March 12-13, 2007
  PLACE: New Delhi, India
  BRIEF DESCRIPTION: Online auctions have challenged many assumptions and results from the traditional economic auction theory. Observed bidder behavior in online auctions often deviates from equilibrium strategies postulated by economic theory. In this research, we consider an online auction as an information system that provides a long-duration, information-rich, dynamic application environment in which users (bidders) interact with the system in a feedback loop, in what we

term reactivity. Bidders react to the observed conditions of the auction and events triggered by actions of other bidders. In this work we propose a new hierarchical characterization model with the purpose of isolating the segments of the auction in which users react to the auction conditions and events. Through this model, it is possible to enrich the auction characterization. Despite the existence of other bidding characterization models, none of them is enough for understanding the factors that characterize and explain the auction dynamics. We present preliminary results which demonstrate the advantages of applying our methodology. The final objective is to gain an understanding of what drives the dynamics of online auctions, the role of reactivity in the auction dynamics, and how the outcome of the auction is affected by the particular dynamics of the system.

- WEBIST07
  CONFERENCE: 3rd International Conference on Web Information Systems and Technologies
  TITLE: "Analyzing eBay Negotiation Patterns"
  AUTHORS: Adriano Pereira, Leonardo Rocha, Fernando Mourão, Thiago Torres, Wagner Meira Jr., Paulo Goes.
  DATE: March 3-6, 2007
  PLACE: Barcelona, Spain
  BRIEF DESCRIPTION: Online auctions have several aspects that violate the common assumptions made by the traditional economic auction theory. An online auction can be seen as an interactive economic information system, where user-system interactions are usually very complex. It is important to note that the interactions are not isolated, but successive interactions become a loop-feedback mechanism, that we call reactivity, where the user behavior affects the auction negotiation and vice-versa. In this paper we describe a new hierarchical characterization model for online auctions and apply this model to a real case study, showing its advantages in discovering some online auction negotiation patterns. The results demonstrate that our characterization model provides an efficient way to open the auction dynamics's "black box". We also propose an abstraction named Auction Model Graph (AMG) which enables the temporal analysis of the negotiation. This work is part of a research to analyze reactivity in e-business, that may contribute to understand the business dynamics and has wide applicability to activities such as designing recommendation agents, service personalization, and site interaction enhancement.

- NAEC07
  CONFERENCE: Networking and Electronic Commerce Research Conference 2007 (NAEC 2007)
  TITLE: "Characterizing Bidding Behavior in Internet Auctions"
  AUTHORS: Adriano Pereira, Leonardo Rocha, Fernando Mourão, Wagner Meira Jr., Paulo Goes.
  DATE: October 18-21, 2007
  PLACE: Lake Garda, Italy
  BRIEF DESCRIPTION: The Internet has developed a networked community characterized by the possibility of people who are physically and temporally separated to interact with each other. One of the most popular services provided by this environment is the online auctions, whose characteristics are significantly different from traditional auctions. For example, electronic auctions typically last much longer and their participants are dispersed in terms of when they follow (and thus bid) the auction, and where they are located. Although there are several relevant studies of online auctions, they do not consider the dynamics of the negotiation, which is handled as a "black box". Our research has pointed out that the negotiation patterns vary significantly across auctions and the same bidder has different behavior profiles in different auction negotiations. These and other findings motivate us to consider an online auction as a dynamic application environment in which bidders interact with the system in a feedback loop, in what we call reactivity. Considering this new concept, in this work we describe a new hierarchical organization and a multilevel characterization methodology for Internet auctions, that we apply to a real case study to characterize bidding behavior. The results show how our characterization methodology explains in detail the bidding behavior. Moreover, the different bidding behavior profiles can be used to understand, for example, why some auctions present fast price growing than other ones, and why some winners are defined earlier in some auction negotiations than others, where the winner is defined only in the last minute. This work is part of our ongoing research to analyze reactivity in e-business.

- LAWEB07
  CONFERENCE: 5th Latin American Web Congress - LAWEB
  TITLE: "Characterization of Online Auctions: Correlating Negotiation Patterns and Bidding Behavior"
  AUTHORS: Adriano Pereira, Leonardo Rocha, Fernando Mourão,Wagner Meira Jr.,

Paulo Goes.
DATE: October 31st - November 2nd, 2007
PLACE: Santiago, Chile
BRIEF DESCRIPTION: Online auctions have become a major electronic commerce channel in terms of revenue, reaching an enormous and very diverse population of participants all over the world. Modeling the factors that drive the dynamics of an auction, that is, the interactions that happen during its negotiation, is crucial to improve the customer experience for both sellers and buyers. Auction dynamics can be seen as complex and non-isolated interactions, where successive interactions become a loop-feedback mechanism. In what we call reactivity, the user behavior affects the auction negotiation and vice-versa. In this work, we develop a characterization approach for online auctions to capture the reactivity concept. Using the characterization, we are able to model both auction negotiation patterns and bidding behavior and correlate them. Our approach is novel, and the results start to shed light into measuring reactivity in online auctions. Our aim is to explain how the auction negotiation affects the bidders' behavior and vice-versa, and to relate the correlation auction pattern - bidding behavior to outcome measures.

- JWE - Journal under review
  CONFERENCE: JWE - Journal of Web Engineering
  TITLE: "Characterizing and Modeling Reactivity to Improve Web System's QoS"
  AUTHORS: Adriano PEREIRA, Leonardo SILVA, Wagner MEIRA Jr., Walter SANTOS
  SUBMITTED IN: December, 2006.
  CODE: JWE061129

- ITM - Journal under review
  CONFERENCE: Journal of Special Topics in Information Technology and Management
  TITLE: "A Hierarchical Model and Characterization Methodology for Online Auctions"
  AUTHORS: Adriano Pereira, Leonardo Rocha, Fernando Mourão, Paulo Goes, Wagner Meira Jr.
  SUBMITTED IN: June, 2007.

Appendix A

Case Study - eBay

In this section we apply our methodology to another case study, using another dataset from eBay. The dataset consists of 1963 auctions of the video game console *PlayStation*. PlayStation 2 (PS2) is Sony's second video game console, the successor to the PlayStation and the predecessor to the PlayStation 3. Its development was announced in March 1999, and it was first released in Japan on March 4, 2000, in North America on October 26, 2000 and in Europe on November 24, 2000. PS2 is part of the sixth generation era, and has become the fastest selling and arguably the most dominant home console of video game history, with over 115 million units shipped worldwide by December 2006.

A.1 Applying Methodology to General Characterization

In this section we apply our methodology to another case study, using another dataset from eBay. The dataset consists of 1963 auctions of the video game console *PlayStation*. PlayStation 2 (PS2) is Sony's second video game console, the successor to the PlayStation and the predecessor to the PlayStation 3. Its development was announced in March 1999, and it was first released in Japan on March 4, 2000, in North America on October 26, 2000 and in Europe on November 24, 2000. PS2 is part of the sixth generation era, and has

become the fastest selling and arguably the most dominant home console of video game history, with over 115 million units shipped worldwide by December 2006.

The dataset consists of auctions from 11/13/2002 to 01/16/2003, that have attracted 31316 bids. A preliminary analysis of the data shows that: all auctions achieve success in selling the product; the number of distinct sellers is 676, showing that auctions are not concentrated among a small number of sellers; there is a significant number of bids per auctions (15.95), which indicates the level of competition during the negotiation; and the average winner price is U$185.45.

Table A.1 presents some important auction information that is relevant to understand auction dynamics. From Table A.1 we can see that the average number of sessions per sequence is small, just 1.57, since there is a tendency to find one or more sequences with one session in general. On the other hand, the average number of sequences per auction (6.56) shows that the dynamics of the negotiation is rich, which motivates our analysis. Another aspect we analyze is the active and inactive times of the auctions. The active time, defined as before as the total time of auction activity, is very small, only 2.2%.

#Auctions	**#Seq**	**#Ses**	**#(Ses/Seq)**	**#(Seq/Auct)**	**#(Ses/Auct)**	T_{act}	T_{inact}
1963	12887	20296	1.57	6.56	10.33	2.2%	97.8%

Table A.1: Auction Characterization - General Statistics

Figures A.1 and A.2 show the histograms of patterns for auction sequences and sessions, respectively. As described by Table 5.9, sequences are represented by 3 letters and sessions by 5 letters. As can be seen, thirteen sequence patterns have occurred (only patterns MZw e FZw have not happened). For sessions, 6 patterns (identified as OPTrw, OPTrW, OPTRw, OPTRW, MPtRw, and MPTRw) have not been observed in the dataset. As can be seen in the auction sequences histogram - Figure A.1 - the most popular patterns are: MNW, MNw, MSW, FNW, and INW. The pattern INW is an initial sequence. The patterns MNW, MNw and MSW are intermediary sequences. And the patterns FNW is a final one. In terms of competition, from this group only the pattern MSW is characterized by competition. Regarding the changing winner aspect, only MNw do not cause changes in the winner bidder. This result is similar to one observed in the first case study of *nintendo* auctions.

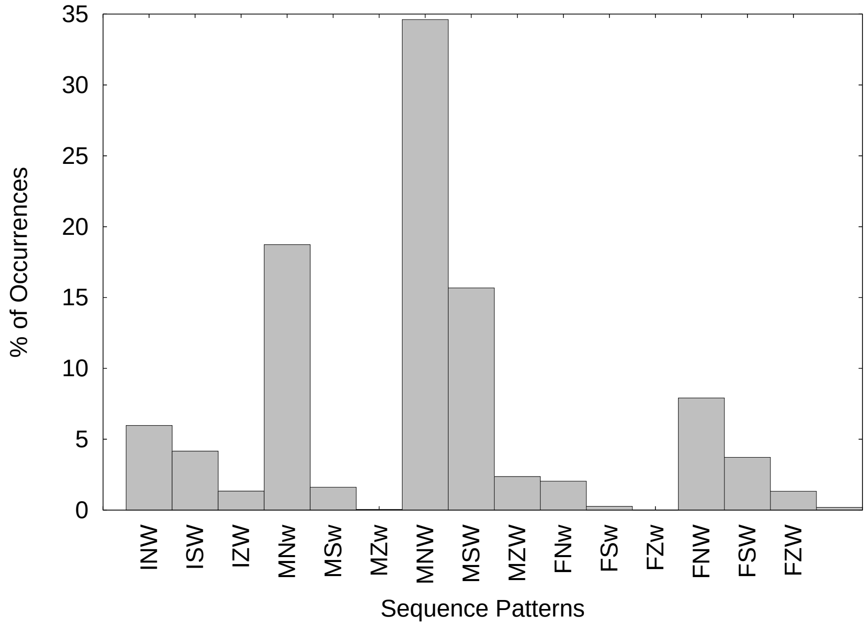


Figure A.1: Histograms of Auction Sequences

Table A.2 presents the statistics for auctions' sequences classification. As can be seen, the group of auctions has 12%, 53%, and 35% of initial, intermediary, and final sequences, respectively. In terms of competition, 73% of the sequences have no competition. And 83% of the sequences cause change in the winner bidder.

This general analysis and previous case study are examples of how our model can help understanding the auction dynamics and is the basis for reactivity modeling.

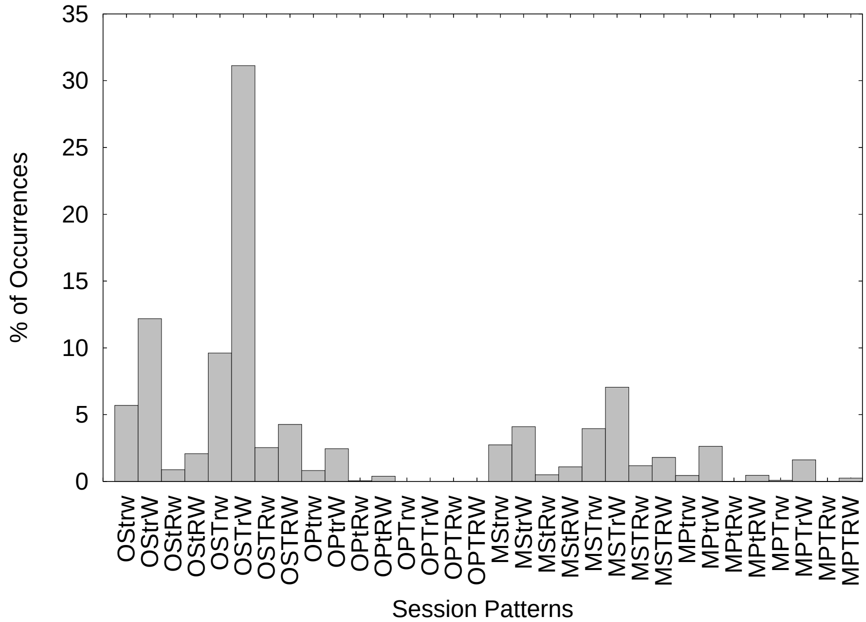


Figure A.2: Histograms of Auction Sessions

Time			Competition			Change W	
I	**M**	**F**	**N**	**S**	**Z**	**w**	**Y**
12%	53%	35%	73%	21%	6%	17%	83%

Table A.2: Auction's Sequences: Classification

A.2 Applying Methodology to Identify Auction Negotiation Patterns

Following our methodology, we present the characterization of the auction negotiation. We apply k-means clustering to identify similar auctions. To determine the number of meaningful clusters representing typical negotiation patterns in the dataset, we computed the dissimilarity measures beta-CV e beta-VAR in addition to the k-means error coefficient and visual inspection. Figures A.3 and A.4 show the beta-CV and beta-VAR analysis. The analysis pointed out 4 as the best number of clusters for auctions.

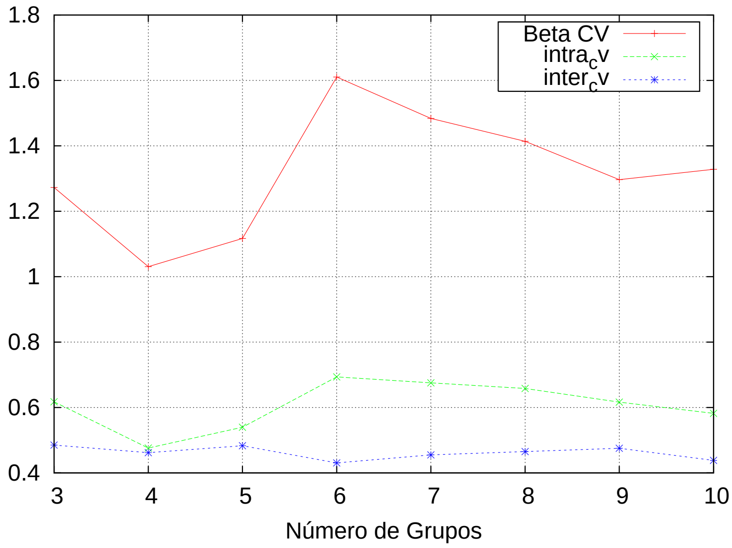


Figure A.3: Auctions - beta-CV and beta-VAR

Table A.3 shows the frequency distribution of the 15 possible sequences for the clusters. The last row of the table shows the percentage of auctions that falls in each cluster. We can describe each cluster as:

A0: represents auctions with medium level of activity (sequences). Most auctions in this cluster have two sequences. They have only 25.1% of competition, divided in 78% of successive and 22% of zigzag competition. 85.6% of its sequences change the winner. Only 7.6% of auctions present this pattern.

A1: group of auctions with a large number of sequences and high competition level (45.1%). The number of sequences that changes the winner is also high, almost 86%. It is a popular pattern, with 31.7% of auctions. It changes the winner in 70%. It is the most competitive auction pattern.

A2: auctions with very small number of sequences, almost all of them unique and with no competition in 97.4%. Almost all of them change the winner (99.7%), as expected, once the first sequence always changes the winner in eBay. This pattern occurs in 22.7% of the auctions.

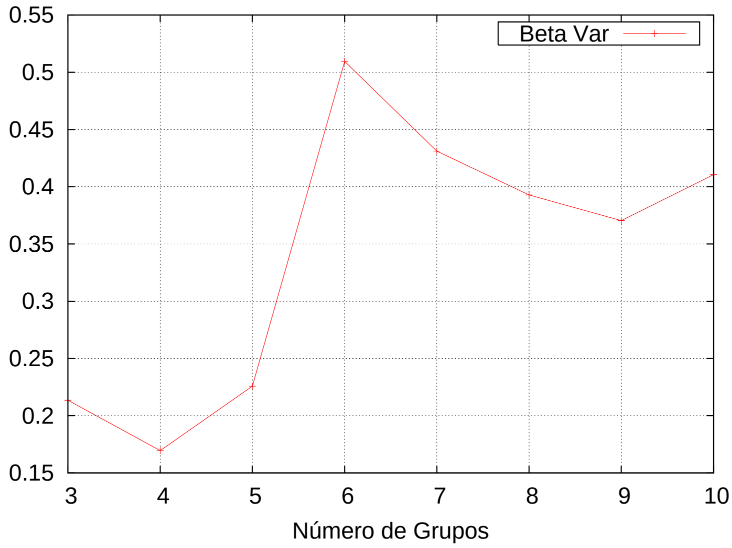


Figure A.4: Auctions - beta-CV and beta-VAR

A3: group of auctions with a large number of sequences, but with low competition level (around 27.0%). It is similar to *A1*, however the number of sequences that changes the winner is much higher, 83%. It is the most popular pattern, with 38% of auctions.

Once we have determined the four groups, we analyze the relationships between auction inputs and outputs with the negotiation. Table A.4 presents some important aspects for each cluster. It presents two auction negotiation inputs (starting bid and duration) and four outputs (number of bids and number of bidders, 1st Price and 2nd Price). Every measurement represents the average value of the attribute.

A0 has a high starting price (U$114.5) and a medium duration (3.7 days). However these auctions attracted a significant number of bidders that places an average of 11.5 bids.

A1 is a group with the lowest starting price (U$65.3) and highest duration (4.2 days). These auctions have attracted many bidders (9.8), who together places 18.2 bids. These characteristics and their typical negotiation pattern lead to a high final price.

A2 has the highest *starting price* and the shortest *duration*. In terms of its dynamics, we previously identified low activity and competition. However, it is interesting to note that these auctions achieve a high winner price (the *AVG 2nd price* is US$179.3). These can be explained by the fact that they present a high *starting price* and a significant amount of

Sequence	**Clusters**			
	A0	**A1**	**A2**	**A3**
1 (I-N-W)	35.1	4.8	0.0	6.2
2 (I-S-W)	0.0	6.0	0.8	4.3
3 (I-Z-W)	0.0	2.5	1.3	1.6
4 (M-N-w)	6.1	24.8	0.0	13.7
5 (M-S-w)	0.6	2.2	0.0	1.3
6 (M-Z-w)	0.2	0.2	0.00	0.00
7 (M-N-W)	13.6	16.9	0.0	47.1
8 (M-S-W)	7.0	20.5	0.0	12.1
9 (M-Z-W)	2.2	2.6	0.0	2.3
10 (F-N-w)	7.2	2.6	0.0	1.7
11 (F-S-w)	0.2	0.4	0.0	0.3
12 (F-Z-w)	0.0	0.0	0.0	0.0
13 (F-N-W)	12.8	5.7	97.4	4.5
14 (F-S-W)	11.9	5.2	0.0	4.1
15 (F-Z-W)	3.0	5.2	0.0	1.2
Frequency (%)	7.6	31.7	22.7	38.0

Table A.3: Distribution of Cluster's Sequences

bids (10 bids).

A3 and *A1* have similar characteristics, but different behavior in terms of competition profile. *A3* presents the highest number of bidders and bids, 9.9 and 18.3, respectively. These auctions have the highest first and second prices, despite the differences between them and the auctions from *A1* are very small.

Despite the similarities observed in auctions of groups *A3* and *A1*, we identify that they are different in terms of their auction negotiation evolution. In *A3*, almost half sequences are intermediary with no competition and change winner. In *A1*, this pattern is identified only in 16.9% and they have more sequences of type *MNw* and also sequences with successive competition. That is why the competition in *A1* is 70% higher than in auction of *A3*.

It is important to emphasize that this second case study presents more homogeneity since the dataset consists of products with the same specification, different from products

Aspects	**Clusters**			
	A0	**A1**	**A2**	**A3**
Starting Price (US$)	114.5	65.3	121.9	67.9
Duration (days)	3.7	4.3	2.9	4.2
#Bids	11.5	18.2	10.1	18.3
#Bidders	6.9	9.8	5.8	9.9
1st Price (US$)	184.7	185.2	182.9	185.8
2nd Price (US$)	179.9	182.4	179.3	182.4

Table A.4: Auction analysis

of the first case study (that has old and new products, and accessories, for example).

A.3 Applying Methodology to Identify Bidding Behavior Profiles

This section describes our bidding behavior characterization that follows the proposed methodology already presented in this doctoral dissertation.

According to our methodology, we apply a clustering technique to identify similar bidding behavior profiles based on the distribution of session patterns. We also consider the time-locality and competition aspects from the sequence in which the session is inserted to characterize the bidders. As previously explained, we also employ the attributes *ToE* and *ToX* to our feature vector.

Figures A.5 and A.5 show beta-CV and beta-VAR for bidding behavior clusters of our case study. Beta-CV denotes the intra-CV/inter-CV for the clusters. Whereas the intra-CV measures the coefficient of variation for the similarities intra-cluster, the inter-CV measure the similarities between different clusters. Thus, beta-CV is a measure of the quality of the clusters generated. The more stable the beta CV the better quality in terms of the grouping obtained. The ratio between the intra and intercluster variance, denoted beta-VAR is also useful in determining the quality of the clustering process (the smaller its values the better is the clustering set). We analyze each number of clusters using these measures and also the k-means error coefficient as well as visual inspection. The analysis pointed out 7 as the best number of clusters for bidding behavior.

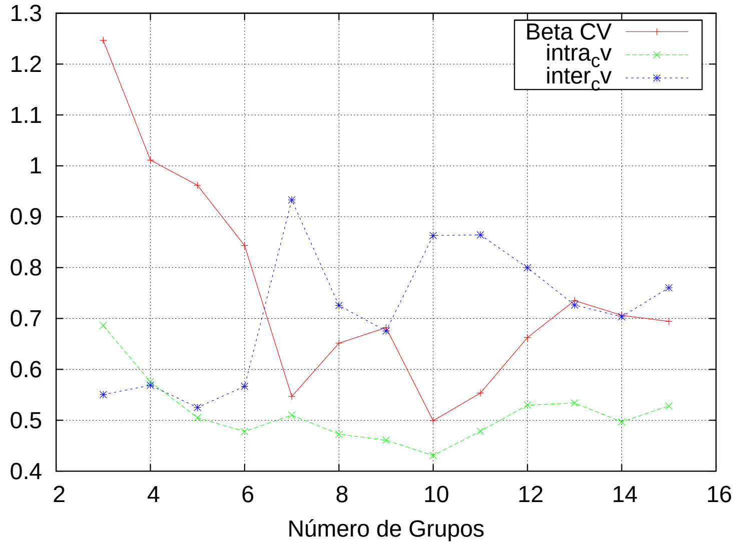


Figure A.5: Bidding Behavior Clustering - beta-CV

Next we present an analysis of each of the cluster, where we adopt the notation Bn to identify each bidding behavior cluster, where n varies from 0 to 6. Following our analysis, we present the characterization of each of these groups of bidders:

- **B0**: bidders who act in intermediary (99%) negotiation sequences, typically without competition (86%). Most of their sessions have more than one bid (84%), are triggered (90%), non-recurrent (73%), and change winner (92%). They act during in the middle of the auction negotiation (55-62% of duration time), placing 2.9 bids in average. These bidders represent 5.2% of bidding behaviors.

- **B1**: these bidders mainly act in initial (22%) and intermediary (78%) sequences of the auctions and in competitive situations (63%) with successive type predominance

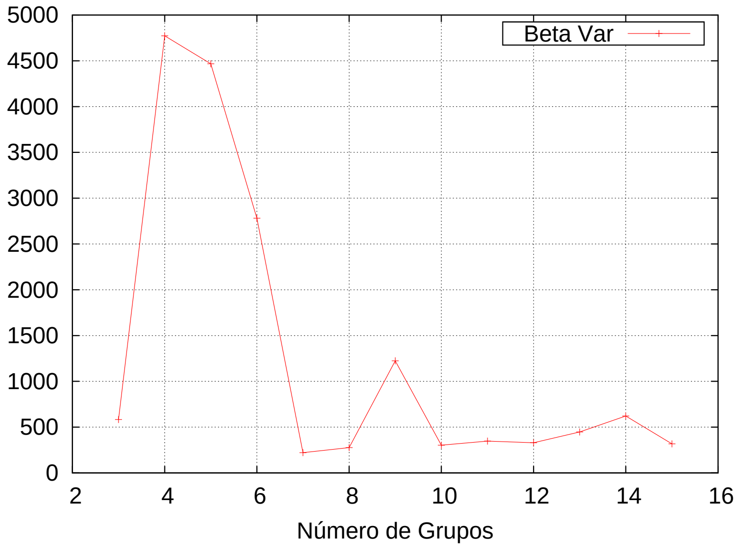


Figure A.6: Bidding Behavior Clustering - beta-VAR

(almost 80% of the competitions). Most of their sessions have only one bid (69%), are triggered (55%), non-recurrent (82%), and change winner (75%). They act in earlier stages of the auction negotiations (25-29% of duration time), placing 1.7 bids in average. They represent 26.6% of the bidders, the most frequent bidding behavior.

- **B2**: bidders who act typically in the first sequence (85%) of the auctions and never in the last one, in situation with high competition (90%) and of successive type, placing more than one one bid per session in 64%. Their sessions are triggered in 47%, they are non-recurrent in 85%, and change winner in 62%. They usually place 1.9 bids per auction. This behavior happens in 6.7% and during the earliest phases of the negotiation (from 14 to 18% of auction timing duration).

- **B3**: these bidders mainly act in the first sequences (76%) of the auctions and sometimes in intermediary sequences (23%), in situations with no competition (89%), placing only one bid (92%) per session. As expected from the previous characteristics, their sessions are triggered in 91%. Moreover, they are non-recurrent in 76%, change winner in 90%, and place only one bid in average. This behavior happens in 4.2% and during the initial phase of the negotiation (from 28 to 36% of auction timing duration).

- **B4**: group of bidders that act typically in the intermediary (24%) and final (70%) sequences, in competition situations (73%), with predominance of zigzag competition type (54%). Their sessions have one bid in 72% of occurrences. Also, 51% of their session are triggered and they have not participated yet in negotiation in every case. They change winner in 75% of their sessions and place on average 1.45 bids. They act during the final moments of the auction duration (after 98% of timing duration). These group is the third most frequent one, happening in 24.8% of the behaviors.

- **B5**: bidders who place bids in intermediary sequences (99%) and the rest in final ones, and their sequences are balanced in terms of competition (45% of no competition and 55% of competition). All competitions have successive pattern and 85% of their sessions have one bid. 2/3 of their sessions are triggered, 84% are recurrent, and 81% change the winner. They act late in the auction negotiation (between 82 and 86% of duration time), placing typically 1.3 bids in average. They represent the second most frequent behavior, with 25.2%.

- **B6**: bidders that act in intermediary sequences (100%), in scenarios with no competition (99%). Their sessions are triggered, however they do not change winner (96%). Only 7% of them have already participated in the current negotiation before (recurrent). They act typically after the middle of the auction negotiation, from 59 to 61% of negotiation timing duration. This behavior happens in 7.4%. They place 1.1 bids in average.

In Table A.5, the *Aspects* columns are measures of outcomes for each cluster and *F* presents the percentage of occurrence for each one. Comparing the results from Figure 5.27 and Table 5.16, it is possible to understand some interesting aspects about auction negotiation and bidding behavior. *BR* measures the bidder reputation in terms of experience in eBay auctions, *NoB* is the number of bids, δ *1st* is first price variation, δ *2nd* is second price variation, *ST* is the bidder session time (average duration), *S/B* is the average number of sessions per bidder and, finally, *W* is the percentage of bidders that are winners. The attributes *ToE* and *ToX* have already been explained.

Id	**Aspects**									**F**
	BR	*NoB*	δ**1st**	δ**2nd**	**ST**	**S/B**	**W**	*ToE*	*ToX*	(%)
B0	16.49	2.88	9.59	18.54	211.40	1.05	5	5.49	6.20	5.2%
B1	30.25	1.74	19.22	16.57	184.84	1.02	1	2.49	2.89	26.6%
B2	27.10	1.90	15.23	25.29	74.34	1.01	< 1	1.44	1.83	6.7%
B3	37.51	1.08	19.90	72.25	1.30	1.04	4	2.79	3.61	4.2%
B4	41.23	1.45	6.69	27.90	179.71	1.01	37	9.80	9.82	24.8%
B5	28.57	1.29	9.28	7.28	31.02	1.03	8	8.21	8.58	25.2%
B6	37.84	1.07	0.22	11.62	3.76	1.04	1	5.93	6.12	7.4%

Table A.5: Bidding Behavior Analysis

Analyzing Table A.5, we see that the most frequent clusters are *B1* (26.6%), *B5* (25.2%), and *B4* (24.8%). Together they correspond to almost 87% of the amount of bidders. The rarest bidding behavior profiles are *B3* and *B0*, with 4.2% and 5.2%, respectively.

In terms of auction's winner, *B4* has the highest rate (37%). Analyzing session duration, we identify that bidders from *B0* have longer sessions, around 3.5 minutes in average. The bidders from *B1* and *B4* also stay longer active, around 3 minutes in average. As expected, the shortest sessions, shorter than 5 seconds, belongs to bidders who usually have session with only one bid, such as *B3* and *B6*.

In terms of second price, which defines the winner price to pay in eBay, bidders from *B3* have the highest inter-bidding price (US$72.3), followed by *B4* and *B2*, with US$27.9 and US$25.3, respectively.

It is interesting to note that the number of sessions per bidder, in all clusters, is close to 1, showing that these dataset is characterized by bidders who typically act in only one time during the negotiation, with only one instantaneous bid or monitoring the negotiation during few minutes.

Moreover, bidders from *B4*, *B3* and *B6* present the highest reputation ratio, while *B0* has the lowest one.

This analysis shows the completeness and accuracy of our bidding behavior characterization. Moreover, our approach has uncovered two additional important facts that motivate us to further correlate bidding behavior with auction negotiation patterns:

- There are bidders who belongs to the same categories proposed in the literature, who present different behavior profiles using our new hierarchical model.
- There are bidders who present different behaviors in different auctions.

Analyzing all these bidding behaviors, we try to identify how to explain a behavior, what causes the bidder to act in one way or another one. We conclude that some bids may be explained by auction parameters (such as the *starting price*), other are determined by the competition scenario of one auction sequence. Nevertheless, we know that part of the bidding behavior is reactive (e.g., a bidder reacts to a competitor's bid which overcome his/her winner bid, placing another bid in the same auction sequence, or a bidder who reacts to the last-minute chance, trying to win the auction negotiation) and may be modeled and the other one is not.

The next challenge is to work on the correlations between bidding behavior and auction negotiation patterns with the objective of identifying these correlations, as will be described in Section A.4.

A.4 Correlating Auction Negotiation Patterns and Bidding Behavior

This section presents an analysis of the correlation between the auction negotiation patterns and bidding behavior.

Table A.6 presents the relative number of occurrences of each bidding behavior profile over the auction negotiation patterns of eBay case study.

In *A0* we observe predominance of *B5* (30.7%), that is mainly characterized by small increments in terms of second price. Bidders from *B4* also have a significant participation in this group of auctions (26.1%). The high starting price (US$114.5) can explain why

Id	**Auctions**			
	A0	**A1**	**A2**	**A3**
B0	4.0	5.5	4.7	5.2
B1	18.2	27.8	23.5	27.7
B2	5.9	6.3	6.0	7.3
B3	8.2	4.0	4.5	3.7
B4	26.1	24.1	29.6	23.7
B5	30.7	24.7	24.1	25.3
B6	6.9	7.6	7.6	7.1

Table A.6: Bidding Behaviors versus Auction Negotiations

these auctions attract less bidders who act in initial stages than other ones (such as *A1* and *A3*. This auction pattern attracts mainly the bidders who act at the end of the auction negotiation.

Auctions from *A2* present similar characteristics with small differences to *A0*, in all aspects. They have the highest average starting price (U$121.9), the shortest duration (2.9 days), the smallest number of bidders and bids (10.1 and 5.8, respectively), and the shortest first and second prices (U$182.9 and U$179.3, respectively). Comparing to *A0*, they attract 30% more bidders from *B1*, who bids in the beginning, and 20% less bidders from *B5*, who acts at the end of the negotiation.

A1 and *A3* represent auctions with similar characteristics in terms of starting price (US$65.3 and US$67.9), which are the lowest starting prices comparing to the other groups, duration (4.3 and 4.2 days), which are the longest durations among all groups, high number of bidders (9.8 and 9.9) and bids per auction (18.2 and 18.3), and high winning prices (US$182.4 and US$182.4), which are the highest values. Moreover, these groups are very similar in terms of number of sequences. These groups are very popular, together they represent almost 70% of the auction negotiations. The difference between them relates to competition. Despite this, the outcomes are similar, as previously showed. Both of them present high participation of bidders from *B1*, which act in the beginning of the negotiation, probably attracted by the lower starting prices. But also, both of them, attract bidders from *B4* and *B5* who act later in the auction negotiation.

The small differences observed in these groups can be explained by the presence of 1% more bidders from *B2* in *A3*, who act typically in the first sequence of the auctions in situation with high competition (90%) and of successive type. Moreover, in auctions of group *A1*, there is a little more bidders from *B0* and *B6*. *B0* corresponds to bidders who act in intermediary negotiation sequences, typically without competition (86%). *B6* represents bidders that act in intermediary sequences (100%), in scenarios with no competition (99%) also.

Despite the differences previously explained, these four auction groups have similar average winning prices (variation is only *1.7%*), despite the high variation in terms of amount of bidders (81%), bids (71%), and duration (48%). We can identify two similar groups in terms of auction negotiation characteristics and bidders who act in the negotiations: *A0* and *A2*, *A1* and *A3*.

As can be seen through our case study, our approach considers detailed aspects related to auction negotiation dynamics, and also identifies the bidding behavior with a wealth of details. Furthermore, we can correlate the auction negotiation patterns with bidding behavior profiles, providing new ways of understanding how the negotiation characteristics affect the bidders and vice-versa. In the next section we present the reactive transitions applied to this dataset, showing the results of modeling reactivity.

A.5 Modeling Reactivity - Reactive Transitions

In this section we present the results for the reactive transitions, as previously explained (Section 5.5.1, that is the best way we find to model reactivity in online auctions.

The general idea of modeling reactivity, as we have previously presented in Chapter 3, is to identify the set of perception aspects that can affect the bidder behavior and also some characteristics that can be measured as a consequence of the bidder's action in the auction negotiation.

After doing many different experiments, we decide to adopt the following aspects to model each reactive transition:

We have four perception criteria:

- Negotiation Time (*Time*): represents the relative time of the auction negotiation perceived by the bidder when placing the bid.
- Winning Price (*WinP*): is the relative winning price of the auction negotiation perceived by the bidder when placing the bid.
- Winning Bidder (*WinB*): represents the perception of the bidder when acting in terms of who is winning the negotiation at this moment (it is me - 1 - or not - 0).
- Competition Type (*Comp*): defines the competition type (no competition - N, successive competition - S, zigzag competition - Z) the bidder perceives when placing the bid.

We have consider two consequences in the auction negotiation as resultant of the action (bid):

- Inter-Bidding Price - (*IBP*): is the variation of price the current bid causes in the auction negotiation.
- Change Winner (*ChW*): indicates if the action changes the winner (1) or not (0).

We discretize some of these measures. The negotiation time can assume the following values:

- I: initial phase of the auction negotiation (up to 0.3 - that is, up to 30% of the negotiation).
- M: middle phase of the auction negotiation (from 0.3 to 0.8).
- F: final phase of the auction negotiation (from 0.8 to 0.99).
- S: sniping phase of the auction negotiation (more than 0.99), that is to capture the last second bidding behavior.

The winning price assume the following values:

- L: low (less than U$150)

- M: medium (from U$150 to U$200)

- H: (more than U$200)

The *IBP* assumes the following values:

- VL: very low (less than U$1)

- L: low (from U$1 to U$5)

- M: medium (from U$5 to U$15)

- H: high (more than U$15)

It is important to say these values are based on the statistical analysis of our dataset. The attributes *winning bidder* and *change winner* assume boolean values (0 or 1). The *competition type* has already been specified in categories. Therefore each reactive transition can be represented by 4 perception attribute values and 2 attributes obtained from the action.

The set of perception criteria can assume 72 (4*3*2*3) possibilities and there are 8 (4*2) options in terms of action's consequence in the auction negotiation.

As already explained in Section 5.5.1, We apply a data mining technique named association rules. Association rules have to satisfy constraints on measures of significance and interestingness. We call these measures *Support* and *Confidence*.

In the next section we present the results of modeling reactivity in online auction using reactive transitions.

A.5.1 Results

In this section we present the results we achieve using the idea of reactive transitions. We divide the results in general (applied to the role dataset without any filtering), aggregated by bidder, aggregated by auction, and also the analysis considering the bidder in auction matching.

A.5.1.1 General Results

Applying the methodology to generate the reactive transitions for the role dataset, setting the minimum support very low and a minimum confidence of 50%, we obtain the results presented in Table A.7.

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
87.5	0.07	S	M	1	Z	→	L	0
73.1	0.06	S	M	1	N	→	L	0
69.6	0.12	S	M	1	S	→	L	0
64.4	0.09	I	L	1	Z	→	VL	0
57.9	0.07	F	M	1	S	→	L	0
54.3	0.16	F	M	1	N	→	L	0

Table A.7: Reactive Transition - General Dataset

For example, the first transition *S M 1 Z* $\rightarrow$ *L 0* indicates that in 87.5% with a support of 0.07% a bidder in the sniping phase of the negotiation time, observing a market price, who is winning the auction negotiation, in a situation of zigzag competition, places a bid causing a low inter-bidding price and no change in the winner bidder. As can be seen, these rules representing the reactive transitions are not representative.

It was expected that these general analysis would not be representative, once is very difficult to identify reactivity without grouping any similarities of bidding behavior, auction negotiation or both. Continuing the analysis in this direction, the next sections present the results aggregated by bidder, auction and both of them, respectively.

A.5.1.2 Bidder Results

In this section we present the main reactive transitions according to the bidding behavior profile. We have choosen some groups of bidders to show the results and also present only the main relevant ones for each group.

For each presented result, the labels in the tables mean: *Conf.* is confidence, *Sup.* is the support, *Time* is the relative time where the auction negotiation is when the bidder acts, *WinP* is the relative winning price of the negotiation, *WinB* is an attribute that indicates whether the bidder is the current winner (1) or not (0), *Comp* indicates the competition type (N for no competition, S for successive, Z for zigzag) perceived by the bidder, *IBP* is the price variation the bid causes, and *ChW* is an attribute that indicates whether the winning bidder changes (1) or not (0).

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.2	F	H	1	N	→	H	0
100.0	0.1	M	H	0	N	→	VL	1
100.0	0.1	I	L	0	Z	→	VL	1
100.0	0.1	F	M	0	Z	→	VL	1
63.6	0.9	F	M	0	S	→	L	1
57.1	1.5	H	1	→	I	L	0	N
50.0	0.1	S	M	0	S	→	VL	1
50.0	0.1	S	M	0	S	→	L	1
50.0	0.1	S	L	0	N	→	VL	1
50.0	0.1	S	L	0	N	→	L	1

Table A.8: Reactive Transition - Bidding Behavior - Cluster B0

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
66.7	0.05	F	M	0	S	→	L	1
57.1	0.01	I	M	0	Z	→	L	1
50.0	0.1	M	L	0	Z	→	L	1
50.0	0.05	M	M	0	Z	→	L	1
50.0	0.05	I	H	0	S	→	H	1

Table A.9: Reactive Transition - Bidding Behavior - Cluster B1

For example, Table A.11 shows the results for group of bidders *B3*. As can be seen, for these bidding behavior profile, we identify a rule that means that: during intermediary (middle) phase (M), in negotiations where the winning price is medium (M), the bidder who acts is not the current winner, in a scenario without competition, the action of the bidder causes a high (H) increase in winning price and the bidder becomes winner at that

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.1	M	M	0	S	→	H	0
100.0	0.1	F	M	1	S	→	L	0
100.0	0.1	F	M	0	Z	→	M	0
100.0	0.1	F	M	0	S	→	M	1
100.0	0.1	F	L	0	S	→	L	1
66.7	0.2	M	M	0	N	→	H	1
63.6	0.7	M	L	0	N	→	H	1
50.0	0.2	F	M	0	N	→	M	0

Table A.10: Reactive Transition - Bidding Behavior - Cluster B2

moment. This typical transition occurs with a Confidence of 100.0% and a Support of 7.7%. The same reactive transition is observed for the similar situation in initial phase of the auction negotiation with a Confidence of 100.0% and a Support of 5.0%. This is a typical behavior of a bidder that is not winning the negotiation and decides to place a bid with a significant price variation, becoming the current winner of the auction.

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	7.7	M	M	0	N	→	H	1
100.0	5.0	I	M	0	N	→	H	1
100.0	0.2	M	H	0	N	→	H	1
100.0	0.2	F	L	1	N	→	VL	0
100.0	0.2	F	H	0	S	→	H	0
88.6	6.3	M	L	0	N	→	H	1
80.0	0.6	F	M	1	N	→	VL	0
72.9	5.6	F	M	0	N	→	H	1
66.7	0.3	F	M	0	S	→	L	0
50.0	0.2	F	L	0	S	→	M	0
50.0	0.2	F	L	0	S	→	L	1

Table A.11: Reactive Transition - Bidding Behavior - Cluster B3

It is interesting to emphasize that these bidders mainly act in the first sequences (76%) of the auctions and sometimes in intermediary sequences (23%), in situations with no competition (89%), placing only one bid (92%) per session. As expected from the previous characteristics, their sessions are triggered in 91%. Moreover, they are non-recurrent in 76%, change winner in 90%, and place only one bid in average. This behavior happens in 4.2% and during the initial phase of the negotiation (from 28 to 36% of auction timing duration).

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.1	S	H	1	Z	→	L	0
100.0	0.1	M	H	0	S	→	M	1
63.6	0.9	F	M	0	Z	→	L	1
56.2	2.5	S	M	0	Z	→	L	1

Table A.12: Reactive Transition - Bidding Behavior - Cluster B4

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
90.0	0.2	M	L	0	Z	→	L	1
75.0	0.1	S	M	0	Z	→	L	1
75.0	0.1	S	H	0	N	→	L	1
70.0	0.4	F	M	0	Z	→	L	1
66.7	0.2	S	H	0	S	→	L	0
61.9	1.0	M	M	0	S	→	L	1
60.0	0.2	S	L	0	N	→	VL	1
60.0	0.1	M	H	0	S	→	L	1
52.0	4.3	F	M	0	S	→	L	1
50.0	0.1	F	L	0	Z	→	L	1

Table A.13: Reactive Transition - Bidding Behavior - Cluster B5

A.5.1.3 Auction Results

In this section we present the main reactive transitions according to the auction negotiation pattern. We have choosen some groups of auctions to show the results and also present only the main relevant reactive transitions (considering their Confidence) for each group. For each presented result, the labels in the tables mean the same that we have already presented in the last section.

Table A.15 shows the results for group of auctions *A0*. Remembering the auction characterization we have already presented in this document, these auctions have medium level of activity (sequences), most of them have two sequences. They have only 25.1% of competition, divided in 78% of successive and 22% of zigzag competition. 85.6% of its

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.1	S	H	0	N	→	L	1
100.0	0.1	M	H	0	S	→	M	0
100.0	0.1	I	H	0	N	→	H	0
100.0	0.1	F	M	1	N	→	L	0
100.0	0.1	F	L	1	N	→	H	0
75.0	0.3	S	M	0	S	→	M	0
68.8	1.0	S	M	0	N	→	L	0
66.7	0.2	M	M	0	Z	→	M	0
66.7	0.2	F	M	0	Z	→	M	0
64.0	1.4	F	L	0	S	→	M	0
60.0	0.3	M	L	0	Z	→	L	0
56.3	0.8	I	M	0	N	→	L	0
52.0	1.2	M	M	0	S	→	M	0
50.0	0.1	F	L	0	Z	→	M	0
50.0	0.1	F	L	0	Z	→	L	0
50.0	0.1	F	H	0	S	→	M	0
50.0	0.1	F	H	0	S	→	H	0

Table A.14: Reactive Transition - Bidding Behavior - Cluster B6

sequences change the winner. Only 7.6% of auctions present this pattern. Despite the interesting rules identified with 100% of confidence, we observe a very small support for these rules.

Table A.16 shows the results for group of auctions *A1*.

Table A.17 shows the results for group of auctions *A2*.

Table A.18 shows the results for group of auctions *A3*.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.00	0.174520	S	M	1	S	→	L	0
100.00	0.116347	F	H	0	S	→	M	0
100.00	0.058173	S	M	1	N	→	L	0
100.00	0.058173	M	M	1	S	→	L	0
100.00	0.058173	M	M	0	Z	→	VL	1
100.00	0.058173	M	H	1	N	→	L	0
100.00	0.058173	F	H	1	N	→	VL	0
100.00	0.058173	F	H	0	Z	→	VL	1
75.00	0.174520	S	M	1	Z	→	L	0
75.00	0.174520	F	M	0	Z	→	L	1
66.67	0.116347	F	M	1	S	→	L	0
60.00	0.349040	M	L	0	Z	→	L	1
50.00	0.872600	S	M	0	Z	→	L	1
50.00	0.174520	S	L	0	N	→	VL	1
50.00	0.174520	M	M	1	N	→	L	0
50.00	0.174520	I	L	1	S	→	L	0
50.00	0.174520	F	M	1	N	→	L	0
50.00	0.116347	S	H	1	S	→	VL	0
50.00	0.116347	S	H	1	S	→	L	0
50.00	0.116347	F	H	0	N	→	L	1

Table A.15: Reactive Transition - Auction Negotiation - Cluster A0

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
81.82	0.079547	S	M	1	Z	→	L	0
72.73	0.070709	S	M	1	N	→	L	0
62.50	0.132579	F	L	1	N	→	L	0
60.00	0.106063	S	M	1	S	→	L	0
52.38	0.097225	M	M	1	N	→	L	0

Table A.16: Reactive Transition - Auction Negotiation - Cluster A1

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.00	0.089425	I	M	1	S	→	M	0
100.00	0.044713	S	M	1	Z	→	L	0
100.00	0.044713	S	M	1	N	→	L	0
83.33	0.111782	F	M	1	S	→	L	0
80.00	0.089425	F	H	0	S	→	L	0
76.19	0.357702	F	M	1	N	→	L	0
73.68	0.625978	I	L	1	Z	→	VL	0
60.00	0.134138	F	L	0	Z	→	M	0
60.00	0.067069	S	M	1	S	→	L	0
60.00	0.067069	F	L	1	N	→	M	0
55.88	0.849542	S	M	0	Z	→	L	1
54.60	4.381847	S	M	0	N	→	H	1
50.00	0.245920	S	L	0	Z	→	L	0
50.00	0.044713	S	H	0	Z	→	L	1
50.00	0.044713	F	M	1	Z	→	M	0

Table A.17: Reactive Transition - Auction Negotiation - Cluster A2

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.00	0.051084	S	M	1	Z	→	L	0
75.00	0.153251	S	M	1	S	→	L	0
66.67	0.058381	S	M	1	N	→	L	0
56.25	0.065679	F	M	1	S	→	L	0

Table A.18: Reactive Transition - Auction Negotiation - Cluster A3

A.5.1.4 Bidder in Auction Results

This section presents some results in the context of bidder in auction. In our case study, we have identified 7 different groups of bidders and 4 groups of auctions. From these groups, we can have 28 different scenarios to analyze.

The number of reactive transitions identified in this 28 bidder in auction scenarios are very high, suggesting the accuracy of the approach. There are 301 reactive transition rules with a confidence higher than 50%. Considering a 100% confidence, we have 138 rules, that represent the modeled reactivity patterns.

We choose some results about the reactive transitions considering the analysis of bidder in auction sets. This aggregation is the most restrictive analysis that we can perform. On the other hand, we can say that is the most confident analysis, once it has the most complete set of knowledge about the environment (auction negotiation) and the agent (bidder behavior).

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	2.7	F	L	0	S	→	VL	1
100.0	2.7	F	M	0	S	→	L	0
60.0	8.1	M	M	0	N	→	L	1
50.0	8.1	F	M	0	N	→	L	1
50.0	2.7	I	L	0	S	→	M	1
50.0	2.7	I	L	0	S	→	L	1

Table A.19: Reactive Transition - Bidder in Auction - Cluster A0B0

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	1.2	M	M	0	S	→	L	1
100.0	0.6	M	L	0	Z	→	L	1
100.0	0.6	M	H	0	N	→	L	1
100.0	0.6	I	M	0	S	→	M	0
66.7	1.2	M	L	0	S	→	L	1
60.0	1.8	M	M	0	N	→	L	1

Table A.20: Reactive Transition - Bidder in Auction - Cluster A0B1

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	1.9	M	L	0	N	→	H	1

Table A.21: Reactive Transition - Bidder in Auction - Cluster A0B2

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	22.7	M	M	0	N	→	H	1
100.0	8.0	I	M	0	N	→	H	1
100.0	4.0	M	L	0	N	→	H	1
100.0	2.7	F	M	1	N	→	VL	0
100.0	1.3	F	M	0	S	→	L	1
100.0	1.3	F	L	0	S	→	L	1
91.7	14.7	F	M	0	N	→	H	1
66.7	2.7	F	L	0	N	→	H	1

Table A.22: Reactive Transition - Bidder in Auction - Cluster A0B3

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.8	F	M	0	Z	→	L	1
100.0	0.4	S	H	0	Z	→	L	1
100.0	0.4	M	M	0	S	→	L	1
100.0	0.8	F	H	0	S	→	M	0
100.0	0.4	M	L	0	N	→	M	0
60.0	2.4	S	M	0	Z	→	L	1
57.1	1.6	S	H	0	S	→	L	1
50.0	2.8	F	M	0	S	→	L	1
50.0	0.4	F	L	0	Z	→	M	1
50.0	0.4	M	M	0	N	→	M	0
50.0	0.4	F	L	0	Z	→	M	0
50.0	0.8	S	L	0	N	→	H	1
50.0	0.4	M	M	0	N	→	VL	1
50.0	0.4	F	H	0	N	→	VL	1
50.0	0.4	F	H	0	N	→	L	1

Table A.23: Reactive Transition - Bidder in Auction - Cluster A0B4

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.4	S	H	0	S	→	L	0
100.0	0.4	M	H	0	N	→	L	0
100.0	0.7	S	L	0	N	→	VL	1
100.0	0.4	S	H	1	S	→	VL	0
100.0	0.4	S	H	0	N	→	L	1
65.2	5.5	F	M	0	S	→	L	1
52.6	3.7	S	M	0	N	→	L	1
50.0	0.4	F	H	0	N	→	M	0
50.0	0.4	I	L	0	N	→	VL	1
50.0	0.4	F	L	0	Z	→	VL	1
50.0	0.4	I	L	0	N	→	H	1
50.0	0.4	F	H	0	N	→	L	1
50.0	0.4	F	L	0	Z	→	H	0

Table A.24: Reactive Transition - Bidder in Auction - Cluster A0B5

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	1.6	S	H	0	N	→	L	1
100.0	1.6	I	L	0	Z	→	H	0
100.0	1.6	I	M	0	N	→	L	0
100.0	1.6	F	L	0	S	→	L	0
100.0	1.6	I	M	0	S	→	M	0
75.0	4.8	M	L	0	S	→	M	0
66.7	3.2	M	M	0	S	→	M	0
66.7	3.2	F	L	0	N	→	M	0
63.6	11.3	F	M	0	N	→	L	0
53.8	11.3	I	L	0	N	→	M	0
50.0	3.2	M	M	0	N	→	L	0
50.0	3.2	I	L	0	S	→	L	0
50.0	3.2	F	M	0	S	→	L	0
50.0	3.2	M	M	0	N	→	M	0
50.0	3.2	F	M	0	S	→	M	0

Table A.25: Reactive Transition - Bidder in Auction - Cluster A0B6

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.3	M	H	0	N	→	VL	1
100.0	0.3	I	L	0	S	→	VL	1
100.0	0.3	F	M	0	Z	→	VL	1
100.0	0.3	F	M	0	S	→	L	1
100.0	0.3	F	L	0	S	→	L	1
60.0	1.0	I	M	0	N	→	M	0
50.0	0.3	F	H	0	N	→	M	0
50.0	0.3	S	M	0	S	→	VL	1
50.0	0.3	S	M	0	S	→	L	1
50.0	0.3	F	H	0	N	→	L	0

Table A.26: Reactive Transition - Bidder in Auction - Cluster A1B0

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.1	I	H	0	S	→	H	1
100.0	0.1	I	H	0	N	→	M	1
100.0	0.1	F	M	1	N	→	L	0
100.0	0.1	F	M	0	S	→	L	1
100.0	0.1	F	L	0	Z	→	M	0
60.0	0.2	M	M	0	S	→	L	0
50.0	0.1	F	L	0	S	→	H	0
50.0	0.1	M	M	0	Z	→	L	0
50.0	0.1	F	L	0	S	→	L	0
50.0	0.1	M	M	0	Z	→	L	1
50.0	0.1	M	L	0	Z	→	L	1
50.0	0.1	M	L	0	Z	→	VL	1

Table A.27: Reactive Transition - Bidder in Auction - Cluster A1B1

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.6	F	M	0	N	→	M	0
100.0	0.6	M	M	0	N	→	H	1
66.7	0.6	M	L	0	N	→	H	1
60.0	0.9	I	M	0	S	→	L	0
50.0	0.3	I	M	0	N	→	H	1
50.0	0.3	F	L	0	N	→	M	1
50.0	0.3	I	M	0	N	→	H	0
50.0	0.3	F	L	0	N	→	L	1

Table A.28: Reactive Transition - Bidder in Auction - Cluster A1B2

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	6.6	M	M	0	N	→	H	1
100.0	2.3	I	M	0	N	→	H	1
100.0	0.9	F	M	0	S	→	L	0
100.0	0.5	F	M	1	N	→	VL	0
100.0	0.5	M	H	0	N	→	H	1
86.4	8.9	M	L	0	N	→	H	1
66.7	3.8	F	M	0	N	→	H	1

Table A.29: Reactive Transition - Bidder in Auction - Cluster A1B3

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.1	S	H	1	Z	→	L	0
100.0	0.1	S	H	1	N	→	L	0
100.0	0.1	M	L	0	Z	→	L	1
100.0	0.1	F	L	1	N	→	VL	0
100.0	0.1	F	H	1	S	→	H	0
75.0	0.2	F	H	0	S	→	L	1
66.7	0.1	M	M	0	N	→	L	1
60.0	1.1	F	M	0	Z	→	L	1
57.1	0.3	S	H	0	Z	→	L	0
52.9	2.0	S	M	0	Z	→	L	1
51.2	1.7	S	L	0	N	→	VL	1

Table A.30: Reactive Transition - Bidder in Auction - Cluster A1B4

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.2	F	L	0	Z	→	L	1
100.0	0.1	S	H	0	Z	→	L	0
100.0	0.1	S	H	0	N	→	VL	1
100.0	0.1	I	M	0	N	→	H	1
100.0	0.1	S	M	0	Z	→	L	1
85.7	0.4	M	L	0	Z	→	L	1
83.3	0.4	F	M	0	Z	→	L	1
66.7	0.1	S	L	0	N	→	VL	1
66.7	0.1	M	M	0	Z	→	L	1
66.7	0.1	M	H	0	S	→	L	1
63.2	0.9	M	M	0	S	→	L	1
60.4	4.7	F	M	0	S	→	L	1
57.1	0.3	F	M	1	N	→	VL	0
50.0	3.0	S	M	0	N	→	L	1
50.0	0.1	F	L	1	N	→	M	0
50.0	0.1	S	H	0	S	→	L	0
50.0	0.1	F	L	1	N	→	L	0
50.0	0.1	I	L	0	N	→	H	1
50.0	0.1	I	L	0	N	→	L	1

Table A.31: Reactive Transition - Bidder in Auction - Cluster A1B5

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.5	S	M	0	S	→	M	0
100.0	0.2	F	H	0	S	→	H	0
100.0	0.2	M	M	0	Z	→	M	0
100.0	0.2	M	H	0	S	→	M	0
100.0	0.2	M	L	0	Z	→	L	0
75.0	0.7	I	M	0	N	→	M	0
66.7	0.5	I	M	0	S	→	L	0
62.5	1.2	S	M	0	N	→	L	0
57.1	1.0	M	M	0	S	→	L	0
50.0	0.2	F	H	0	N	→	M	1
50.0	1.0	I	L	0	Z	→	H	0
50.0	0.2	F	M	0	Z	→	M	0
50.0	0.2	F	M	0	Z	→	L	0
50.0	0.2	F	H	0	N	→	L	0

Table A.32: Reactive Transition - Bidder in Auction - Cluster A1B6

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.9	S	L	0	N	→	L	1
100.0	0.9	S	M	0	N	→	H	0
100.0	0.9	I	M	0	N	→	H	0
100.0	0.9	F	H	1	N	→	H	0
100.0	0.9	I	L	0	S	→	H	1

Table A.33: Reactive Transition - Bidder in Auction - Cluster A2B0

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.2	M	H	0	N	→	H	1
100.0	0.2	I	H	0	N	→	H	1
100.0	0.2	F	H	0	S	→	L	0
100.0	0.2	M	M	0	Z	→	M	1
100.0	0.4	M	L	0	Z	→	L	1
100.0	0.2	F	M	0	S	→	L	1
100.0	0.2	I	L	1	N	→	VL	0
100.0	0.2	F	M	0	N	→	M	0
66.7	0.4	M	M	0	S	→	H	0
50.0	0.2	I	M	0	Z	→	VL	1
50.0	0.8	I	M	0	S	→	L	1
50.0	0.2	I	M	0	Z	→	L	1

Table A.34: Reactive Transition - Bidder in Auction - Cluster A2B1

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	1.4	I	M	0	S	→	M	0
100.0	1.4	I	M	0	N	→	H	0
100.0	0.7	M	M	0	N	→	M	0
100.0	0.7	F	L	0	N	→	M	0
100.0	0.7	M	M	0	S	→	H	0
100.0	0.7	F	M	1	S	→	L	0
100.0	0.7	I	L	0	Z	→	VL	0

Table A.35: Reactive Transition - Bidder in Auction - Cluster A2B2

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	8.9	M	M	0	N	→	H	1
100.0	6.9	I	M	0	N	→	H	1
100.0	1.0	F	L	0	N	→	H	0
100.0	1.0	F	M	1	N	→	M	0
70.0	6.9	F	M	0	N	→	H	1
60.0	3.0	M	L	0	N	→	H	1

Table A.36: Reactive Transition - Bidder in Auction - Cluster A2B3

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.1	S	H	1	S	→	L	0
100.0	0.1	F	H	0	Z	→	L	0
100.0	0.1	M	H	0	S	→	M	1
66.7	0.3	F	M	0	Z	→	VL	1
65.4	25.6	S	M	0	N	→	H	1
53.6	2.2	S	M	0	Z	→	L	1
53.3	1.2	S	L	0	N	→	VL	1
50.0	2.8	F	M	0	S	→	L	1
50.0	0.1	M	L	0	S	→	M	0
50.0	0.6	S	L	0	Z	→	L	0
50.0	0.1	S	H	0	Z	→	L	0
50.0	0.1	M	L	0	N	→	L	0
50.0	0.7	S	H	0	S	→	L	1
50.0	0.1	S	H	0	Z	→	L	1
50.0	0.1	M	L	0	N	→	H	1
50.0	0.1	M	L	0	S	→	H	0

Table A.37: Reactive Transition - Bidder in Auction - Cluster A2B4

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.2	S	L	0	N	→	M	1
100.0	0.4	F	M	1	N	→	L	0
100.0	0.2	I	L	0	S	→	VL	0
100.0	0.2	S	L	0	S	→	VL	1
100.0	0.2	M	M	0	Z	→	VL	1
100.0	0.2	F	L	0	Z	→	VL	1
66.7	2.5	S	M	0	N	→	L	1
66.7	0.4	S	H	0	S	→	L	0
66.7	0.4	F	M	0	Z	→	L	1
54.5	2.2	M	L	0	S	→	L	1
50.9	5.1	F	M	0	S	→	L	1
50.0	0.2	F	H	0	N	→	VL	1
50.0	0.7	M	M	0	S	→	L	1
50.0	0.2	F	H	0	N	→	L	1

Table A.38: Reactive Transition - Bidder in Auction - Cluster A2B5

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.6	S	M	0	S	→	L	0
100.0	0.6	I	M	0	S	→	L	0
100.0	0.6	F	M	1	N	→	L	0
100.0	0.6	F	L	0	Z	→	M	0
100.0	0.6	F	H	0	S	→	M	0
73.3	6.4	M	M	0	N	→	M	0
66.7	2.3	F	L	0	S	→	M	0
66.7	1.2	S	M	0	N	→	L	0
60.0	1.7	I	M	0	N	→	L	0
58.3	4.1	I	L	0	S	→	H	0
56.7	9.9	F	M	0	N	→	L	0
50.0	1.2	I	L	0	Z	→	L	0
50.0	1.2	I	L	0	Z	→	M	0

Table A.39: Reactive Transition - Bidder in Auction - Cluster A2B6

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.3	S	L	0	N	→	VL	1
100.0	0.3	I	L	0	Z	→	VL	1
100.0	0.3	F	H	1	N	→	H	0
66.7	1.7	F	M	0	S	→	L	1
66.7	0.6	M	L	0	S	→	L	1
50.0	0.3	F	H	0	N	→	H	0
50.0	0.3	F	H	0	N	→	L	1
50.0	0.8	I	M	0	N	→	L	0

Table A.40: Reactive Transition - Bidder in Auction - Cluster A3B0

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.1	M	H	0	N	→	M	1
100.0	0.1	M	L	1	N	→	L	0
100.0	0.1	M	H	1	N	→	L	0
100.0	0.1	I	H	1	S	→	L	0
100.0	0.1	F	M	0	S	→	L	0
100.0	0.1	I	M	0	Z	→	L	1
100.0	0.1	M	M	0	Z	→	L	1
100.0	0.1	F	M	0	Z	→	L	1
100.0	0.1	F	H	0	N	→	L	1
100.0	0.1	I	H	0	N	→	M	0
66.7	0.1	F	L	0	N	→	M	0

Table A.41: Reactive Transition - Bidder in Auction - Cluster A3B1

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.2	F	M	0	S	→	M	1
100.0	0.2	F	L	0	S	→	L	1
100.0	0.2	F	M	0	Z	→	M	0
75.0	0.6	M	L	0	N	→	H	1
66.7	0.4	I	M	0	S	→	H	0
50.0	0.2	I	L	0	Z	→	M	0
50.0	0.6	I	M	0	N	→	H	1
50.0	0.4	F	L	0	N	→	L	0
50.0	0.2	I	L	0	Z	→	L	0
50.0	0.2	F	M	0	N	→	L	0
50.0	0.2	F	M	0	N	→	L	1

Table A.42: Reactive Transition - Bidder in Auction - Cluster A3B2

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	6.0	M	L	0	N	→	H	1
100.0	5.6	I	M	0	N	→	H	1
100.0	3.4	M	M	0	N	→	H	1
100.0	0.4	F	M	1	N	→	VL	0
100.0	0.4	F	L	1	N	→	VL	0
100.0	0.4	F	L	0	S	→	M	0
100.0	0.4	F	H	0	S	→	H	0
64.3	3.9	F	M	0	N	→	H	1

Table A.43: Reactive Transition - Bidder in Auction - Cluster A3B3

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.1	M	M	0	S	→	M	0
100.0	0.1	F	H	0	Z	→	M	0
100.0	0.1	M	M	0	Z	→	VL	1
100.0	0.1	S	H	1	Z	→	L	0
100.0	0.1	M	H	0	S	→	M	1
68.0	1.1	F	M	0	Z	→	L	1
66.7	0.1	M	L	0	S	→	L	0
58.8	3.0	S	M	0	Z	→	L	1
50.0	0.2	S	H	0	Z	→	L	1

Table A.44: Reactive Transition - Bidder in Auction - Cluster A3B4

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.2	M	L	0	Z	→	L	1
100.0	0.1	S	H	1	S	→	L	0
100.0	0.1	S	H	0	S	→	L	0
100.0	0.1	S	H	0	N	→	L	1
100.0	0.1	M	M	0	Z	→	H	1
75.0	0.2	F	M	1	N	→	M	0
66.7	1.3	M	M	0	S	→	L	1
66.7	0.1	S	M	0	Z	→	L	1
63.6	0.4	F	M	0	Z	→	L	1
57.1	0.2	F	H	0	N	→	L	1
51.6	1.9	S	M	0	S	→	L	1
50.0	0.1	S	L	0	N	→	VL	1
50.0	0.1	F	L	0	Z	→	VL	1
50.0	0.1	M	H	0	S	→	VL	1
50.0	0.1	I	L	0	N	→	VL	1
50.0	0.1	F	H	0	S	→	VL	1
50.0	0.1	F	L	0	Z	→	L	1
50.0	0.1	M	H	0	S	→	L	1
50.0	0.1	I	L	0	N	→	L	1
50.0	0.1	F	H	0	S	→	L	1

Table A.45: Reactive Transition - Bidder in Auction - Cluster A3B5

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.2	I	H	0	N	→	H	0
100.0	0.2	F	L	1	N	→	H	0
100.0	0.2	M	L	0	Z	→	L	0
100.0	0.2	I	L	0	Z	→	L	0
100.0	0.2	F	L	0	Z	→	L	0
100.0	0.2	S	M	0	S	→	M	0
100.0	0.2	F	M	0	Z	→	M	0
81.8	1.9	F	L	0	S	→	M	0
80.0	0.9	S	M	0	N	→	L	0
70.0	1.5	M	M	0	S	→	M	0
66.7	0.9	I	M	0	N	→	L	0
50.0	0.6	F	H	0	N	→	H	0
50.0	0.4	I	M	0	S	→	H	0
50.0	0.2	M	M	0	Z	→	L	0
50.0	0.4	I	M	0	S	→	M	0
50.0	0.2	M	M	0	Z	→	M	0

Table A.46: Reactive Transition - Bidder in Auction - Cluster A3B6

A.6 Analysis and Conclusion

This appendix describes another complete case study using a dataset from eBay.

After applying the methodology to characterize auction negotiation and bidding behavior, we analyze the correlation between them in online auctions, explaining the observed correlations through a wealth of details that our approach is able to provide. We capture the necessary attributes that characterize reactivity in terms of competition, price changes, winning status, etc., from the bidder's perspective and the overall auction's perspective. The characterization of the relevant reactivity periods is used as input to the identification of patterns of auction negotiation as well as bidding behavior. Using actual bidding data from eBay, we demonstrate the efficacy of our approach in explaining bidding dynamics and the effect on auction outcomes by correlating auction patterns to bidding behavior.

As can be seen, the reactive transitions is an approach that provide a intuitive way to understand the interactions in an online auction, enriching the analysis and also guaranteeing a good accuracy.

It is important to emphasize that this results become possible only after detailing our result's set. We consider in this last analysis the bidder in auction sets, that is, the most restrictive approach that we can have in this study. As expected, we achieve a confident result. This can be explained by the fact that we use the detailed characteristics about the environment (auction negotiation and its state) and the agent (the bidder considering her/his characteristics), therefore this context presents the most complete set of knowledge to analyze reactivity in this case study.

Appendix B

Experimental Methodology

This research work follows the experimental lifecycle represented by Figure B.1.

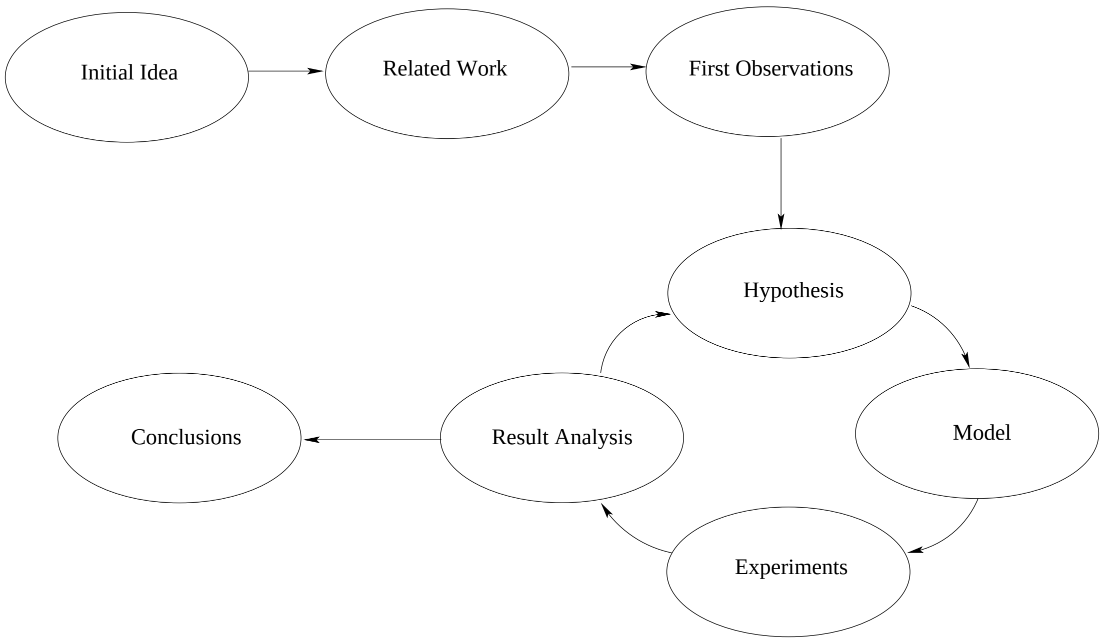


Figure B.1: Experimental Lifecycle

This experimental methodology can be described as:

- Initial idea: defines what the work addresses.
- Related work: identify works that are related to the context of the work in order to

learn more about the subject and identify the differences and the innovations of each other.

- First observations: scientists are usually curious about their surroundings and may notice something and want to understand more about it. This type of curiosity may lead them to observe their surroundings more carefully. As a result, when they document their observations, questions may arise which are then formulated into hypothesis.

- Hypothesis: as first observations are made, questions are formulated and it is important to try to answer these questions, which leads to guesses or hypothesis. If observations do not support these statements, the hypothesis is rejected. A hypothesis must be stated in a way that can be tested by the scientific method. Reviewing similar studies performed by others can also be helpful in this step.

- Model: in order to test the hypothesis, it is necessary to use a known model or to create a new one and data must be collected.

- Experiments: apply the model to validate it through experimentation.

- Result Analysis: when you complete your experiments, examine and organize your findings. Did your experiments give you the expected results? Why or why not? Was your experiment performed with the exact same steps each time? Are there other explanations that you had not considered or observed? Were there errors in your observations? Understanding errors and reporting that a suspected variable did not change the results can be valuable information. If possible, statistically analyze your data. This step is necessary to prove or disprove a hypothesis by experimentation.

- Conclusions: write the conclusions achieved from the experimental lifecycle. An experiment is done to prove or disprove an hypothesis. It is important to Identify the Future Research Needs, that may include areas of related interest that should be studied to better understand the subject.

Appendix C

Complete Set of Results - Online Auctions

This appendix shows the complete set of results about reactivity transitions, according to the case study presented in Section 5.5. The results are divided in *Bidder*, *Auction*, and *Bidder in Auction*, as already explained in Section 5.5.

C.1 Bidder Results

In this section we present the reactive transitions according to the bidding behavior profile.

For each presented result, the labels in the tables mean: *Conf.* is confidence, *Sup.* is the support, *Time* is the relative time where the auction negotiation is when the bidder acts, *WinP* is the relative winning price of the negotiation, *WinB* is an attribute that indicates whether the bidder is the current winner (1) or not (0), *Comp* indicates the competition type (N for no competition, S for successive, Z for zigzag) perceived by the bidder, *IBP* is the price variation the bid causes, and *ChW* is an attribute that indicates whether the winning bidder changes (1) or not (0).

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
66.7	1.8	M	L	1	N	→	VL	0
64.9	0.6	I	H	0	S	→	H	0
64.7	0.3	I	H	1	N	→	H	0
61.0	0.6	F	M	0	S	→	L	0

Table C.1: Reactive Transition - Bidding Behavior - Cluster B0

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
72.7	0.4	I	L	1	N	→	VL	0

Table C.2: Reactive Transition - Bidding Behavior - Cluster B1

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
60.8	0.6	S	M	1	Z	→	VL	0
60.5	0.5	S	M	1	N	→	VL	0

Table C.3: Reactive Transition - Bidding Behavior - Cluster B2

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
60.0	0.3	S	H	1	N	→	L	0

Table C.4: Reactive Transition - Bidding Behavior - Cluster B3

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
65.1	0.5	F	L	1	N	→	VL	0
62.2	0.6	S	M	1	Z	→	VL	0

Table C.5: Reactive Transition - Bidding Behavior - Cluster B4

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
80.6	0.4	M	H	0	N	→	H	1
84.0	1.6	I	H	0	N	→	H	1
82.0	6.0	I	M	0	N	→	H	1
71.4	0.6	M	L	1	N	→	VL	0
67.9	0.6	M	M	0	N	→	H	1

Table C.6: Reactive Transition - Bidding Behavior - Cluster B5

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
68.5	10.9	F	M	0	N	→	L	0
67.7	0.3	S	L	0	N	→	L	0
65.7	0.4	S	M	0	S	→	L	0
62.2	4.4	M	M	0	N	→	L	0
61.9	1.3	I	M	0	N	→	L	0
60.9	1.5	S	M	0	N	→	L	0

Table C.7: Reactive Transition - Bidding Behavior - Cluster B6

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
61.7	1.0	M	L	0	S	→	L	0

Table C.8: Reactive Transition - Bidding Behavior - Cluster B7

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
90.9	1.6	I	M	0	N	→	H	1
61.9	0.5	F	L	1	N	→	L	0
60.0	0.3	I	H	0	N	→	H	1

Table C.9: Reactive Transition - Bidding Behavior - Cluster B8

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
64.0	0.3	M	M	0	S	→	L	0
61.7	1.2	S	M	0	S	→	L	0

Table C.10: Reactive Transition - Bidding Behavior - Cluster B9

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
85.0	1.5	S	L	0	N	→	L	0
80.0	0.4	I	L	0	S	→	VL	0
72.1	5.5	S	M	0	N	→	L	0
65.4	1.5	S	L	0	S	→	L	0
63.4	7.0	S	M	0	S	→	L	0
62.5	0.5	M	M	0	S	→	L	0

Table C.11: Reactive Transition - Bidding Behavior - Cluster B10

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
94.3	6.0	M	M	0	N	→	H	1
83.3	1.3	M	H	0	N	→	H	1
77.7	13.1	F	M	0	N	→	H	1
77.4	0.8	S	M	1	N	→	VL	0
71.4	0.3	F	M	0	Z	→	L	0
71.4	4.2	S	M	0	N	→	H	1
68.0	2.2	F	H	0	N	→	H	1
66.7	0.3	S	M	1	S	→	VL	0

Table C.12: Reactive Transition - Bidding Behavior - Cluster B11

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
75.0	0.5	S	M	1	Z	→	VL	0
73.3	0.3	I	L	0	N	→	L	0

Table C.13: Reactive Transition - Bidding Behavior - Cluster B12

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
59.1	0.5	S	M	1	S	→	VL	0

Table C.14: Reactive Transition - Bidding Behavior - Cluster B13

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
93.3	0.5	M	H	0	N	→	H	1
72.2	13.8	S	H	0	N	→	H	1
70.0	1.1	F	H	0	N	→	H	1

Table C.15: Reactive Transition - Bidding Behavior - Cluster B14

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.3	S	L	0	Z	→	L	0
85.7	0.9	S	L	0	N	→	L	0
75.0	0.5	F	M	0	Z	→	L	0
71.4	0.8	F	H	0	S	→	H	0
71.4	0.8	S	H	0	S	→	L	0
67.7	3.6	F	L	0	S	→	L	0
66.7	0.3	F	L	0	Z	→	L	0
63.6	2.2	S	M	0	S	→	L	0
63.6	1.1	M	L	0	S	→	L	0
63.1	15.8	F	L	0	N	→	L	0
62.8	5.0	S	M	0	N	→	L	0

Table C.16: Reactive Transition - Bidding Behavior - Cluster B15

C.2 Auction Results

In this section we present the main reactive transitions according to the auction negotiation pattern. We have choosen some groups of auctions to show the results and also present only the main relevant reactive transitions (considering their Confidence) for each group. For each presented result, the labels in the tables mean the same that we have already presented in the last section.

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	25.6	S	H	0	N	→	H	1
100.0	1.0	M	H	0	N	→	H	1
100.0	0.4	M	M	0	N	→	H	1
100.0	0.3	I	M	0	N	→	H	1
98.6	38.8	S	M	0	N	→	H	1
96.7	2.0	F	H	0	N	→	H	1
95.4	5.9	F	M	0	N	→	H	1
80.0	0.3	M	L	1	N	→	VL	0

Table C.17: Reactive Transition - Auction Negotiation - Cluster A0

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
95.7	0.8	I	H	0	S	→	H	0
81.0	0.6	S	M	1	Z	→	VL	0
71.4	0.7	S	M	1	S	→	VL	0
68.4	0.5	S	M	1	N	→	VL	0

Table C.18: Reactive Transition - Auction Negotiation - Cluster A1

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
70.2	0.4	M	L	1	N	→	VL	0
60.4	0.4	S	M	1	Z	→	VL	0

Table C.19: Reactive Transition - Auction Negotiation - Cluster A2

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
95.7	0.6	I	M	0	N	→	H	1
84.6	0.3	M	H	0	N	→	H	1
73.3	0.3	F	L	1	N	→	VL	0
69.4	1.5	M	M	0	N	→	H	1
66.1	1.1	S	M	1	N	→	VL	0
65.9	0.7	S	M	1	Z	→	VL	0

Table C.20: Reactive Transition - Auction Negotiation - Cluster A3

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	1.5	S	M	1	S	→	VL	0
100.0	0.5	S	M	1	N	→	VL	0
100.0	0.3	S	L	1	Z	→	VL	0
100.0	0.3	F	L	1	N	→	VL	0
100.0	0.3	M	M	0	N	→	H	1
100.0	0.3	S	H	1	Z	→	L	0
100.0	0.3	F	H	0	S	→	L	0
100.0	0.3	M	H	1	S	→	H	0
100.0	0.3	F	M	0	Z	→	M	1

Table C.21: Reactive Transition - Auction Negotiation - Cluster A4

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
There is not any rule with confindence greater than 50%								

Table C.22: Reactive Transition - Auction Negotiation - Cluster A5

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
58.6	0.4	M	L	1	N	→	VL	0
58.3	0.3	I	L	1	N	→	VL	0
51.1	0.3	S	M	1	S	→	VL	0
50.9	0.5	F	M	1	N	→	L	0

Table C.23: Reactive Transition - Auction Negotiation - Cluster A6

C.3 Bidder in Auction Results

This section presents the complete set of results in the context of bidder in auction. As previously showed, in our case study we have identified 16 different groups of bidders and 7 groups of auctions. From these groups, we can have 112 different scenarios to analyze, however 82 has been occured in the dataset.

The number of reactive transitions identified in this 82 bidder in auction scenarios are very high, suggesting the accuracy of the approach. There are 1251 reactive transition rules with a confidence higher than 50%. Considering a 100% confidence, we have 359 rules.

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	10.5	I	H	1	N	→	H	0
100.0	5.3	M	M	1	N	→	M	0
100.0	21.1	I	M	0	N	→	H	1
100.0	10.5	I	H	0	N	→	H	1
100.0	5.3	M	M	0	N	→	H	1
100.0	5.3	M	H	0	N	→	H	1

Table C.24: Reactive Transition - Bidder in Auction - Cluster A0B0

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	3.8	S	M	0	Z	→	L	0
100.0	3.8	F	L	1	N	→	VL	0
100.0	1.9	S	L	1	N	→	VL	0
100.0	13.2	F	M	0	N	→	H	1
100.0	5.7	S	M	0	N	→	H	1
100.0	5.7	F	H	0	N	→	H	1
100.0	1.9	S	L	0	N	→	H	1
100.0	1.9	M	L	0	N	→	H	1
100.0	3,8	S	M	0	S	→	VL	1

Table C.25: Reactive Transition - Bidder in Auction - Cluster A0B2

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	1.5	S	L	0	S	→	L	0
100.0	0.7	S	M	1	Z	→	VL	0
100.0	0.7	M	L	0	S	→	L	1
100.0	0.7	M	L	0	Z	→	VL	1
100.0	0.7	F	M	1	N	→	L	0

Table C.26: Reactive Transition - Bidder in Auction - Cluster A0B4

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	45.0	S	M	0	N	→	H	1
100.0	30.0	S	H	0	N	→	H	1
100.0	5.7	F	M	0	N	→	H	1
100.0	2.1	F	H	0	N	→	H	1
100.0	1.1	M	H	0	N	→	H	1
100.0	0.4	M	M	0	N	→	H	1

Table C.27: Reactive Transition - Bidder in Auction - Cluster A0B14

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	10.3	I	H	0	S	→	H	0
100.0	1.4	M	M	0	S	→	L	0
100.0	0.9	F	M	1	S	→	VL	0
100.0	0.9	I	H	1	S	→	H	0
100.0	0.9	I	H	1	N	→	H	0
100.0	0.9	F	L	0	S	→	L	0
100.0	0.5	I	M	0	Z	→	H	0
100.0	0.5	I	L	1	S	→	VL	0
100.0	0.5	S	M	0	S	→	L	0
100.0	0.5	M	L	1	S	→	L	0
100.0	0.5	I	L	0	Z	→	VL	1
100.0	0.5	F	M	0	S	→	L	1

Table C.28: Reactive Transition - Bidder in Auction - Cluster A1B0

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	6.5	F	H	0	S	→	L	0
100.0	6.5	S	M	0	S	→	L	1
100.0	3.2	F	M	0	Z	→	M	0
100.0	3.2	M	M	0	N	→	VL	1
100.0	3.2	F	L	0	S	→	VL	1
100.0	3.2	S	M	1	N	→	L	0
100.0	3.2	S	H	1	Z	→	L	0
100.0	3.2	F	M	1	S	→	L	0
100.0	3.2	S	M	1	S	→	VL	0
100.0	3.2	S	H	0	S	→	M	1
100.0	3.2	M	M	0	S	→	H	0

Table C.29: Reactive Transition - Bidder in Auction - Cluster A1B1

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	1.7	S	M	1	N	→	VL	0
100.0	0.2	S	L	1	Z	→	VL	0
100.0	0.2	F	M	1	N	→	VL	0
100.0	0.2	F	L	0	Z	→	VL	1
100.0	0.2	S	L	1	S	→	L	0
100.0	0.2	F	M	1	S	→	L	0

Table C.30: Reactive Transition - Bidder in Auction - Cluster A1B2

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	3.2	F	L	0	N	→	VL	1
100.0	1.6	F	L	0	Z	→	L	0
100.0	1.6	S	M	0	N	→	VL	1
100.0	1.6	S	L	0	Z	→	VL	1
100.0	1.6	S	L	0	N	→	L	1
100.0	1.6	M	M	0	N	→	L	1
100.0	1.6	F	H	0	N	→	H	0
100.0	1.6	S	M	1	N	→	VL	0
100.0	1.6	S	L	1	N	→	VL	0
100.0	1.6	F	L	1	N	→	VL	0
100.0	1.6	I	H	0	S	→	H	1

Table C.31: Reactive Transition - Bidder in Auction - Cluster A1B3

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
87.5	0.9	S	M	1	Z	→	VL	0

Table C.32: Reactive Transition - Bidder in Auction - Cluster A1B4

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	11.8	F	M	0	N	→	L	0
100.0	2.9	M	L	0	S	→	H	0
100.0	2.9	M	M	0	S	→	M	0
100.0	2.9	I	L	0	S	→	L	0
100.0	2.9	I	L	0	N	→	L	0

Table C.33: Reactive Transition - Bidder in Auction - Cluster A1B6

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	6.3	F	L	0	S	→	L	0
100.0	3.1	M	L	0	S	→	M	0
100.0	3.1	M	H	0	S	→	H	0
100.0	3.1	F	H	0	S	→	H	0
100.0	3.1	S	H	0	S	→	L	0
100.0	3.1	M	L	0	N	→	L	0
100.0	3.1	I	L	0	S	→	L	0
100.0	3.1	F	L	0	N	→	L	0

Table C.34: Reactive Transition - Bidder in Auction - Cluster A1B7

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	5.4	M	M	0	S	→	L	0
100.0	2.7	S	M	0	N	→	L	1
100.0	0.9	I	L	0	S	→	M	0
100.0	0.9	F	L	1	N	→	M	0
100.0	0.9	S	H	0	Z	→	M	1
100.0	0.9	M	L	1	N	→	L	0
100.0	0.9	I	L	0	N	→	L	0
100.0	0.9	F	L	0	Z	→	L	0

Table C.35: Reactive Transition - Bidder in Auction - Cluster A1B9

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	3.5	S	H	0	N	→	VL	0
100.0	3.5	I	L	0	S	→	VL	0
100.0	3.5	M	M	0	S	→	H	0
100.0	3.5	S	M	0	N	→	L	0
100.0	3.5	I	L	0	N	→	L	0

Table C.36: Reactive Transition - Bidder in Auction - Cluster A1B10

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	3.7	F	L	0	N	→	L	0
100.0	1.9	M	L	0	N	→	L	0
100.0	1.9	F	M	1	S	→	L	0
100.0	1.9	F	H	0	S	→	L	1
100.0	1.9	S	H	0	N	→	H	0

Table C.37: Reactive Transition - Bidder in Auction - Cluster A1B12

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	1.1	S	M	1	Z	→	L	0
100.0	1.1	S	H	1	S	→	L	0
100.0	3.2	M	L	0	N	→	VL	1
100.0	1.1	S	L	0	N	→	VL	1
100.0	1.1	S	M	1	S	→	VL	0
100.0	1.1	S	L	1	N	→	VL	0
100.0	2.1	F	H	0	S	→	L	1
100.0	1.1	S	H	0	Z	→	L	1
100.0	1.1	I	L	0	S	→	L	1
100.0	1.1	F	M	0	Z	→	L	1
100.0	1.1	F	L	0	Z	→	L	1

Table C.38: Reactive Transition - Bidder in Auction - Cluster A1B13

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	1.0	F	H	0	S	→	VL	1
100.0	1.0	S	L	0	N	→	L	1
100.0	1.0	F	M	0	S	→	L	1
100.0	1.0	M	H	0	N	→	H	1

Table C.39: Reactive Transition - Bidder in Auction - Cluster A1B14

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	7.7	M	L	0	S	→	VL	0
100.0	15.4	F	M	0	N	→	L	0
100.0	7.7	S	M	0	N	→	L	0
100.0	7.7	S	H	0	Z	→	VL	1

Table C.40: Reactive Transition - Bidder in Auction - Cluster A1B15

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.3	S	M	0	S	→	L	0

Table C.41: Reactive Transition - Bidder in Auction - Cluster A2B0

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.4	S	L	1	N	→	VL	0

Table C.42: Reactive Transition - Bidder in Auction - Cluster A2B1

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
83.3	0.3	S	L	1	S	→	VL	0

Table C.43: Reactive Transition - Bidder in Auction - Cluster A2B2

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.4	F	L	1	N	→	VL	0
100.0	0.3	F	H	1	S	→	L	0
100.0	0.3	S	L	1	Z	→	VL	0
100.0	0.3	S	H	1	S	→	VL	0
100.0	0.3	M	L	1	N	→	VL	0
100.0	0.3	F	M	1	N	→	VL	0

Table C.44: Reactive Transition - Bidder in Auction - Cluster A2B3

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
78.6	4.2	M	L	1	N	→	VL	0
77.3	2.4	F	M	1	S	→	L	0

Table C.45: Reactive Transition - Bidder in Auction - Cluster A2B4

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.4	I	L	1	N	→	VL	0

Table C.46: Reactive Transition - Bidder in Auction - Cluster A2B5

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**

Table C.47: Reactive Transition - Bidder in Auction - Cluster A2B6

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.3	S	L	0	Z	→	H	0
100.0	0.6	S	H	0	Z	→	M	0
100.0	0.3	F	M	1	S	→	M	0
100.0	0.3	F	L	0	Z	→	L	0

Table C.48: Reactive Transition - Bidder in Auction - Cluster A2B7

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.5	M	L	1	S	→	VL	0
100.0	0.2	M	M	1	N	→	VL	0
100.0	0.2	F	M	1	S	→	VL	0
100.0	0.2	S	L	0	S	→	VL	1
100.0	0.2	F	M	1	Z	→	L	0
100.0	0.2	F	H	1	S	→	L	0
100.0	0.2	F	H	1	N	→	L	0
100.0	0.2	F	L	1	N	→	M	0

Table C.49: Reactive Transition - Bidder in Auction - Cluster A2B8

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.5	S	M	1	S	→	VL	0
100.0	0.5	S	M	1	N	→	L	0

Table C.50: Reactive Transition - Bidder in Auction - Cluster A2B9

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.6	M	H	0	N	→	H	0
100.0	0.2	F	H	0	Z	→	H	0
100.0	0.2	I	M	0	N	→	M	0
100.0	0.2	F	M	0	Z	→	VL	1
100.0	0.2	S	M	1	Z	→	VL	0
100.0	0.2	S	H	1	S	→	VL	0
100.0	0.2	M	L	1	N	→	VL	0
100.0	0.2	F	M	1	N	→	VL	0
100.0	0.2	F	M	1	S	→	L	0

Table C.51: Reactive Transition - Bidder in Auction - Cluster A2B10

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.4	S	L	0	Z	→	VL	1
100.0	5.3	M	M	0	N	→	H	1
100.0	0.7	M	H	0	N	→	H	1
100.0	0.4	M	L	1	N	→	VL	0
100.0	0.3	S	L	1	Z	→	VL	0

Table C.52: Reactive Transition - Bidder in Auction - Cluster A2B11

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
83.3	0.3	F	M	1	N	→	L	0
81.8	0.5	S	M	1	Z	→	VL	0

Table C.53: Reactive Transition - Bidder in Auction - Cluster A2B12

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
70.0	0.3	S	L	0	Z	→	VL	1

Table C.54: Reactive Transition - Bidder in Auction - Cluster A2B13

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	1.5	F	M	0	N	→	L	1
100.0	0.3	F	H	0	N	→	M	1
100.0	0.3	F	M	0	Z	→	L	1

Table C.55: Reactive Transition - Bidder in Auction - Cluster A2B14

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.8	S	L	0	N	→	L	0
100.0	0.4	M	H	0	N	→	H	0
100.0	0.4	I	H	0	N	→	H	0
100.0	0.4	S	M	1	Z	→	L	0
100.0	0.4	S	H	0	Z	→	L	0

Table C.56: Reactive Transition - Bidder in Auction - Cluster A2B15

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	4.0	M	M	1	N	→	VL	0
100.0	4.0	M	L	1	N	→	L	0
100.0	4.0	M	M	0	N	→	H	1
100.0	2.0	F	H	1	N	→	M	0
100.0	2.0	M	H	0	N	→	H	1
100.0	2.0	S	L	0	N	→	L	1
100.0	2.0	M	H	1	N	→	H	0

Table C.57: Reactive Transition - Bidder in Auction - Cluster A3B0

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.3	F	H	1	N	→	L	0

Table C.58: Reactive Transition - Bidder in Auction - Cluster A3B2

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	19.1	F	L	0	N	→	VL	1
100.0	4.8	M	L	0	N	→	VL	1
100.0	4.8	F	H	0	N	→	H	1

Table C.59: Reactive Transition - Bidder in Auction - Cluster A3B3

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.3	M	L	0	N	→	M	1
100.0	0.2	F	M	1	S	→	L	0

Table C.60: Reactive Transition - Bidder in Auction - Cluster A3B4

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	21.8	I	M	0	N	→	H	1
100.0	10.3	M	M	0	N	→	H	1
100.0	6.9	I	H	0	N	→	H	1
100.0	4.6	M	H	0	N	→	H	1
100.0	1.2	I	L	1	N	→	H	0

Table C.61: Reactive Transition - Bidder in Auction - Cluster A3B5

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	1.5	M	H	0	N	→	H	0
100.0	1.5	S	L	0	N	→	VL	0

Table C.62: Reactive Transition - Bidder in Auction - Cluster A3B6

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	14.3	F	L	0	N	→	VL	0
100.0	14.3	F	H	0	N	→	H	0

Table C.63: Reactive Transition - Bidder in Auction - Cluster A3B7

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	12.5	F	M	0	S	→	L	0
100.0	12.5	F	L	0	S	→	L	0
100.0	8.3	M	M	0	N	→	H	1
100.0	8.3	I	L	0	N	→	H	1
100.0	4.2	S	M	1	Z	→	VL	0
100.0	4.2	M	L	1	N	→	VL	0
100.0	4.2	M	L	0	N	→	VL	1
100.0	4.2	I	M	0	N	→	H	1
100.0	4.2	I	H	0	N	→	H	1
100.0	4.2	F	L	0	N	→	L	0
100.0	4.2	F	H	1	N	→	L	0
100.0	4.2	F	M	0	N	→	L	1

Table C.64: Reactive Transition - Bidder in Auction - Cluster A3B8

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	3.2	M	M	1	N	→	M	0
100.0	3.2	F	L	1	N	→	M	0
100.0	3.2	F	H	1	N	→	H	0

Table C.65: Reactive Transition - Bidder in Auction - Cluster A3B9

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	20.0	F	L	0	N	→	L	0
100.0	20.0	S	M	0	N	→	L	0
100.0	10.0	S	H	0	N	→	H	0
100.0	10.0	F	M	0	N	→	H	0

Table C.66: Reactive Transition - Bidder in Auction - Cluster A3B10

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	4.6	M	M	0	N	→	H	1
100.0	0.6	M	H	0	N	→	H	1
100.0	0.4	S	M	1	S	→	VL	0
100.0	0.2	F	M	1	N	→	VL	0

Table C.67: Reactive Transition - Bidder in Auction - Cluster A3B11

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	1.0	S	L	0	Z	→	VL	1
100.0	0.3	M	M	0	N	→	VL	1
100.0	0.3	M	H	0	N	→	VL	1

Table C.68: Reactive Transition - Bidder in Auction - Cluster A3B14

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	18.2	F	H	0	N	→	H	0
100.0	9.1	S	M	0	N	→	VL	0
100.0	9.1	M	M	0	N	→	L	0

Table C.69: Reactive Transition - Bidder in Auction - Cluster A3B15

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
50.0	50.0	M	L	0	N	→	VL	1
50.0	50.0	M	L	0	N	→	H	1

Table C.70: Reactive Transition - Bidder in Auction - Cluster A4B0

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	5.6	F	M	0	S	→	L	0
100.0	1.6	S	H	0	N	→	H	1
100.0	1.6	F	H	0	N	→	H	1
100.0	0.8	S	L	0	S	→	L	0
100.0	0.8	S	M	0	Z	→	VL	1
100.0	0.8	S	L	0	Z	→	VL	1
100.0	0.8	M	L	0	N	→	VL	1
100.0	0.8	F	M	0	Z	→	M	1
100.0	0.8	S	M	1	N	→	VL	0
100.0	0.8	F	L	1	N	→	VL	0

Table C.71: Reactive Transition - Bidder in Auction - Cluster A4B2

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	2.8	S	M	1	S	→	VL	0
100.0	0.5	S	M	1	N	→	VL	0
100.0	0.5	S	L	1	Z	→	VL	0
100.0	0.5	M	H	1	S	→	H	0
100.0	0.5	S	H	1	Z	→	L	0
100.0	0.5	F	H	0	S	→	L	0
100.0	0.5	M	M	0	N	→	H	1

Table C.72: Reactive Transition - Bidder in Auction - Cluster A4B4

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	4.0	S	H	0	S	→	M	1
100.0	2.0	S	L	0	Z	→	VL	1
100.0	2.0	F	H	0	N	→	VL	1
100.0	2.0	F	L	0	Z	→	VL	0

Table C.73: Reactive Transition - Bidder in Auction - Cluster A4B14

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
75.0	0.5	M	H	1	N	→	H	0
72.7	0.7	M	L	0	Z	→	L	0

Table C.74: Reactive Transition - Bidder in Auction - Cluster A5B0

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.5	S	H	1	Z	→	VL	0
100.0	0.3	S	M	1	Z	→	VL	0
100.0	0.3	I	L	1	N	→	VL	0
100.0	0.3	F	M	0	S	→	VL	0
100.0	0.3	F	L	0	Z	→	VL	0
100.0	0.3	F	H	0	Z	→	H	1
100.0	0.3	S	M	1	S	→	L	0
100.0	0.3	M	M	0	Z	→	L	0
100.0	0.3	F	M	1	S	→	H	0
100.0	0.3	I	M	1	N	→	M	0

Table C.75: Reactive Transition - Bidder in Auction - Cluster A5B1

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
66.7	0.4	S	M	1	N	→	VL	0
60.0	0.5	S	M	1	Z	→	VL	0

Table C.76: Reactive Transition - Bidder in Auction - Cluster A5B2

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
100.0	0.2	S	H	1	Z	→	L	0

Table C.77: Reactive Transition - Bidder in Auction - Cluster A5B3

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
77.0	0.3	M	H	0	S	→	L	0

Table C.78: Reactive Transition - Bidder in Auction - Cluster A5B4

Conf.	Sup.	Time	WinP	WinB	Comp		IBP	ChW
80.0	0.5	M	M	1	N	→	VL	0

Table C.79: Reactive Transition - Bidder in Auction - Cluster A5B5

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
75.0	0.5	S	L	0	N	→	L	0
69.3	11.6	F	M	0	N	→	L	0

Table C.80: Reactive Transition - Bidder in Auction - Cluster A5B6

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
76.5	1.5	M	H	0	N	→	H	0
75.0	0.7	M	L	0	S	→	L	0

Table C.81: Reactive Transition - Bidder in Auction - Cluster A5B7

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.4	S	L	0	N	→	H	0

Table C.82: Reactive Transition - Bidder in Auction - Cluster A5B8

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.2	F	L	1	S	→	VL	0

Table C.83: Reactive Transition - Bidder in Auction - Cluster A5B9

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.8	I	H	0	N	→	H	0
100.0	0.8	S	L	0	N	→	L	0
100.0	0.4	S	H	1	Z	→	L	0
100.0	0.4	M	M	0	S	→	L	0

Table C.84: Reactive Transition - Bidder in Auction - Cluster A5B10

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	1.8	M	H	0	N	→	H	1
100.0	0.4	S	L	0	S	→	VL	0
100.0	0.2	F	L	0	Z	→	VL	1
100.0	0.2	S	H	1	S	→	L	0
100.0	0.2	F	L	1	Z	→	L	0
100.0	0.2	F	L	1	S	→	L	0
100.0	0.2	S	M	1	N	→	H	0
100.0	0.2	M	H	1	N	→	H	0
100.0	0.2	F	H	1	S	→	M	0
100.0	0.2	S	M	1	S	→	VL	0
100.0	0.2	S	L	1	N	→	VL	0
100.0	0.2	S	L	0	N	→	VL	0
100.0	0.2	M	L	1	S	→	VL	0

Table C.85: Reactive Transition - Bidder in Auction - Cluster A5B11

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.6	I	L	0	Z	→	VL	1
100.0	0.6	I	L	0	N	→	L	0
100.0	0.4	S	H	1	S	→	M	0
100.0	0.2	M	H	0	S	→	H	1
100.0	0.2	F	M	0	Z	→	L	1
100.0	0.2	F	L	0	Z	→	L	1
100.0	0.2	F	H	0	Z	→	L	1
100.0	0.2	I	M	0	S	→	VL	1
100.0	0.2	S	H	1	Z	→	L	0
100.0	0.2	M	H	0	N	→	M	0
100.0	0.2	S	L	1	S	→	VL	0
100.0	0.2	F	M	1	Z	→	VL	0
100.0	0.2	S	H	1	N	→	H	0
100.0	0.2	I	M	0	N	→	H	0

Table C.86: Reactive Transition - Bidder in Auction - Cluster A5B12

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.2	F	L	1	N	→	VL	0

Table C.87: Reactive Transition - Bidder in Auction - Cluster A5B13

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.2	M	M	0	N	→	M	1
100.0	0.2	S	L	0	Z	→	VL	1
100.0	0.2	M	M	0	S	→	L	1
100.0	0.2	F	L	0	S	→	L	1
100.0	0.2	F	H	0	S	→	L	1

Table C.88: Reactive Transition - Bidder in Auction - Cluster A5B14

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	2.6	M	L	0	S	→	L	0
100.0	1.3	M	M	0	N	→	H	0
100.0	0.7	S	M	1	Z	→	VL	0
100.0	0.7	F	L	1	Z	→	VL	0
100.0	0.7	F	H	0	S	→	H	0
100.0	0.7	S	M	0	Z	→	L	0
100.0	0.7	S	L	0	Z	→	L	0
100.0	0.7	M	H	0	N	→	L	0

Table C.89: Reactive Transition - Bidder in Auction - Cluster A5B15

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
76.3	1.6	M	L	1	N	→	VL	0
75.0	0.3	F	M	0	Z	→	L	0

Table C.90: Reactive Transition - Bidder in Auction - Cluster A6B0

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.4	S	M	1	S	→	VL	0

Table C.91: Reactive Transition - Bidder in Auction - Cluster A6B1

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
56.7	0.5	S	M	1	Z	→	VL	0
52.2	0.4	S	M	1	N	→	VL	0
52.1	1.1	F	M	0	S	→	L	0
51.9	10.2	S	M	0	S	→	L	0

Table C.92: Reactive Transition - Bidder in Auction - Cluster A6B2

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
57.5	0.4	F	M	1	N	→	L	0
56.0	0.3	S	M	1	S	→	VL	0
50.0	0.3	F	H	1	N	→	VL	0

Table C.93: Reactive Transition - Bidder in Auction - Cluster A6B3

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
65.6	0.5	F	L	1	N	→	VL	0
62.8	0.6	S	M	1	Z	→	VL	0
61.9	0.3	F	L	1	S	→	VL	0

Table C.94: Reactive Transition - Bidder in Auction - Cluster A6B4

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
81.8	1.6	I	H	0	N	→	H	1
80.0	1.7	M	M	0	N	→	H	1
80.0	0.4	M	H	0	N	→	H	1
80.0	0.7	M	L	1	N	→	VL	0

Table C.95: Reactive Transition - Bidder in Auction - Cluster A6B5

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
80.0	0.8	S	M	0	S	→	L	0
80.0	0.3	S	M	0	Z	→	L	0

Table C.96: Reactive Transition - Bidder in Auction - Cluster A6B6

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
75.0	0.4	S	H	0	Z	→	L	0

Table C.97: Reactive Transition - Bidder in Auction - Cluster A6B7

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.6	F	H	0	S	→	L	0
100.0	0.3	F	L	0	Z	→	L	0
100.0	0.2	F	L	1	N	→	L	0

Table C.98: Reactive Transition - Bidder in Auction - Cluster A6B8

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
60.9	1.0	S	M	0	S	→	L	0

Table C.99: Reactive Transition - Bidder in Auction - Cluster A6B9

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.7	M	M	0	S	→	L	0
100.0	0.4	F	H	0	Z	→	M	0
100.0	0.4	S	M	0	Z	→	L	0
100.0	0.4	M	M	1	N	→	L	0
100.0	0.4	M	M	0	Z	→	L	0
100.0	0.4	M	H	0	S	→	L	0
100.0	0.4	F	M	0	Z	→	L	0
100.0	0.4	F	L	0	Z	→	L	0

Table C.100: Reactive Transition - Bidder in Auction - Cluster A6B10

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	0.3	F	L	1	N	→	VL	0
90.0	6.8	M	M	0	N	→	H	1

Table C.101: Reactive Transition - Bidder in Auction - Cluster A6B11

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
77.8	0.7	I	M	0	S	→	L	0
75.0	0.3	S	L	1	S	→	VL	0
75.0	0.3	S	H	1	Z	→	VL	0
71.4	0.5	S	M	1	Z	→	VL	0

Table C.102: Reactive Transition - Bidder in Auction - Cluster A6B12

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
76.9	0.6	S	H	1	Z	→	L	0
70.0	0.4	M	M	0	Z	→	L	0

Table C.103: Reactive Transition - Bidder in Auction - Cluster A6B13

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
64.7	1.8	S	L	0	S	→	VL	1
60.0	2.9	F	M	0	N	→	L	1

Table C.104: Reactive Transition - Bidder in Auction - Cluster A6B14

Conf.	**Sup.**	**Time**	**WinP**	**WinB**	**Comp**		**IBP**	**ChW**
100.0	2.8	S	M	0	S	→	L	0
100.0	1.4	S	L	0	N	→	L	0
100.0	1.4	F	L	0	S	→	L	0
100.0	0.5	M	M	0	S	→	M	0
100.0	0.5	I	M	0	N	→	M	0
100.0	0.5	F	H	0	Z	→	M	0
100.0	0.5	S	M	0	Z	→	VL	1
100.0	0.5	S	L	0	Z	→	L	0
100.0	0.5	S	H	0	Z	→	L	0
100.0	0.5	I	L	0	S	→	L	0
100.0	0.5	F	L	0	Z	→	L	0
100.0	0.5	F	H	1	N	→	VL	0

Table C.105: Reactive Transition - Bidder in Auction - Cluster A6B15

Bibliography

[1] Tpc - transaction processing council. tpc benchmark w. `http://www.tpc.org/tpcw/`.

[2] T. F. Abdelzaher and N. Bhatti. Web content adaptation to improve server overload behavior. In *WWW '99: Proceeding of the eighth international conference on World Wide Web*, pages 1563–1577, New York, NY, USA, 1999. Elsevier North-Holland, Inc.

[3] T. F. Abdelzaher and N. Bhatti. Web content adaptation to improve server overload behavior. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1563–1577, 1999.

[4] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington D.C., May 1993.

[5] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, pages 487–499, Santiago, Chile, September 1994.

[6] V. Akula and D. Menascé. Two-level workload characterization of online auctions. *Electronic Commerce Research and Applications Journal*, 6(2):192–208, 2007.

[7] J. M. Almeida, J. Krueger, D. L. Eager, and M. K. Vernon. Analysis of educational media server workloads. In *Proceedings of the 11th international workshop on Network and operating systems support for digital audio and video*, pages 21–30. ACM Press, 2001.

[8] J. M. Almeida, J. Krueger, and M. K. Vernon. Characterization of user access to streaming media files. In *Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 340–341. ACM Press, 2001.

[9] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira. Characterizing reference locality in the WWW. In *Proceedings of the IEEE Conference on Parallel and Distributed Information Systems (PDIS)*, Miami Beach, FL, 1996.

[10] S. Ambler. *The Object Primer: Agile Model Driven Development with UML 2*. Cambridge University Press, 2004.

[11] S. Anderson, D. Friedman, G. Milam, and N. Singh. Seller strategies on ebay. Industrial Organization 0412004, Economics Working Paper Archive EconWPA, Dec. 2004.

[12] D. Ariely and I. Simonson. Buying, bidding, playing, or competing? value assessment and decision dynamics in online auctions. *Journal of Consumer Psychology*, 13(1):113–123, 2003.

[13] M. F. Arlitt and C. L. Williamson. Web server workload characterization: The search for invariants. In *Measurement and Modeling of Computer Systems*, pages 126–137, 1996.

[14] M. F. Arlitt and C. L. Williamson. Internet web servers: workload characterization and performance implications. *IEEE/ACM Trans. Netw.*, 5(5):631–645, 1997.

[15] Y. Auctions. Yahoo!, Inc., 2003. http://auctions.yahoo.com/.

[16] A. Avritzer, J. Kondek, D. Liu, and E. J. Weyuker. Software performance testing based on workload characterization. In *WOSP '02: Proceedings of the third international workshop on Software and performance*, pages 17–24. ACM Press, 2002.

[17] S. Ba and P. A. Pavlou. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *MIS Quarterly*, 26(3), 2002.

[18] P. Bajari and A. Hortacsu. The winner's curse, reserve prices, and endogenous entry: Empirical insights from ebay auctions. *RAND Journal of Economics*, 34(2):329–55, 2003.

[19] A. Balachandran, G. Voelker, P. Bahl, and P. Rangan. Characterizing user behavior and network performance in a public wireless lan. In *Proceedings of ACM SIGEMTRICS'02*, 2002.

[20] R. Bapna. When snipers become predators: can mechanism design save online auctions? *Commun. ACM*, 46(12):152–158, 2003.

[21] R. Bapna, P. B. Goes, and A. Gupta. Replicating online yankee auctions to analyze auctioneers' and bidders' strategies. *Information Systems Research*, 14(3):244–268, 2003.

[22] R. Bapna, P. B. Goes, A. Gupta, and Y. Jin. User heterogeneity and its impact on electronic auction market design: An empirical exploration research essays. *MIS Quarterly*, 28(1), 2004.

[23] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 151–160. ACM Press, 1998.

[24] N. Bhatti, A. Bouch, and A. Kuchinsky. Integrating user-perceived quality into web server design. In *Proceedings of the 9th International World Wide Web Conference*, 2000.

[25] N. Bhatti and R. Friedrich. Web server support for tiered services. *IEEE Network*, 13(5):64–71, 1999.

[26] H.-H. Bock. Data mining tasks and methods - classification: the goal of classification. pages 254–258, 2002.

[27] F. Bry and P.-L. Patranjan. Reactivity on the web: paradigms and applications of the language xchange. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1645–1649, New York, NY, USA, 2005. ACM Press.

[28] M. Busari and C. Williamson. Prowgen: a synthetic workload generation tool for simulation evaluation of web proxy caches. *Comput. Networks*, 38(6):779–794, 2002.

[29] R. Carter and L. Cherkasova. Detecting timed-out client requests for avoiding livelock and improving web server performance. In *Proc. of 5th IEEE Symposium on Computers and Communications*, pages pp 2–7, Antibes, FRANCE, 2000. IEEE Computer Society.

[30] S. Ceri, F. Daniel, V. Demaldé, and F. M. Facca. An Approach to User-Behavior-Aware Web Applications. In D. Lowe and M. Gaedke, editors, *Web Engineering, 5th International Conference, ICWE 2005, Sydney, Australia, July 27-29, 2005, Proceedings*, volume 3579 of *Lecture Notes in Computer Science*, pages 417–428. Springer, 2005.

[31] D. Chakravarti, E. A. Greenleaf, A. Sinha, A. Cheema, J. C. Cox, D. Friedman, T. H. Ho, R. M. Isaac, A. A. M. A. Rapoport, M. H. Rothkopf, J. Srivastava, and R. Zwick. Auctions: Research opportunities in marketing. *Marketing Letters*, 13(3):279–293, 2002.

[32] P. Chatterjee, D. Hoffman, and T. Novak. Modeling the clickstream: Implications for web-based advertising efforts, 1998.

[33] L. Cherkasova, Y. Fu, W. Tang, and A. Vahdat. Measuring and characterizing end-to-end internet service performance. *ACM Trans. Inter. Tech.*, 3(4):347–391, 2003.

[34] L. Cherkasova and P. Phaal. Predictive admission control strategy for overloaded commercial web server. In *MASCOTS '00: Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, page 500, Washington, DC, USA, 2000. IEEE Computer Society.

[35] L. Cherkasova and P. Phaal. Session-based admission control: A mechanism for peak load management of commercial web sites. *IEEE Trans. Comput.*, 51(6):669–685, 2002.

[36] S. Club. SamsClub, Inc., 2006. http://www.samsclub.com/.

[37] C. Costa, I. Cunha, A. Borges, C. Ramos, M. Rocha, J. Almeida, and B. R. Neto. Analyzing client interactivity in streaming media. In *Proceedings of the 13th World Wide Web Conference*, 2004.

[38] C. Costa, I. Cunha, A. Borges, C. Ramos, M. Rocha, J. Almeida, and B. Ribeiro-Neto. Analyzing client interactivity in streaming media. In *Proceedings of the 13th World Wide Web Conference*, 2004.

[39] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. In *Proceedings of the 1996 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 160–169. ACM Press, 1996.

[40] T. Davies. Universal law series - the law of cause & effect, 2005. http://www.articlestree.com/self-improvement/universal-law-series-the-law-of-cause-tx180562.html.

[41] U. Dholakia and K. Soltysinski. Coveted or overlooked? the psychology of bidding for comparable listings in digital auctions. *Marketing Letters*, 12(3):225–237, August 2001.

[42] M. do Planejamento. Brasil, 2006. http://www.comprasnet.gov.br/.

[43] R. F. Easley and R. Tenorio. Jump bidding strategies in internet auctions. *Manage. Sci.*, 50(10):1407–1419, 2004.

[44] EBay. eBay, Inc., 2007. http://www.ebay.com/.

[45] S. Elnikety, E. Nahum, J. Tracey, and W. Zwaenepoel. A method for transparent admission control and request scheduling in e-commerce web sites. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 276–286. ACM Press, 2004.

[46] F. M. Facca, S. Ceri, J. Armani, and V. Demaldé. Building reactive web applications. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1058–1059, New York, NY, USA, 2005. ACM Press.

[47] D. Feitelson. Workload modeling for performance evaluation, 2002.

[48] P. A. Fishwick. Simpack: Getting started with simulation programming in c and c++. In *Winter Simulation Conference*, pages 154–162, 1992.

[49] G. Franco and W. Meira, Jr. Usar: a user behavior characterization model. In *Proceedings of the PhD and MSc Workshop of the 2nd Latin American Web Congress and the 10th Brazilian Symposium on Multimedia and the Web (LA-WebMedia 2004)*, Ribeirão Preto, SP, Brazil, 2004. IEEE Computer Society.

[50] D. F. García and J. García. Tpc-w e-commerce benchmark evaluation. *Computer*, 36(2):42–48, 2003.

[51] S. Garner. Weka: The waikato environment for knowledge analysis. In *Proceedings of the New Zealand Computer Science Research Students Conference*, pages 57–64, 1995.

[52] J. H. Gilkeson and K. Reynolds. Determinants of internet auction success and closing price: An exploratory study. *Psychology & Marketing*, 20(6):537–566, 2003.

[53] B. Grosof. Representing e-business rules for the semantic web: Situated courteous logic programs in ruleml. In *Workshop on Information Technologies and Systems (WITS-2001)*, New Orleans, LA, USA, 2001.

[54] B. N. Grosof. Representing e-commerce rules via situated courteous logic programs in ruleml. *Electronic Commerce Research and Applications*, 3(1):2–20, 2004.

[55] J. Hartigan. Clustering Algorithms. *John Wiley and Sons, Inc.*, 1975.

[56] T. Henderson. Latency and user behaviour on a multiplayer game server. In *Proceedings of the Third International COST264 Workshop on Networked Group Communication*, pages 1–13. Springer-Verlag, 2001.

[57] T. Henderson. Latency and user behaviour on a multiplayer game server. In *Proceedings of the Third International COST264 Workshop on Networked Group Communication*, pages 1–13. Springer-Verlag, 2001.

[58] H. Hlavacs, E. Hotop, and G. Kotsis. Workload generation by modeling user behavior. In *Proceedings of OPNETWORKS 2000*, 2000.

[59] H. Hlavacs and G. Kotsis. Modeling user behavior: A layered approach. In *MASCOTS*, pages 218–225, 1999.

[60] A. Iyengar, E. MacNair, and T. Nguyen. An analysis of web server performance. In *In Proceedings of GLOBECOM '97*, Phoenix, Arizona, USA, 1997.

[61] W. Jank and G. Shmueli. Functional data analysis in electronic commerce research. *Statistical Science*, 21:155, 2006.

[62] W. Jank, G. Shmueli, and S. Wang. Dynamic, real-time forecasting of online auctions via functional models. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 580–585, New York, NY, USA, 2006. ACM Press.

[63] S. Jin and A. Bestavros. Gismo: a generator of internet streaming media objects and workloads. *SIGMETRICS Perform. Eval. Rev.*, 29(3):2–10, 2001.

[64] J. Kagel. Auctions: A survey of experimental research. In J. Kagel and E. Roth, editors, *The Handbook of Experimental Economics*, volume 1571. Princeton University Press, 1995.

[65] M. A. Kamins, X. Dèrze, and V. S. Folkes. A field study of the effects of minimum and reserve prices on internet auction. *Journal of Consumer Research*, 30(4):622–628, 2004.

[66] M. Kearns, Y. Mansour, and A. Y. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 282–293, Providence, Rhode Island, USA, 1997.

[67] B. Krishnamurthy, Y. Zhang, C. Wills, and K. Vishwanath. Design, implementation, and evaluation of a client characterization driven web server. In *WWW'03: Proceedings of the twelfth international conference on World Wide Web*, pages 138–147. ACM Press, 2003.

[68] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 611–617, New York, NY, USA, 2006. ACM Press.

[69] J. Laffont, H. Ossard, and Q. Vuong. Econometrics of first-price auctions. *Econometrica*, 63(4):953–80, July 1995.

[70] R. S. Lillie. Scientific Books: The Origin and Evolution of Life, on the Theory of the Action, Reaction and Interaction of Energy. *Science*, 48:472–474, Nov. 1918.

[71] D. Lucking-Reiley. Using field experiments to test equivalence between auction formats: Magic on the internet. *American Economic Review*, 89(5):1063–1080, 1999.

[72] R. E. Madsen, S. Sigurdsson, L. K. Hansen, and J. Larsen. Vocabulary pruning for improved context recognition. In *Proceedings of the International Joint Conference on Neural Networks*, pages 80–85. IEEE Press, aug 2004.

[73] P. P. Mcafee and J. Mcmillan. Auctions and bidding. *Journal of Economic Literature*, 25(2):699–738, 1987.

[74] D. Menasce and V. Akula. Towards workload characterization. In *Proceedings of the IEEE 6th Annual Workshop on Workload Characterization (WWC-6)*, Austin, Texas, USA, 2003. IEEE Computer Society.

[75] D. Menascé, V. Almeida, and L. Dowdy. *Performance by Design*. Prentice Hall, 2004.

[76] D. Menascé, V. Almeida, R. Riedi, F. Ribeiro, R. Fonseca, and W. M. Jr. A hierarchical and multiscale approach to analyze e-business workloads. *Perform. Eval.*, 54(1):33–57, 2003.

[77] D. Menascé, V. Almeida, R. Riedi, F. Ribeiro, R. Fonseca, and W. Meira, Jr. In search of invariants for e-business workloads. In *Proceedings of the 2nd ACM conference on Electronic commerce*, pages 56–65. ACM Press, 2000.

[78] D. A. Menascé. Testing e-commerce site scalability with tpc-w. In *Int. CMG Conference*, pages 457–466, 2001.

[79] D. A. Menascé and V. A. F. Almeida. *Scaling for E Business: Technologies, Models, Performance, and Capacity Planning*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[80] D. A. Menascé;, V. A. F. Almeida, R. Fonseca, and M. A. Mendes. A methodology for workload characterization of e-commerce sites. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 119–128, New York, NY, USA, 1999. ACM Press.

[81] D. A. Menascé, V. A. F. Almeida, R. Riedi, F. Ribeiro, R. Fonseca, and W. Meira, Jr. A hierarchical and multiscale approach to analyze e-business workloads. *Perform. Eval.*, 54(1):33–57, 2003.

[82] P. Milgrom. Auctions and bidding: A primer. *The Journal of Economic Perspectives*, 3(3):3–22, 1989.

[83] D. Mosberger and T. Jin. httperf–tool for measuring web server performance. *SIGMETRICS Perform. Eval. Rev.*, 26(3):31–37, 1998.

[84] N. K. Mukhi and P. Plebani. Supporting policy-driven behaviors in web services: experiences and issues. In *ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing*, pages 322–328, New York, NY, USA, 2004. ACM Press.

[85] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[86] E. M. Nahum, M.-C. Rosu, S. Seshan, and J. Almeida. The effects of wide-area conditions on www server performance. In *SIGMETRICS '01: Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 257–267. ACM Press, 2001.

[87] M. Newman, D. Watts, and S. Strogatz. Random graph models of social networks. In *Proc. Natl. Acad. Sci.*, volume 99, pages 2566–2572, USA, 2002.

[88] M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67:026126, 2003.

[89] D. Olshefski, J. Nieh, and D. Agrawal. Inferring client response time at the web server. In *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 160–171. ACM Press, 2002.

[90] N. W. Paton, editor. *Active Rules in Database Systems*. Springer, New York, 1999.

[91] C. B. G. Pecotche. *The Mechanism of Conscious Life.* Editora Logosófica - São Paulo - Brazil, 1997.

[92] A. Pereira, G. Franco, L. Silva, and W. Meira, Jr. A hierarchical characterization of user behavior. In *Proceedings of the 2nd Latin American Web Congress and the 10th Brazilian Symposium on Multimedia and the Web (LA-WebMedia 2004)*, Ribeirão Preto, SP, Brazil, 2004. IEEE Computer Society.

[93] A. Pereira, G. Franco, L. Silva, and W. Meira, Jr. A hierarchical characterization of user behavior. In *Proceedings of the 2nd Latin American Web Congress (LA-Web 2004)*, Ribeirão Preto, SP, Brazil, 2004. IEEE Computer Society.

[94] A. Pereira, G. Franco, L. Silva, W. Meira, Jr., and W. Santos. The *usar* characterization model. In *Proceedings of the IEEE 7th Annual Workshop on Workload Characterization (WWC-7)*, Austin, Texas, USA, 2004. IEEE Computer Society.

[95] A. Pereira, F. Mourao, P. Goes, and W. Meira, Jr. Reactivity in online auctions. In *Workshop Reactivity on the Web at EDBT 2006*, volume 4254 of *LNCS*, pages 909–918, Munich, Germany, 2006. Springer.

[96] A. Pereira, L. Rocha, F. M. ao, T. Torres, P. Góes, and W. M. Jr. A hierarchical characterization model for online auctions. In *ICISTM'07: Proceedings of the First International Conference on Information Systems, Technology and Management*, New Delhi, INDIA, 2007.

[97] A. Pereira, L. Rocha, F. M. ao, T. Torres, W. M. Jr., and P. Góes. Analyzing ebay negotiation patterns. In *WEBIST'07: Proceedings of the 3rd International Conference on Web Information Systems and Technologies*, Barcelona, SPAIN, 2007.

[98] A. Pereira, L. Silva, and W. Meira, Jr. Evaluating the impact of reactive workloads on the performance of web applications. In *Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC)*, Phoenix, Arizona, USA, 2006. IEEE Computer Society.

[99] A. Pereira, L. Silva, W. Meira, Jr., and W. Santos. Assessing the impact of user reactivity on the performance of web applications. Technical Report RT.DCC 007/2005, Department of Computer Science, Federal University of Minas Gerais, Brazil, April 2005.

[100] A. Pereira, L. Silva, W. Meira, Jr., and W. Santos. Assessing reactive qos strategies for internet services. In *Proceedings of the IEEE/IPSJ Symposium on Applications and the Internet (SAINT2006)*, Phoenix, Arizona, USA, 2006. IEEE Computer Society (IEEE-CS) and Information Processing Society of Japan (IPSJ).

[101] A. Pereira, L. Silva, W. Meira, Jr., and W. Santos. Assessing the impact of reactive workloads on the performance of web applications. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS-2006)*, Austin, Texas, USA, 2006. IEEE Computer Society.

[102] J. Preece, Y. Rogers, and H. Sharp. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, Inc., 2002.

[103] J. Preece, Y. Rogers, H. Sharp, and D. Benyon. *Human-Computer Interaction*. Addison-Wesley Longman Ltd., Essex, UK, UK, 1994.

[104] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *The Economics of the Internet and E-Commerce. Amsterdam: Elsevier Science B.V.*, pages 127–157, 2002.

[105] A. Roth and A. Ockenfels. Last-minute bidding and the rules for ending second-price auctions: Evidence from ebay and amazon auctions on the internet. *American Economic Review*, 92(4):1093–1103, 2002.

[106] M. H. Rothkopf and R. M. Harstad. Modeling competitive bidding: a critical essay. *Manage. Sci.*, 40(3):364–384, 1994.

[107] P. Selvridge, B. Chaparro, and G. Bender. The world wide wait: Effects of delays on user performance. In *Proceedings of the IEA 2000/HFES 2000 Congress*, 2000.

[108] G. Shmueli and W. Jank. Modeling the dynamics of online auctions: A modern statistical approach. *Economics, Information Systems & Ecommerce Research II: Advanced Empirical Methods*, 2006. Forthcoming.

[109] H. A. Simon. *Models of Bounded Rationality.* MIT Press, 1982.

[110] H. A. Simon. *Models of my life.* Basic Books, Inc., New York, NY, USA, 1991.

[111] Specweb99 benchmark, 1999. http://www.specbench.org/osg/web99/.

[112] F. Stefano. Simon's bounded rationality. origins and use in economic theory. CESMEP Working Papers 200509, University of Turin, July 2005. available at http://ideas.repec.org/p/uto/cesmep/200509.html.

[113] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat. Medisyn: a synthetic streaming media service workload generator. In *Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, pages 12–21. ACM Press, 2003.

[114] B. Urgaonkar and P. Shenoy. Cataclysm: policing extreme overloads in internet applications. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 740–749, New York, NY, USA, 2005. ACM Press.

[115] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin. A hierarchical characterization of a live streaming media workload. In *Proceedings of the second ACM SIGCOMM Workshop on Internet measurment*, pages 117–130. ACM Press, 2002.

[116] A. J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Softw.*, 3(3):253–256, 1977.

[117] Webbench 5.0 benchmark, 2002. http://www.veritest.com/benchmarks/webbench/.

[118] B. Wellman. A computer network is a social network. *SIGGROUP Bull.*, 19(3):41–45, 1998.

[119] J. Widom and S. Ceri. *Active Database Systems: Triggers and Rules For Advanced Database Processing.* Morgan Kaufmann, 1996.

[120] N. Ye, E. S. Gel, X. Li, T. Farley, and Y.-C. Lai. Web server qos models: applying scheduling rules from production planning. *Computers & Operations Research*, 32:1147–1164, 2005.