

The Marketplace of User Interface Real Estate

Luigi Troiano and Gerardo Canfora
University of Sannio, Department of Engineering
RCOST Viale Traiano
82100 – Benevento, Italy
{troiano,canfora}@unisannio.it

Abstract

Financial markets suggest new ways to design, organize and evolve user interfaces. In this paper we present an approach based on marketplace simulations, where UI resources are sold and acquired, attempting to prove that this metaphor can innovate the way we think about UI design and deployment, relying on self-ruling emerging properties of complex systems.

1. Introduction

Imagine a user interface able to self-organize components. Imagine you are watching a movie, when video is resizing and an upcoming message gains space on the bottom of the screen. You see that is an advertisement, so you ignore it, and video after a few, goes again in full screen mode. A new message from office arrives, so you pay attention to it. The message gains more space on the screen, and video becomes first smaller, and finally goes on pause. Nothing urgent from office, and you can continue to enjoy your movie, that goes again in full screen mode.

In this interface components are not cooperating as expected, they are competing to gain interface resources. This competition is ruled by a marketplace in which interface assets are regularly on sale, and applications attempt to acquire them spending credits. Coordination is not coming out by cooperation. Instead, it is expressed as equilibrium among competitors. The problem of agents competing for limited resources is quite old, and widely studied by Economics to model social behavior.

This inspires a novel approach in conceiving user interfaces. Not a set of resources used by applications according only to their needs, but a set of assets that must be acquired in order to perform application functionalities. The UI real estate is made of different assets, such as screen regions, audio channels, keyboard, pointer and other input devices.

2. The UI marketplace

The marketplace is where the demand and the offer for UI assets meet. The UI orchestration is reached according to rules governing the marketplace. There are different models from real world we can adopt for ruling the UI marketplace. Stock Exchange provides a good example from which to take a move. In this market, trades are based on an auction market paradigm where a potential buyer bids a specific price for a stock and a potential seller asks a specific price for the stock. When the bid and ask prices match, a sale takes place on a first come first served basis if there are multiple bidders or askers at a given price. Exchange is aimed at facilitating the trading of securities between buyers and sellers. A slightly different marketplace is the Foreign Exchange. Foreign exchange is an out-of-the-counter (OTC) market where brokers/dealers negotiate directly with one another. There is no central exchange or clearing house. The market participants (generally large international banks) continually provide the market with both bid (buy) and ask (sell) prices. The bid/ask spread is the difference between the price at which a bank or market maker will sell (ask, or offer) and the price at which a market-maker will buy (bid) from a wholesale customer.

We started our investigation with a simple auction marketplace. Its structure is depicted in Fig.1. There are different types of auction. The most well known are: (i) Open Ascending Price (English) auction, where participants bids each against one another; (ii) Open Descending Price (Dutch) auction, where auctioneer starts from a high initial asking price that is lowered until there is a participant who accepts the ask price; (iii) First-Price Sealed-Bid (FPSB) auction, where participants make a simultaneous bid without knowing the bids made by the others; (iv) Sealed-Bid Second-Price (Vickrey) auction, similar to FPSB but the winner pays the second highest bid plus an increment. There are many other variants of these types, concerning the time limits, the number of participants, the minimum and maximum bid price, etc. We chose a FPSB auction, as

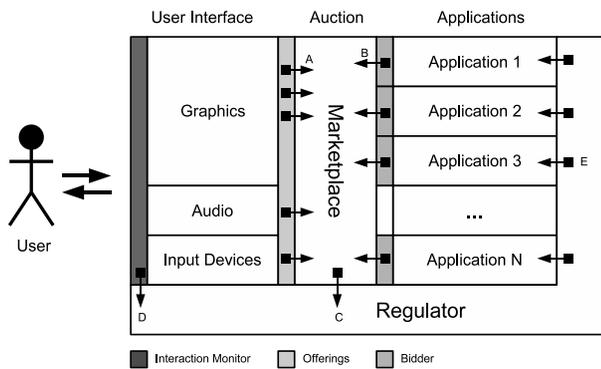


Figure 1. Structure of the marketplace

it is a scheme fast and scalable, with a minimum ask price and bid priority.

2.1. Auction

The UI assets can be different according to the interface features. Typical assets are screen regions, audio channels and input devices. Each asset is characterized by properties that make it more or less suitable for the application needs. For example, a region could be too small for placing a UI widget. Moreover, some assets can result more appealing than others. For instance, eye tracking studies have proved that the upper-left quadrant gets more the user attention. The important thing to underline is that applications do not know the number and qualities of assets at design time.

Applications gain the control of UI assets for a limited time. When the asset is released it goes on the market for sale (A). The Marketplace decide for a minimum ask price and the asset becomes part of proposals given to the application Bidders. Bidders make an asset assessment aimed at valuing quality properties such as centrality and size for screen regions, volume level (i.e. foreground or background) for the audio channel, etc. According to this appraisal, they decide which proposals to make bid on (B). They also declare for how many time slots the asset will be acquired. Bids are collected by the Marketplace that decide the auction winner, collecting the credit from the application credit book it owns. Applications holding assets also pay a maintaining tax to the marketplace according to the period they keep the asset. Taxes and auction revenues are given to the Regulator (C). The user interacting with applications get benefit of them, so a credit is collected by the Regulator according to the user interaction (D). Different techniques are available to measure the user interaction, such as eye tracking, click stream analysis, and input stream analysis. In our initial investigation we chose a click stream analysis as a fast way to prove our concept. The Regulator

is in charge of redistributing collected credits, on the basis of two main policies (E). The first (Capitalism) is the applications have been used more, should participate to the capital gain redistribution proportionally to the user interactivity. The second (Welfare State) should assist application with a lower credit availability.

2.2. Market rules

In order to make the whole system feasible and functioning, some rules should be considered. We propose the following:

- Application can express a high/normal level of interest for an asset. So, assets are assigned first considering bids with a high level of interest, then looking at bids with normal level of interest. Application can express a high level of interest for one proposal at a time.
- The minimum asset ask price is determined by considering bids on the asset within a time interval. This helps to get comparable bids from applications.
- Holding taxes should be related to the asset value the application gained the control for. In our case we chose a taxation of 10%, paid at each time slot, until the application does not release the asset, or the time limit is reached.
- Two or more winning bids can have the same value. The asset can be assigned considering different criteria, such as how many auctions an application have lost in order to give priority to those are waiting longer, and/or which auction registered first.
- The capital revenue should be distributed among all applications, in order to give to each participants enough credits to gain assets in the future, but preventing application from monopolizing assets. We chose to give 40% to winners and 60% to losers. In addition losers have been divided in two categories: short-term losers (only at recent auctions) obtaining the 18%, and long-term losers (losing auctions reiteratively) obtaining the 42
- Bankruptcy is an event inevitable in every financial system. Regulator should have policies for supporting bankrupted applications, for instance having a fund for this kind of applications.
- The system can register inflation and deflation trends in the asset prices. This will ask the Regulator to have policies for facing this event, such as injecting credit liquidity in the market, or extra-taxing bids.

- Capital growth should be limited, in order to avoid a liquidity crisis on the market. Regulator can employ policies in order to trim the capital given to applications.

2.3. Emerging behavior

The system can be studied with tools and methods coming from Economics and Finance. In particular, we need first to understand the rationale driving applications and the market itself, as the ultimate goal is to provide a usable set of applications to the user.

The competition among market participants allows to reach this goal, without the need of programming explicitly the coordination among UI components, or asking the user to orchestrate the interface according to his/her needs and tasks. The system behavior emerges as a complex interaction among user and applications within the market, where both the user and applications are able to satisfy complementary needs. Indeed, if on one side the user is aimed at interacting with applications in order to achieve his/her tasks, on the other side applications have been designed and implemented in order to deliver functionalities to the user. On the contrary, application would become useless. This cycle of needs is able to drive the market to the goal it has been thought for.

Each application attempts in isolation to capture the user attention gaining the control of parts of the user interface. This is to answer its purpose. For this reason, applications are ready to compete and pursue their goal. If there is space enough for all applications, there is no need to compete, so there is no need of orchestrating the interface components. Competition is based on spending credits to get access to the UI assets. After, the application gains some space in the UI, it is able to interact with the user. The user will judge if this attempt was worthy or not. The judgment is determined by his/her behavior, interacting or not with the application. This will produce a capital gain, that can be distributed back to applications in order to reward those are really required by the user in that moment. This will provide more resources to interact with the user again. Applications that are not useful will see a reduction of their resources, suggesting to revise their strategy in gaining control of the user interface, or driving them to bankrupt, thus stopping their activity with the user.

2.4. Expected Benefits

The approach we are presenting, provides benefits from both the application user and designer point of view. Designers are not forced to think about how to organize components in their application, and applications are not forced to cooperate within a framework in order to get the compo-

nents orchestration. Designers are only demanded to provide a bidding strategy to their applications, and asset assessment criteria on which to base the bidding decision. Indeed, some assets could be more valuable than others according to the application needs.

On the other side, users can interact with a set of applications without being obliged to control and orchestrate explicitly the components deployed at the interface. Applications becomes more independent on GUI layout. They can automatically adapt to different layouts, as they are able to estimate the quality of assets according to their properties, despite how assets are arranged together.

3. An example of application

We conducted a preliminary experimentation in order to see how the system behaves. We considered the deployment of three applications: Media Player, Mailer and Instant Messaging. Each application is provided with different interface components. In particular, the Media Player has a title selection list, a video display and a control panel. The Mailer has a login dialog, a message list, a message window and a compose dialog. The Instant Messaging application has a contact list and chat dialogs.

For the sake of simplicity, we only considered as UI assets the screen regions in which a graphical layer has been divided as depicted in Fig.2.

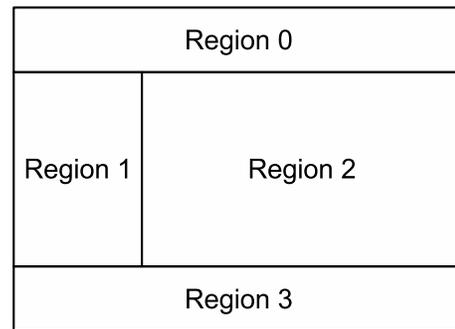


Figure 2. The UI real estate.

As sometimes regions are not assigned to any application, i.e. there could be no demand for them, the layout has also a feature that makes regions able to expand to neighbors.

Each application responds to events such as selecting a video title, or receiving a new message in the mailbox. Some of these events will lead to interact with the user. Experimentation was conducted with an event generator in order to simulate a user session. An extract of events sequence is given in Fig.3. In this figure we can note how the different application components are moving among the layout regions. The scenario we considered has an initial

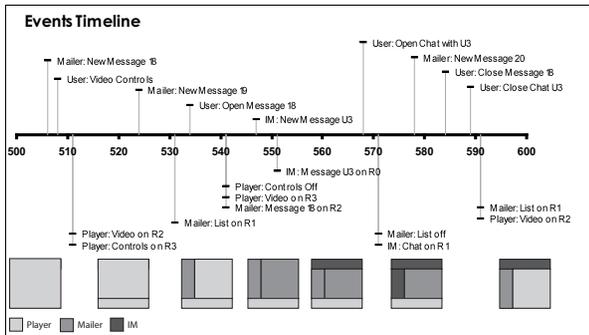


Figure 3. Events timeline between 500-600 sec.

activity for choosing a video and start to watch it, with mail messages arriving in the mailbox. Later chat session starts, whilst the movie ends. The capital time series of each application is depicted in Fig.4. Although all applications started

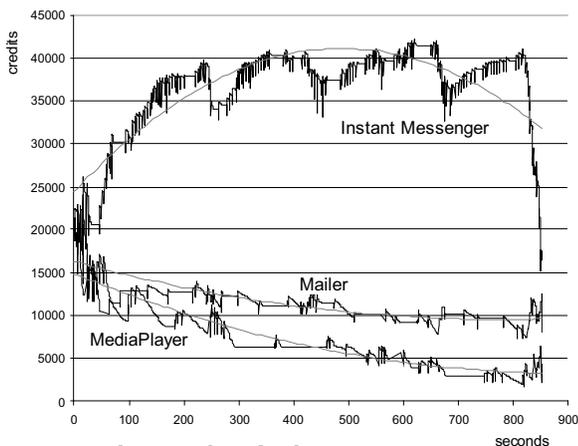


Figure 4. Application capital time series.

with an initial capital of 20k credits, the interaction with the user led applications whose interaction is more frequent to spend more, thus to erode their capital. We can also observe how there is a higher capital dynamics at the beginning, due to the initial competition aimed at gaining space on the interface layout. After, dynamics became smaller, as application already gained space and they mostly were paying holding taxes. This was because the bidding strategy we implemented is proportional to the resource availability. We think that a better strategy should consider price dynamics and the behavior of the other applications.

Experimentation also proved that the auction approach is feasible, not affecting performances too much, keeping the overhead below the 10% with an auction rate of 1 per second.

4. Related work

Intelligent and adaptive user interfaces have been of interest for researchers since long time [2], although research activities have been mostly reduced by the belief that an automatic reconfiguration the user interface may disorient the user. Instead we share the belief that adaptive user interfaces can still play a role for highly interactive systems, as those targeted by Rich Internet Applications [3]. In the past most effort has been devoted to find a coherent theory for adaptivity in human-computer interaction, and to prototype complex but often rigid systems. In the recent years a lot of attention has been paid to emergence [1] as a new approach for designing and engineering complex systems. Autonomic Computing [5] is a prominent example of this new way of intending the system complexity. Our research is aimed at driving interface adaptivity without a pre-made model of users.

5. Conclusions and future works

In this paper we presented an approach based on auction markets in order to orchestrate the user interface usage made by applications. The novelty of this approach resides in the fact that coordination of resources is not reached by an over-imposed cooperative scheme, but by competition in gaining the control of limited resources such as those provided by the user interface. Many interesting questions arise, both concerning theory and practice. For example, we believe that the Efficient Market Hypothesis [4] can be applied to the UI marketplace. Finding a set of constraints and requirements able to make the market efficient is important, as it makes the application behavior independent from each other, and it would open the way to build robust bidding strategies. The approach based on auction markets can be extended to other problems, where a demand of resources must face limitation of them. Examples are the service orchestration and the job allocation in Grid Computing.

References

- [1] P. W. Anderson. More is different: Broken symmetry and the nature of the hierarchical structure of science. *Science*, 177(4047):393–396, Aug. 1972.
- [2] H. Dieterich, U. Malinowski, T. Kuhme, and M. Schneider-Hufschmidt. *Adaptive User Interfaces: Principles and Practice*, chapter State of the Art in adaptive user interfaces, pages 13–48. 1993.
- [3] J. Duhl. Rich internet applications. White paper, IDC, Nov. 2003.
- [4] E. Fama. The behavior of stock market prices. *Journal of Business*, 38:34–105, 1965.
- [5] A. Ganek and T. Corbi. The dawning of the autonomic computing era. *IBM Systems Journal*, 42(1):5–18, 2003.