E-Commerce and Usability

Shawren Singh
University of South Africa, South Africa

ELECTRONIC COMMERCE

The term "Electronic Commerce" (EC) conjures various interpretations. Figure 1.1 shows some of the different types of EC, of which there are many, such as Business-to-Business (B2B); Business-to-Consumer (B2C); Consumer-to-Business (C2B); Consumer-to-Consumer (C2C); People-to-People (P2P); non-business EC; intrabusiness (organisational) EC; business-to-employees (B2E); government-to-citizen (G2C); exchange-to-exchange (E2E); collaborative commerce; ultimate commerce (u-commerce) and mobile commerce (m-commerce).

A certain basic infrastructure is required for any type of EC to function efficiently. Key components of this infrastructure are networks, Web servers, Web servers' support and software, electronic catalog, Web page design and construction software, transactional software and Internet access components (Turban, Rainer & Potter, 2001).

The infrastructure on which the EC application is built will affect the users' experience of that application. It is generally accepted that any EC that does not provide the user with such experience will not thrive (Brandt, 1999). The traditional approaches of enticing a purchase in brick-and-mortar commerce, such as atmosphere, placement of goods, lightning and so forth, cannot be applied to online commerce. Nielsen (1999) contends that a "bad" user interface is one of the reasons for EC failure. Interaction and participation are the emotional hooks for EC, and the developers of EC sites should bear this in mind.

DEVELOPMENT OF EC APPLICATION

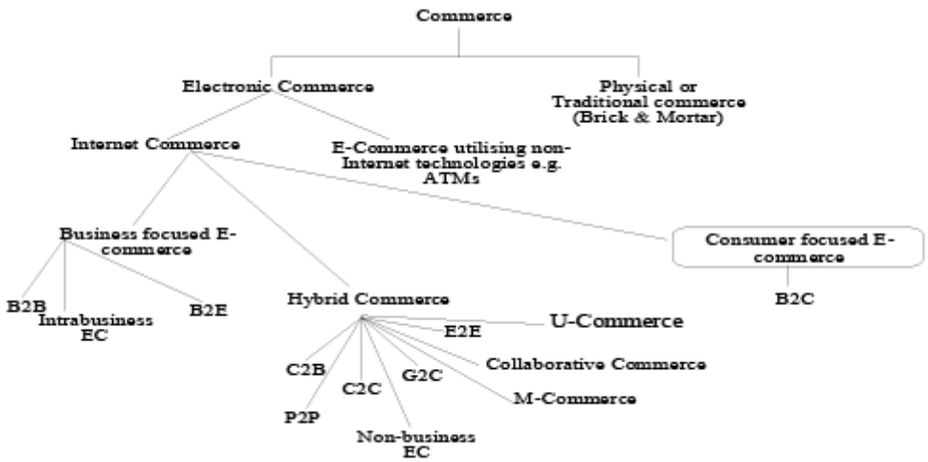
As companies realised that their EC ventures were not as successful as they had anticipated and were prone to failure, they began to investigate alternate development strategies to deal with this rapidly changing environment. One such approach that has won favour amongst Web application developers is agile development methodologies (ADMs). ADMs do not have prescriptive processes and do not define detailed procedures on how to create a given type of model. Instead, they provide advice on how to be effective as a modeller. As opposed to the traditional development approaches, ADMs are not hard and fast. ADMs can also be seen as a craft and not a science (Ambler, 2003).

With regard to the World Wide Web (WWW), Pressman (2000, p. 8) states: "What worries me is that this major new technology has become a breeding ground for important Web applications that are hacked in much the same way as important application software was hacked a few generations back—in the 1960s and 1970s."

Pressman (2000) goes on to say that the current basic Web development philosophy is that Web applications must be developed within days or weeks. The argument is that time frames do not allow for anything but a rush to the finish line. Web applications are constantly evolving. The argument, then, is why spend time on specifying what is needed and designing how to build it when everything will change anyway? Web applications are inherently different from other application software. The argument is that the content (text, graphics, images, audio and video, for example) is inextricably integrated with procedural processing.

Pressman believes that people who use Web applications are more tolerant of errors. The argument is that users really want exciting Web sites that are up and running in days. He argues that it is almost impossible to know what Web applications users really want, because the demographics of Web visitors are so hard to predict. We believe that as Web applications are becoming an integral part of life, users' fault tolerance is becoming much lower. The people who build Web applications are different. Web developers are free thinkers, who certainly would feel unduly constrained by the old ways. In fact, talk of a disciplined approach, other than "build it, test it to death (if time permits), and then put it online," usually results in grimaces all around. The development of

Figure 1. Types of commerce (adapted from Chan, Lee, Dillon and Chang (2001)



applications for the WWW, therefore, has its own set of unique problems. No current theory adequately addresses how to effectively create Web sites for online selling.

USABILITY

Usability refers to how usable a system is from a user's point of view. Usability concerns are not about presentation, but a whole gamut of exchanges that form human-computer interaction. Below are some issues that need to be considered to improve usability.

COGNITION, PERCEPTION AND PHYSIOLOGY

We can divide human (user) resources into three categories (Kotze, 2000): *Perception*, which refers to the way in which humans detect information in their environment; c*ognition,* which refers to the way in which humans process that information; and *physiology,* which refers to the way in which humans move and interact with physical objects in their environment.

A vital foundation for interactive-system designers is an understanding of the cognitive, perceptual and physiological abilities of the user. The human ability to interpret sensory input rapidly and to initiate complex actions makes modern computer systems possible. *Perception* involves the use of our senses to detect information (Kotze & Johnson, 2001). In computerized systems, this mainly involves using senses to detect audio output, senses to detect visual output and tactile feedback. This is affected by many factors, such as change in output (loudness/size); maximum and minimum detectable levels; field of perception (can the user see the display?); fatigue; Circadian rhythms (biological rhythms); problems with background noise; and so forth. *Cognition* involves various cognitive processes, including (Kotze & Johnson, 2001) short-term memory; long-tem memory and learning; problem solving; decision making; attention and scope of concern; search and scanning; time perception; perceptual or mental load; anxiety; and fear.

When we operate a system, we gradually move from general knowledge to rules and then to skills. Users with greater expertise will be able to enter the process at a higher level. Ideally, we all want to work at the highest skill level. We do not want to spend time thinking about the use of previous systems or sifting through our general knowledge. The more we work at the knowledge and rule level, the more uncertain we are about things. Users do not want to be forced to make guesses. Guessing introduces inefficiency and can consume a great deal of time in "repair" tasks when things go wrong; for instance, if we delete a file by accident.