This paper describes the Commitment, Concurrency and Recovery (CCR) features found in the ISO Common Application Service Element (CASE) for CCR and the SYNCPT verb of IBM's LU6.2. The International Standard for CCR is contained in ISO 8649 Part 3 and ISO 8650 Part 3.

1. History and objectives

The use of "two-phase commitment" to handle "simultaneous" updates of two or more separate resources has been known for many years in database work, and has become a common feature of "transaction" protocols.

Much of the early impetus in ISO came from the needs of the JTM work (see Chapter 14), where the distributed nature of JTM activity led to a requirement for commitment. The JTM work also emphasised the need for well-defined protection against crashes of systems, and the interaction of this with "commitment" handshakes.

Finally, the crucial interaction with concurrency controls was recognised and the term "CCR" - Commitment, Concurrency, and Recovery control was born. CCR was absent from SNA protocols until the introduction of LU6.2. The IBM and ISO approaches to CCR were developed independently of each other, but there was a very strong similarity of function and mechanism between the SYNCPT verb of LU6.2 and the ISO CCR Standard. This similarity was increased when, at a late stage in the development of ISO CCR, a detailed comparison with the LU6.2 SYNCPT verb functions was undertaken. This showed some features of LU6.2 which could usefully be added to the ISO Standard. These were

a) heuristic commitment; and

b) a single protocol round-trip exchange to end one atomic action and start the next one.
These have been incorporated in the final International Standard in a way which enhances the current LU6.2 features.

The result permits a program interface to be developed for transaction processing which could be supported by either the LU6.2 protocol or the ISO CCR protocol.

CCR is designed to ensure successful coordination and completion of activities distributed across several open systems, taking into account the possibility of network failures and system crashes.

CCR is often described as coordinating across several connections synchronisation of communications activity which, for a single connection, is handled in ISO by the session layer. A dual, but perhaps more useful, focus, is to regard it as coordinating the information processing at the nodes linked by the connections.

The ISO work neared stability at the end of 1983, but at this time there was a growing interest in use of CCR for other applications - transaction processing, remote database access, directory updates, which introduced elements of instability.

At the time of writing this paper (July 1986) the final form of the initial ISO CCR Standard is not certain. There is also instability in the precise way the ISO CCR specifications are to be included in other standardisation work, revolving around the so-called "cooperating main service" issue.

This paper describes the main features of CCR, present in the second ISO Draft International Standard, and broadly present also in the LU6.2 SYMCPPT verb. Where differences exist, the ISO approach is described first, and LU6.2 is covered in later sections.

2. Application of CCR facilities

CCR is primarily concerned with exception cases. It worries about what happens if a crash (loss of information, release of concurrency controls) occurs at critical points in an activity, or if one update succeeds but a related update cannot be done because the network has gone down.

This means that almost any application using CCR can be run without CCR, and will work a lot of the time. Nonetheless, for almost all applications, use of CCR gives a degree of reliability which is generally highly desirable.