Thesis/Dissertation Sheet

Surname/Family Name	:	**Lyons**
Given Name/s	:	**Anna Mary**
Abbreviation for degree as give in the University calendar	:	**PhD**
Faculty	:	**Engineering**
School	:	**Computer Science**
Thesis Title	:	**Mixed-Criticality Scheduling and Resource Sharing for High-Assurance Operating Systems**

Abstract 350 words maximum: (PLEASE TYPE)

Criticality of a software system refers to the severity of the impact of a failure. In a high-criticality system, failure risks significant loss of life or damage to the environment. In a low-criticality system, failure may risk a downgrade in user-experience. As criticality of a software system increases, so too does the cost and time to develop that software: raising the criticality also raises the assurance level, with the highest levels requiring extensive, expensive, independent certification.

For modern cyber-physical systems, including autonomous aircraft and other vehicles, the traditional approach of isolating systems of different criticality by using completely separate physical hardware, is no longer practical, being both restrictive and inefficient. The result is mixed-criticality systems, where software applications with different criticalities execute on the same hardware. Sufficient mechanisms are required to ascertain that software in mixed-criticality systems is sufficiently isolated, otherwise, all software on that hardware is promoted to the highest criticality level, driving up costs to impractical levels. For mixed-criticality systems to be viable, both spatial and temporal isolation are required.

Current aviation standards allow for mixed-criticality systems where temporal and spatial resources are strictly and statically partitioned in time and space, allowing some improvement over fully isolated hardware. However, further improvements are not only possible, but required for future innovation in cyber-physical systems.

This thesis explores further operating systems mechanisms to allow for mixed-criticality software to share resources in far less restrictive ways, opening further possibilities in cyber-physical system design without sacrificing assurance properties. Two key properties are required: first, time must be managed as a central resource of the system, while allowing for overbooking with asymmetric protection without increasing certification burdens. Second, components of different criticalities should be able to safely share resources without suffering undue utilisation penalties. The implementation platform is the seL4 microkernel, which is built for the safety-critical, high assurance domain.