For many years now both in my talks and my writings I have been suggesting that real progress in software technology requires a scientific base and framework such as those that support other engineering technologies. More recently, while working in the Eureka ESF project I came across the terms framework and reference models. From the contexts in which they were being used it seemed to me that these two terms were essentially synonymous. I did not fully understand what precisely what was meant in that context. What basic concepts underlay them? What issues or problems were they intended to address or solve. What did appear clear was that they were not being used in my sense of the science surrounding and supporting the technology. Instead they appeared to be the output of a process of classification and categorisation. As such they provide the knowledge and reflect the understanding necessary for the building of a theory, truly an essential part of its development. But is that all. I had hoped that the papers presented at this session and the discussion that must follow would clarify the matter, extending the terminology beyond my interpretation. But perusal of early versions of the papers does nothing to resolve my semantic problem. If anything they confirm my understanding though the authors appear to have in mind uses other than the development of a theory supporting software technology. But I am still not certain that I have got it right, that I really understand the meaning to be attached to framework and reference model as used by the software engineering community.

Of the three papers we are to have presented that by Christie et al is the most general. Its generality is implied by the title A Reference Model for Software Technology. It is confirmed by the content. They describe their intent as presentation of the beginnings of a taxonomy of concepts underlying software process technology. This does not appear far removed from my understanding of the common usage of reference model as categorisation and classification. They do however provide the further elucidation that it is concepts that are to be addressed. As such their contribution could be an important input to development of a theory the need for which is now widely acknowledged. They do not provide clear indication how or for what other purposes their framework might be used. The second paper by Nguyen et al also appears to consider framework as a categorisation. However, as described by the title, Towards a Rigorous Approach for Managing Process Evolution, their concern, is the description and use of a specific framework. In that they already indicate their belief and seek to demonstrate a practical use to which frameworks may be used. By providing this example they appear to reply to at least one of the questions I posed above. As to their proposed use of the framework for the management of process (and system)
evolution, I am curious to discover from their discussion where there is more than
conjecture at this stage, whether they are laying foundations for further research
or whether the paper reflects real experience. In Conradi’s position paper, SPIQ:
A Revised Agenda for Software Process Support, framework as categorisation
is implicit rather than explicit. The paper makes an interesting classification of
those involved in or influencing process improvement and the contribution they
can make. It (in its early draft version) presents useful insights and then outlines
a research program to be undertaken jointly with Norwegian software houses and
based on these insights that is to seek improvements in their individual processes
and hence to achieve a National software process improvement program.

I look forward to the clarification and insight that will, hopefully, follow the
presentations and ensuing discussion. As a budding theoretician and philoso-
pher I expect to take away some interesting and useful concepts and insights
from these three presentations and the ensuing discussion. I hope that the prac-
titioners amongst us will reap equal benefit.