SERVICE ENGINEERING VERSUS SOFTWARE ENGINEERING*
A FOUNDATIONAL STUDY

Mícheál Mac an Airchinnigh¹ and Hans-Jürgen Kugler²

¹ Department of Computer Science, Trinity College, Dublin, Generics (Software) Limited and K&M Technologies Limited, Ireland. e-mail: mmaa@cs.tcd.ie
² European Software Institute, E-48016 Zamudio (Bilbao), Bizkaia, Spain. Tel: +34-4-420 9519

Abstract. The proposed marriage of service engineering and software engineering is expected to produce offspring that will do justice to both professions. Specifically, it is believed that the market in telecommunication services is better served by those in the software business and that telecommunication engineers can focus on network provisioning. The so-called service creation environment (SCE) is expected to play the role of catalyst in this brave new world.

Much has already been written on the subject. But experience has shown that much still needs to be written. For, even though there are commercially available, though proprietary, SCEs, the question: What is an SCE?, is still being asked.

In this paper, foundational issues are addressed. The nature of an SCE is examined specifically to exhibit the peculiar inherent characteristics of service engineering and the necessary conclusion is reached that the service/software union is wholly determined by the practice of formal methods, counterbalanced and complemented by certification in a service validation centre, of which a testing environment shall be a key component.

Keywords: engineering, environment, foundation, genus, service, software, species, testing.

1 Genus, Species and Differentiæ

One of the key activities of SCORE (RACE 2017, Service Creation in an Object-oriented Reuse Environment) is the determination of the nature of a service creation environment (SCE). More specifically it attempts to answer the question

• • What is a service creation environment?

* The results reported in this paper have arisen from work partially funded by the RACE Project Ref: 2017, Service Creation in an Object-oriented Reuse Environment (SCORE). The opinions expressed are those of the authors.
The need for such a determination, ultimately to be expressed as a set of requirements, appears to be driven by telecommunications' concerns and inspired by the realisation that greater productivity in service creation can be generated by software firms who will act as service providers.

Undoubtedly, the confidence that telecommunications' firms place in software firms has been generated by the emergence of

(a) the software engineering environment (SEE) for the development and support of complex software systems, and
(b) the expanding PC market in software products.

However, the limits to such confidence are to be severely scrutinised and tested in this paper. Specifically, one may take the view that service engineering is but one of the many forms of software engineering.

Let us consider then that a software engineering firm, which is completely independent of a network operator or provider, develops, among other products, services for deployment on a network. Currently, such a service may be deployed only under the control of the network provider or operator, i.e., it is subject to the control of their service management system. At first glance, this seems reasonable and sound, both from a pragmatic and a legal perspective. But there is another opposing scenario. Suppose that service providers were permitted to manage their own services, and to deploy new services at will? What would be the corresponding implications for the nature of the processes used to develop such services?

In this paper we propose to examine the nature of service engineering and contrast it with software engineering. We shall argue that there are distinctions between service engineering and software engineering and shall explore the precise nature of these distinctions. It may be fruitful to locate our prospective analysis in the analogic setting of darwinian evolution. In other words, the principals whose behavioural processes we observe—the service/software engineers—carry out their tasks in an environment which both moulds them and, is in turn formed by them. Consequently, in such a darwinian framework and concomitant universe of discourse, the service creation environment must play a foundational role in our purposes, being in a sense part of the engineer's habitat.

To establish a firm foundational basis for the specification of an SCE model, we resort to the classical Aristotelian framework of genus, species, and differentiae[1] which underlies all classification (and, thus object-oriented).

---

3 As a working hypothesis we shall suppose that a service engineer is a telecommunications engineer (and, therefore, an engineer in the real sense) who is responsible, in some sense, for the creation, deployment and management of services.

4 In spite of the vast amount of attention given to object-oriented technologies, including the early work carried out in the domain of Artificial Intelligence, it is remarkable that little if any reference is made to the neo-Aristotelian corpus on the subject, accumulated over two thousand years. We have no desire to make the same mistake.