Blume
Software Engineering LLC

Object-orientation was intended to model the real world, both the concrete and abstract. For example, you want to leverage the idea of a "container", something you put other things in - as a place to keep them. You might use a coffee cup to hold your coffee - or you might use it to hold pens and pencils, thus turning the coffee cup into a pen/pencil cup so you can keep all your pens and pencils in a specialized container that helps you keep your desk organized. In order to do that, however, you must abstract the "coffee cup" AND the "pen/pencil cup" into a "container". Both coffee cups and pen/pencil cups derive, they "inherit" from the general, more abstract CONCEPTUAL object known as a "container". This introduction of the conceptual nature of how we apprehend and perceive objects now puts us in the area of cognitive science where we start to look at HOW we view the objective world, both abstract and concrete in a hierarchy of classes - under which we organize and make sense of our world. As it turns out, our cognitive skills are involved with software development, the “knower” has a role in the creation of software systems, making the developer himself/herself a component of the process.

In David Brown's "An introduction to Object-Oriented Analysis" (ISBN 0-471-11028-0, 1997) the author reveals that "According to Webster (1975), an object is "something that is, or is capable of being, seen, touched or otherwise sensed." "Or, alternatively, it is 'something physical or mental of which a subject is cognitively aware."1 He goes on to relate that The Oxford Dictionary has yet another definition - which tends to support the previous two. [1] Brown then elucidates on page 117 of his book that Object-orientation works because it organizes the objective world in the same way that our brains work, in terms of our cognitive function that identifies objects, generalizes and also specializes them into classifications like “coffee cup” is a container, container can be specialized as a pen/pencil cup and so on.

The “trick” of object-orientation then becomes one of decomposition of the world around us, how we abstract objects. That requires an object decomposition. When you think about it in terms of object-oriented analysis, design and programming, that becomes quite profound. Where people are having difficulty with OOT (Object-Oriented Technology) is in the way they break down the “problem” for which they're trying to provide a solution. That old “functional decomposition” seems to keep getting in the way of a proper object decomposition because the old functional decomposition mindset is still so prevalent. People are still thinking in terms of algorithms, a very granular level to approach the problem – a procedural analysis – instead of its objects and an object-oriented analysis.

OO is difficult and “does not work” for those who are procedurally oriented or focused on the functional approach to problem-solving, cognitively. On the SOA side, those who are building architectures from a Service-Oriented approach – might arrive at those services from a procedural approach, but I would guess they do so with great difficulty. Instead, since the current technology[2] that helps us implement a Service-Oriented architecture is itself object-oriented internally, it would be easier to build services with an object-oriented approach instead of a functional or procedural approach. SOA is, after all, merely a higher level of organization, a larger architectural approach to software development. The thing to remember is that once you “go OO”, EVERYTHING must be “OO”, including your problem domain decomposition and your design. Nearly every author I've read on this topic – including David Brown - has pointed out that your whole approach to software

[1] Pg 107 - “An Introduction to object-oriented Analysis” - David Brown
[2] In the “Microsoft Stack”, that technology is WCF, Windows Communications Foundation.

development must be object-oriented, or you are not doing OO. You cannot build an object-oriented system, by thinking in terms of functionality, procedures or methods. Your very THINKING must change from a procedural approach to an object-oriented one. You may even be required to rearrange your consciousness, itself. If you want to know more about that, you'll be studying quantum physics, biochemistry, neurophysiology and everything in between. For that, you should watch this video presentation by Stuart Hameroff: https://youtu.be/hKAVgq99o_w

Happy object-orientation!