
NpubMaterial

Release 0.1

Matheus

Dec 09, 2021

CONTENTS

1	Contents	3
1.1	Introduction	3
1.2	Modelling	4
1.3	Generating	4
1.4	Publishing	4
1.5	Retrieving	4
1.6	Example Projects	4
1.7	Useful References	4

In order to help new stakeholders better understand the content of Nanopublications, **NpubMaterial** provides a compilation of materials related to the conceptual part of nanopublications and the practical application of using nanopublications in different domains.

In addition to the introductory conceptual part of nanopublications, this material presents an overview of the steps that can be taken to model, generate, publish and retrieve data and metadata following the concept of nanopublications. Finally, it is also possible to find some projects and references that use the concept of nanopublications as the foundation of their proposals.

To better understand the principles that underpin the concept of nanopublications, check out the [Introduction](#) section.

Note: This project is under active development.

CONTENTS

1.1 Introduction

Seven years before the conception of FAIR, the nanopublication concept was created, as an approach with the similar goal of reusability as the digital object. The purpose of the nanopublications, firstly, was a way to publish researchers papers as small annotated statements to be citable, attributable, and reviewable like the scientific peer-reviewed publication, but more easily by aggregating and identifying through the Web, enabling better discoverability, interoperability, and reusability. Although, the intention of nanopublications has been broadening out by an approach of publishing digital objects on minimal scientific statements with their associated context and provenance. A nanopublication is composed using the W3C Resource Description Language (RDF), the Web Ontology Language, and the nanopublication schema, that consists of:

Assertion: Contains the small meaningful atomic unit of information.

Provenance: Is the provenance of the assertion, in order words, provide information about the assertion in the manner of how the assertion was generated, who generated it, when it was generated, which type of process or machine was used to generate it, and any other important information.

Publication Info: Contains the provenance of the whole nanopublication. When the nanopublication was created, by whom it was created, the type of usage license, and any provenance to identify this nanopublication's composition.

The core section in a nanopublication is the assertion, which can be represented in RDF as a triple with the Subject, Predicate, and Object, like a small normal phrase. To illustrate the scenario, figure 3.3 represents a nanopublication, extracted from the work of Mons and Velterop (2009). In this case, the assertion “Malaria is transmitted by mosquito.” represents two concepts (Malaria and Mosquito) and a relationship (Transmitted by). By translating it to an RDF triple it can be displayed as Subject (Malaria), Predicate (Transmitted by), and Object (Mosquito). The assertion graph is represented inside the blue box in figure 3.2.

Besides, nanopublications have the provenance graph about the assertion in the orange box and the publication info-graph about the nanopublication in the yellow box. In order to link the assertion, provenance and publication info graphs and identify the nanopublication itself, the nanopublication has a head graph represented by grey box. Nowadays the nanopublications can be published in a decentralized peer-to-peer server network, called nanopub server, where they are signed with Trusty URIs with the objective to make the nanopublications immutable, permanent, verifiable, and decentralized.

Since the first nanopublication proposition, a series of projects were developed, most of them in the life science area. Kuhn's article (2018) summarizes some of these works that produced more than 10 million nanopublications.

1.2 Modelling

—

1.3 Generating

Generating

1.4 Publishing

1.5 Retrieving

1.6 Example Projects

1.7 Useful References