El punto de partida para esta revisión son los *reviews* de Malekshahi [1] y Roopa [2] muy completos.

# Social IoT

Primero debemos intentar definir o al menos crear un consenso de qué consideramos como SIoT. La idea parte de combinar las redes sociales SN (Social Networks) con el IoT.

Moviéndonos en el tiempo y comparando a varios autores se puede resumir el SIoT como un escenario donde los objetos que se comunican entre si, teniendo como base la internet (IoT), son capaces de establecer relaciones sociales con otros objetos, de forma autónoma, sin la intervención de los humanos [3]–[5]. Esta idea en un principio pudiera sonar contradictoria, pues la naturaleza de la socialización es inherente a las personas, pero no a los objetos.

Prácticamente todos los autores consultados coinciden que la génesis del SIoT está en Kleinberg y su planteamiento de el fenómeno del mundo pequeño (*small world phenomenon*), aunque en realidad este trabajo describe el fenómeno de las redes sociales como tal. Según [3] las primeras ideas sobre la socialización de los objetos fue planteada por Holmquist et al. [6] en el 2001, año en que el IoT y las SN estaban en su infancia, no obstante este trabajo plantea el establecer relaciones temporales entre objetos de una WSN (*Wireless Sensor Network*) para prestar un servicio.

El término de Web 2.0 acuñado por O´Reilly [7] (aunque ya considerado obsoleto por él mismo [8] desde el 2011) separa paradigmas donde los actores de Internet dejan de ser consumidores pasivos de contenidos a ser generadores e interactuar en igualdad de condiciones distinguiendo entre la idea de estar conectado a la internet y participar de la internet.

En este caso la SIoT trata de imitar el comportamiento humano cuando trata de establecer relaciones en función de construir su propia red social [4]. A su vez este paradigma no excluye a las personas, de hecho las integra en el modelo de relaciones [9]–[11]. Al final, una red social de dispositivos y personas interconectadas basada en intereses y motivaciones comunes es capaz de proveer nuevos y mejores servicios a usuarios finales.

El SIoT extrae entonces de las SNs su estructura y describe como los objetos deben comunicarse, seleccionar a sus amigos y hacer asociaciones del tipo Amigos de mis Amigos (o FaAF *Friends of A Friend*). Varios trabajos abordan formas para la selección de amigos [12]–[14]. Farhadi et al. [15] hacen un *review* al respecto (este documento no lo tengo).

Cada objeto que se considere “amigo” de otro objeto tendrá en cuenta diversos parámetros como **escalabilidad**, **interoperabilidad** y **confianza** que serán definidos por sus propietarios en función de su comportamiento y los servicios a los que están destinados.

Malekshahi [1], desde el mismo resumen establece los principales componentes o aristas del SIoT (*Social Internet of Things*):

* Arquitectura
* Gestión de las Relaciones (*Relation Management*)
* Gestión de la Confianza (*Trust Management*)
* Servicios Web
* Información

Basados en esta idea intentaremos resumir dichas aristas, poner en claro los conceptos en torno a ellas y llevar una discusión sobre los requerimientos, limitaciones y desafíos recogidos en la literatura consultada sobre SIoT.

Además haremos una revisión de los *frameworks* e implementaciones de sistemas SIoT…..

Intentaremos entonces buscar toda la literatura disponible sobre SIoT entre 2011 y diciembre de 2021 en las publicaciones de revistas y conferencias indexadas en reputadas bases de datos bibliográficas internacionales.

Tabla 1: Bases de datos bibliográficas consultadas

|  |  |
| --- | --- |
| Database | URL |
| ScienceDirect | http://www.sciencedirect.com |
| IEEE Xplorer | http://ieeexplorer.ieee.org |
| ACM | http://dl.acm.org |
| John Wiley | http://onlinelibrary.wiley.com |
| Springer | http://links.springer.com |
| … | … |

Además de las bases de datos citadas se intentará procesar artículos citados en los ya consultados y de momento ningún documento queda excluido, siempre que se pueda verificar la calidad de la publicación. Siendo así serán procesados documentos en otros idiomas además del inglés, principalmente español, portugués, italiano y alemán.

## Arquitectura y Protocolos

Tanto Malekshahi [1] y Roopa [2] coinciden en que no existe un modelo en concreto consensuado o una arquitectura para el SIoT. Encontramos la misma situación en el propio IoT como punto de partida, aunque en este caso hay bastante consenso en un modelo de 3 capas [16], [17]:

* Capa de percepción (*Perception Layer*): Es la capa más baja e incluye los elementos que recogen información sobre el entorno físico, básicamente los sensores.
* Capa de red (*Network Layer*): Incluye la forma en que los datos son movidos desde el lugar de recolección hasta el punto de consumo. Hablamos de los *stack* de red que intervienen (protocolos), los dispositivos de red terminales e intermedios, medios de transmisión, mecanismos de acceso al medio, ruteo, etc.
* Capa de aplicación: (*Aplication Layer*): Aquí es donde residen los consumidores de datos y gestores de servicios, conocido de forma primaria como Plataforma IoT. Es aquí donde el espectro se abre al SIoT.

Es evidente que este modelo no es suficiente para incluir al SIoT. Si bien, como nos hace notar Malekshahi [1], los autores plantean una arquitectura por capas, si podemos concluir que es la forma más común y probablemente la más simple y organizada y en la que nos centraremos para definir y/o adoptar un modelo.

Uno de los puntos de división parece ser el alcance en la descripción de la arquitectura. En otras palabras, que elementos se incluyen explícitamente y cuales se abstraen. En este sentido algunos pueden excluir las capas del IoT tratarlo como un elemento abstracto (considerando que la forma que se manejan los objetos físicos, obtienen los datos y se conectan a la red es problema de la IoT). Así Ortiz et al. [18] describen una arquitectura (yo diría al otro extremo) con elementos completamente globales y abstractos como Actores, Sistemas Inteligentes, Interfaces e Internet.

Otras ideas parte de la base de ver al objeto como una entidad virtual u entidades equivalentes a objetos físicos en el mundo real. Este concepto es tratado por Voutyras et al. de esta misma manera (VE *Virtual Entity*) [19] que presenta su arquitectura para el proyecto Cosmos [20], que provee un *framework* “para la gestión descentralizada y autónoma de los objetos basado en principios orientados a los servicios, la interacción, la localización y la reputación, inspirados en las tecnologías de los medios sociales”. Hay que señalar que Voutyras et al. no se plantea una arquitectura por capas, sino relacional. No obstante esta idea de definir la abstracción como tal es fundamental y ha sido parte de otras arquitecturas.

Por otra parte los modelos más completos (creo) incluyen las capas de la IoT de una forma u otra y las enriquecen.

Bernal et al. [21] presenta una arquitectura muy detallada que incluye cada componente que es en realidad para un proyecto en concreto llamado SocIoTal. Su complejidad y ajuste al proyecto no la hace viable para generalizar, pero si muestra una anatomía a partir de la cual se pueden separar y agrupar muy bien los distintos componentes o capas de una posible arquitectura más global para el SIoT.

Gulati & Kaur [4] muestran una arquitectura de 4 capas como referencia para el SIoT. Al modelo original del IoT se le agrega una capa de gestión de la socialización que incluye la abstracción de los dispositivos.

Hasta el momento la arquitectura más simple, explicita y completa que he visto es la que propone el propio Malekshahi [1] que consta de 5 capas:

* Aplicación: Aplicaciones de usuario.
* Interacción social: SIoT
* Comunicación: *Stack* de red, *gateways*, ruteo, acceso al medio…
* Abstracción de componentes (Object ID Management (ID Mng), Object Profiling (OP), and Owner Control (OC) [3])
* Entidad: Objeto físico.

Básicamente se incluyen dos capas nuevas al IoT que son las señaladas:

* **Interacción social**: En esta capa se establecen las relaciones entre los objetos y entre los objetos y los humanos. Esta capa es la extensión “social” de:
* **Abstracción de componentes**: Es donde el componente físico se asocia a su par digital y, probablemente lo fundamental para el SIoT, adquiere otros “vectores” de información y responsabilidades y políticas.

Y son precisamente los componentes fundamentales del SIoT que plantean Atzori et al. [22] para la abstracción del objeto en esta capa:

* Gestión de la Identificación ID (*ID* *Management*): Asigna un ID que identifique universalmente todos los objetos.
* Perfil de Objeto OP (*Object Profiling*): Incluye información estática y dinámica sobre el objeto. Está organizada en dos clases, la básica y la principal.
* Control del propietario OC (*Owner Control*): es necesario que el propietario defina políticas específicas para regular cualquier posible operación que realice el objeto (información a compartir, relaciones permitidas, etc.). Para ello, se pueden utilizar diferentes lenguajes de definición de políticas de seguridad y control de acceso ya disponibles.
* Descubrimiento de servicios SD (*Service Discovery*): sustituye a la presencia social y su fin es encontrar qué objetos pueden proporcionar el servicio requerido, de la misma manera que los humanos buscan amistades e información en los SNs. En efecto, para descubrir el servicio, el objeto consulta su red de relaciones sociales.
* Composición de servicios SC (*Service Composition*): permite la interacción entre objetos y sustituye al modelo de participación. El servicio explota las relaciones de los objetos para encontrar el servicio deseado, que luego es activado por este componente. Se puede tener tanto un enfoque reactivo o proactivo para la composición de servicios. Este componente también incluirá la funcionalidad de procesamiento de la información obtenida de diferentes objetos y obtener la respuesta más fiable a una consulta sobre la base de diferentes visiones.
* Gestión de la confianza TM (*Trustworthiness Management*): tiene como objetivo entender cómo la información proporcionada por otros miembros debe ser procesada. La fiabilidad se construye sobre la base del comportamiento del objeto y está estrictamente relacionada el RM (*Relationship Management*). La fiabilidad puede estimarse mediante nociones conocidas del SNs como la centralidad y el prestigio, y, de nuevo, se construye sobre la base de la estructura social del objeto.
* API de Servicios (*Service APIs*): este componente es análogo al requerido en las SNs

Los modelos por capas tienen un problema o limitación (tal vez) en este caso, y es que se espera cuando se observa la estructura que exista un orden entre la solicitud del servicio (por las capas superiores) y la prestación de dicho servicio por las capas inferiores. Entonces, este modelo me crea una posible contradicción entre la abstracción de componentes y la comunicación. El caso es que un componente abstracto se “comunica” con su entidad y a la vez con otros componentes abstractos y a la vez con las aplicaciones de usuarios y como los usuarios a través de las aplicaciones de usuario. Eso quiere decir que la capa de comunicación “no queda” como una capa intermedia entre dos niveles o capas, sino que hay relaciones múltiples entre otras varias capa a través de la capa de comunicación. … en fin… hay que establecer mejor esto. Ahora, lo que si me parece que es correcto y suficiente para describirlo todo son estos 5 componentes o capas aunque después deban ser subdivididos.

Entonces una forma de organizar esta idea en un gráfico es más o menos así:

|  |  |
| --- | --- |
| App | *Networking* |
| Social |
| Abstracción |
| Entidad (dev) |

Figura 1: Arquitectura que..... me parece más lógica y clara

… y difiere de Malekshahi [1] tan solo en la forma de representación.

Ahora, sigue quedando “extraño” algo. No sé aún si la pregunta es correcta o muy ingenua, pero:

*¿Quién y cómo se construyen las aplicaciones a partir la información extendida (de perfil) de otros objetos y sus relaciones?*

El porqué y el para qué parecen bastante claros, mientras más información exista sobre un fenómeno proveniente de múltiples fuentes (incluso si algunas no son lo suficientemente confiables) mejor (y de forma más certera) se podrá manejar el fenómeno y desarrollar mejores aplicaciones en torno a él.

Ahora, en las SNs humanas digamos que la socialización “ocurre” generando información de forma independiente que luego las aplicaciones explotan. Algunas aplicaciones “fuerzan” de hecho la socialización de más información que otras explotan (como las sugerencias de amistades), pero en cualquier caso la socialización en sí no sería un app desde el punto de vista de la explotación de información para la satisfacción de un servicio. Si este planteamiento resulta verás, entonces podemos afirmar (por ahora) que la el modelo representado en la Figura 1 es correcto.

Así algunos autores como Atzori et al. [3] o Dutta et al. [23] proponen observar la arquitectura desde dos ángulos o niveles, desde el cliente y el servidor. A su vez cada uno estos niveles se divide en sub-capas. Esta separación puede ser confusa y considero que no contribuye a la descripción del fenómeno, sin embargo se hace interesante mencionar porque intenta lidiar con el conflicto mencionado en los párrafos anteriores, aunque no veo claro como es que lo resuelve, de hecho, no veo claro si lo resuelve.

En el caso de Dutta et al. [23] la “dimensión” servidor se compone por una capa de red, una de herramientas y componentes y otra de interface, mientras que la cliente incluye las 3 capas más elementales más una abstracción de componentes. Este modelo aparentemente (no me queda muy claro) carga las aplicaciones a la abstracción del objeto, y los mecanismos y políticas de socialización (“*the core functionality of Social Internet of things*”) las sitúa en la capa de servicios en el servidor. Esto intenta dejar clero de quién es “responsable” de las aplicaciones, pero me parece limitado (al menos lo que me permitió comprender el artículo) y con componentes de más.

Atzori et al. [3] expone un modelo más amplio donde cada una de las partes (cliente y servidor) tiene casi las misma responsabilidades (subcapas) de *networking* y aplicación, pero la cliente suma una subcapa de *sensing*. No obstante el cliente se divide en 2, Gateway y Object, y cada una de las subcapas de divide en otras subcapas según sea servidor, Gateway u Object, que muchas son opcionales. En fin, que genera un modelo denso y complejo buscando repartir responsabilidades que pueden o no tener y al final…...

### Conclusión parcial

Me quedo con el modelo planteado por Malekshahi [1] modificado en la Figura 1 y con la duda de que sea correcta la reflexión sobre su veracidad.

Tal vez necesite revisar más literatura del tema.

## SIoT Processes

Pensar en objetos que mantienen relaciones parecidas a las de naturaleza humana para llegar a un propósito, alcanzar un objetivo con cierta autonomía induce a reformular el alcance de al menos dos características, **autonomía** y **propósito** que van más allá de lo que se había considerado para las máquinas. Se debe pensar que los objetos no solo deberán ser capaces de socializar, necesitan también el entorno y los mecanismos donde estos intercambios ocurren.

Nitti et al. [24] plantea que, de acuerdo con el paradigma SIoT, “cada objeto tiene que almacenar y gestionar la información relacionada con las amistades, implementar las funciones de búsqueda y, eventualmente, emplear herramientas adicionales como el módulo de relaciones de confianza para evaluar la fiabilidad de cada amigo.” A esto le debemos sumar que dicho objeto, además, que deberá ser encuestado por sus amigos que buscarán otros servicios que podrán ofrecer otros amigos que deberá “recomendar”. Evidentemente, esto necesita de un poder de respuesta que va mucho más allá de un nodo sensor y potencialmente carga en demasía su representación digital (*Thing*).

Por su parte Saleem et al. [25] propone la explotación del SIoT a partir de un sistema de recomendación que funciona una entidad intermedia entre las aplicaciones y las plataformas de interoperabilidad, en este caso refiere a oneM2M y FIWARE. Pese a que presenta un ejemplo muy elocuente, realmente no queda claro como funciona este esquema de recomendación. Lo que si es relevante observar es como se separa el componente SIoT de los dispositivos.

Ya Atzori et al. [3] presenta un conjunto de actividades que debe realizar un objeto que ingrese a un sistema SIoT:

* Entrada del objeto al sistema.
* Descubrimiento de servicio y su composición
* Establecimiento de relaciones
* Proveer servicios

La Figura 2 representa las principales actividades.

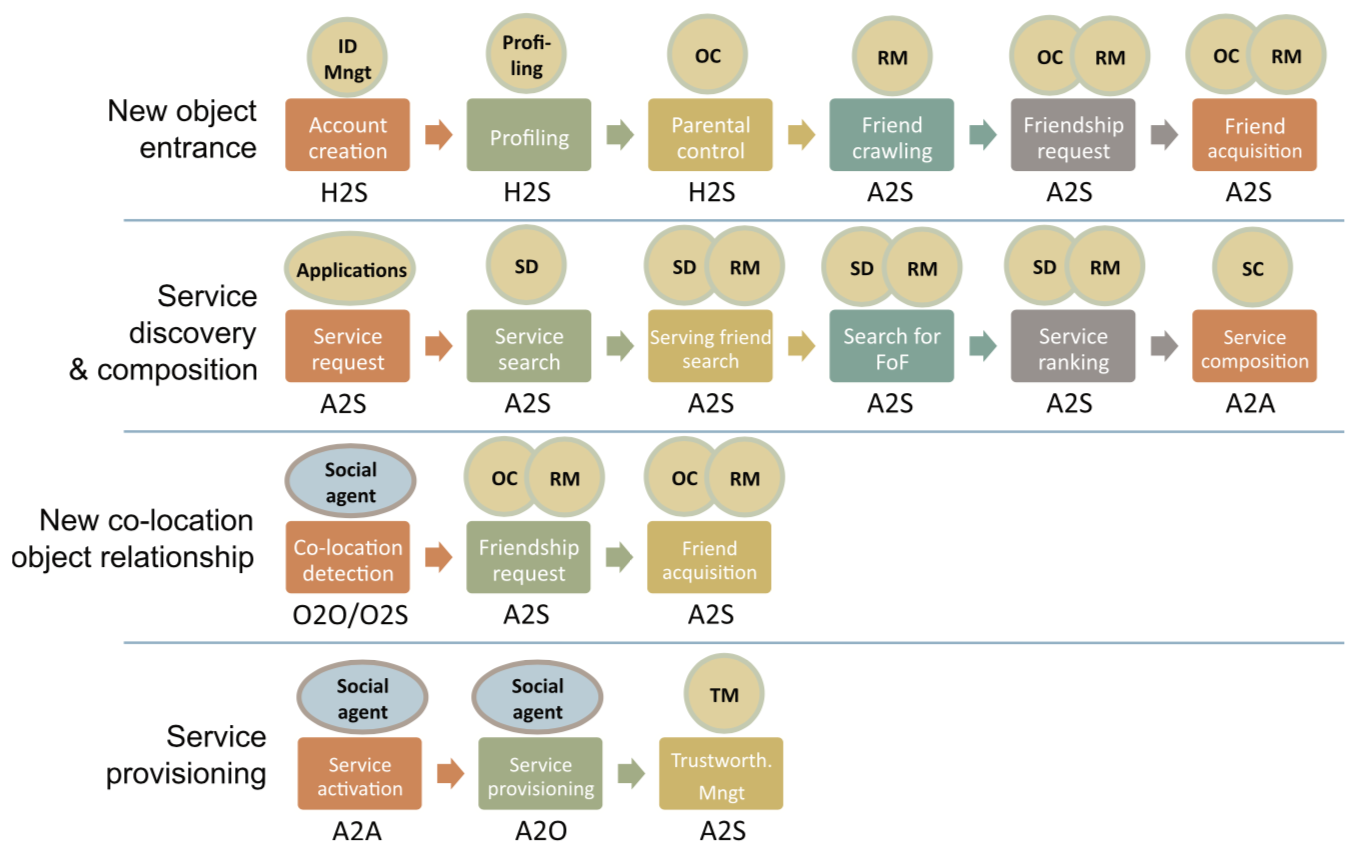


Figura 2: Principales actividades de los objetos en SIoT [3]

## Navegabilidad de la Red

En el modelo de la Web un individuo, desde un punto, persigue un dato o servicio que se encuentra en otro punto remoto. La cultura pop diría que dicho individuo “navega”, siguiendo enlaces de un punto a otro, haciendo una búsqueda de dicha información o servicio hasta encontrarla. Si se parte a través de un directorio o un buscador como Google, dicha búsqueda será centralizada; si se parte de preguntar a amigos, y a sus amigos, o sea, un red social, se tiene una búsqueda descentralizada.

Uno de los objetivos fundamentales de la Web es acortar la distancia entre la entidad que necesita de la información y aquella que la provee. Este fenómeno es abordado por Kleinberg [26] donde especifica que para que una red sea “navegable” esta debe contar con caminos cortos entre todos (o casi todos) los pares de nodos. Luego, en [27] concluiría, partiendo del experimento de Milgram [28], que en una red social existen pistas estructurales que ayudan a los individuos a guiar un mensaje eficientemente hacia su destino sin que estos tengan necesariamente conocimiento de la estructura de la red.

Roopa et al. incluya la navegabilidadcomo una de las *thrust areas* planteándolo como la forma en que los objetos alcanzan un servicio a través, no solo del descubrimiento de servicios, sino explotado la relación con otros objetos. Así propone un mecanismo basado en la estructura social para que el objeto pueda decidir cual será el próximo salto (el próximo enlace) basado en su amistad como objetos de características similares.

Nitti et al. [24] hace un análisis del fenómeno y presenta una evaluación experimental. (……..)

Nitti & Atzory [12] proponen y comparan la estrategia de *caching* (de forma centralizada) con la de selección de amigos para evaluar este parámetro y concluyen que…

## *Relation Management*

La base del SIoT son las relaciones entre los dispositivos. A diferencia de las personas, en las que las relaciones sociales son intrínsecas, los objetos deben incorporar idea de que son las relaciones y un mecanismo para establecerlas y actualizarlas. El *Relation Management* (RM) es el bloque que indica como los objetos seleccionan a sus posibles amigo, inician, actualizan o terminan relaciones.

Primero se sebe establecer cuales son los tipos de relaciones. Byun et al. [30] resume el modelo de forma como se muestra en la Figura 3.

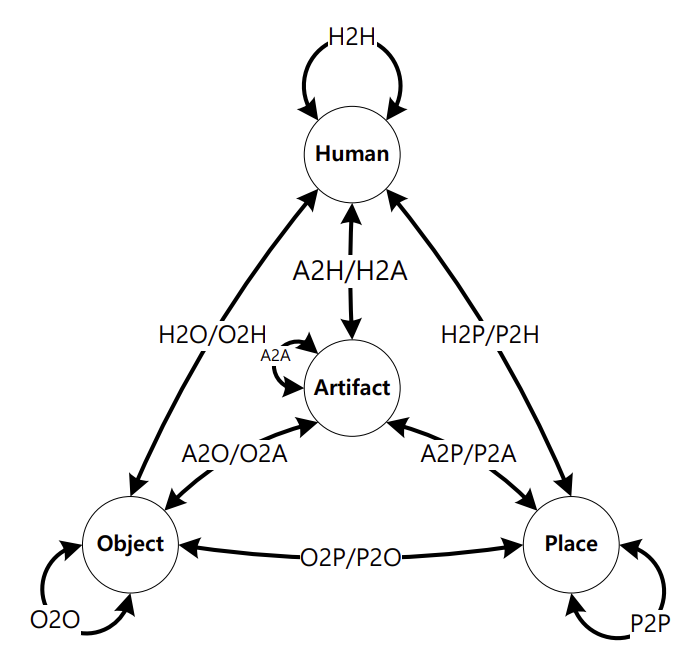


Figura 3: IoT Social Model [30]

Donde:

* *Human:* clase que representa a una persona o un grupo de personas;
* *Objetc:* clase que indica una cosa inteligente cosa inteligente (por ejemplo, aire acondicionado, regulador de intensidad, sensor de temperatura, etc.) o su grupo;
* *Place:* clase que representa un lugar explícito donde las personas y los objetos inteligentes interactúan entre sí.
* *Artifact:* Para Byun et al. [30] todos los contenidos que las personas, los objetos y los lugares producen como *Artifacts*. Esto permite obtener metadatos para invocar servicios en un objeto y obtener un contexto del mundo real con sólo utilizar el gráfico. En consecuencia, esto hace posible inferir un contexto más complejo y controlar los derechos de acceso con sólo utilizar el gráfico.

Todas estas relaciones en las SN se establecen y actualizan en función de las actividades y características de los sujetos (humanos u objetos) que participan. En el caso de las cosas, estas relaciones son explotadas por un módulo de descubrimiento de servicios para localizar los objetos que pueden proporcionar los servicios requeridos, de manera similar a como los humanos buscan información y amistades. [25]

Estas relaciones son muy amplias y genéricas. Hay varios artículos que abordan al RM conceptualmente y más en detalle para la SIoT, pero de una forma u otra parten siempre de los trabajos de Atzori, principalmente [3]. Este básicamente divide las relaciones entre objetos y humanos. Dentro de los objetos observa 5 categorías de Relaciones entre Objetos OR (*Object Relationships*):

* Relación Parental, POR (*Parental OR*): Esta relación se establece entre objetos con un mismo centro de fabricación, por ejemplo una serie en concreto de un fabricante.
* Propiedad sobre OR, OOR (*Owner OR*): Esta relación se establece entre objetos con un propietario específico, como las aplicaciones heterogéneas en el teléfono de alguien.
* Colega OR, C-WOR (*Co-Work OR*): Esta relación se establece entre objetos que trabajan juntos para un propósito específico, incluso en diferentes lugares.
* Social OR (SOR): Esta relación se basa en la relación social que se establece entre los propietarios de los objetos, que se traduce en la colaboración entre sus dispositivos.
* Co-Localizado, CLOR (*Co-Locate OR*): Esta relación se forma entre objetos que se encuentran en un mismo lugar. Pueden ser homogéneos o heterogéneos. *in certain cases, such C-LORs are estab- lished between objects that are unlikely to cooperate with each other to achieve a common goal. Neverthe- less, they are still useful to fill the network with ‘‘short’’ links.*

Atzori aclara que estas relaciones se establecen si la intervención humana, de ahí que Roopa et al. [2] establezca dos categorías principales, las relaciones UO (*User Object*) y las OO (*Object Object*). Además establece divisiones dentro de estas relaciones y las áreas de aplicación de las mismas.

### *Friendship Selection*

Gulati & Kaur [31] proponen una aproximación al RM a través de agentes que manejan las relaciones entre los distintitos actores. Dicho “agentes sociales” los adjunta al objeto a penas este se registra en el sistema.

## Trust Management (TM)

## Semántica en SIoT

Un desafío que está presente en todos los *reviews*, ya sea implícita o explícitamentees la heterogeneidad de los diferentes componentes que forman parte de la IoT. Como condición fundamental para establecer relaciones e intercambiar datos es el entendimiento entre los objetos, así que este desafío deriva en otro, que es garantizar la **interoperabilidad semántica**.

Rahman & Hussain [32] presentan una extensiva al respecto de los varios esfuerzos por lograr la interoperabilidad de los sistemas y divide los en 3 categorías, las **ontologías**, el ***middleware*** y la **web semántica**. El estudio hace notar como en la literatura consultada no se pone el foco en la interoperabilidad semántica entre la heterogeneidad de los datos, señalando distintos problemas y desafíos a los que se enfrentan los distintos modelos presentados.

Cuando una *Thing* interactúa con sus pares o con aplicaciones para buscar un servicio necesitará de anotaciones semánticas (adicionales a los datos) específicas que le indiquen cuales son los amigos de sus amigos, que relación existe entre ellos, que información de perfil tienen entre otras, que le permitan evaluar a la *Thing*,a través de un mecanismo de selección de amigos, si deberá establecer relación con dicho objeto o aplicación y consumir sus servicios.

Atzori et al. [3] refiere la necesidad de las ontologías y los servicios semánticos para la representción de los atributos funcionales y no funcionales. En el modelo que presentan utilizan anotaciones semánticas en las consultas del módulo de descubrimiento, aunque no especifican más al respecto. Resulta interesante notar que en el resto de los trabajos consultados de Atzori, Nitti y Mache apenas se mencione la palabra semántica.

Por su parte Gulati & Kaur se plantean una “perspectiva semántica” [4] del SIoT donde proponen (aparentemente) una ontología basada en OWL que es bastante limitada. Ya en [31] delinean mejor el concepto en el ámbito de IIoT (*Industrial IoT*). Este trabajo está circunscrito al entorno industrial por lo que continúa con limitaciones en entornos más amplios.

Byun et al. [30] propone Lilliput, una plataforma de SIoT basada en ontologías. Lilliput no incluye las relaciones establecidas por Atsori [3] (ver 1.4) ampliamente referidas, pero establece una ontología para describir las relaciones más básicas que puede ser utilizado como punto de partida para trabajos más completos.

…

## Conclusión Parcial

No hay una ontología para SIoT…

Fundamental el RM, la búsqueda y selección de amigos…

Amigos como mecanismo de recomendación….

--- Aquí me detuve para comprender mejor el fenómeno de la cosas, la web y el conocimiento y comparto con usted algunas consideraciones ---

Pensar en cosas que mantienen relaciones parecidas a las de naturaleza humana para llegar a un propósito, alcanzar un objetivo con cierta autonomía. Aquí aparecen dos características, **autonomía** y **propósito** que digamos que van más allá de lo que se había considerado para las máquinas.

Voy a intentar ilustrar con un ejemplo parecido al de María [25], más particular, para poder analizarlo. Supongamos un invernadero (bueno, es en lo que estamos trabajando por acá).

Tratemos al invernadero GH (*GreenHouse*) como una *Thing* que su vez contiene otras *Things* que son los sensores y actuadores. Simplificando, un GH está orientado a hacer crecer y tener frutos de una determinada variedad de plantas. Entonces un GH puede exponer como datos (su TD/perfil) el tipo de planta que crece en él (que a su vez pueden ser *Things* o DT cuyas características las maneja un tercero), características de su suelo, historiales de la gestión ambiental, acciones fitotécnicas y fitosanitarias, progreso del crecimiento, cosecha……. En fin, todos los datos que se pueda recoger del mismo. Como notará algunos datos son generado por los sensores, otros son proveídos por personas.

Ahora, que puede “querer” un GH. Bueno, supongo que cómo tener un mejor rendimiento de lo que sea que esté produciendo. Supongamos que existe una red social de GHs que se comparten toda la información posible lo que está pasando con ellos y sus “experiencias”. Sea experiencia la información vista como conocimiento. Una experiencia puede ser vista como: Seleccionando primero un GH en condiciones similares y con similar ciclo de cultivo, en tal parte del ciclo del cultivo, con tales condiciones ambientales, debo aplicar tal producto, ¿cuánto aplicaste tú en ese momento y cómo te resultó?

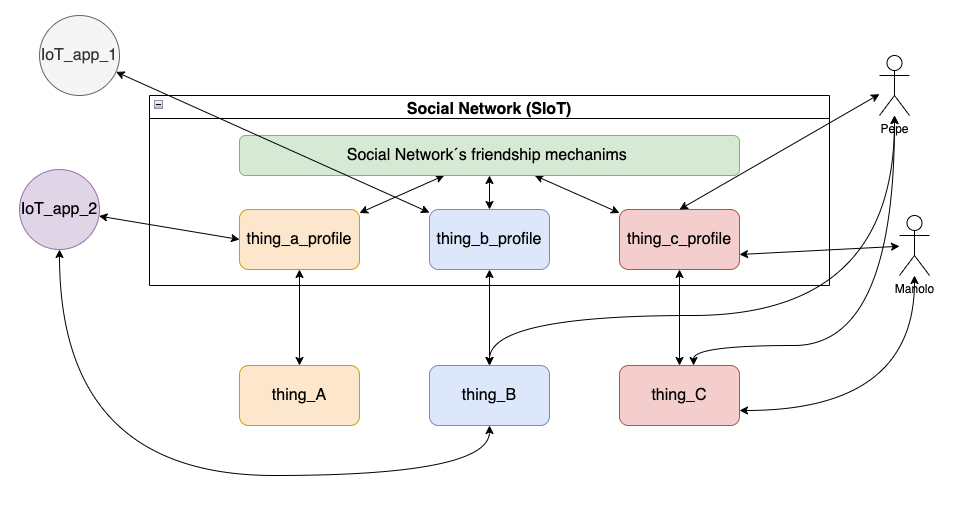
Aquí hay al menos dos escenarios posibles para satisfacer la necesidad del GH (que se traduce o en realizar autónomamente una acción en concreto o sugerirle al productor realizar una acción):

* O consultar a cada uno de los GH periódicamente contrastando “experiencias”.
* O consultar periódicamente a una app “experta” que es quién se dedica a concentrar la información de los GH y aprender cuales son las mejores experiencias.

La cuestión es básicamente quién procesa la información. El modelo puede ser abierto a los dos casos siempre y cuando el poder de cómputo del hospedero de la *Thing* lo permita u otras condiciones. Lo que si es claro que dejarle a la *Thing* el procesamiento de dicha información puede complicar mucho el modelo.

En el caso concreto de Saleem [25] se plantea continuamente que la explotación de las relaciones sociales es a partir del intercambio de los perfiles que son accedidos por las “aplicaciones IoT” que aparentemente son las que explotan dichos perfiles. El SIoT él lo ve (y me parece obvio) como una forma de proveer servicios de recomendación para las aplicaciones de IoT, que a su vez comparten sus datos y conocimientos en forma de servicios para otras aplicaciones mejorando el rendimiento de las aplicaciones.

O sea que la estaría en proveer información hacia la SN desde las *Thing*. Hay que tener en cuenta que a las *Thing* (o en el *WoT Runtime*) deben tener la capacidad de ingresar en la SN (ver 1.3 que describe Atzori) o de ser “descubiertas” por este. El caso es que de alguna manera quedará como usuario de la SN con un perfil que constituye una segunda capa de homogenización semántica. Entonces las apps buscan los perfiles afines a la solución de sus problemas dentro de la SN y “contactan” directamente a la *Thing* en caso que sea necesario. Muy a lo torpe, algo así:



…

Marche et. al. [33] presenta:

* una definición del proceso genérico de búsqueda de servicios en el IoT Social y modelado del comportamiento de los objetos cuando interactúan con otros pares en la red para el intercambio de información y servicios.
* Definición de un modelo de generación de consultas, que es capaz de simular la correlación entre objetos y aplicaciones y representa una herramienta fundamental para probar la interacción entre pares en la red. El modelo propuesto se utiliza a continuación para evaluar las ventajas del enfoque social en términos de navegabilidad global.
* Creación de un conjunto de datos, que no sólo incluye la información y las posiciones de los objetos como se hace en [34], sino también los servicios y aplicaciones que ofrecen y utilizan. Los datos recogidos proceden de los dispositivos instalados en la ciudad de Santander en España y de los datos sobre la movilidad de las personas. Esto se pone a disposición de la comunidad investigadora para probar los algoritmos de gestión de (S)IoT (por ejemplo, gestión de relaciones, búsqueda de servicios, gestión de la confianza), con especial atención a la navegabilidad de la red.

# Representación del conocimiento el la web (Web Semántica)

Independientemente de que puedan existir otras formas de representar el conocimiento, nos vamos limitar exclusivamente a los que ha definido el W3C para la Web Semántica.

En las personas el conocimiento representable (es decir, el universo del discurso) se complementa con el conocimiento no expresable (sensaciones, percepciones, sentimientos no verbalizables, conocimiento inconsciente, conocimiento tácito, etc.). Estos elementos irrepresentables, sin embargo, participan en los procesos de razonamiento y toma de decisiones, que son procesos cognitivos en la gestión del conocimiento.

En la Figura 5 aparece representada una pirámide donde se muestra la jerarquía y las comunicaciones existentes entre las diferentes capas que forman un modelo de información semántico.



## Describiendo los Recursos (RDF)

*Resource Description Framework* (RDF) es un modelo de gráfico destinado a describir formalmente los recursos web y los metadatos para permitir el procesamiento automático de dichas descripciones. Desarrollado por el W3C, RDF es el lenguaje básico de la Web Semántica.

Un documento estructurado en RDF es un conjunto de triples. Un triple RDF es una asociación ( sujeto , predicado , objeto ):

* "sujeto" representa el recurso a describir;
* "predicado" representa un tipo de propiedad aplicable a este recurso;
* "objeto" representa un dato u otro recurso: es el valor de la propiedad.

El sujeto y el objeto, en el caso de los recursos, pueden identificarse mediante un URI o ser nodos en blanco . El predicado está necesariamente identificado por un URI.

Por ejemplo, podemos decir “el cielo es de color azul” conformando una afirmación que se puede traducir en RDF donde cielo es el sujeto, color el predicado y azul el objeto, ya que es valor particular para la propiedad particular color del cielo.

Resumiendo, RDF es la forma de “representar” el conocimiento en un lenguaje controlado.

RDF integra información de manera formal para que una máquina pueda entenderla. El objetivo de RDF es proporcionar un mecanismo de codificación e interpretación para representar recursos para software . En otras palabras, para que el software pueda acceder y utilizar información que de otro modo no podría utilizarse.

No obstante RDF no es ni define una sintaxis, formato o un dialecto XML. RDF es simplemente una estructura de datos formada por nodos y organizada en un gráfico, considerado como un Diagrama entidad relación….

Serialización RDF: -🡪

#### RDF/XML

RDF/XML es una sintaxis , definida por el W3C , para expresar (es decir, serializar ) un gráfico RDF como un documento XML. [35] RDF / XML a veces se denomina engañosamente simplemente RDF porque se introdujo entre las otras especificaciones W3C que definen RDF y fue históricamente el primer formato estándar para la serialización W3C RDF.

#### RDF/JSON

<https://www.w3.org/TR/rdf-json/>

#### JSON-LD

<https://www.w3.org/TR/json-ld/>

JSON-LD, o *JavaScript Object Notation for Linked Data*

JSON-LD está diseñado en torno al concepto de "contexto" para proporcionar asociaciones de JSON a un modelo RDF . Este contexto le permite vincular propiedades de objetos en un documento JSON a conceptos en una ontología.

Para vincular la sintaxis JSON-LD a RDF, JSON-LD permite convertir los valores a un tipo específico o etiquetarlos con un lenguaje. Un contexto puede incrustarse directamente en un documento JSON-LD o colocarse en un archivo separado y referenciarse desde diferentes documentos (desde documentos JSON tradicionales a través de un enlace de encabezado HTTP).

{

"@context": {

**"name"**: "http://xmlns.com/foaf/0.1/name",

**"homepage"**: {

"@id": "http://xmlns.com/foaf/0.1/workplaceHomepage",

"@type": "@id"

},

**"Person"**: "http://xmlns.com/foaf/0.1/Person"

},

"@id": "http://me.example.com",

"@type": "Person",

**"name"**: "John Smith",

**"homepage"**: "http://www.example.com/"

}

### SPARQL

SPARQL (pronunciado *sparkle*) es un lenguaje y protocolo de consulta que le permite buscar, agregar, modificar o eliminar datos RDF disponibles en Internet . Su nombre es un acrónimo recursivo que significa **S**PARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage .

SPARQL fue considerado ya en 2007 como una de las tecnologías clave de la Web Semántica por Tim Berners-Lee, el inventor de la Web Semántica, quien explica "Intentar utilizar la Web Semántica sin SPARQL equivale a explotar una base de datos relacional sin SQL ".

### Representando a una Cosa, el *Thing Description*

Un *Thing Description* o TD…

*Interations Affordances* Capacidades de Interacción

## Representación del Conocimiento en la Web

un requisito para un recurso RDF (como un sujeto o un predicado) es que sea único. Los recursos presentados deben ser únicos para permitir la identificación exacta de los recursos descritos. El predicado debe ser único para reducir las posibilidades de confundir la noción de título o editor con el software. Si el software reconoce http://purl.org/dc/elements/1.1/title (una definición del concepto de título establecido por la Dublin Core Metadata Initiative), también debe tener en cuenta que este título es diferente de un título de propiedad. o un título honorífico o simplemente las letras del título juntas

Una persona puede formular una afirmación como “el cielo es azul”, o “color cielo”, donde por motivos aparentemente lógicos se puede omitir “color” y “cielo” respectivamente, esto no es tan obvio para un sistema. Por tanto, las actuaciones cognitivas de un agente informático se basarán en parte en el campo de las representaciones al que tendrá acceso, es decir, concretamente en el campo de las representaciones que se habrán formalizado.

## Ontologías

En informática, una ontología es precisamente la herramienta que permite representar un cuerpo de conocimiento [36] en una forma que pueda ser utilizada por una computadora.

Una ontología define un conjunto de primitivas de representación con las que modelar un dominio de conocimiento o discurso. Las primitivas de representación de primitivas de representación suelen ser clases (o conjuntos), atributos (o propiedades) y relaciones (o relaciones entre los miembros de la clase). Las definiciones de las primitivas de representación incluyen información sobre su significado y restricciones sobre su aplicación lógicamente coherente. [37]

En *stack* de los estándares de la Web Semántica, las ontologías se denominan como una capa explícita. En concreto, la ontología SSN (*Semantic Sensor Network*) [38] es el pivote de las demás ontologías descritas para el WoT [39], las cuales, según [40], se han expresado como especializaciones de la misma.

En la práctica, la interoperabilidad semántica en el modelo de TD se traduce en mecanismos de anotación como documentos TD (instancias serializadas del TD en JSON-LD) asociadas a términos RDF (*Resources Description Framework*) definidos en un vocabulario específico del dominio [39], por lo que el modelo TD ha sido diseñado encima de una ontología Web, llamada simplemente ontología TD [41]. Dichas ontologías han sido formalmente alineadas, según [41], con otros vocabularios WoT SOSA [42] y Schema.org [43]. Además, [39] describe una alineación formal con SSN [38].

### Ontología TD

La ontología TD es una axiomatización idiomática RDF del modelo de información TD, que puede utilizarse para describir las *Things* en WoT y sus capacidades de interacción. La ontología TD importa la ontología de controles hipermedia [44].

Aunque el modelo de información TD también incluye términos para los esquemas de datos y las configuraciones de seguridad, la ontología TD no obliga a utilizar el esquema JSON y los vocabularios de seguridad WoT correspondientes. Por lo tanto, se pueden aprovechar otros lenguajes de esquemas como SHACL para describir WoT *Things* [41].

### SSN y SOSA

Las ontologías SSN (*Semantic Sensor Network*) [45] y SOSA (*Sensor, Observation, Sample, and Actuator*) [42] se han propuesto ofrecer perspectivas flexibles pero coherentes para representar las entidades, relaciones y actividades que intervienen en la detección, el muestreo y la actuación. [38]

SOSA proporciona un núcleo ligero para SSN y pretende ampliar el público objetivo y las áreas de aplicación que pueden hacer uso de las ontologías de la Web Semántica. Al mismo tiempo, SOSA actúa como un nivel mínimo de interoperabilidad, es decir, define aquellas clases y propiedades comunes para las que se pueden intercambiar datos de forma segura en todos los usos de SSN, sus módulos y SOSA.

En este sentido ambas ontologías aparecen bajo la misma recomendación del W3C [38] formulando una ontología compuesta por varios módulos que utilizan declaraciones *owl:import*, donde se distingen dos métodos en función de la direccionalidad de la segmentación: una segmentación vertical y una segmentación horizontal. La Figura 3 muestra en un esquema como se unen todas las piezas para SSN y SOSA a través de OWL.

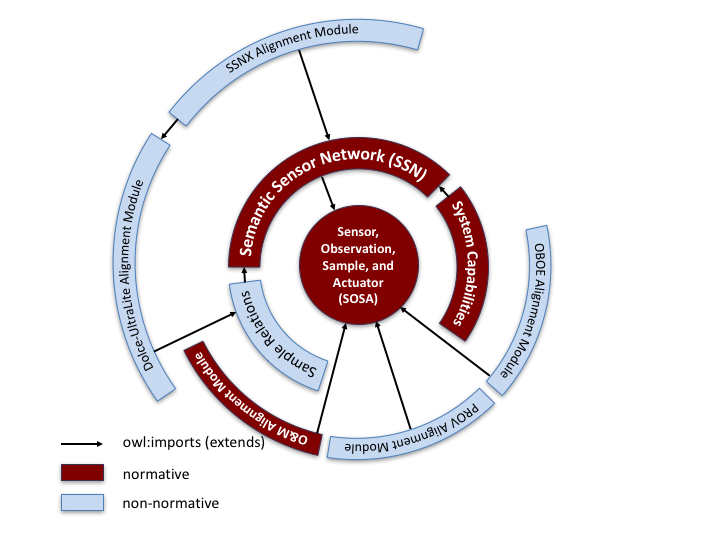


Figura 4: Las ontologías SOSA y SSN y sus módulos vertical y horizontal [38]

## Esquemas

¿?

### RDF *Schema*

RDFS proporciona elementos básicos para la definición de ontologías o vocabularios destinados a estructurar recursos RDF , pudiendo estos últimos ser almacenados en bases de datos relacionales denominadas *triplestores* y manejados mediante consultas en lenguaje SPARQL.

|  |  |  |
| --- | --- | --- |
| Un perro es un animal | ex:dog1 rdf:type ex:animal. | Esquema básico de vinculación |
| cat1 es un gato | ex:cat1 rdf:type ex:cat. |
| Los gatos son animales | ex:cat rdfs:subClassOf ex:animal. |
| Los animales viven en el zoo | zoo:host rdfs:range ex:animal. |
| El zoo1 alberga a el cat2 | ex:zoo1 zoo:host ex:cat2. |

### OWL

*Web Ontology Language* (OWL) es un lenguaje de representación del conocimiento construido sobre el modelo de datos RDF. Proporciona los medios para definir ontologías web estructuradas. Su segunda versión se convirtió en una recomendación del W3C a final de 2012.

## La Semántica en Web de las Cosas

La Figura 4 presenta la tendencia de la IoT mientras se construyen las tecnologías de la web semántica. En primer lugar, se establece la interconexión entre las cosas del mundo real con Internet. Después de establecer la conexión, hay que ocuparse de los problemas de heterogeneidad. Para ello, se puede utilizar un protocolo común como CoAP, MQTT o HTTP para la transferencia de datos. El siguiente paso es conectar las cosas físicas a la Web a través de Internet, en la WoT.

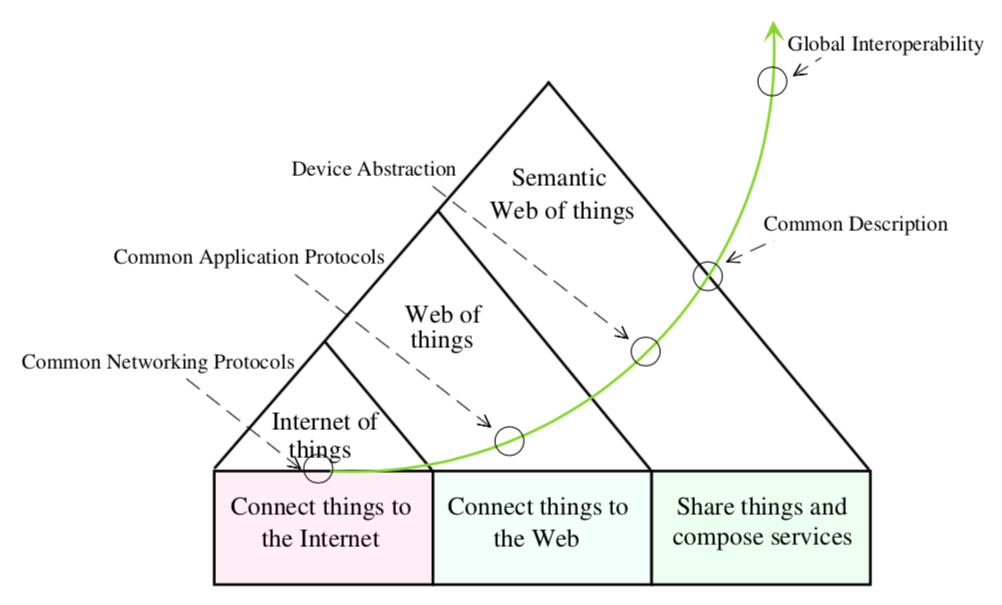


Figura 5: La tendencia muestra que el IoT va incluso más allá de la WoT en su búsqueda de la interoperabilidad [46]

### La Web de las Cosas

## El Descubrimiento

Una aplicación tendrá una necesidad que será satisfecha de alguna manera por un servicio. El desafío es cómo dicha aplicación encuentra **autónomamente** el servicio que satisface su necesidad.

El descubrimiento de servicios se basa en:

* un punto de entrada a una entidad (.well\_know/core) y un sistema de enlaces (LD) a partir de los cuales se exponen los servicios. Estos enlaces pueden llevar a otras entidades que exponen su vez sus servicios.
* Un sistema de etiquetados a modo de sintaxis (serialización xml/json). La debe poder identificar en el documento qué es un dato y donde está.
* Un semántica común entre las entidades. El app debe poder identificar la variable que necesita sin ambigüedades y no otra con un significado distinto.

Supongamos el escenario que el app tenga una “lista” de servicios, y solo esa lista. El app solo consumirá dichos servicios.

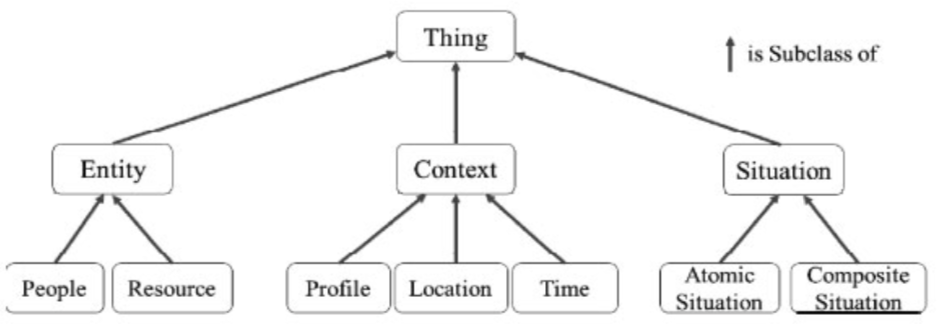
Supongamos el escenario donde el app tenga una lista de entidades. El app encuestará a cada una de las entidades con las que tenga compatibilidad semántica por los servicios y luego, si encuentra el que le satisface, lo consume.

Supongamos el escenario anterior donde las entidades contengan además enlaces a otras entidades con servicios similares y la semántica compartida le permita a la app entender, cachear las nuevas entidades y encuestarlas en busca del servicio. Aquí el proceso de búsqueda se expande y termina cuando ya no hayan más entidades, o cuando se llega a un límite, o cuando se encuentra el servicio que satisface el requerimiento.

Supongamos un escenario donde las entidades tengan un usuario, un perfil y un grupo de amigos en una red social. Un app que necesite un servicio (que a su vez tiene un usuario, un perfil y una serie de amigo en la misma red social) empiece a buscarlo revisando entre los perfiles de sus amigos, para luego seguir a partir los amigos de mis amigos y así…

## ¿¿Anotaciones??

Lo siguiente es una estructura ontológica simple (o taxonómica, aún no se) leida en tesis master Alexandra cuerva y dice que la sacó de *J. Singh, N. Hassanzadech, S. Rea y D. Pesch, «Semantics-Empowered Middleware Implementation for Home Ecosystem Gateway,» 11th EEE International Workshop on Managing Ubiquitous Communications and Services, 2014*.



Parece que puede servir como punto de partida para un caso de estudio o+++

----

Estoy encontrando algo creepy aquí (bueno, y en lo que he leído hasta aquí) y es que todo está muy sentado en la base de seguir intereses comunes, objetos con características similares ++, entonces me hace preguntar (claro, debo entender mejor los planteamientos que hacen) en que si en vez de buscar la “verdad” a partir información de “amigos” con “intereses comunes” no estamos reforzando errores. No se si me hago entender, pongo un ejemplo: Yo siempre he evitado hablar en otro idioma (inglés o portugués en mi caso) con personas que no son nativas o muy bien preparadas porque he llegado a la conclusión que cuando haces eso, y comentes un error, y no te lo corrigen, lo que haces es reforzarlo por ignorancia. Aislándose en círculos de intereses comunes hoy hay mucha gente que con toda la información del mundo cree que la tierra es plana. Este fenómeno podría pasar con los objetos si no se tiene en cuenta…… creo….

# Ambientes RESTful con Altas Restricciones (CoRE)

El modelo arquitectónico *Representational State Transfer* (REST) ha dominado el escenario de Internet [47] en su forma más o menos “pura”, RESTful, así como la forma en que se crean las relaciones y se apunta hacia los recursos utilizado enlaces (*Web Links*).

Los ambientes con altas restricciones como los construidos sobre nodos (ej. de MCU de 8bits) y redes (como la 6LowPAN [48, p. 6]) de recursos restringidos tienen múltiples características [49] de los mismos que los hacen diferentes a los web tradicionales. Algunas características que los representa son:

* Limitado poder de computo: Donde muchas funciones no pueden ser llevadas a cabo. Se trata de obtener los datos y procesarlos en otra parte.
* Tamaños limitados de *payload*: Cada bit transmitido es un costo energético y en memoria por lo que los *payloads* suelen ser muy pequeños. Esto restringe significativamente toda la metainformación que requiera cualquier protocolo.
* *Dutty cycles*: Los dispositivos no trabajan continuamente sino a ciclos. Esto implica retardos y la posibilidad de manejo de muy pocas conexiones simultáneas (a veces solo una).
* Redes muy complejas y con pérdidas: Con el fin de ahorrar potencia los dispositivos transmiten a distancias cortas. Esto los fuerza a cooperar formando mallas que pueden ser muy complejas. Esto enlentece más aun la comunicación y aumenta la probabilidad de pérdidas.

Se puede deducir que se debe intentar acceder lo menos posible a dichos dispositivos. CoRE (*Constrained RESTful Environments*) describe el modelo arquitectónico REST para dichos ambientes que están pensados principalmente para comunicaciones M2M.

## *CoRE Link Format*

CoRE *Link Format* descrito en la RFC 6690 [50] estandariza los enlace web para ambientes restringidos, teniendo en cuenta sus características. Una URI (*Universal Resource Identifier*) bien conocida (*“/.well-known/core*”) es definida como el punto de entrada por defecto (mecanismo de descubrimientos en CoRE) para hacer las solicitudes sobre los otros enlaces. Esta especificación es válida para CoAP, http u otro protocolo de transferencia Web.

En HTTP el *Link Header*  puede ser utilizado para llevar la información de los enlaces del recurso en la respuesta. En CoRE, debido a que su principal caso de uso para los enlaces es el descubrimiento de recursos, los enlaces van en el *payload* (no en los *headers*). Por ello, la serialización del Formato de Enlace CoRE (*CoRE Link Format*) se lleva como una representación de recursos de una URI bien conocida. El formato de enlace CoRE reutiliza el formato de la serialización *HTTP Link Header* definido en la RFC5988 [50].

# Bibliografía

[1] M. Malekshahi Rad, A. M. Rahmani, A. Sahafi, and N. Nasih Qader, “Social Internet of Things: vision, challenges, and trends,” *Hum.-Centric Comput. Inf. Sci.*, vol. 10, no. 1, p. 52, Dec. 2020, doi: 10.1186/s13673-020-00254-6.

[2] R. M.s., S. Pattar, R. Buyya, V. K.r., S. S. Iyengar, and L. M. Patnaik, “Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions,” *Comput. Commun.*, vol. 139, pp. 32–57, May 2019, doi: 10.1016/j.comcom.2019.03.009.

[3] L. Atzori, A. Iera, G. Morabito, and M. Nitti, “The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization,” *Comput. Netw.*, vol. 56, no. 16, pp. 3594–3608, Nov. 2012, doi: 10.1016/j.comnet.2012.07.010.

[4] N. Gulati and P. D. Kaur, “When Things Become Friends: A Semantic Perspective on the Social Internet of Things,” in *Smart Innovations in Communication and Computational Sciences*, Singapore, 2019, pp. 149–159. doi: 10.1007/978-981-10-8971-8\_15.

[5] S.-E. Lee, M. Choi, and S. Kim, “How and what to study about IoT: Research trends and future directions from the perspective of social science,” *Telecommun. Policy*, vol. 41, no. 10, pp. 1056–1067, Nov. 2017, doi: 10.1016/j.telpol.2017.09.007.

[6] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, “Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts,” 2001, pp. 116–122.

[7] T. O’Reilly, “What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software,” Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 1008839, Aug. 2007. Accessed: Feb. 24, 2022. [Online]. Available: https://papers.ssrn.com/abstract=1008839

[8] “Tim O’Reilly: «El concepto web 2.0 está obsoleto»,” *abc*, Nov. 22, 2011. https://www.abc.es/tecnologia/abci-ficod-201111220000\_noticia.html (accessed Feb. 24, 2022).

[9] D. Hussein, S. N. Han, G. M. Lee, N. Crespi, and E. Bertin, “Towards a dynamic discovery of smart services in the social internet of things,” *Comput. Electr. Eng.*, vol. 58, pp. 429–443, Feb. 2017, doi: 10.1016/j.compeleceng.2016.12.008.

[10] R. Ma, K. Wang, T. Qiu, A. K. Sangaiah, D. Lin, and H. B. Liaqat, “Feature-based Compositing Memory Networks for Aspect-based Sentiment Classification in Social Internet of Things,” *Future Gener. Comput. Syst.*, vol. 92, pp. 879–888, Mar. 2019, doi: 10.1016/j.future.2017.11.036.

[11] S. Rho and Y. Chen, “Social Internet of Things: Applications, architectures and protocols,” *Future Gener. Comput. Syst.*, vol. 82, pp. 667–668, May 2018, doi: 10.1016/j.future.2018.01.035.

[12] W. Mardini, Y. Khamayseh, and M. H. Khatatbeh, “Genetic algorithm for friendship selection in social IoT,” in *2017 International Conference on Engineering MIS (ICEMIS)*, May 2017, pp. 1–4. doi: 10.1109/ICEMIS.2017.8273022.

[13] M. Nitti, L. Atzori, and I. P. Cvijikj, “Friendship Selection in the Social Internet of Things: Challenges and Possible Strategies,” *IEEE Internet Things J.*, vol. 2, no. 3, pp. 240–247, Jun. 2015, doi: 10.1109/JIOT.2014.2384734.

[14] A. Rehman, A. Paul, A. U. Rehman, F. Amin, R. M. Asif, and N. Rahmatov, “An Efficient Friendship Selection Mechanism for an Individual’s Small World in Social Internet of Things,” in *2020 International Conference on Engineering and Emerging Technologies (ICEET)*, Feb. 2020, pp. 1–6. doi: 10.1109/ICEET48479.2020.9048234.

[15] B. Farhadi, A. Masoud Rahmani, P. Asghari, and M. Hosseinzadeh, “Friendship selection and management in social internet of things: A systematic review,” *Comput. Netw.*, vol. 201, p. 108568, Dec. 2021, doi: 10.1016/j.comnet.2021.108568.

[16] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017, doi: 10.1109/JIOT.2017.2683200.

[17] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du, “Research on the architecture of Internet of Things,” in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, Aug. 2010, vol. 5, pp. V5-484-V5-487. doi: 10.1109/ICACTE.2010.5579493.

[18] A. M. Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, “The Cluster Between Internet of Things and Social Networks: Review and Research Challenges,” *IEEE Internet Things J.*, vol. 1, no. 3, pp. 206–215, Jun. 2014, doi: 10.1109/JIOT.2014.2318835.

[19] O. Voutyras, P. Bourelos, S. Gogouvitis, D. Kyriazis, and T. Varvarigou, “Social monitoring and social analysis in internet of things virtual networks,” in *2015 18th International Conference on Intelligence in Next Generation Networks*, Feb. 2015, pp. 244–251. doi: 10.1109/ICIN.2015.7073838.

[20] “iot-cosmos | Cultivate resilient smart Objects for Sustainable city applicatiOnS.” http://iot-cosmos.eu/ (accessed Feb. 18, 2022).

[21] J. Bernal Bernabe *et al.*, “SocIoTal — The development and architecture of a social IoT framework,” in *2017 Global Internet of Things Summit (GIoTS)*, Jun. 2017, pp. 1–6. doi: 10.1109/GIOTS.2017.8016286.

[22] L. Atzori, A. Iera, and G. Morabito, “SIoT: Giving a Social Structure to the Internet of Things,” *IEEE Commun. Lett.*, vol. 15, no. 11, pp. 1193–1195, Nov. 2011, doi: 10.1109/LCOMM.2011.090911.111340.

[23] D. Dutta, C. Tazivazvino, S. Das, and B. K. Tripathy, *Social Internet of Things (SIoT): Transforming smart object to social object*. 2015.

[24] M. Nitti, L. Atzori, and I. P. Cvijikj, “Network navigability in the social Internet of Things,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Mar. 2014, pp. 405–410. doi: 10.1109/WF-IoT.2014.6803200.

[25] Y. Saleem, N. Crespi, M. H. Rehmani, R. Copeland, D. Hussein, and E. Bertin, “Exploitation of social IoT for recommendation services,” in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, VA, USA, Dec. 2016, pp. 359–364. doi: 10.1109/WF-IoT.2016.7845500.

[26] J. Kleinberg, “Small-world phenomena and the dynamics of information,” in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, Cambridge, MA, USA, Jan. 2001, pp. 431–438.

[27] J. Kleinberg, “Navigation in a small world,” *Nature*, 2000, doi: 10.1038/35022643.

[28] J. B. Travers and S. Milgram, “An experimental study of the small world problem,” *Sociometry*, vol. 32, no. 4, pp. 225–443, 1969.

[29] Michele Nitti and L. Atzori, *What the SIoT needs: A new caching system or new friendship selection mechanism?* 2015. doi: 10.1109/WF-IoT.2015.7389092.

[30] J. Byun, S. H. Kim, and D. Kim, “Lilliput: Ontology-Based Platform for IoT Social Networks,” in *2014 IEEE International Conference on Services Computing*, Jun. 2014, pp. 139–146. doi: 10.1109/SCC.2014.27.

[31] N. Gulati and P. D. Kaur, “Towards socially enabled internet of industrial things: Architecture, semantic model and relationship management,” *Ad Hoc Netw.*, vol. 91, p. 101869, Aug. 2019, doi: 10.1016/j.adhoc.2019.101869.

[32] H. Rahman and M. I. Hussain, “A comprehensive survey on semantic interoperability for Internet of Things: State-of-the-art and research challenges,” *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 12, p. e3902, 2020, doi: 10.1002/ett.3902.

[33] C. Marche, L. Atzori, V. Pilloni, and M. Nitti, “How to exploit the Social Internet of Things: Query Generation Model and Device Profiles’ Dataset,” *Comput. Netw.*, vol. 174, p. 107248, Jun. 2020, doi: 10.1016/j.comnet.2020.107248.

[34] C. Marche, L. Atzori, and M. Nitti, “A Dataset for Performance Analysis of the Social Internet of Things,” in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2018, pp. 1–5. doi: 10.1109/PIMRC.2018.8580830.

[35] F. Gandon, G. Schreiber, and D. Beckett, “RDF 1.1 XML Syntax.” https://www.w3.org/TR/rdf-syntax-grammar/ (accessed Mar. 25, 2022).

[36] N. Guarino, “Formal ontology, conceptual analysis and knowledge representation,” *Int. J. Hum.-Comput. Stud.*, vol. 43, no. 5, pp. 625–640, Nov. 1995, doi: 10.1006/ijhc.1995.1066.

[37] T. Gruber, “Ontology,” in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 1963–1965. doi: 10.1007/978-0-387-39940-9\_1318.

[38] A. Haller, K. Janowicz, S. Cox, D. L. Phuoc, K. Taylor, and M. Lefrançois, “Semantic Sensor Network Ontology,” Dec. 08, 2017. https://www.w3.org/TR/vocab-ssn/ (accessed Jun. 21, 2021).

[39] V. Charpenay and S. Käbisch, “On Modeling the Physical World as a Collection of Things: The W3C Thing Description Ontology,” in *The Semantic Web*, Cham, 2020, pp. 599–615. doi: 10.1007/978-3-030-49461-2\_35.

[40] V. Charpenay, S. Käbisch, and H. Kosch, “Introducing Thing Descriptions and Interactions: An Ontology for the Web of Things,” 2016.

[41] Victor Charpenay, Maxime Lefrançois, María Poveda Villalón, and Sebastian Käbisch, “Thing Description (TD) Ontology,” Mar. 11, 2022. https://www.w3.org/2019/wot/td (accessed Jul. 02, 2021).

[42] K. Janowicz, A. Haller, S. J. D. Cox, D. Le Phuoc, and M. Lefrançois, “SOSA: A lightweight ontology for sensors, observations, samples, and actuators,” *J. Web Semant.*, vol. 56, pp. 1–10, May 2019, doi: 10.1016/j.websem.2018.06.003.

[43] “Schema.org - Schema.org.” https://schema.org/ (accessed Mar. 26, 2022).

[44] V. Charpenay and M. Kovatsch, “Hypermedia Controls Ontology,” Mar. 11, 2022. https://www.w3.org/2019/wot/hypermedia (accessed Mar. 26, 2022).

[45] M. Compton *et al.*, “The SSN ontology of the W3C semantic sensor network incubator group,” *J. Web Semant.*, vol. 17, pp. 25–32, Dec. 2012, doi: 10.1016/j.websem.2012.05.003.

[46] A. Jara, A. C. Olivieri, Y. Bocchi, M. Jung, W. Kastner, and A. Gómez-Skarmeta, “Semantic Web of Things: an analysis of the application semantics for the IoT moving towards the IoT convergence,” *Int J Web Grid Serv*, 2014, doi: 10.1504/IJWGS.2014.060260.

[47] M. Amundsen, S. Ruby, and L. Richardson, “RESTful Web APIs [Book],” Sep. 2013. https://www.oreilly.com/library/view/restful-web-apis/9781449359713/ (accessed Jun. 23, 2021).

[48] G. Montenegro, C. Schumacher, and N. Kushalnagar, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals,” Internet Engineering Task Force, Request for Comments RFC 4919, Aug. 2007. doi: 10.17487/RFC4919.

[49] C. Bormann, M. Ersue, and A. Keranen, “Terminology for Constrained-Node Networks,” Internet Engineering Task Force (IETF), Request for Comments, May 2014. Accessed: Nov. 10, 2021. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7228

[50] Z. Shelby, “Constrained RESTful Environments (CoRE) Link Format,” Internet Engineering Task Force, Request for Comments RFC 6690, Aug. 2012. doi: 10.17487/RFC6690.