

MIDDLEWARE OF INTERNET OF THINGS

Tugas Mata Kuliah Kapita Selekt

Dosen Pengampu : Deris Stiawan, M.T., Ph.D.



Oleh:

Mardiah 09011281320005

Program Studi Sistem Komputer

Fakultas Ilmu Komputer

Universitas Sriwijaya

2016

A. Service-Oriented middlewares

Paradigma desain berorientasi membangun layanan software atau aplikasi dalam bentuk jasa. Service Oriented Computing (SOC) berdasarkan Service-Oriented Architecture (SOA) pendekatan dan telah digunakan secara tradisional dalam sistem IT perusahaan. Karakteristik SOC, seperti netralitas teknologi, loose coupling, service usability, service composability, service discoverability [92], juga berpotensi bermanfaat untuk aplikasi IOT. Namun, ultra jaringan IOT ini skala besar, perangkat sumber daya terbatas, dan mobilitas karakteristik membuat service discovery dan komposisi menantang.

Sebuah service oriented middleware (SOM) memiliki potensi untuk mengurangi tantangan ini melalui penyediaan fungsi yang sesuai (seperti ditunjukkan pada Gambar. 6) untuk deploying, penerbitan / Menemukan dan mengakses layanan pada saat runtime. SOM juga menyediakan dukungan untuk komposisi layanan adaptif ketika layanan tidak tersedia. Sejumlah besar layanan-middlewares IOT berorientasi tersedia. Middlewares ini dapat dikategorikan sebagai standalone SOM untuk IOT [93] - [97] atau middleware layanan yang disediakan oleh awan platform komputasi sebagai layanan (PaaS) Model [98] - [100]

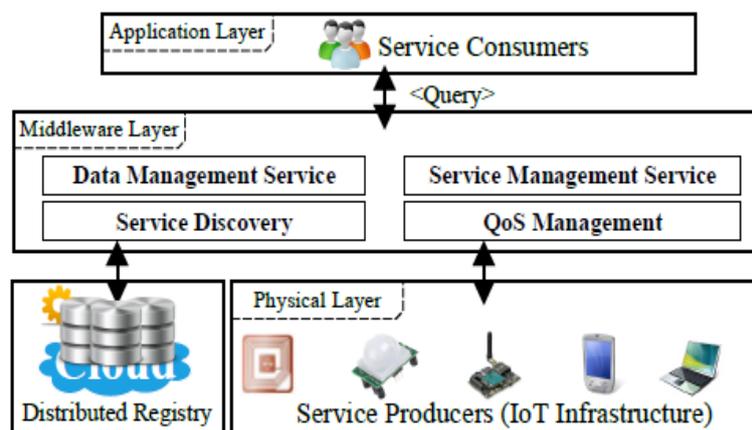
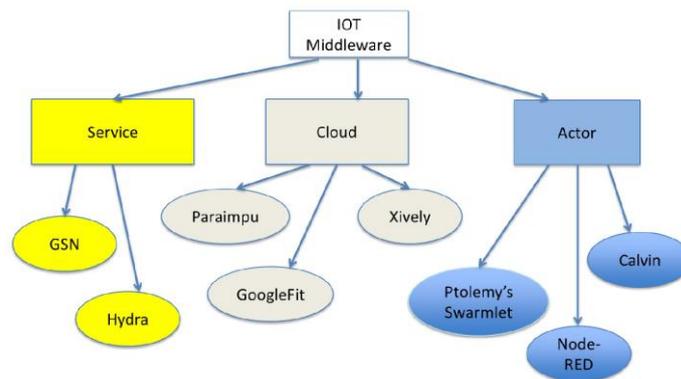


Fig. 6. General design model for a Service-Oriented Middleware.

Hydra [101], yang saat ini dikenal sebagai LinkSmart [102], adalah middleware untuk kecerdasan ambien (AMI) layanan dan sistem. Hal ini dibangun pada SOA dan model-driven arsitektur. Arsitekturnya terdiri dari sejumlah komponen manajemen, termasuk manajer layanan, manajer event, device manager, manager storage, manajer konteks, dan manajer keamanan. Komponen-komponen ini dikelompokkan ke dalam aplikasi dan perangkat elemen, yang masing-masing memiliki lapisan semantik, lapisan layanan, lapisan jaringan, dan lapisan keamanan. Hydra menyediakan interoperabilitas tingkat sintaksis dan semantik menggunakan layanan web semantik. Selain sejumlah persyaratan fungsional (misalnya, manajemen data, manajemen acara, manajemen sumber daya), mendukung konfigurasi ulang dinamis dan konfigurasi diri. Manajer sumber daya, perangkat, dan kebijakan Hydra membuatnya ringan dengan mengoptimalkan konsumsi energi di perangkat sumber daya terbatas. Keamanan didistribusikan dan kepercayaan sosial komponen menawarkan komunikasi yang aman dan terpercaya dalam perangkat. Solusi keamanan dan privasi menggunakan virtualisasi dan pelaksanaan mekanisme berbasis WS

diperkaya dengan resolusi semantik [101]. Namun, virtualisasi yang dapat memperkenalkan masalah keamanan (misalnya, serangan sisi saluran). Juga, semantik keamanan dan interoperabilitas solusi berbasis ontologi cenderung tidak cocok di IOT karena, saat ini, tidak ada ontologi standar untuk skala besar yang ultra IOT.

Pada bagian ini, kami menyediakan lebih dalam analisis mendalam dari kemampuan semua tiga jenis IOT middleware dalam hal i) abstraksi mereka menyediakan untuk menghubungkan dan mengakses perangkat fisik, ii) layanan yang mereka berikan untuk komposisi fleksibel perangkat dan layanan IOT, iii) dukungan untuk penemuan layanan, iv) penanganan privasi dan keamanan data yang dikumpulkan, dan v) dukungan yang mereka berikan untuk menciptakan dan penyediaan aplikasi IOT. Gambar 5 memberikan gambaran dari berbagai IOT middleware yang kita pelajari. Banyak IOT sistem, kerangka, atau middleware yang terus dikembangkan. Cakupan lengkap dari semua dari mereka adalah mustahil. Kami bertujuan untuk survei setidaknya dua sistem IOT middleware di masing-masing gaya arsitektur, dan membatasi penelitian kami untuk sistem middleware IOT yang dapat mendukung pengumpulan data dan analisis aplikasi IOT.



Gambar. 5. Ikhtisar Berbagai IOT Middleware

Hydra: The Hydra sistem [22], [23] adalah proyek 4 tahun yang didanai oleh Uni Eropa untuk mengembangkan middleware berorientasi layanan untuk embedded system jaringan. Proyek ini telah diubah namanya menjadi LinkSmart sejak 2014. Layanan Web disediakan sebagai abstraksi utama untuk menggabungkan perangkat fisik yang heterogen ke dalam aplikasi dan untuk mengendalikan semua jenis perangkat fisik terlepas dari teknologi jaringan mereka seperti Bluetooth, RF, ZigBee, RFID, WiFi, dll Hydra-enabled perangkat dan layanan yang aman dan terpercaya melalui keamanan didistribusikan dan sosial unit trust perhitungan disediakan oleh middleware.

The Hydra middleware memungkinkan pengembang untuk menggabungkan perangkat fisik yang heterogen ke dalam aplikasi mereka. Ini menyediakan mudah digunakan antarmuka layanan web untuk mengendalikan semua jenis perangkat fisik terlepas dari teknologi antarmuka jaringan. Hal ini didasarkan pada Model Driven Architecture semantik untuk pemrograman yang mudah dan juga menggabungkan alat untuk perangkat dan penemuan layanan, komunikasi peer-to-peer dan diagnostik. Perangkat Hydra-enabled menawarkan komunikasi yang aman dan terpercaya melalui didistribusikan keamanan dan kepercayaan sosial komponen middleware. Hydra Software Development Kit (SDK),

Pengembangan Perangkat Kit (DDK) dan IDE (Integrated Development Environment) akan memungkinkan para pengembang untuk membuat jaringan tertanam aplikasi AMI baru dan perangkat cepat dan biaya efektif.

Dalam Hydra Ambient Intelligence (AMI) aplikasi, perangkat fisik, sensor, aktuator atau subsistem dapat dianggap sebagai layanan web yang unik. Sebuah hal baru utama dalam pendekatan Hydra adalah bahwa middleware menyediakan dukungan untuk menggunakan perangkat sebagai layanan baik dengan menanamkan jasa dalam perangkat dan dengan layanan proxy untuk perangkat. Hal baru lainnya adalah bahwa middleware mendukung konfigurasi ulang dinamis dan konfigurasi diri, yang merupakan sifat yang sangat diperlukan dalam setiap aplikasi AMI [9], [10]. Untuk membantu pengembang aplikasi dalam mengatasi berbagai macam perangkat mobile dan stasioner dan jaringan, middleware Hydra menyembunyikan perangkat-dependent dan The Hydra middleware juga dapat mengelola komunikasi dalam jaringan Hydra, rute data, menyediakan manajemen sesi dalam komunikasi dan sinkronisasi entitas yang berbeda dalam jaringan.

Implementasi Novel di middleware Hydra adalah penggunaan peer-to-peer (P2P) teknologi jaringan untuk mengidentifikasi dan memanfaatkan layanan yang tersedia di jaringan, bahkan jika mereka berada di belakang firewall atau NAT, [11]. P2P digunakan sebagai alternatif untuk komunikasi WS antara Hydra-enabled rincian devices.network-dependent dan menyediakan antarmuka yang luas terbuka untuk layar, port komunikasi, fasilitas input dan manajemen memori dari masing-masing kelas dari perangkat.

Hydra membahas tiga domain aplikasi: otomatisasi rumah, kesehatan dan pertanian. Demonstrasinya ini berkonsentrasi pada efisiensi energi pada tingkat perangkat di rumah otomatisasi. Prototipe kami disajikan pada tahun 2009 [14] menerapkan serangkaian efisiensi energi konteks-sadar aplikasi cerdas ambient di "Casa Domótica" di mana rumah yang lengkap tersedia untuk menguji aplikasi HYDRA. Sebuah contoh HYDRA middleware dikerahkan juga di rumah, mengendalikan semua sensor ditemukan, aktuator dan perangkat tersebut. Konteks yang diperhitungkan dalam prototipe adalah lokasi pengguna dan saat hari dimana aksi terjadi (apakah ada siang hari atau tidak). Aplikasi yang dikembangkan prototipe adalah:

- Beri aku cahaya: sistem menyediakan pengguna dengan cahaya yang diperlukan mengenai konteksnya. Misalnya, jika pengguna di ruang tamu dan siang hari, sistem akan menonaktifkan cahaya. Jika tidak, lampu akan diaktifkan
- Selamat tinggal, Aplikasi mengurus mematikan semua lampu di rumah dan menolak semua perangkat yang berada dalam mode stand-by Energi-konsumsi.
- Informasi Pengguna mendapatkan real time informasi tentang konsumsi energi saat mereka berinteraksi dengan peralatan. Setiap perangkat memiliki profil energi, yang memungkinkan pengguna untuk memeriksa penggunaan dari perangkat tertentu dalam rumah.

Hydra demikian lebih cocok untuk aplikasi IOT tingkat perusahaan yang membentuk jangka panjang dan kopling ketat dengan set statis perangkat IOT yang sudah didukung oleh platform

TABLE I
IOT MIDDLEWARE FUNCTIONALITIES

Middleware	Functionalities							
	Device abstraction	Network connectivity	Composition	Monitoring and visualization	Service discovery	Security and privacy	Persistency	Stream processing
Hydra	Web service	ZigBee, Bluetooth, RFID, Wi-Fi, HTTP, SOAP	SDK tool kit with Semantic Model Driven Architecture	Vendor GUI application	OWL and SAWSDL for device description	Distributed security and social trust	Unknown	Event manager
GSN	Virtual sensor/XML deployment descriptor	Push/pull, SOAP, HTTP	Specific application created by programmer	Vendor Web application	Key-value store registry for devices	Login account, electronic signature	SQL database	SQL query on data stream
Google Fit	Fit Programmable API or RESTful resource	Bluetooth, Wi-Fi, HTTP	Fitness based devices and data	Vendor Web application	Not supported	Permission and user control module	Cloud storage	Sensors API
Xively	RESTful resource	Bluetooth, Wi-Fi, Socket, MQTT, HTTP	Not supported	Vendor Web application	Not supported	User control on how data is shared	Cloud storage	RDF stream processing
Paraimpu	Sensor, Actuator, Connection	Pull/push, HTTP	Compose one sensor with one actuator using Connection	Paraimpu workspace	New sensor can announce its presence	Access token needed for access, user data shared according to the stated privacy law	Mongo DB	JSON query
Calvin	Actor	Wi-Fi, Bluetooth, i2c	CalvinScript	Vendor Web application	Provide search via Calvin Control APIs	Not supported	Distributed Hash Table (DHT)	Data flow processing
Node-RED	Node	MQTT	Drag and drop visual tool	Browser-based flow editing tool	Key word based search provided for node	User permission with password securely hashed using the bcrypt algorithm	Local file system	Node streams
Ptolemy Accessor Host	Accessor	MQTT, Bluetooth, Wi-Fi, HTTP, UDP, CoAP, Websockets	Drag and drop visual tool based on Virgil	Java-based GUI application	Import and export facility	Key management and Taint analysis	Local file system	Discrete event director

The Hydra kerangka middleware memfasilitasi pengembang untuk membangun efisien scalable, embedded system sementara menawarkan antarmuka layanan web untuk mengendalikan segala jenis perangkat terlepas dari teknologi jaringan. Menerapkan arsitektur berorientasi layanan, Hydra terdiri dari satu set komponen canggih (disebut manajer) mengurus komunikasi P2P antar perangkat, keamanan, acara penanganan, kesadaran konteks dll Setiap manajer dikhususkan untuk tugas yang ditetapkan dalam middleware dan mengekspos layanan untuk dikonsumsi oleh komponen lainnya.

Gambar 3 memberikan gambaran tentang arsitektur Hydra, menampilkan semua manajer yang merupakan bagian dari middleware, beberapa yang penting (misalnya Network manager) dan lain-lain yang menyediakan fungsionalitas opsional (misalnya Konteks Manager atau Manajer Storage). Setiap manajer memiliki peran clearlydefined, menawarkan satu set layanan yang akan digunakan oleh manajer lain atau komponen tingkat aplikasi. Selanjutnya, sebagai Hydra bertujuan mendukung pengembangan sistem terdistribusi, manajer dapat digunakan pada host yang berbeda, berkomunikasi melalui layanan web. Karena itu Hydra mendukung pengembangan berbagai macam aplikasi, dari hanya menghubungkan dua komputer untuk lingkungan meresap penuh mendukung keamanan, penyimpanan dan kesadaran konteks didistribusikan lebih banyak perangkat yang heterogen, tergantung pada kebutuhan domain-spesifik.

Hydra membedakan dua jenis perangkat : mereka yang mampu menjadi tuan rumah bagian dari middleware, dan mereka yang tidak. Konfigurasi minimum hanya mencakup Network Manager, memungkinkan komunikasi P2P antara perangkat. Perangkat yang tidak mampu menjadi tuan rumah Manajer Jaringan, karena sumber daya yang terbatas atau sejenisnya, dapat terhubung ke jaringan Hydra oleh proxy perangkat lunak yang berjalan pada perangkat yang lebih kuat. Contoh perangkat dibatasi

sumber daya yang Manager Jaringan telah successfully digunakan pada adalah Playstation 3 dan Android ponsel G1 ponsel [1].

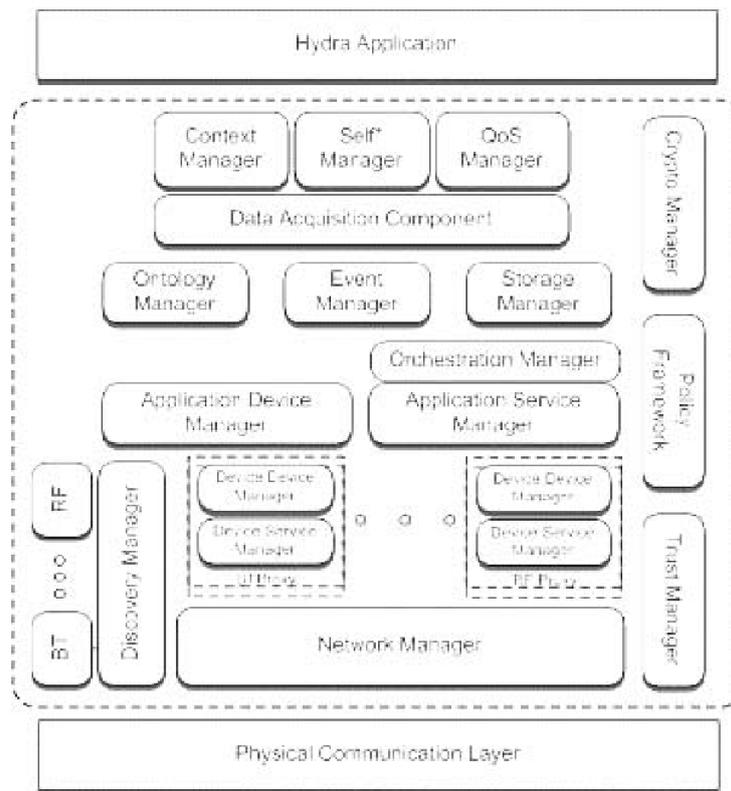


Figure 3. Hydra Architecture

The Network Manager memungkinkan jaringan komunikasi antara perangkat di dalam jaringan Hydra. Ini menciptakan jaringan P2P overlay yang mengimplementasikan SOAP Tunneling sebagai mekanisme transportasi untuk panggilan layanan web [10], yang memungkinkan komunikasi langsung antara semua perangkat di dalam jaringan Hydra, tidak peduli apakah mereka muncul di belakang firewall atau NAT (Network Address Translator). Hydra menggunakan mekanisme kustom pengalamatan yang memberikan pengenalan unik (Hydra-ID, HID) untuk layanan, yang memungkinkan perangkat untuk secara transparan mempublikasikan dan jasa digunakan kapan saja di mana saja tanpa batas-batas jaringan atau endpoint tetap. Jika perangkat ingin mengkonsumsi layanan dari perangkat lain, Manajer Jaringan dari kedua perangkat merawat routing panggilan layanan web, menggunakan layanan HIDs '.

IkhtisarFP6 Hydra proyek (<http://www.hydramiddleware.eu>) adalah Proyek Terpadu 4 tahun bahwa middleware dikembangkan untuk Sistem Jaringan Tertanam. Hydra middleware (saat ini bernama LinkSmart) memungkinkan pengembang untuk menggabungkan perangkat fisik yang heterogen ke dalam aplikasi mereka dengan menawarkan mudah-ke-menggunakan layanan web antarmuka untuk mengendalikan semua jenis perangkat fisik terlepas dari teknologi jaringan seperti Bluetooth, RF, ZigBee, RFID, WiFi, dll Hydra menggabungkan alat untuk Perangkat dan Service Discovery, Model Semantic Driven Architecture, komunikasi P2P, dan Diagnostik. Hydra diaktifkan perangkat dan layanan dapat aman dan terpercaya melalui didistribusikan keamanan dan kepercayaan sosial komponen middleware.

Proyek lain menggunakan hasil dari proyek HYDRA adalah proyek FP7 Adapt4EE. The LinkSmart middleware digunakan di sini untuk menghubungkan sensor di gedung-gedung yang nyata untuk mendapatkan informasi tentang perilaku hunian dalam bangunan tersebut. Perilaku penghuni kemudian disimulasikan dalam bangunan baru yang diusulkan dari domain yang sama untuk memprediksi Key Performance Indicators bangunan tersebut. Indikator kinerja digabungkan dari efisiensi energi, kenyamanan manusia dan nilai-nilai kinerja bisnis. Arsitektur sistem Adapt4EE untuk tahap pengukuran (di mana sebagian besar teknologi semantik dalam proyek yang digunakan) digambarkan dalam skema pada Gambar 2.

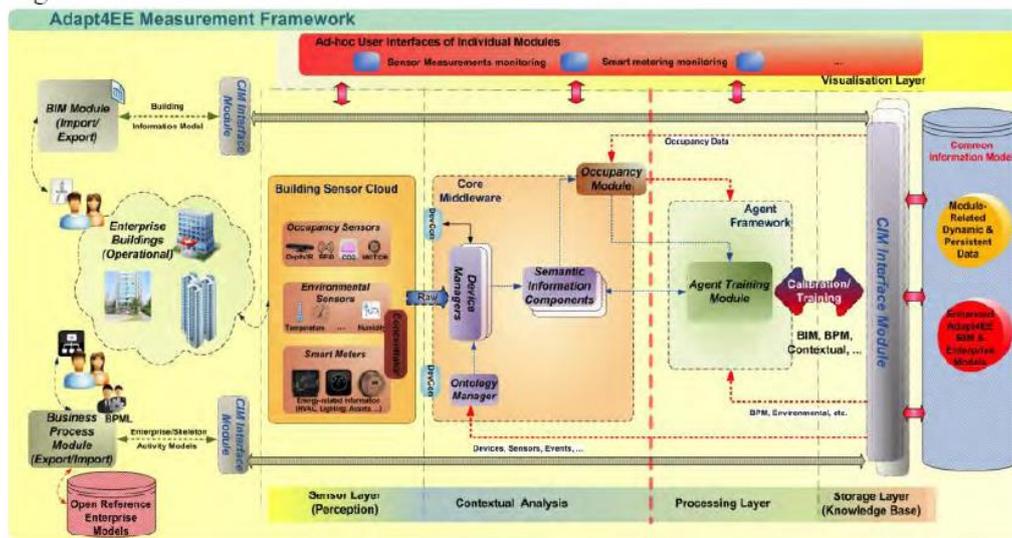


Figure 2 Adapt4EE Measurement Framework

LinkSmart ontologi

Model semantik Driven Architecture (MDA) dari middleware LinkSmart mencakup seperangkat model (ontologi dalam kasus ini) yang digunakan pada desain serta runtime. Semantik sistem dapat dilihat dari dua perspektif: Desain-waktu: selama fase desain (pengembangan), deskripsi semantik bertindak sebagai elemen menghemat tenaga kerja untuk pengembang bekerja pada sebuah aplikasi menggunakan perangkat LinkSmart diaktifkan. Deskripsi semantik dapat digunakan untuk menghasilkan kode untuk memanggil layanan, untuk query perangkat yang mengimplementasikan layanan, dan untuk bekerja dengan data layanan beroperasi. Run-time: selama runtime model semantik sebagian besar digunakan untuk tujuan penemuan. Aplikasi harus bekerja hanya dengan jenis perangkat dan layanan yang pengembang yang dipilih dan akan bekerja dengan deskripsi yang lebih abstrak dari perangkat. Dalam proses penemuan perangkat, informasi penemuan diselesaikan menggunakan deskripsi perangkat. Sistem akan mengidentifikasi template yang paling cocok untuk perangkat dan akan membuat yang baru, unik run-time misalnya mewakili perangkat fisik tertentu. Informasi semantik tambahan dari deskripsi perangkat / layanan dapat digunakan dalam aplikasi untuk menyelesaikan berbagai persyaratan keamanan atau untuk memilih jasa yang sesuai dengan kualitas layanan (QoS) tindakan.

Tujuan dari LinkSmart Aplikasi Ontologi Manager (**Gambar 4**) adalah untuk menyediakan antarmuka yang seragam untuk menggunakan perangkat Ontologi dan semua model terkait dalam middleware LinkSmart. Manajer juga mempertahankan contoh run-time dari perangkat LinkSmart. The

Ontologi Manajer dikembangkan dan diimplementasikan di Java, menggunakan kerangka OSGi sebagai platform layanan, kemudian refactored menjadi modul layanan berbasis web independen (sebagai LinkSmart pindah dari kerangka OSGi teknologi P2P lebih fleksibel).

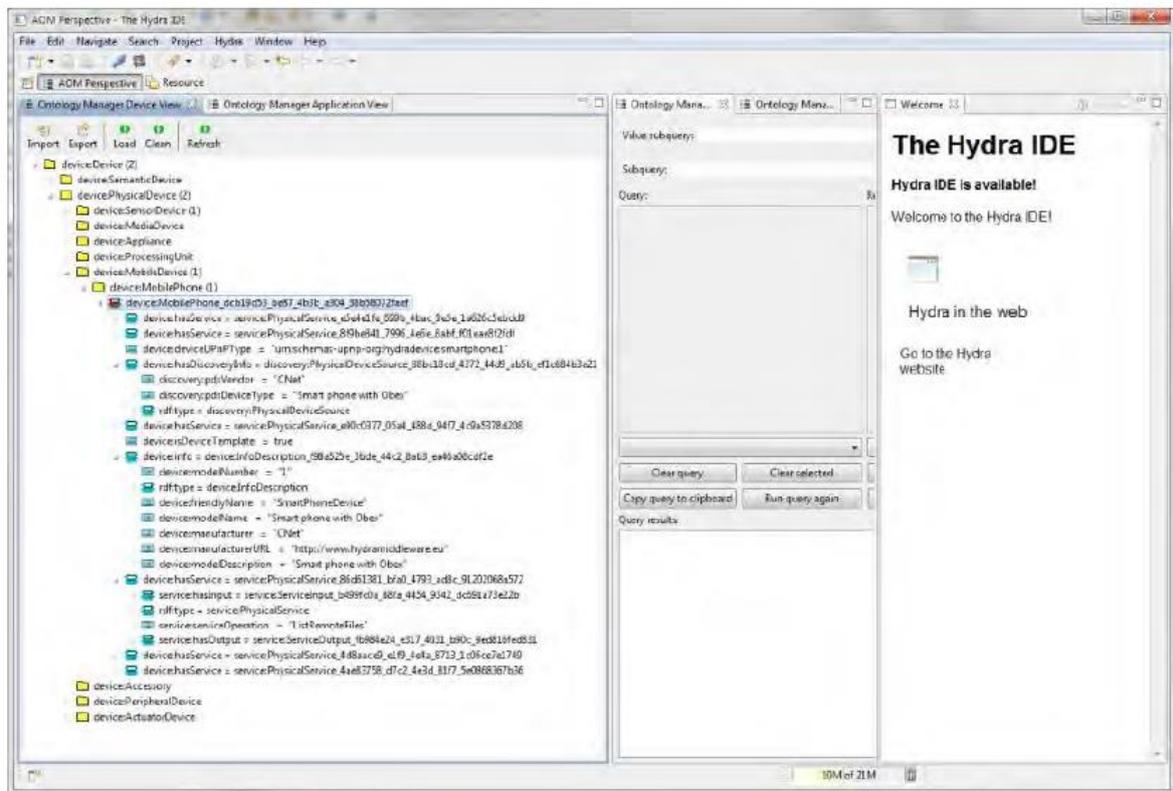


Figure 4 LinkSmart Ontology Manager