

아키텍처 기술 언어를 사용한 센서 네트워크의 적응형 소프트웨어 설계

김현정⁰ 최호진 고인영
한국정보통신대학교 공학부
{hena080⁰, hjchoi, iko}@icu.ac.kr

A Design of Adaptive Software for Sensor Networks Using an Architecture Description Language

Hyun-Chong Kim⁰, Ho-Jin Choi, In-Young Ko
School of Engineering, Information and Communications University

요 약

수많은 경량의 센서 노드들이 무선통신으로 연결되어 사람이 접근하기 어려운 환경에서도 응용서비스를 수행하는 센서 네트워크에 대한 관심이 고조되고 있다. 노드의 다양성과 사람의 접근불가성으로 인해 센서 네트워크는 무인 운용이 가능하도록 설계되고 배치되어야 하며 이를 위한 적응형 소프트웨어에 대한 연구가 활발히 진행되고 있는 추세이다. 본 논문은 소프트웨어 아키텍처 기술 언어를 사용하여 노드 장애나 환경 변화에 동적으로 대처하여 스스로를 재구성하는 센서 네트워크의 적응형 소프트웨어 설계를 제시한다.

1. 서론

통신기술의 발달과 하드웨어의 소형화 추세에 힘입어, 수많은 경량의 센서 노드들이 무선통신으로 연결되어 사람이 접근하기 어려운 환경에서도 응용서비스를 수행하는 센서 네트워크에 대한 관심이 고조되고 있다. 센서 네트워크의 노드들은 가형 전장터에서 탱크의 움직임을 감지한 다던가 하는 임무를 수행하기 위해 사람이 접근하기 힘든 필드에 임의로 뿌려지게 되는데 [3], 이러한 노드의 다양성과 사람의 접근불가성으로 인해 센서 네트워크는 무인 운용이 가능하도록 설계되고 배치되어야 하며 이를 위해서는 적응성있는 소프트웨어의 설계가 필수적이다.

파손 또는 배터리 소진으로 특정 센서 노드가 동작을 멈추면 인접한 노드들이 해당 기능을 대체하여야 하며 전체 네트워크는 재구성되어야 한다. [1]과 [2]는 이러한 재구성 방법으로 정적인 구조의 소프트웨어 아키텍처를 제안하고 있다. 즉, 모든 가능한 재구성 시나리오를 미리 프로그래밍해두는 것인데, 이 방법은 사용자의 변화하는 요구사항을 수용할 수 없다는 단점이 있다. 또한 이 방법들은 동일한 기능의 소프트웨어를 탑재한 동종의 센서 노드들로 구성된 센서 네트워크를 전제로 하고 있어서, 다양한 기능의 이종 센서 노드들로 구성된 네트워크의 재프로그래밍 메커니즘은 지원하지 못하고 있다.

본 논문은 이종 노드들로 구성된 센서 네트워크를 전제로, 소프트웨어 아키텍처 기술 언어를 사용하여 노드 장애나 환경 변화에 동적으로 대처하여 스스로를 재구성하는 센서 네트워크의 적응형 소프트웨어 설계를 제시한다.

2. 관련 연구

센서 노드들이 배치된 환경과 변화하는 요구에 맞게 적응성을 발휘하도록 설계하는 자기조직(self-organizing) 메커니즘에 대한 대표적인 연구로 [4]와 [5]를 들 수 있다. [4]는 스스로 재구성하는 시스템을 위한 소프트웨어 아키텍처를 제안하고 이 아키텍처를 이용하여 네트워크의 센서 노드들을 구성하는 알고리즘을 제시하고 있다. [5]는 데이터 수집, 전달 및 라우팅과 같은 센서 기능들을 코디네이트하는 역할 기반의 계층적인 자기조직 메커니즘을 제안하고 있다. 앞서 언급한 [1]과 [2]는 프로그래밍 관점에서 센서 노드들을 재프로그래밍하는 방법을 제시하고 있는데, 이종 센서 노드들에 대한 지원은 배제되어 있다. 소프트웨어 아키텍처 관점에서 센서 네트워크를 다룬 연구로는 [6]이 있는데, 적응성 문제는 다루지 않았고, 다만 센서 네트워크 소프트웨어의 개발 방법으로 서비스 지향적인 소프트웨어 아키텍처를 채택하고 있다.

3. 아키텍처 기반의 적응형 소프트웨어 설계

본 논문에서 추구하는 센서 네트워크를 위한 적응형 소프트웨어의 설계는 [7, 8, 9]에서 제안하는 컴포넌트-커넥터 기반의 소프트웨어 아키텍처 개념을 채택한다. 즉, 소프트웨어 아키텍처는 상호작용하는 컴포넌트들을 그래프 구조로 표현한 것인데, 컴포넌트라 칭하는 그래프의 노드는 시스템의 컴퓨팅 요소나 데이터 저장소를 나타내고, 커넥터라 칭하는 그래프의 에지는 이들 컴포넌트간의 상호작용을 나타낸다.

3.1. 아키텍처 기술 언어를 사용한 센서 네트워크의 표현

그림 1은 이종 센서 노드들로 구성된 센서 네트워크를 예시하고 있다. 각 노드를 구성하는 소프트웨어 컴포넌트가 노드별로 차이가 있으며 센서 노드들은 임무 수행의 필요에 따라 상호작용을 갖는다.

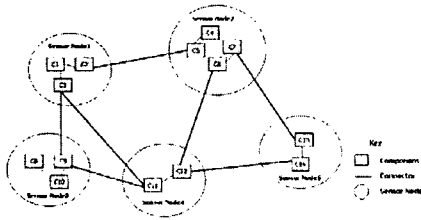


그림 1. 이종 노드로 구성된 센서 네트워크

본 연구는 소프트웨어 아키텍처 기술 언어(Architecture Description Language; ADL)의 일종인 Acme[11]를 사용하여 각 센서 노드 내부의 소프트웨어 컴포넌트들 간의 구조를 표현하고 이를 지역구조뷰(local configuration view)라 명명한다. 지역구조뷰는 해당 노드를 이루는 컴포넌트와 커넥터 및 인접한 노드들로 구성된 것이다. 그림 2는 Acme 언어의 보조도구인 AcmeStudio를 통해 가시화된 센서 노드 한 개의 지역구조뷰를 예시하고 있다.

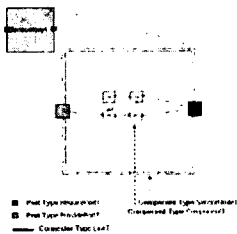


그림 2. 센서 노드의 지역구조뷰

또한, 전체 센서 네트워크의 구조 역시 Acme로 표현하고 이를 전역구조뷰(global configuration view)라 명명한다. 전역구조뷰에서는 각 센서 노드가 아키텍처 상의 컴포넌트로, 노드들 간의 연결(무선통신)이 커넥터로 표현된다.

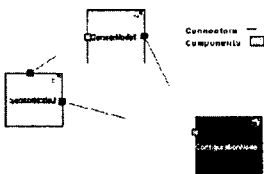


그림 3. 센서 네트워크의 전역구조뷰

그림 3은 AcmeStudio를 통해 가시화된 간단한 네트워크의 전역구조뷰를 보여주고 있고, 이 뷰를 발생시킨 Acme 소스가 그림 4에 예시되어 있다.

```

01 System SensorNodes : SensorNodeFam = new SensorNodeFam extended with {
02   Property threshold_Latency : float = 10.0;
03   Property PowerLevel : int = 100;
04   Component SensorNode1 : SensorNodeT = new SensorNodeT extended with (...);
05   Component SensorNode2 : SensorNodeT = new SensorNodeT extended with (...);
06   Component ConfigurationNode : ConfigurationNodeT
07     = new ConfigurationNodeT extended with {
08     Port providePort : ProvidePortT = new ProvidePortT;
09     ...
10 };
11 Connector LinkT0 : LinkT = new LinkT extended with {
12   Role provider : ProviderRoleT = new ProviderRoleT;
13   Role requirer : RequirerRoleT = new RequirerRoleT extended with {
14     Property Latency : float = 9.0;
15     Property threshold_Latency : float = 10.0;
16 };
17 };
18 Attachment SensorNode2.requirePort to LinkT0.require;
19 Attachment SensorNode1.providePort to LinkT0.provider;
20 Connector LinkT1 : LinkT = new LinkT extended with (...);
21
22 Attachment SensorNode1.requirePort to LinkT1.require;
23 Attachment ConfigurationNode.providePort to LinkT1.provider;
24 Connector LinkT2 : LinkT = new LinkT extended with (...);
25
26 Attachment SensorNode2.RequirePortT0 to LinkT2.require;
27 Attachment ConfigurationNode.providePort to LinkT2.provider;
28 };
    
```

그림 4. Acme 로 기술된 센서 네트워크의 전역구조뷰

3.2. 장애복구 전략의 표현

파손 또는 배터리 소진으로 특정 센서 노드가 동작을 멈추면 해당 기능을 대체할 노드를 찾고 전체 네트워크는 재구성되어야 한다. 가령, 다음과 같은 시나리오를 생각해 보자.

각 센서 노드는 감지한 데이터를 인접한 노드들에게 주기적으로 전송한다. 인접한 노드들이 이 데이터를 소정의 시간 내에 받지 못하면 해당 노드는 장애가 있는 것으로 간주되고 네트워크에서 제거된다 (그림 5).

이러한 유형의 장애복구 전략은 [10]에서 제시한 것과 같은 Acme 스크립트로 표현이 가능하다. 예를 들면, 그림 6은 위의 시나리오에 대처하는 장애복구 전략을 스크립트로 표현한 것이다. 가령, 이 스크립트의 처음 세 줄은 이 장애복구 전략이 실행될 조건을 표시하고 있는데, 첫째 줄의 "invariant r.Latency<=threshold_Latency"은 제약조건을 표현하고, 둘째 줄의 "!->" 연산자는 좌변의 제약조건이 위배되었을 시 우변의 복구 전략을 실행 시키라는 내용을 표현하고 있다. 이 경우, 복구 전략은 "reconfiguration(r)" 이라는 매개변수화된 프로시저로 두 가지 전술을 가지고 있는 것을 알 수 있다.

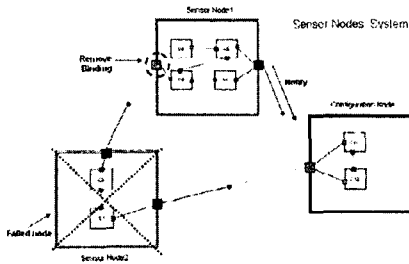


그림 5. 장애복구 시나리오

```

01 invariant r.Latency <= threshold_Latency
02 | →
03 reconfiguration(r);
04
05 strategy reconfiguration(badRole:RequirerRoleT)={
06 begin repair-transaction;
07 let badSensorNode:SensorNodeT=
08 select one sen: SensorNodeT in self.Components |
09 exists p: RequirePortT in sen.Ports | attached(badRole, p);
10 notifyProblem(badSensorNode);
11 If(findSubstitute(badSensorNode, badRole)){
12 commit repair-transaction;
13 }else If(removeBinding(badSensorNode, badRole)){
14 commit repair-transaction;
15 }else{
16 abort(ModelError);
17 }
18 }
19
20 tactic findSubstitute(sensorNode:SensorNodeT,
21 role: RequirerRoleT):boolean={
22 let substituteNode: SensorNodeT =
23 findSubstituteNode(sensorNode);
24 If(substituteNode != nil){
25 removeBinding(role);
26 sensorNode.connectSubstitute(substituteNode);
27 notifyChange(sensorNode, substituteNode);
28 return true;
29 }else{
30 abort(NoSubstituteFound);
31 }
32 }
33
34 tactic removeBinding(sensorNode: SensorNodeT,
35 role: RequirerRoleT):boolean={
36 removeBinding(role);
37 notifyRemoval(sensorNode);
38 return true;
39 }
    
```

그림 6. 장애복구 전략의 표현

4. 결론

본 논문은 이종 노드들로 구성된 센서 네트워크를 전제로, 노드 장애나 환경 변화에 동적으로 대처하여 스스로 재구성하는 적응형 소프트웨어 설계의 일환으로 아키텍처 기술 언어가 어떻게 사용될 수 있는가를 보였다. 향후 연구로는 장애복구 전략을 다변화하여 연구의 유용성을 높이고, 예시된 바와 같은 스크립트를 센서 노드 내에서 동적으로 재구성하여 실행 가능한 프로그램 코드로 변환하는 방법을 연구하고자 한다.

감사의 글

본 연구는 정보통신부의 대학 IT연구센터 사업에 의해 지원되었음.

참고문헌

- [1] Boulis, A., Han, C-C., Srivastava, M.B.: Design and Implementation of a Framework for Efficient and Programmable Sensor Networks. In: Proceedings of the First International Conference on Mobile Systems, Applications, and Services (MobiSys). May 2003.
- [2] Levis, P., Culler, D.: Mate: A Tiny Virtual Machine for Sensor Network. In: Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X), 2002
- [3] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E.: A Survey on Sensor Networks. IEEE Communications Magazine. August 2002
- [4] Subramanian, L., Katz, R.H.: An Architecture for Building Self-Configurable Systems. IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc 2000), Boston, August 2000
- [5] Kochhal, M., Schwiebert, L., Gupta, S.: Role-based Hierarchical Self-Organization for Wireless Ad hoc Sensor Networks. In: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, 2003, pp 98 – 107
- [6] Blumenthal, J., Handy, M., Golasowski, F., Haase, M., Timmermann, D.: Wireless Sensor Networks –New Challenges in Software Engineering. In: Proceedings of 9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Lissabon, Portugal, September 2003
- [7] Oreizy, P., et al.: An Architecture-Based Approach to Self-Adaptive Software. IEEE Intelligent Systems, May/Jun. 1999, 14(3):54-62
- [8] Georgiadis, L., Magee, J., Kramer, J.: Self-Organizing Software Architecture for Distributed Systems. In: Proceedings of the first workshop on Self-healing systems, 2002, pp. 33 – 38
- [9] Garlan, D., Schmerl, B.: Model-based adaptation for self-healing systems. In: Proceedings of the first workshop on Self-healing systems, Charleston, South Carolina, November 18-19, 2002
- [10] Schmerl, B., Garlan, D.: Exploiting Architectural Design Knowledge to Support Self-repairing Systems. 14th International Conf. On Software Engineering and Knowledge Engineering, Ischia, Italy, 2003
- [11] Garlan, D., Monroe, R., Wile, D.: Acme: an architecture description interchange language. In: Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research, November 1997