As an outline, I was thinking:

* Abstract
* Introduction
  + Motivating the need for verification in component based software applications, especially ones catering to mission/safety critical systems
    - Example, if required to show motivation
  + Brief summary of work presented in the paper
* Related Research
  + Other modeling and analysis methods similarly developed for component based software systems.
* Our Component Model
  + Brief description of our component model
    - Brief summary of IAP
    - What is a component?
    - What are the currently available ports?
    - What are the primary interaction patterns?
    - Brief summary of connectors decouple transport mechanism from execution
    - State variables and timers 🡪 might remove this.
    - Component Operations
      * Whats an operation?
      * What are its properties?
      * How are component operations scheduled?
      * Component Executor thread is single-threaded
        + Why?
* Our OS Temporal Partition Scheduler
  + Brief summary of our F6OS scheduler
    - Components are grouped into actors
    - ARINC-653 style temporal partitions
      * Temporal isolation between actors
      * Security needs addressed for safety critical systems.
    - Brief summary of minor frames, major frames and hyperperiod
* Interactions between the two schedulers
  + Briefly describe hierarchical nature of scheduling
  + How’re applications built from re-useable components.
* Modeling and Analysis
  + Colored Petri Nets
    - Extension of petri net
    - Reasons to choose CPN
    - Level of abstraction used
    - Component-Thread Life cycle
    - Handling multiple threads
    - Detecting Missed deadlines
    - Properties of components captured
    - Assumptions and Guarantees
    - How the model scales -> Size of state space for a simple example and how it does not explode for complicated examples
    - What information the developer needs to provide
    - Snapshots of the CPN model  I assume this is a must.
  + Brief summary of analysis:
    - State space generation
    - Queries used once state space is generated
      * Potential power of such queries
      * Maybe time taken to obtain results of state space search?
    - Model checker library available with CPNTools
      * Brief summary of usefulness of model checker
      * Using strongly connected components of state space to efficiently run through state space
  + Examples to show verification
    - What examples are demonstrated
    - What are the scenarios
      * Temporal partition schedule chosen
      * Expected deadlock is detected
      * Expected deadline violation is detected
        + Hopefully, the model is sophisticated enough to detect a deadlock in the future realized by delay propagation
* Conclusions and Future Work
  + What are the lessons learnt
  + How this compares against some of the work presented in related research
  + What is missing and how to improve the modeling and verification
  + How can this be made less difficult for developers